



INAOE

Simulador de la trayectoria orbital de un satélite con orientación fija, utilizando un cuadrorotor

por

Raziel César Campos Sánchez

Tesis sometida como requerimiento parcial para obtener el grado de

Maestro en Ciencias, en el área de Ciencia y Tecnología del Espacio

por el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Diciembre, 2016

Tonantzintla, Puebla

Supervisor:

Dr. Rogerio Enríquez Caldera

Coordinación de Ciencia y Tecnología del Espacio

INAOE

©INAOE 2016

Todos los derechos reservados

El autor(a) otorga al INAOE permiso para la reproducción y distribución del
presente documento



A mi Madre, por la vida y todo el amor.

Índice general

Agradecimientos	XIII
Resumen	xv
1. Introducción	1
1.1. Estado del arte	2
1.1.1. Herramientas para el desarrollo de sistemas satelitales	2
1.1.2. Cuadrorotor	5
1.1.3. Técnicas de control	7
1.2. Justificación	8
1.3. Objetivos y metas	9
1.4. Metodología	9
1.5. Contribuciones	10
1.6. Estructura de la tesis	10
2. Marco Teórico	13
2.1. Cuerpo Rígido	13
2.1.1. Representación de la orientación	15
2.1.2. Ecuaciones de movimiento del cuerpo rígido	21

2.2. Cuadrotor	24
2.2.1. Modelo Matemático de un cuadrotor	24
2.2.2. Técnicas de Control de Cuadrotores	25
2.2.3. Componentes Físicos y Electrónicos	28
2.3. Trayectorias Orbitales	28
2.3.1. Problema de los dos cuerpos	29
2.3.2. Órbitas y algunas de sus propiedades	32
2.3.3. Elementos Orbitales	33
3. Sistema del Simulador	37
3.1. Descripción general del sistema	37
3.2. Interfaz Gráfica de Usuario	38
3.2.1. Especificaciones técnicas del desarrollo de la interfaz	39
3.3. Elementos orbitales a trayectoria orbital escalada	39
3.3.1. Especificaciones técnicas del cálculo de la trayectoria	43
3.4. Características del vehículo cuadrotor comercial	43
3.4.1. Especificaciones técnicas del vehículo	44
3.5. Funcionamiento del simulador con un vehículo real o simulado.	44
3.6. Elementos orbitales a partir de la trayectoria orbital simulada	46
3.7. Consideraciones de variación en las características físicas del vehículo y la carga útil	46
4. Plataforma de implementación de controladores y control difuso	47
4.1. Plataforma de implementación de controladores para la tarjeta 'Pixhawk'	48
4.2. Controlador difuso en tiempo real	49
4.2.1. Datos de entrenamiento	49
4.2.2. Parámetros de entrenamiento de ANFIS	52

5. Experimentos y resultados	55
5.1. Especificaciones de funcionamiento de la Interfaz de control del simulador	55
5.1.1. Algoritmo para la obtención de la trayectoria orbital a partir de los elementos orbitales un satélite	56
5.1.2. Comunicación con el vehículo	57
5.2. Simulación orbital	57
5.2.1. Simulación orbital con un vehículo virtual	57
5.2.2. Simulación orbital con un vehículo real	62
5.2.3. Comparativa entre simulación con vehículo real y virtual	66
5.3. Plataforma de implementación de algoritmos de control	67
5.3.1. Controlador difuso en tiempo real	67
6. Conclusiones y trabajo futuro	69
A. Interfaz Gráfica para el control de la simulación	71

Índice de figuras

1.1. Simulador de Navegación Aeroespacial.[?, p.173]	3
1.2. Software para el desarrollo de sistemas satelitales	3
1.3. Instalaciones de ESA-ESTEC para pruebas de satélites	4
1.4. Simulador vuelo satelital para la validación experimental de la orientación de satélites desarrollado en la UNAM.[?]	5
1.5. Cuadrorotor marca 3DR Robotics.	5
1.6. Modelado del sistema de navegación de 5 niveles.	6
1.7. Movimientos del cuadrorotor en función de la velocidad de los motores.	7
2.1. Movimiento de un cuerpo rígido respecto de un marco de referencia inercial.	14
2.2. Red Neuronal Generalizada [GNN][?, p.1].	27
2.3. 2 cuerpos analizados bajo un marco de referencia inercial [?, p.60].	30
2.4. Órbita elíptica [?, p.87].	32
2.5. Elementos orbitales bajo un marco de referencia geocentrico ecuatorial [?, p.87].	34
3.1. Diagrama del funcionamiento general del sistema propuesto, donde \mathbf{R} y \mathbf{V}	38
3.2. Interfaz gráfica de usuario para el control del simulador de trayectoria orbital.	39
3.3. Diagrama para convertir elementos orbitales a trayectoria local escalada.	41
3.4. Esquema de control implementado en la tarjeta 'Pixhawk' para cada grado de liber- tad del vehículo.[?]	43

3.5. Interfaz gráfica de control durante la conexión con el sistema de navegación de un vehículo virtual.	45
4.1. Sistema de vuelo experimental desarrollado en 'Simulink' para la implementación rápida de algoritmos de control en la tarjeta controladora 'Pixhawk'	48
4.2. Señal de cada uno de los motores y el promedio. El eje vertical representa el valor PWM de cada uno de los motores, el eje horizontal representa los ciclos del procesador	50
4.3. Valor PWM de los motores debida a la estabilización de la orientación, el eje vertical representa el valor PWM de cada uno de los motores, el eje horizontal representa los ciclos del procesador.	51
4.4. Señal proveniente del radio control normalizada, el eje vertical indica la posición de los joysticks y el eje horizontal representa los ciclos del procesador.	51
4.5. Coeficiente de requerimiento de fuerza de cada grado de libertad, el eje vertical indica la cantidad de fuerza requerida y el eje horizontal representa las muestras de cada ciclo del procesador.	51
4.6. Contribución de fuerza para cada grado de libertad debida a las estabilizaciones del vuelo, el eje horizontal representa el valor en PWM y el eje horizontal representa las muestras de cada ciclo del procesador.	52
4.7. Superficies de control resultantes para cada grado de libertad, los ejes horizontales muestran la posición del grado de libertad en radianes y el requerimiento del radio control. El eje vertical indica la salida PWM generada.	53
5.1. Interfaz de usuario para el control de la simulación mostrando la gráfica de la órbita del satélite GPS-24876.	56
5.2. Interfaz gráfica de control funcionando en conjunto con un vehículo virtual, el vehículo virtual se ejecuta sobre el software 'SITL' bajo el sistema operativo 'Linux - Ubuntu'	58
5.3. Trayectoria simulada de la órbita de la estación espacial internacional realizada por un vehículo virtual. La visualización de la trayectoria vista desde arriba, se realiza en un mapa geo-referenciado	58

5.4. Posición del vehículo virtual bajo el marco de referencia NED durante la simulación orbital. El eje vertical indica la posición en metros y el eje vertical representa cada muestra del ciclo del procesador.	59
5.5. Orientación del vehículo virtual durante la simulación orbital. El eje vertical indica la posición angular de cada eje medido en grados y el eje vertical representa cada muestra del ciclo del procesador.	60
5.6. Gráficas de desempeño de la trayectoria orbital simulada con un vehículo virtual. . .	61
5.7. Trayectoria de la órbita simulada realizada por el vehículo real visualizada en un mapa geo-referenciado.	63
5.8. Posición del vehículo real bajo el marco de referencia NED durante la simulación orbital. El eje vertical indica la posición en metros y el eje horizontal representa cada muestra del ciclo del procesador.	63
5.9. Orientación del vehículo real durante la simulación orbital. El eje vertical indica la posición angular de cada eje medido en grados y el eje vertical representa cada muestra del ciclo del procesador.	64
5.10. Gráficas de desempeño de la trayectoria orbital simulada con un vehículo real. . . .	65
5.11. Registro de datos del sistema de vuelo implementado.	67
5.12. Superficies de control resultantes para cada grado de libertad	68

Índice de tablas

3.1. Valores de los parámetros de los controladores bajo el esquema presentado en la figura 3.4 para cada grado de libertad del vehículo	44
5.1. Datos de la trayectoria orbital simulada realizada por un vehículo virtual	61
5.2. Elementos orbitales de la trayectoria realizada por el vehículo virtual. Semi-eje mayor, excentricidad, inclinación, argumento de perigeo, ascensión recta del nodo ascendente. Los datos corespondientes a angulos están dados en grados.	62
5.3. Datos de la trayectoria orbital simulada realizada por un vehículo real	66
5.4. Elementos orbitales de la trayectoria realizada por el vehículo real. Semi-eje mayor, excentricidad, inclinación, argumento de perigeo, ascensión recta del nodo ascendente. Los datos corespondientes a angulos están dados en grados.	66
5.5. Comparativa entre los errores de las simulaciones con vehículo real y virtual.	66

Agradecimientos

Infinitas gracias a mi Madre, por siempre darme lo mas valioso en esta vida, el amor.

Gracias a mi familia, por su ayuda y comprensión.

Gracias a mi pareja Maricarmen, por su compañía, cariño y paciencia.

Gracias a Montse, por mostrarme el bello mundo de la programacion, donde quiera que estés...

Gracias a todos mis maestros que compartieron sus conocimientos conmigo.

Agradezco a CONACyT por su apoyo mediante la beca CVU 623274.

Resumen

Los avances científicos y tecnológicos en el área espacial han propiciado algunos de los más grandes avances en la humanidad. En la actualidad, los sistemas satelitales y todos los servicios que proporcionan son algunas de las herramientas más importantes para el avance de la sociedad, es por esto que el desarrollo de estos sistemas es llevado a cabo por unas pocas agencias espaciales ya que es un proceso multidisciplinario y generalmente costoso, lo que dificulta el acceso a los temas espaciales a países en desarrollo creando una brecha cultural, científica y tecnológica.

En los últimos años la aparición de dispositivos electrónicos de bajo costo y alto rendimiento, han hecho posible que el desarrollo de tecnología tenga un costo relativamente reducido. Es por esto que ahora es posible participar en actividades espaciales sin requerir de herramientas especializadas, aprovechando las ventajas que se obtienen del desarrollo de sistemas satelitales.

En este trabajo se propone la realización de un sistema de simulación de la trayectoria orbital de un satélite llevada a cabo por un dron comercial. Este simulador tiene como objetivo principal servir como plataforma de experimentación, validación e integración de subsistemas satelitales, en un ambiente más adecuado a las circunstancias cinemáticas presentes en órbita.

La trayectoria de simulación se obtiene a partir de los elementos orbitales del satélite, posteriormente se adecua y se envía al vehículo, en este caso un cuadrotor. Existen dos formas fundamentales de funcionamiento del simulador: utilizar un vehículo real o un vehículo virtual. El correcto funcionamiento se logra utilizando un interfaz gráfica de usuario que integra los elementos necesarios para realizar la simulación, estos son: Introducción de elementos orbitales y obtención de la trayectoria, control de la simulación y conexión con el sistema de navegación del vehículo.

El lenguaje de programación 'Python' fue elegido para la codificación de la interfaz, por ser de código libre y funcionar bajo los sistemas operativos más comunes 'Windows' y 'Linux'. El protocolo de comunicación implementado es utilizado por la tarjeta controladora de vuelo 'Pixhawk', utilizada en este trabajo, entre otras tarjetas de vuelo comerciales.

Una característica importante de los vehículos cuadrotores comerciales es la reducida capacidad de manipulación del sistema de navegación, por lo tanto no posible realizar cambios en el vehículo sin ver afectado el funcionamiento. Para evitar este problema y darle robustez al sistema del simulador, se propone la construcción de una plataforma de implementación de controladores para la tarjeta controladora 'Pixhawk'. Esta plataforma consta, en una primera etapa, de un sistema que es capaz de realizar el vuelo mediante comandos provenientes del radio control, utilizando controladores implementados por el usuario con capacidad de registrar los datos producidos por el sistema. El sistema resultante es codificado utilizando 'Matlab-Simulink' y la toolbox 'Pixhawk-Simulink'.

EL simulador de trayectoria es probado realizando simulaciones de trayectoria en sus dos formas de operación: con un vehículo real o uno virtual. Los resultados muestran un desempeño satisfactorio en ambos experimentos, los datos para el análisis son obtenidos a partir de los datos del registro de vuelo, comparando los resultados con la trayectoria de referencia generada por la interfaz.

Así mismo la plataforma es probada con controladores difusos obtenidos a partir de datos de vuelo del sistema de navegación de fábrica del cuadrotor utilizando la herramienta 'Adaptative Neuro-Fuzzy Inference System, ANFIS', los controladores difusos muestran un desempeño aceptable a nivel de simulación.

La codificación del sistema de vuelo para la implementación de controladores fue exitosa, mostrando la capacidad de implementar controladores de diferentes tipos, de manera sencilla y correcta.

El sistema de simulación de trayectoria orbital fue correctamente implementado y probado mostrando resultados satisfactorios en todas sus etapas de funcionamiento, brindando nuevas alternativas para el desarrollo de sistemas satelitales y debido a su diseño es accesible a una gran parte de la población.

Capítulo 1

Introducción

Desde la puesta en órbita del primer satélite artificial el 'Sputnik 1' el 4 de Octubre de 1957 [?], el ser humano ha hecho uso de los satélites para resolver diversas problemáticas de la sociedad y su desarrollo. Las aplicaciones de un satélite son muy variadas y van desde las comunicaciones hasta la observación del cosmos, además de las importantes ventajas que ofrece la observación de la tierra desde el espacio con los sistemas de percepción remota.

Los satélites son sistemas complejos y muy costosos, están constituidos por múltiples instrumentos necesarios para las tareas de navegación, control, comunicación y la misión específica que vayan a desempeñar. La principal desventaja de los satélites es que una vez que se encuentran en órbita es muy poco factible repararlos, debido a esto, antes de que un satélite pueda ser construido y lanzado, todos sus sistemas pasan cuidadosamente por los procesos de diseño, implementación, experimentación, validación, integración y finalmente son sometidos a rigurosas pruebas para asegurar el correcto funcionamiento durante su vida útil. [?]

Las herramientas necesarias para el desarrollo de sistemas satelitales son muy variadas, van desde un pequeño sensor hasta edificios especializados a simular el ambiente espacial. Esto puede representar un elevado costo, lo que dificulta el acceso y la creación de sistemas satelitales a países en desarrollo. Una forma de evitar este problema son los simuladores virtuales, aunque solo resuelvan el problema de manera parcial debido a sus limitantes cuando se busca la implementación física del sistema.

En este trabajo se presenta una herramienta alternativa que puede servir como auxiliar en el diseño, experimentación, integración y validación de sub-sistemas satelitales, mediante la simulación

de la trayectoria orbital de un satélite a partir de sus Elementos Orbitales. Para realizar todo lo anterior, la trayectoria es escalada y transformada para poder ser realizada por un UAV (Unmanned Aerial Vehicle), en este caso un cuadrotor. Las relaciones entre la velocidad del vehículo y la posición así como su orientación a lo largo de la trayectoria orbital, son conservadas en la simulación.

El vehículo será capaz de llevar una carga útil dependiente del sub-sistema a experimentar, cuyas dimensiones dependerán de las especificaciones técnicas del cuadrotor utilizado el simulador.

El sistema de fábrica de los vehículos comerciales ofrece una reducida tolerancia a modificaciones al sistema, por lo tanto para dar robustez al simulador como plataforma de validación e integración y aumentar el rango de la carga útil a experimentar, se propone el desarrollo de una plataforma de implementación de controladores. El control difuso es la técnica utilizada con fines de probar su correcta codificación en la tarjeta controladora del vehículo. Los controladores difusos son entrenados y simulados con datos de vuelos de prueba reales para finalmente ser comparados con la respuesta de los controladores del sistema de fábrica.

1.1. Estado del arte

A continuación se muestra el estado del arte para las diversas etapas de este trabajo. Primero, se presentan algunas de herramientas virtuales para el desarrollo y simulación de sistemas satelitales así como los más avanzados sistemas de pruebas físicas de algunos de los países con mayor desarrollo tecnológico en el área espacial que actualmente se desarrollan en los centros de ensamble, integración y prueba de satélites (AIT por sus siglas en inglés). Posteriormente un panorama general de los UAVs, en específico de los cuadrotos comerciales: funcionamiento y técnicas de control.

1.1.1. Herramientas para el desarrollo de sistemas satelitales

Software de diseño satelital y simuladores virtuales

Este tipo de herramientas son las más comunes y utilizadas debido a su accesibilidad y los mínimos requerimientos técnicos para su funcionamiento. Existen diversos tipos de software auxiliares en diversas tareas por ejemplo los de diseño del satélite y sus características físicas y dinámicas, de diseño de órbitas considerando distintos modelos gravitacionales [ver Figura. 1.1], de constelaciones satelitales para cobertura de comunicaciones y toma de imágenes de alta definición, entre otros. Algunas herramientas más sofisticadas permiten el diseño y simulación de sistemas sa-

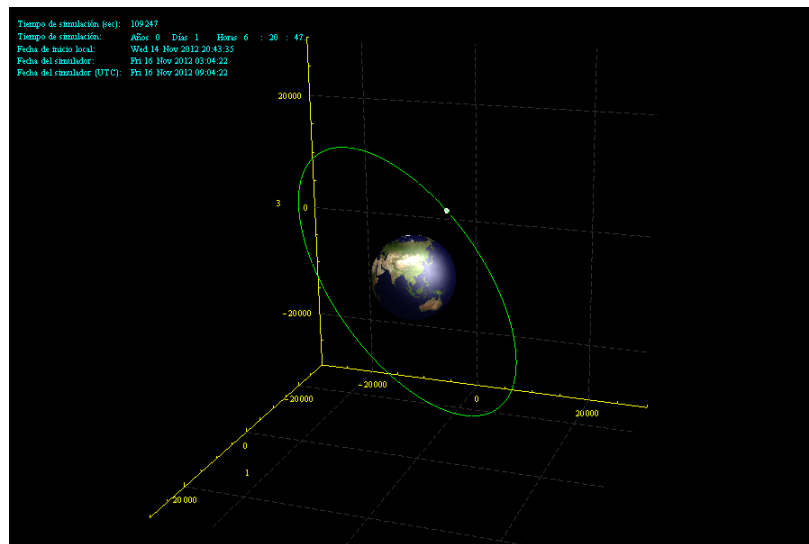
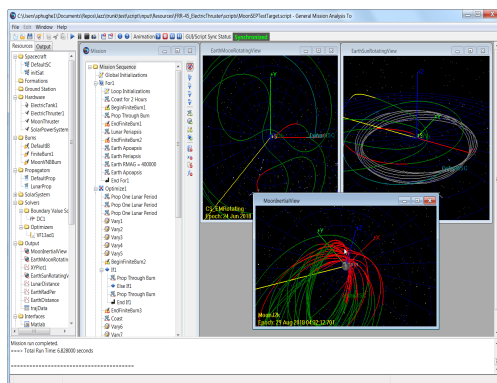
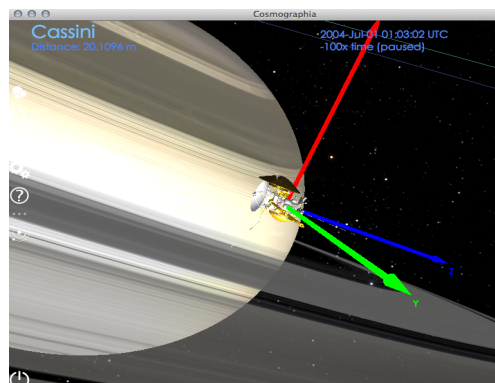


Figura 1.1: Simulador de Navegación Aeroespacial.[?, p.173]

telitales completos con estación terrena y satélite en órbita[?], también existen programas de código libre en forma de API (Application Programming Interface), disponible para diversos lenguajes de programación capaces de realizar un análisis completo de una misión espacial como el 'General Mission Analysis Tool (GMAT)' [?] o el software 'SPICE'[?]. En la Figura. 1.2 podemos imágenes de simulaciones realizadas con algunos de estos programas.



(a) Simulación de trayectorias orbitales utilizando el software GMAT.[?]



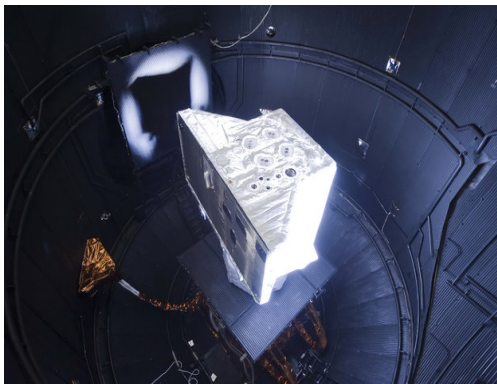
(b) Simulación de la sonda espacial cassini, utilizando el software SPICE.[?]

Figura 1.2: Software para el desarrollo de sistemas satelitales

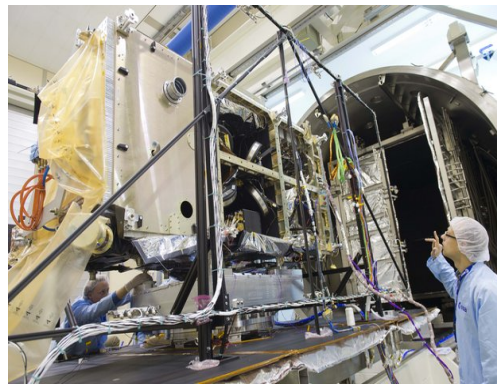
Pruebas a sistemas satelitales físicos

Las pruebas realizadas a un satélite que se pondrá en órbita consideran los fenómenos físicos más importantes del ambiente espacial: carga eléctrica del vehículo, rayos cósmicos que puedan interferir con algún componente electrónico, la fricción atmosférica en órbitas de baja altitud (LEO - Low Earth Orbit), efectos de la radiación sobre los dispositivos electrónicos, interferencia solar e intermitencia en los datos de telemetría, la desgasificación de los materiales al exponerse al vacío principalmente cuando hay instrumentación óptica presente.[?]

Este tipo de simuladores son los más complejos que existen, ya que es necesario reproducir los ambientes extremos que pueden presentarse en órbita, los más comunes son el ambiente térmico, electromagnético, acústico y de vacío. Las instalaciones requeridas para la realización de estas simulaciones son muy grandes y están equipadas con instrumentos de alta tecnología desarrollados específicamente para esta función. Regularmente, este tipo de instalaciones pertenece a grandes compañías privadas o a las agencias espaciales más importantes del mundo [?], [?], [?], [?], entre otros. En la Figura 1.3 podemos observar dos ejemplos de este tipo de instalaciones.



(a) 'Large Space Simulator', ESTEC-ESA.[?]



(b) 'Thermal vacuum test chamber', ESTEC-ESA.[?]

Figura 1.3: Instalaciones de ESA-ESTEC para pruebas de satélites

También existen herramientas que pueden ser auxiliares en el diseño y pruebas para la implementación de sistemas satelitales en movimiento, teniendo especial cuidado en los sistemas de orientación. En México existe desarrollo tecnológico en el área de orientación como podemos ver en este simulador de vuelo satelital desarrollado por Mendoza et al [?][ver Figura 1.4].



Figura 1.4: Simulador vuelo satelital para la validación experimental de la orientación de satélites desarrollado en la UNAM.[?]



Figura 1.5: Cuadrotor marca 3DR Robotics.

1.1.2. Cuadrotor

Desde principios del siglo XX el ser humano ha hecho el intento por desarrollar un vehículo multirrotor capaz de quedar suspendido en el aire pero a pesar de los esfuerzos los resultados no fueron satisfactorios [?]. Recientemente el desarrollo de procesadores más pequeños y potentes, sensores más precisos y el aumento en la capacidad de almacenamiento de energía han hecho posible el desarrollo de vehículos aéreos no tripulados, ya sean de ala fija o multirrotores. Los cuadrotores son vehículos que constan de cuatro rotores unidos por una estructura mecánica sencilla, usualmente en configuración de 'X' o '+' y son capaces de volar gracias al empuje que generan los rotores colocados en cada extremo. Podemos observar un ejemplo de cuadrotor comercial en la Figura 1.5.

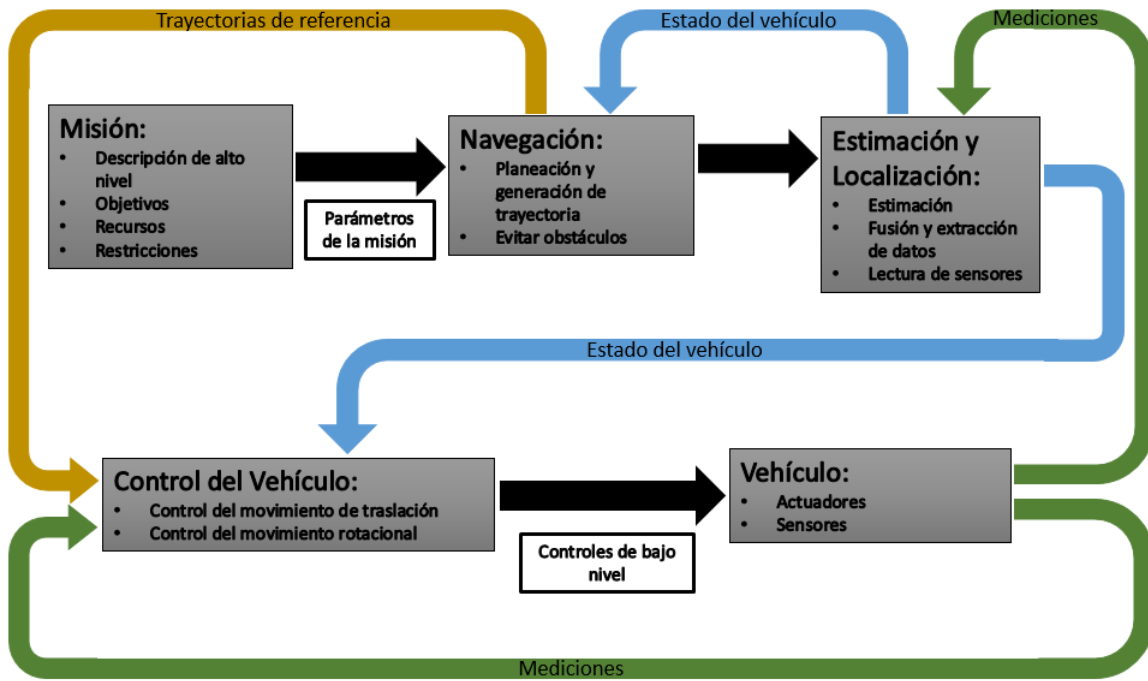


Figura 1.6: Modelado del sistema de navegación de 5 niveles.

Estos vehículos tienen una amplia diversidad de aplicaciones, en gran parte por la capacidad de despegue vertical y una alta maniobrabilidad que les permite realizar tareas que para otro tipo de robot o incluso para un humano pueden ser difíciles o peligrosas. La simplicidad mecánica contrasta con la complejidad del problema del control ya que es un sistema no lineal y sub-actuado, es decir, se controlan los 6 grados de libertad existentes $[x, y, z, \theta, \phi, \psi]$ donde x, y, z especifican la trayectoria y los ángulos roll(alabeo), pitch(cabeceo) y yaw(guiñada) con tan solo 4 actuadores.

El sistema de navegación completo de un vehículo autónomo consta de 5 niveles [?]. Actualmente se pueden conseguir vehículos comercialmente que generalmente incluyen motores, hélices, estructura, componentes electrónicos y el controlador con un sistema de vuelo desarrollado por el vendedor. Algunas compañías permiten el acceso a las diferentes partes del sistema de vuelo, aunque con serias limitaciones.

El funcionamiento de un cuadrotor está basado principalmente en el empuje que genera cada motor y la diferencia de torque producido por cada par de motores opuestos, las adecuadas combinaciones de las velocidades angulares de cada motor nos permiten movernos en los 6 grados de libertad existentes. En la Figura 1.7 podemos observar las configuraciones adecuadas para producir el movimiento.

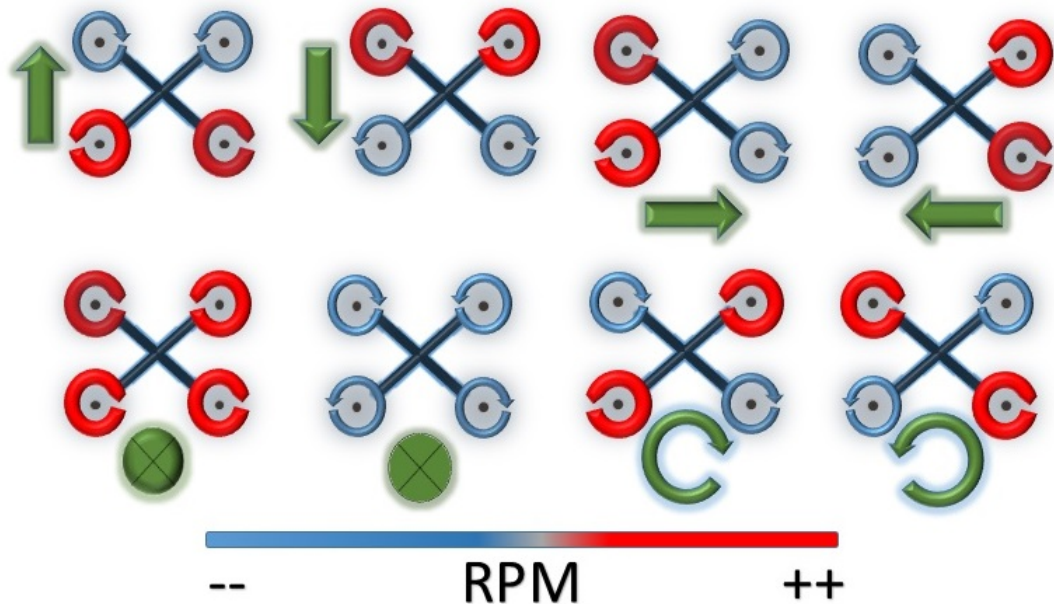


Figura 1.7: Movimientos del cuadrotor en función de la velocidad de los motores.

La principal ventaja que ofrece este tipo de vehículos es su estabilidad, por el contrario la principal desventaja es la dificultad que representa coordinar los cuatro motores al mismo tiempo.

1.1.3. Técnicas de control

Para que un cuadrotor pueda volar debe contar con un control que establezca las velocidades de los cuatro motores de forma coordinada. Actualmente esto se hace con una tarjeta controladora que tiene programado un control y está conectada a los motores y sensores que permiten la retroalimentación de los estados del sistema.

Un problema que se presenta al momento de controlar el vuelo es la complicada dinámica del sistema, incertidumbres en el modelo y el ruido proveniente de los sensores incluyendo la *IMU* (*Inertial Measurement Unit*), que en general tratan de disminuirse por medio de sistemas de estimación en los seis grados de libertad involucrados.

El principal obstáculo al controlar vehículos cuadrotores comerciales es la limitada capacidad de memoria y procesamiento que se tiene a bordo, y debido a la dificultad de obtener un modelo exacto de la planta del sistema, generalmente se utilizan estrategias de control independientes del modelo. El sistema de vuelo de las tarjetas de vuelo comerciales contiene un PD programado como estrategia de control, y con fines de precaución la respuesta es usualmente lenta.

A pesar de la versatilidad de los vehículos cuadrotores comerciales y sus aplicaciones de alto nivel, se debe tener en cuenta una consideración básica al momento de elegirlos para realizar tareas complejas: La capacidad de manipulación de a nivel de misión contrasta con la capacidad restringida de manipular el vehículo a un nivel de motores y sensores, por el contrario, si tenemos acceso libre a nivel de sensores y actuadores y se desea programar una misión autónoma, comúnmente debemos de programar el sistema de vuelo casi en su totalidad. Los sistemas comerciales aumentan su costo con la capacidad del usuario para manipular los distintos niveles de control del vehículo.

1.2. Justificación

La participación de los países emergentes en las actividades referentes al espacio es un aspecto que las agencias espaciales más importantes del mundo están promoviendo [?], debido a esto, es necesario generar herramientas de bajo costo que permitan tener familiaridad con los sistemas espaciales para generar mayor accesibilidad a esta área.

Las herramientas existentes para este propósito son en su mayoría, simuladores virtuales ya que solo se requiere una computadora para su funcionamiento. La principal limitación de este tipo de herramientas es que no siempre se puede llevar el trabajo hecho en el simulador a la implementación de un sistema físico, lo que dificulta el desarrollo científico y tecnológico. Debido a esto, se vuelve de gran utilidad una herramienta que sirva de plataforma para diseñar, experimentar e integrar algunos de los sub-sistemas más importantes que componen un satélite, como el de navegación, sensado, control, comunicaciones, entre otros. Con esto se busca extender las posibilidades de contribuir con el desarrollo científico y tecnológico además de hacer que las tecnologías espaciales sean accesibles para los países en desarrollo.

Otro aspecto importante que se debe tener en cuenta son las ventajas de la utilización de tecnología comercial de bajo costo ya que brinda eficiencia y versatilidad para implementar sistemas que se adecuen a la solución de los problemas planteados, además de aumentar su grado de reproducibilidad y disminuir el costo.

La exploración espacial, con todas las necesidades y posibilidades que plantea, ha sido responsable de algunos de los mayores avances de la humanidad, además de impulsar la creación de herramientas tecnológicas que ayuda al desarrollo de un país. Esto puede comprobarse con la historia de los países desarrollados que han pasado muchos años creando y perfeccionando sus tecnologías.

1.3. Objetivos y metas

La simulación propuesta tiene como principal objetivo implementar una herramienta que sea capaz de simular la forma en la que se mueve un satélite, respetando las variaciones de velocidad en función de la posición a lo largo de su trayectoria orbital con la finalidad de brindar un contexto mas adecuado a las circunstancias reales de movimiento.

Facilitar la reproducibilidad separando la abstracción de la implementación, utilizando software libre para hacer al sistema accesible a más usuarios.

Crear herramientas tecnológicas reutilizables que puedan ser ajustadas y optimizadas para satisfacer nuevas necesidades.

Aumentar la robustez del funcionamiento del simulador de trayectoria orbital diseñando una plataforma de implementación de controladores, de manera que éste simulador de trayectoria orbital pueda ser utilizado como plataforma genérica de experimentación, validación e integración de sub-sistemas satelitales con cargas útiles y variación en las características de los vehículos.

Una de las metas más importantes para este trabajo es comprobar que se pueden obtener ventajas de la utilización de tecnología de bajo costo obteniendo resultados de calidad y mayor posibilidad de acción ante la necesidad de hacer al espacio mas accesible globalmente.

Otra meta importante es generar experiencia en el desarrollo de sistemas completos: Teoría, diseño, implementación y control.

1.4. Metodología

La trayectoria orbital de un satélite será calculada a partir de sus elementos orbitales, después será transformada y escalada para poder realizar la simulación en la Tierra con un vehículo cuadrotor comercial. El cálculo de la trayectoria se realizara utilizando una interfaz gráfica de usuario desarrollada en 'Python' que permite variar los parámetros de la simulación y contiene los protocolos para realizar la comunicación con el vehículo por medio de radios de telemetría. Para realizar la simulación se utiliza el sistema de navegación tal cual viene programado de fábrica.

Luego para el aumento de la robustez del sistema se desarrollará, como primera etapa, un sistema de vuelo que sirva como plataforma de implementación de controladores para el vehículo, buscando su futura utilización en conjunto con el simulador. Esta plataforma se codificará en la

tarjeta controladora 'Pixhawk' de la marca 3DR, basada en arduino y será codificado en 'Simulink', utilizando la toolbox de soporte 'Pixhawk-Simulink'.

Para probar la viabilidad de la plataforma de implementación de controladores se desarrollará un control difuso para la estabilización de un vehículo cuadrotor y cuya ley de control sera inferida con la herramienta ANFIS.

1.5. Contribuciones

Este trabajo contribuye con el desarrollo de una herramienta de simulación de trayectoria orbital que permite realizar pruebas de experimentación, integración y validación de instrumentación abordo de satélites, de una manera más apegada a las circunstancias cinemáticas reales en comparación con los simuladores virtuales, brindando confianza en las pruebas y con una inversión relativamente baja en comparación con otros sistemas de simulación de sistemas satelitales físicos. Este trabajo también fomenta al mismo tiempo la noción de cómo se mueven los vehículos cuando se encuentran en el espacio.

Así mismo se contribuye con la creación de una plataforma para la implementación de controladores en la tarjeta controladora 'Pixhawk'.

Una última contribución es el desarrollo de un sistema de control difuso para la estabilización del vuelo del vehículo cuadrotor que mejora la respuesta del controlador de fábrica en la simulación.

1.6. Estructura de la tesis

En el capítulo 1 se presenta el planteamiento del problema con un panorama general de los satélites, las herramientas existentes para su desarrollo, además de algunos tipos de sistemas de simulación satelital y de las características más importantes de los cuadrotos. En el capítulo 2 se analizará la cinemática y dinámica del cuerpo rígido para entender como es que se utiliza un cuadrotor como plataforma de simulación de sistemas satelitales. También analizaremos el modelo matemático de un cuadrotor así como algunas técnicas de control utilizadas en sistemas comerciales. Así como el problema de los dos cuerpos y como se representa una trayectoria orbital mediante los elementos orbitales. En el capítulo 3 se describe el simulador de trayectoria satelital junto con los elementos que la conforman. En el capítulo 4 se presenta la plataforma de implementación de

controladores con un controlador difuso en la tarjeta controladora 'Pixhawk'. En el capítulo 5 se presentan los experimentos y los resultados obtenidos para finalizar con el capítulo 6 donde se exponen las conclusiones y el trabajo a futuro.

Capítulo 2

Marco Teórico

Para comprender el funcionamiento del simulador de trayectoria orbital se presentan las bases teóricas de las diversas fases que componen al sistema. Primero se muestra un panorama general del movimiento del cuerpo rígido, respecto del cuadrotor, se presentan las etapas matemática, mecánica y electrónica que se requieren para realizar un vuelo. Finalmente se presenta el problema gravitatorio de los dos cuerpos y como se puede representar una órbita alrededor de un cuerpo central, mediante los elementos orbitales.

2.1. Cuerpo Rígido

Podemos definir un cuerpo rígido como un sistema de puntos materiales en el cual la distancia entre todos los pares de puntos que lo conforman se mantiene constante durante el movimiento[?]. Tomando en cuenta esto, podemos aproximar a un satélite tanto como un cuadrotor como cuerpos rígidos si observamos las trayectorias desde un marco de referencia inercial, esto es, observar el movimiento desde un punto fijo o moviéndose a una velocidad constante.

Para describir como se mueve un cuerpo rígido a lo largo del tiempo establecemos un vector [ver Figura 2.1] que va del origen de un sistema de referencia inercial $\{I\}$ al cuerpo rígido, en ese punto se establece un segundo marco de referencia y el origen de un vector G que se encuentra en el marco de referencia móvil $\{B\}$ y representa la posición de un punto p en el cuerpo. Ahora solo debemos analizar la variación de las componentes de dicho vector, tanto en traslación como en rotación para obtener el movimiento total del vector. Las componentes de dicho vector dependerán

de la forma de parametrización utilizada[?]:

$$dG_{\text{espacio}} = (dG)_{\text{traslación}} + (dG)_{\text{rotación}} \quad (2.1)$$

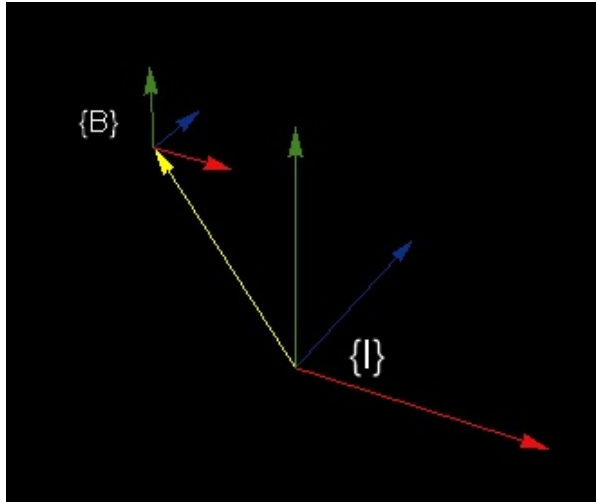


Figura 2.1: Movimiento de un cuerpo rígido respecto de un marco de referencia inercial.

Desde el punto de vista del cuerpo rígido un satélite en órbita cuenta con 6 grados de libertad: 3 de posición y 3 de orientación referente a un sistema de referencia geocéntrico [?, p.196]. De igual manera si consideramos un vehículo volador cuenta con los mismos 6 grados de libertad bajo un sistema de referencia inercial con origen en un punto sobre la superficie terrestre.

Para que un vehículo dentro de la tierra pueda simular las características del movimiento orbital, tenemos que hacer algunas consideraciones respecto al satélite: la velocidad del vehículo varía dependiendo de su posición dentro de la trayectoria orbital, además las condiciones de micro gravedad hacen posible que en el satélite no existan restricciones de movimiento en ningún grado de libertad a menos que la misión lo requiera.

En la Tierra la mayor dificultad que presenta este planteamiento es la de poder realizar movimientos suaves a lo largo del eje Z, debido a que la gravedad afecta al libre movimiento en este grado de libertad. Por lo tanto se propone utilizar un vehículo cuadrotor por su capacidad de vuelo estacionario, que permite emular las condiciones cinemáticas que se presentan durante el recorrido de la trayectoria orbital de un satélite. Además se cuenta con la capacidad de controlar la orientación del vehículo cuadrotor a lo largo de la trayectoria al igual que un satélite lo hace. Para esto será necesario establecer restricciones en dos grados de libertad del cuadrotor (roll y pitch), que para obtener un vuelo estable, se necesitan mantener cercanas a cero.

En las páginas siguientes se muestran los modelos cinemáticos y dinámicos del cuerpo rígido. A continuación se muestra la herramienta matemática utilizada para representar la orientación y algunas formas de parametrización para encontrar la orientación de un cuerpo rígido.

2.1.1. Representación de la orientación

Para poder describir la orientación de un cuerpo rígido, primero se deben establecer dos marcos de referencia ortogonales, uno inercial o fijo en tierra y otro móvil, donde el origen del segundo coincide con el centro de gravedad del vehículo. El objetivo es poder describir la orientación del vehículo o marco de referencia móvil $\{B\}$, en función del marco de referencia fijo en tierra $\{I\}$. En la Figura 2.1 podemos observar un ejemplo de lo anterior.

Partimos del hecho que un cambio en la orientación de un vector puede ser representado mediante una transformación ortogonal, por lo tanto, el objeto matemático utilizado para representar rotaciones de un cuerpo rígido es la '*Matriz de Rotación*'. Este es un operador lineal que cumple con la condición de ortogonalidad, es decir, el determinante vale +1, ya que de ser diferente podría tratarse de una traslación, un escalamiento o una combinación de estas. Por otra parte si el determinante llegara a valer -1 se trataría de una rotación más una inversión de los ejes, lo cual es físicamente imposible para un cuerpo rígido. Las matrices que cumplen con las características para expresar una rotación, definen un grupo llamado especial ortogonal $SO(3)$ [?, p.14]. Para un análisis detallado de las propiedades de la matriz de transformación ver [?, p.167-183].

En la ecuación 2.2 la matriz de rotación R opera una transformación ortogonal sobre un vector v referido a un marco de referencia $\{I\}$ y se obtiene un vector equivalente v' referido a un marco de referencia $\{B\}$.

$$v' = Rv$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.2)$$

Los componentes de la matriz de rotación dependen del método de parametrización que utilicemos, algunos de estos son: Cosenos Directores, Ángulos de Euler, Eje y Ángulo de Giro, Cuaterniones y Parámetros de Cayley-Klein, las ventajas y desventajas de cada uno de estos métodos

se presentan mas adelante.

Matriz de Cosenos Directores

Los cosenos directores proporcionan la forma "natural" de construir la matriz de rotación ya que sus parámetros son las proyecciones de las bases de un sistema de referencia en el otro. Cuando los dos marcos de referencia tengan la misma orientación la matriz de rotación se convierte en la matriz identidad. La matriz de cosenos directores que mapea la posición del vector en el primer marco de referencia al segundo, queda constituida de la siguiente forma:

$$R = \begin{bmatrix} \cos(\theta x'x) & \cos(\theta x'y) & \cos(\theta x'z) \\ \cos(\theta y'x) & \cos(\theta y'y) & \cos(\theta y'z) \\ \cos(\theta z'x) & \cos(\theta z'y) & \cos(\theta z'z) \end{bmatrix} \quad (2.3)$$

Ángulos de Euler

Tomando en cuenta que se puede construir una matriz para representar una rotación alrededor de un eje, se puede lograr obtener cualquier orientación en un espacio ortogonal de tres dimensiones mediante una secuencia de tres rotaciones coordinadas alrededor de tres ejes, en términos prácticos hablamos de la multiplicación de tres matrices de rotación, una por cada base. Existe la restricción que impide elegir el mismo eje dos veces consecutivas, debido a que esto significaría una sola rotación de un cierto ángulo sobre el mismo eje, esto nos da como resultado 12 posibles combinaciones diferentes en el orden de los ejes [?, p.455].

La rotación más intuitiva y fácil de visualizar es la convención $X - Y - Z$ o $1 - 2 - 3$, que consta de un giro de ϕ grados respecto al eje X , seguido de un giro de θ grados respecto al eje Y , finalmente un giro de ψ grados respecto al eje Z . La matriz de rotación para esta convención es:

$$R_{123}(\phi, \theta, \psi) = R_1(\phi)R_2(\theta)R_3(\psi) = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\phi & -S_\theta \\ S_\phi S_\theta C_\psi - C_\phi S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\theta S_\phi \\ C_\phi S_\theta C_\psi + C_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi S_\psi & C_\theta C_\phi \end{bmatrix} \quad (2.4)$$

Para una descripción completa de las convenciones que existen ver [?, p.22].

La representación de rotaciones utilizando ángulos de Euler tiene la desventaja de presentar singularidades para ciertas configuraciones de orientación específicas, ésta se presenta en la convención 123 cuando por ejemplo, el eje X de un marco de referencia está traslapado con el eje Y'

del segundo marco. Para cada convención la singularidad se presenta en orientaciones diferentes, esto es útil cuando el diseño del vehículo restringe naturalmente ciertas posiciones, por ejemplo, un helicóptero generalmente no requiere dar giros completos sobre el eje X o el eje Y .

Cuaterniones

A diferencia de la aeronáutica, en la navegación espacial no existen restricciones en la orientación que puede tener un vehículo. Por este motivo es necesaria una forma de representación que no presente singularidades para ninguna orientación, esto es posible con el uso de los *cuaterniones* que constan de 4 parámetros.

Los cuaterniones son números hipercomplejos que pertenecen a \mathfrak{R}^4 , inventados por Sir William Rowan Hamilton entre 1844 y 1850. Este objeto matemático es una generalización de los números reales y de lo que hoy se conoce como vectores. Estos números tienen la forma $a+bi+cj+dk$, donde a, b, c, d son números reales, las tres unidades imaginarias i, j, k son ortogonales entre si y tienen la restricción [?]:

$$i^2 = j^2 = k^2 = -1 \quad (2.5)$$

Básicamente se trata de un sistema numérico para representar puntos en un espacio en \mathfrak{R}^3 , similar a como los números complejos describen el plano en \mathfrak{R}^2 [?], los cuaterniones se pueden denotar mediante sus partes real y vectorial. Lo anterior implica que podemos tener cuaterniones puramente vectoriales cuando la parte real es cero y por el contrario podemos tener cuaterniones puramente reales haciendo que todas sus componentes vectoriales sean igual a cero. De esta manera podemos observar que tanto los números reales como los vectores pueden ser representados en forma de cuaternión.

Generalmente los cuaterniones suelen escribirse en forma de vector columna de la forma:

$$Q = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (2.6)$$

De la misma forma en que un número complejo puede representar una rotación una rotación en el plano cuando su modulo equivale a la unidad, un cuaternión también puede representar una

rotación en un espacio en \mathbb{R}^3 cuando su modulo es igual a uno, es decir:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (2.7)$$

Tomando en cuenta la ecuación 2.7 y basado en el teorema de Euler que dice que: 'El corrimiento general de un cuerpo rígido con un punto fijo, es una rotación alrededor de un cierto eje', podemos concluir que cualquier sucesión de rotaciones pueden ser representadas por un cuaternión unitario: donde su componente real indica un ángulo de rotación y la componente vectorial indica el eje sobre el cual se esta realizando dicha rotación.

Por ejemplo a la sucesión de rotaciones 3-1-2 le corresponde un cuaternión unitario y éste puede ser parametrizado a partir de los ángulos de Euler de la siguiente manera [?]:

$$Q = \begin{bmatrix} \cos(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\psi}{2}) - \sin(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\psi}{2}) - \sin(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\psi}{2}) + \sin(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\psi}{2}) + \sin(\frac{\phi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\psi}{2}) \end{bmatrix} \quad (2.8)$$

Para poder expresar algebraicamente una rotación con cuaterniones se debe efectuar la operación de multiplicación de cuaterniones que se denota con el simbolo: \otimes . Ésta operación está compuesta por productos escalares y vectoriales de los parámetros de los cuaterniones a operar: uno es el cuaternión (unitario) de rotación y el otro es el vector expresado en forma de cuaternión. La composición del cuaternión resultante se obtiene con la siguiente expresión: [?]

$$Q' \otimes Q = \begin{bmatrix} q'_0 q_0 - \mathbf{q}' \cdot \mathbf{q} \\ q_0 \mathbf{q}' + q'_0 - \mathbf{q}' \times \mathbf{q} \end{bmatrix} \quad (2.9)$$

Los cuaterniones también tienen una forma matricial de representación y se denomina matriz antisimétrica, se trata de una forma de construir una matriz a partir de un vector. El álgebra para tratar cuaterniones se vuelve muy simple cuando se utiliza su equivalente matricial, ya que solo se trata de la multiplicación de matrices por vectores. La matriz antisimétrica de un cuaternión \mathbf{Q} se

denota como $S(\mathbf{Q})$, las componentes de la matriz se obtienen con la siguiente expresión [?, p.15]:.

$$S(\mathbf{Q}) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \quad (2.10)$$

Para realizar la rotación utilizando cuaterniones, tenemos que convertir los vectores a cuaterniones puramente vectoriales y hacer la operación de multiplicación entre ellos. Expresando los cuaterniones en forma vectorial (2.6) o en forma matricial (2.10), la rotación de un vector v con componentes $\{x, y, z\}$ queda determinado por cualquiera de las siguientes expresiones equivalentes:

$$\begin{aligned} \mathbf{V}'_q &= \begin{bmatrix} 0 \\ x' \\ y' \\ z' \end{bmatrix} = \mathbf{Q} \otimes \mathbf{V}_q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ x \\ y \\ z \end{bmatrix} \\ \mathbf{V}'_q &= \begin{bmatrix} 0 \\ x' \\ y' \\ z' \end{bmatrix} = S(\mathbf{Q})\mathbf{V}_q = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ x \\ y \\ z \end{bmatrix} \end{aligned} \quad (2.11)$$

Donde \mathbf{V}_q es el cuaternión puramente vectorial construido a partir del vector v . En caso de requerirlo, es posible construir la matriz de rotación \mathbf{R} a partir de los parámetros de un cuaternión unitario con la siguiente expresión:[?, p.20]

$$\mathbf{R} = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \quad (2.12)$$

Parámetros de Cayley-Klein

Ésta forma de representación cuenta con dos parámetros que son dos números complejos de la forma $x + iy$. La métrica Cayley-Klein fue descubierta entre 1847 y 1928 y se les llama así debido al nombre de sus inventores 'Arthur Cayley y Felix Klein' [?, p.196].

Básicamente se trata de un sistema de cuatro parámetros, dos reales y dos complejos, que

tiene lugar en espacio bidimensional complejo y cuya estructura sirve para facilitar la integración de sistemas giroscópicos complicados. La forma de representación de este objeto matemático es una matriz compleja unitaria de 2×2 cuyos componentes se pueden obtener a partir del cuaternión y se ordenan de la siguiente forma: [?, p.470]

$$\mathbf{H} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \begin{bmatrix} q_0 + iq_3 & q_2 + iq_1 \\ -q_2 + iq_1 & q_0 - iq_3 \end{bmatrix} \quad (2.13)$$

Por otro lado, si queremos expresar un vector v que se encuentra en un espacio \mathfrak{R}^3 bajo esta forma de parametrización, construimos su matriz hermítica de la siguiente forma: [?, p.191]

$$\mathbf{V}_H = \begin{bmatrix} z & x - iy \\ x + iy & -z \end{bmatrix} \quad (2.14)$$

La operación que se realiza para poder rotar un vector expresado en forma de matriz hermítica, es una transformación de semejanza. Partiendo de lo anterior decimos que la rotación de un vector v utilizando los parámetros Cayley-Klein queda determinada por la siguiente expresión:

$$\mathbf{V}'_H = \mathbf{H}\mathbf{V}_H\mathbf{H}^\dagger \quad (2.15)$$

Donde \mathbf{H}^\dagger es la matriz adjunta de \mathbf{H} y por ser unitaria es igual a su inversa.

También podemos obtener los parámetros Cayley-Klein a partir de los ángulos de Euler con la siguiente expresión:

$$\begin{aligned} \alpha &= \exp^{i(\psi+\phi)/2} \cos\left(\frac{\theta}{2}\right) \\ \beta &= i \exp^{i(\psi-\phi)/2} \sin\left(\frac{\theta}{2}\right) \\ \gamma &= i \exp^{-i(\psi-\phi)/2} \sin\left(\frac{\theta}{2}\right) \\ \delta &= \exp^{-i(\psi+\phi)/2} \cos\left(\frac{\theta}{2}\right) \end{aligned} \quad (2.16)$$

Por el contrario si deseamos obtener la matriz de rotación en función de los parámetros

Cayley-Klein lo hacemos con la siguiente expresión:

$$\mathbf{R} = \begin{bmatrix} \frac{1}{2}(\alpha^2 - \gamma^2 + \delta^2 - \beta^2) & \frac{i}{2}(\gamma^2 - \alpha^2 + \delta^2 - \beta^2) & \gamma\delta - \alpha\beta \\ \frac{1}{2}(\alpha^2 + \gamma^2 - \beta^2 - \delta^2) & \frac{i}{2}(\alpha^2 + \gamma^2 + \beta^2 + \delta^2) & -i(\alpha\beta + \gamma\delta) \\ \beta\delta - \alpha\gamma & i(\alpha\gamma + \beta\delta) & \alpha\gamma + \beta\delta \end{bmatrix} \quad (2.17)$$

2.1.2. Ecuaciones de movimiento del cuerpo rígido

Cinemática del Cuerpo Rígido

La descripción del movimiento de un cuerpo rígido se puede realizar tomando en cuenta un vector (ver ecuación 2.1) con origen en el marco de referencia inercial $\{I\}$ que va unido al origen del marco de referencia del cuerpo $\{B\}$, donde establecemos un vector G .

Cuando el cuerpo se mueva, usualmente el vector cambiará sus componentes, dependiendo desde donde se analice el movimiento. Un observador que se encuentre unido a un marco de referencia inercial observará que las componentes del vector cambian con la traslación y con la rotación del cuerpo. En cambio, un observador cuyo marco de referencia coincida con el marco de referencia unido al cuerpo solo notará cambios en las componentes del vector si el cuerpo realiza movimiento rotacional.

La variación en el tiempo del vector G debido a la traslación es simplemente la derivada de sus componentes $[x, y, z]$ bajo el marco de referencia inercial es decir su velocidad de traslación:

$$\mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (2.18)$$

Si tomamos en cuenta que la orientación de un vector puede estar determinada por una matriz de rotación, el análisis del movimiento rotacional se consigue encontrando la derivada de la matriz de rotación. Para esto es necesario basarnos en el álgebra de Lie, que trata de las derivadas matrices del grupo Especial Ortogonal en R^3 ($\mathbf{SO}(3)$), el cual dice que un punto $\mathbf{p}(t)$ sobre el cuerpo rígido que gira alrededor de un eje \mathbf{a} tendrá una variación en el tiempo equivalente a $[\mathbf{a},$

p.15]:

$$\frac{d}{dt}\mathbf{p}(t) = \mathbf{a} \times \mathbf{p}(t) = \mathbf{A}\mathbf{p}(t) \quad (2.19)$$

Tomando en cuenta la ecuación anterior (2.21) y considerando que el vector de velocidades angulares define el eje de rotación instantáneo, decimos que el cambio en las componentes debido al movimiento rotacional es:

$$\begin{aligned} \frac{d}{dt}e_1^{bi}(t) &= S(\omega_I(t))e_1^{bi}(t) \\ \frac{d}{dt}e_2^{bi}(t) &= S(\omega_I(t))e_2^{bi}(t) \\ \frac{d}{dt}e_3^{bi}(t) &= S(\omega_I(t))e_3^{bi}(t) \end{aligned} \quad (2.20)$$

Donde $e_i^{bi}(t)$ representa cada componente del vector G referido al marco de referencia inercial $\{I\}$. El término $S(\omega_I(t))$ es la matriz anti-simétrica del vector de velocidades angulares $\omega_I(t)$ referido igualmente al marco de referencia inercial. La matriz antisimétrica del vector de velocidades angulares $\omega_I(t) \in R^3$ con componentes $\{x_x, x_y, x_z\}$ está dado por:

$$S(x) = \begin{bmatrix} 0 & -x_z & x_y \\ x_z & 0 & -x_x \\ -x_y & x_x & 0 \end{bmatrix} \quad (2.21)$$

A partir de los términos en las ecuaciones (2.20), podemos decir que la variación de la matriz de rotación $R(t)$ se puede encontrar con la siguiente expresión [?, p.23]

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)S(\omega(t)) \quad (2.22)$$

De esta manera solo tenemos que multiplicar al vector de orientación por la nueva matriz de rotación. Así que para describir el movimiento de un cuerpo rígido sin considerar las fuerzas que lo producen utilizamos las ecuaciones (2.18) y (2.22).

Dinámica del Cuerpo Rígido

Las fuerzas que generan el movimiento de un cuerpo rígido pueden ser encontradas mediante la aplicación de la 2da. ley de Newton. Simplificando el problema consideramos un cuerpo rígido con toda su masa concentrada en el centro de gravedad, la expresión para la dinámica del movimiento de traslación es:

$$m\dot{\mathbf{v}} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (2.23)$$

Para el análisis del movimiento rotacional debemos encontrar una expresión similar a la ecuación 2.23. Comenzamos considerando en que el vector de velocidad angular $\omega(t)$ y el vector momento cinético $\mathbf{L}(t)$, están relacionados mediante un transformación lineal [?, p.243]

$$\mathbf{L}(t) = \mathbf{R}(t)\mathbf{I}\omega(t) \quad (2.24)$$

Donde $R(t)$ es la matriz de rotación, el operador matricial \mathbf{I} se denomina matriz de inercia y es el equivalente a la masa m en la expresión (2.23), así como la velocidad angular ω lo es a la velocidad \mathbf{v} . Para encontrar el término equivalente a la fuerza que se define con el producto $m\dot{\mathbf{v}}$, vamos a encontrar la derivada en el tiempo de la expresión (2.24) para encontrar la aceleración angular. Tomando en cuenta lo anterior, aplicamos la 2da. ley de Newton y obtenemos:

$$\frac{d}{dt}\mathbf{L}(t) = \mathbf{R}(t)(\tau(t) + \tau_{ext}(t)) \quad (2.25)$$

Sustituyendo la ecuación (2.24) en (2.25) y aplicando derivada, llegamos a la expresión para el torque:

$$\mathbf{I}\dot{\omega}(t) = -S(\omega(t))\mathbf{I}\omega(t) + \tau(t) \quad (2.26)$$

Esta expresión es el equivalente rotacional de la fuerza $\mathbf{F} = m\mathbf{a}$, los torques externos son despreciados.

El modelo dinámico para describir el movimiento del cuerpo rígido contemplando las fuerzas que lo producen, está compuesto por la ecuación (2.23) para la traslación y la expresión (2.26) para la dinámica rotacional.

2.2. Cuadrotor

2.2.1. Modelo Matemático de un cuadrotor

Debido a la forma de como está constituido el vehículo necesitamos tomar en cuenta ciertas restricciones en cuanto al movimiento se refiere. Para producir un desplazamiento en el eje X o en el eje Y necesitamos producir un cambio en los ángulos de orientación Pitch y Roll respectivamente, para un movimiento en el eje Z se requiere aumentar la velocidad angular de todos los motores en la misma proporción. Esto quiere decir que se trata de un sistema sub-actuado, donde se controlan 6 grados de libertad con solo cuatro actuadores.

El modelo cinemático de un cuadrotor se obtiene a partir de las ecuaciones cinemáticas del cuerpo rígido. Comenzamos estableciendo el marco de referencia inercial $\{I\}$ con sus componentes $\{x, y, z\}$ y el marco de referencia unido al centro de gravedad del cuadrotor $\{B\}$ con componentes $\{e_1, e_2, e_3\}$, con e_3 en dirección contraria al eje z .

Las principales fuerzas que actúan sobre el cuadrotor son la fuerza de gravedad y el empuje producido por los motores, ambas fuerzas van en sentido contrario una con la otra. Utilizamos la ecuación (2.23) para obtener la expresión que describe la dinámica de traslación del cuadrotor:

$$m\dot{\mathbf{v}} = mg\mathbf{e}_3^i - \mathbf{T}(t)\mathbf{R}(t)\mathbf{e}_3^i \quad (2.27)$$

Donde m representa la masa y g la gravedad. \mathbf{e}_3^i es el eje \mathbf{e}_3 visto o transformado al marco de referencia inercial $\{I\}$. \mathbf{T} es el vector de empuje que solo tiene componente el en eje \mathbf{e}_3 . $\mathbf{R}(t)$ es la matriz de transformación.

La dinámica rotacional se encuentra dada por la ecuación (2.26), donde el torque $\tau(t)$ es producido por la rotación de los motores. De esta manera el modelo dinámico completo de un cuadrotor queda determinado con el siguiente conjunto de expresiones:

$$\mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)S(\omega(t)) \quad (2.28)$$

$$m\dot{\mathbf{v}} = mg\mathbf{e}_3^i - \mathbf{T}(t)\mathbf{R}(t)\mathbf{e}_3^i$$

$$\mathbf{I}\dot{\omega}(t) = -S(\omega(t))\mathbf{I}\omega(t) + \tau(t)$$

El empuje $\mathbf{T}(t)$ y el torque $\tau(t)$ están determinados en función de la velocidad de giro de cada uno de los motores y una constante dependiente de la hélice y la densidad del aire. El empuje total generado esta dado por:

$$\mathbf{T}(t) = |\mathbf{T}_{mot1}(t)| + |\mathbf{T}_{mot2}(t)| + |\mathbf{T}_{mot3}(t)| + |\mathbf{T}_{mot4}(t)| = K_T \sum \omega_{moti}^2(t) \quad (2.29)$$

Así mismo el torque producido depende de la velocidad de giro de los motores y de una constante de arrastre aerodinámico dada igualmente por la hélice:

$$\tau(t) = K_\tau \sum \omega_{mot}^2(t) \quad (2.30)$$

Las expresiones (2.29) y (2.30) se pueden combinar en una sola expresión matricial. Consideramos que el torque es igualmente proporcional a la distancia d del motor al centro de gravedad:

$$\begin{bmatrix} \mathbf{T}(t) \\ \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -dK_T & dK_T & dK_T & -dK_T \\ dK_T & -dK_T & dK_T & -dK_T \\ -K_\tau & -K_\tau & K_\tau & K_\tau \end{bmatrix} \begin{bmatrix} \omega_{mot1}^2(t) \\ \omega_{mot2}^2(t) \\ \omega_{mot3}^2(t) \\ \omega_{mot4}^2(t) \end{bmatrix} \quad (2.31)$$

2.2.2. Técnicas de Control de Cuadrotoros

EL sistema de control de un cuadrotor ha sido ampliamente estudiado en la literatura, se han analizado diversas técnicas de control óptimo, PID, control difuso, LQR, entre otros [?], [?],

[?], [?]. Tomando en cuenta la consideración de que la complejidad de un sistema limita nuestra capacidad para realizar afirmaciones sobre él, los controladores dependientes del modelo requieren mucho más capacidad de procesamiento y memoria para poder ejecutarse en tiempo real.

En el trabajo [?] se presenta una comparativa del desempeño de controladores LQ vs PID y se muestra que el control LQ no es capaz de realizar un vuelo autónomo a diferencia del control con PID. Este hecho justifica las técnicas de control utilizada en cuadrotores comerciales de bajo costo, como el utilizado para este trabajo. Por lo tanto podemos concluir que las técnicas de control independientes del modelo presentan ventajas ante la resolución de problemas de control no lineales cuando se cuenta con poco recursos computacionales.

Los controladores PID tienen la principal desventaja de que es muy complicada la sintonización de las ganancias ya que el sistema, por ser no lineal, es inestable. En el trabajo de [?] se presenta el control difuso como alternativa ante esta problemática, los experimentos fueron hechos para probar si ciertas tarjetas controladoras tendrían la capacidad de ejecutar un vuelo autónomo con dichos controladores. Otra ventaja es que estos controladores difusos tienen tolerancia con las imprecisiones provenientes del modelo o de la lectura de los sensores.

Cuando se desea la implementación de un algoritmo de control en este tipo de vehículos, existen alternativas dependientes del fabricante de la tarjeta. Algunos de los sistemas de cuadrotores comerciales cuentan con un 'software development kit, SDK' por sus siglas en ingles, que contiene las herramientas necesarias para acceder y modificar ciertas partes del del sistema de vuelo [ver figura 1.6]. En este trabajo accederemos al sistema de orientación del vehículo mediante Matlab-Simulink.

Control Difuso

Aplicar control a sistemas descritos matemáticamente con ecuaciones diferenciales, tiene complicaciones cuando los sistemas son complejos o no se tiene un buen grado de conocimiento de ellos. Debido a esto se desarrollaron estrategias para establecer reglas de control a partir de implicaciones difusas por ejemplo [?] o [?]. La evolución de estas estrategias permitió crear sistemas capaces de modelar un sistema matemáticamente, de manera cada vez mas precisa y menos compleja.

La manera de describir un sistema de manera difusa se realiza mediante implicaciones difusas R , esto quiere decir que se utilizan variables difusas de entrada x que son asociadas a un variable lingüística perteneciente a un conjunto difuso $\mathbf{A}(x)$ mediante una función de pertenencia \mathbf{A} . Estos

valores lingüísticos son operados para formar premisas del tipo:

$$R: \text{ Si } f(x_1 \text{ es } A_1, \dots, x_k \text{ es } A_k) \text{ entonces } y = g(x_1, \dots, x_k) \quad (2.32)$$

Cada premisa difusa es una regla local de funcionamiento del sistema y genera un valor de salida y . Estos valores son operados de para establecer una salida de control al sistema, por ejemplo la función de salida utilizada en [?, p.117] es:

$$y = \frac{\sum (w^i \times y^i)}{\sum w^i} \quad (2.33)$$

Donde y^i es la salida producida por cada implicación difusa, w^i son las ponderaciones o pesos dados a cada implicación difusa \mathbf{R}^i .

ANFIS

El desarrollo de los métodos de descripción mencionados en la sección anterior dio paso a la inteligencia artificial y al desarrollo y aplicación de las redes neuronales para modelar sistemas de manera automática y formal como lo hace 'Adaptative Neuro-fuzzy Inference System, ANFIS' por sus siglas en inglés [?].

Esta herramienta nos permite obtener una ley de control difuso a partir de datos de entrada-salida de un sistema dado, se basa en una estructura llamada red neuronal generalizada multicapa (GNN, por sus siglas en inglés) [ver figura 2.2]:



Figura 2.2: Red Neuronal Generalizada [GNN][?, p.1].

Donde cada nodo puede realizar una función específica que esta determinada por ciertos parámetros, si no existen parámetros en el nodo, este se denota por un círculo. La particularidad principal de la GNN es que cada nodo puede ser una red neuronal individual de varias capas, que a su vez tiene la capacidad de aproximar cualquier función continua en un intervalo pequeño, a partir de la superposición de una función sigmoideal [?].

El sistema ANFIS funciona mediante datos de entrada y salida de un sistema. A partir de los datos de entrada y una configuración inicial de parámetros de la GNN, se calcula una salida y los parámetros de ésta son ajustados para disminuir el error entre el valor obtenido y el valor deseado. La convergencia del sistema se acelera mediante la aplicación del algoritmo del gradiente descendiente y el filtro Kalman [?].

2.2.3. Componentes Físicos y Electrónicos

Físicamente los cuadrotores constan de una estructura simétrica, usualmente en forma de 'X', con una helice fija a un motor en cada esquina. La tarjeta controladora se coloca en el centro de la estructura con la finalidad de que los sensores se encuentren lo más cerca posible del centro de gravedad del vehículo. La batería se coloca al centro y por debajo de la tarjeta. A partir de la estructura y componentes antes mencionados se pueden implementar herramientas y sensores como cámaras o brazos robóticos que permitan realizar tareas más complejas como podemos ver en [?].

Los motores utilizados en este tipo de vehículos son de corriente directa y pueden alcanzar grandes velocidades debido a que no cuentan con escobillas en el rotor. Las hélices pueden variar en tamaño según la necesidad del vehículo y la carga a soportar.

El componente electrónico principal es la tarjeta controladora que consta de un microcontrolador con sensores embebidos como acelerómetros, giróscopos, brújula, GPS, sonar, entre otros. Esto es crucial para garantizar un nivel suficiente de autonomía del vehículo. Para controlar la velocidad de los motores se utilizan señales PWM provenientes del controlador programado en la de la tarjeta de control y para la etapa de potencia se utilizan dispositivos llamados *Electronic Speed Controller (ESC)*, por sus siglas en inglés.

2.3. Trayectorias Orbitales

Gracias a las observaciones de Tycho Brae y los cálculos de Johanes Kepler, hoy sabemos que los planetas describen trayectorias elípticas alrededor del sol, que se encuentra colocado en uno de los focos de la elipse. Lo mismo sucede con la luna orbitando alrededor de la tierra o los satélites artificiales puestos en órbita por el hombre. Todo estos fenómenos obedecen a la ley de gravitación universal, que establece que dos objetos que se encuentran a una cierta distancia entre si, experimentarán una fuerza central mutua que será proporcional a su masa e inversamente

proporcional a cuadrado de la distancia que las separa, es decir:

$$\mathbf{F} = G \frac{m_1 m_2}{\mathbf{r}^2} \quad (2.34)$$

Donde \mathbf{F} es la fuerza que experimentan los cuerpos, m_1 y m_2 son las masa, \mathbf{r} es la distancia entre ambos cuerpos y G es la constante de gravitación universal [?, p.16]. Al punto donde se ejerce la fuerza se le llama centro de gravedad del sistema y se encuentra sobre la recta que une los dos centros de masa de los objetos.

Considerando un marco de referencia con origen en el centro de masa de uno de los cuerpos, el punto donde se encuentra el centro de gravedad esta determinado por:

$$G_0 = \frac{m_1 d_{12} + m_2 d_{12}}{m_1 + m_2} \quad (2.35)$$

Donde d_{12} indica la distancia del primer cuerpo al segundo. Las ecuaciones para describir el movimiento provocado por la fuerza de gravedad se presentan en la siguiente sección.

2.3.1. Problema de los dos cuerpos

Sabemos que la descripción del movimiento depende desde donde se observe, por lo tanto, las ecuaciones dinámicas para el problema de los dos cuerpos dependen del marco de referencia que tomemos en cuenta. Bajo un marco de referencia inercial [ver figura 2.3] la descripción se realiza mediante la posición del centro de gravedad del sistema, denotado por G_0 , las ecuaciones de posición, velocidad y aceleración son:

$$\mathbf{R}_{G_0} = \frac{m_1 \mathbf{R}_1 + m_2 \mathbf{R}_2}{m_1 + m_2} \quad (2.36)$$

$$\mathbf{v}_{G_0} = \dot{\mathbf{R}}_{G_0} = \frac{m_1 \dot{\mathbf{R}}_1 + m_2 \dot{\mathbf{R}}_2}{m_1 + m_2} \quad (2.37)$$

$$\mathbf{a}_{G_0} = \ddot{\mathbf{R}}_{G_0} = \frac{m_1 \ddot{\mathbf{R}}_1 + m_2 \ddot{\mathbf{R}}_2}{m_1 + m_2} \quad (2.38)$$

Donde \mathbf{R}_{G_0} es el vector de posición del centro de gravedad y \mathbf{R}_1 y \mathbf{R}_2 son los vectores de posición de los cuerpos bajo un marco de referencia inercial. Para calcular la fuerza que ejerce un

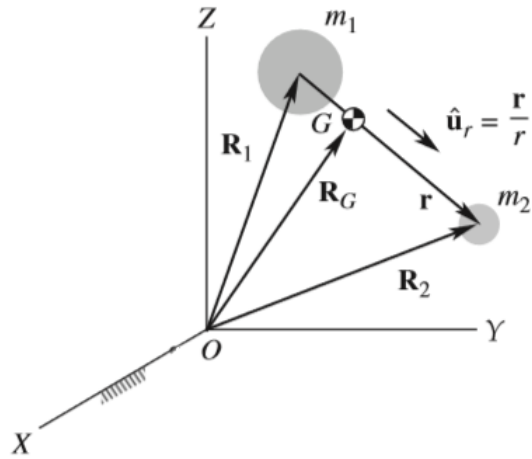


Figura 2.3: 2 cuerpos analizados bajo un marco de referencia inercial [?, p.60].

cuerpo sobre otro, necesitamos obtener un vector que indique la posición de un cuerpo respecto al otro [ver figura 2.3]:

$$\mathbf{r} = \mathbf{R}_2 - \mathbf{R}_1 \quad (2.39)$$

Como sabemos, \mathbf{r} define la dirección donde se aplica la fuerza mutua entre los cuerpos, por lo tanto la fuerza que ejerce un cuerpo sobre el otro se obtiene a partir de la ecuación 2.34, dando paso a las siguientes expresiones:

$$\mathbf{F}_{12} = \frac{Gm_1m_2}{r^2} \mathbf{u}_r \quad (2.40)$$

$$\mathbf{F}_{21} = -\frac{Gm_1m_2}{r^2} \mathbf{u}_r \quad (2.41)$$

Donde \mathbf{u}_r es el vector unitario que determina la dirección de la fuerza entre ambos cuerpos y esta dada por [ver figura 2.3]:

$$\mathbf{u}_r = \frac{\mathbf{r}}{r} \quad (2.42)$$

Contemplando que:

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (2.43)$$

A partir de las ecuaciones 2.40, 2.41, 2.43 podemos establecer las ecuaciones de movimiento para cada uno de los cuerpos:

$$\ddot{\mathbf{R}}_1 = Gm_2 \frac{\mathbf{r}}{r^3} \quad (2.44)$$

$$\ddot{\mathbf{R}}_2 = -Gm_1 \frac{\mathbf{r}}{r^3} \quad (2.45)$$

Para simplificar las ecuaciones utilizamos un marco de referencia situado en uno de los cuerpos, generalmente el de mayor masa. Bajo este nuevo marco de referencia r se convierte en el vector de posición relativa, por lo tanto podemos derivar de la ecuación 2.39 para obtener la posición, velocidad y aceleración relativa entre los cuerpos y se calculan mediante las siguientes expresiones:

$$\ddot{\mathbf{r}} = \ddot{\mathbf{R}}_2 - \ddot{\mathbf{R}}_1 \quad (2.46)$$

$$\ddot{\mathbf{r}} = -\frac{G(m_1 + m_2)}{r^2} \mathbf{u}_r \quad (2.47)$$

Donde el término $G(m_1 + m_2)$ se le llama parámetro gravitacional μ :

$$\mu = G(m_1 + m_2) \quad (2.48)$$

Combinando las expresiones 2.47 y 2.48 tenemos la ecuación de movimiento relativo a uno de los cuerpos:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \quad (2.49)$$

2.3.2. Órbitas y algunas de sus propiedades

En el problema de los dos cuerpos la trayectoria de uno visto desde el otro, es una sección cónica cuya forma esta determinada principalmente por la excentricidad, a esta trayectoria se le llama órbita. Básicamente la excentricidad de una curva nos dice que tanto se aleja de ser un círculo, es decir, la excentricidad para un círculo es $e = 0$, para una elipse $0 < e < 1$, para una parábola $e = 1$ y $e < 1$ para hipérbolas.

Cuando hablamos de órbitas es conveniente conocer algunos puntos y líneas características que son útiles para su descripción, en la figura 2.4 podemos observarlos dentro una órbita elíptica. Los focos estan denotados por F y F' , el objeto central o de mayor masa se encuentra colocado en F . El periapsis P es el punto de la trayectoria mas cercano al cuerpo central, el punto mas alejado se llama apoapsis A . La línea que une al apoapsis, al periapsis y que pasa por el foco de la órbita, es el eje de los apses. La distancia denotada por a es el semi-eje mayor, b el semi-eje menor. La posición de un objeto se caracteriza mediante el ángulo θ que se forma entre la línea de los apses y el vector de posición del objeto. r_A y r_B son los radios del apoapsis y de periapsis respectivamente.

Estos datos están referidos a un marco de referencia perifocal que tiene origen en el foco F y como variable de posición el ángulo θ , que es igual a 0 cuando el objeto orbitante se encuentra en el periapsis.

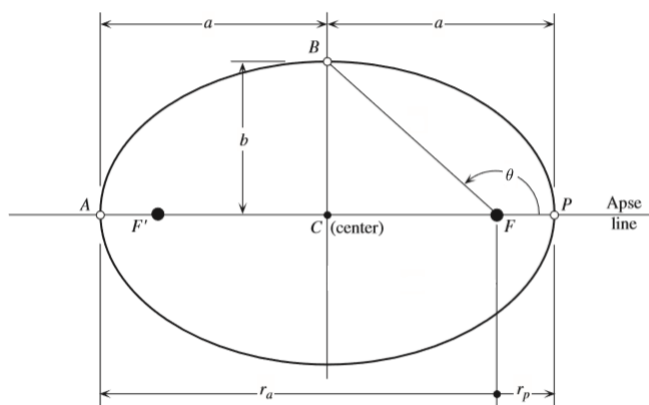


Figura 2.4: Órbita elíptica [?, p.87].

Para analizar el movimiento desde uno de los cuerpos utilizamos la ley de conservación del momento angular. El momento angular del segundo cuerpo relativo al primero, o el de mayor masa,

es el producto vectorial entre su posición y velocidad:

$$\mathbf{h} = \mathbf{r} \times \dot{\mathbf{r}} \quad (2.50)$$

Donde \mathbf{r} esta determinado por la ecuación 2.39. La ley de conservación de momento angular establece que la cantidad en la expresión 2.50 es constante en el tiempo:

$$\frac{d\mathbf{h}}{dt} = 0 \quad (2.51)$$

Para que esto sea posible se requiere que la velocidad y la posición varíen durante la trayectoria o como lo establece la primera ley de Kepler: el vector de posición de un planeta al sol, barre áreas iguales en tiempos iguales. Así mismo de la ecuación 2.50 podemos decir que la órbita se encuentra en un plano perpendicular al vector de momento angular \mathbf{h} .

La expresión que utilizamos para determinar la posición de un objeto utilizando el parámetro gravitacional μ , el momento angular \mathbf{h} , la excentricidad e y la anomalía verdadera θ , se denomina ecuación de la órbita y se define con la siguiente expresión [?, p.77]

$$\mathbf{r} = \frac{h^2}{\mu} \frac{1}{1 + e \cos \theta} \quad (2.52)$$

Donde $h^2 = \mathbf{h} \cdot \mathbf{h}$. Para conocer el periodo de una órbita circular o elíptica, utilizamos cualquiera de las siguientes expresiones equivalentes [?, p.89]:

$$T = \frac{2\pi}{\sqrt{\mu}} a^{3/2} \quad (2.53)$$

$$T = \frac{2\pi}{\mu^2} \left(\frac{h}{\sqrt{1 - e^2}} \right)^3 \quad (2.54)$$

Un conjunto de ecuaciones funcionales para cada tipo de órbita (circular, elipse, parábola, hipérbola) definidas en un plano, bajo un marco de referencia perifocal se encuentra en [?, p.104-106]

2.3.3. Elementos Orbitales

Aunque el análisis del movimiento orbital se puede realizar en un plano, es conveniente referir las órbitas a un marco de referencia tridimensional inercial con origen en el cuerpo de mayor masa.

En general la forma de una órbita en un plano puede estar dada por dos parámetros: el momento angular \mathbf{h} y la excentricidad e . La posición dentro esta órbita requiere de un parámetro adicional, la anomalía verdadera θ . Solo se requieren otros 3 parámetros de que definan la orientación de la órbita respecto a un marco de referencia elegido.

Para órbitas alrededor de la tierra podemos considerar un marco de referencia geocentrico ecuatorial que esta fijo respecto al equinoccio vernal. En la figura 2.5 podemos observar cada uno de estos elementos orbitales.

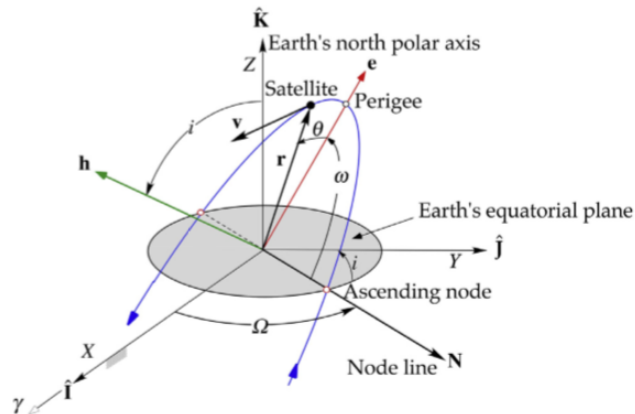


Figura 2.5: Elementos orbitales bajo un marco de referencia geocentrico ecuatorial [?, p.87].

El primero de los 3 elementos restantes se trata de la ascensión recta del nodo ascendente, que es el punto donde la órbita pasa a la parte superior del ecuador. Este punto se encuentra sobre la línea de nodos definida por la intersección del plano ecuatorial con el plano de la órbita. En otros términos es una rotación alrededor del eje Z .

El siguiente elemento orbital es la inclinación i , determinada por el ángulo de inclinación entre el plano ecuatorial y el plano de la órbita o el ángulo entre el vector de momento angular y el eje Z . Esta es una rotación alrededor del eje X .

Finalmente tenemos el argumento de perigeo de la órbita, el cual se define como el ángulo entre el plano ecuatorial y vector que une al origen con el perigeo, nuevamente se trata de una rotación alrededor del eje Z .

En resumen a partir de los elementos orbitales podemos obtener una configuración única de órbita y posición dentro de la misma. Debido a las equivalencias entre el momento angular y el semi-eje mayor, generalmente los elementos orbitales de un satélite están dado por cualquiera de

los siguientes conjuntos:

$$\{h, i, \Omega, e, \omega, \theta\} \quad (2.55)$$

$$\{a, i, \Omega, e, \omega, \theta\} \quad (2.56)$$

Como se observa en las expresiones 2.55 y 2.56 la descripción completa del movimiento se realiza mediante 6 parámetros, es decir, estamos describiendo el movimiento de un cuerpo rígido en un espacio tridimensional.

Capítulo 3

Sistema del Simulador

El simulador de trayectoria orbital desarrollado en este trabajo consta de una interfaz gráfica de usuario que transforma la trayectoria orbital de un satélite a una trayectoria escalada bajo un marco de referencia local, para ser reproducida por un vehículo cuadrotor.

En este capítulo se muestra la estructura general del sistema planteado así como cada uno de los pasos a seguir para lograr el funcionamiento del simulador de trayectoria orbital. Como primer punto, se presenta el proceso para generar los datos de la trayectoria orbital a partir de los elementos orbitales de un satélite, después se muestra la interfaz gráfica de usuario desarrollada para integrar los diversos subsistemas de los que se compone el simulador. Para finalizar se realizan algunas consideraciones acerca de la capacidad de modificar características físicas del vehículo y las cargas útiles en el sistema propuesto.

Finalmente se calculan los elementos orbitales para la trayectoria simulada resultante.

3.1. Descripción general del sistema

El funcionamiento general del sistema se compone de dos partes principales: la interfaz de control con los elementos necesarios para su funcionamiento y el vehículo comercial, real o virtual, con su sistema de navegación de fábrica. En la figura 3.1 podemos observar un diagrama a bloques del planteamiento del sistema:

Donde \mathbf{R} y \mathbf{V} son los vectores de posición y velocidad del satélite, los parámetros de la simulación son la duración del ciclo simulado y la cantidad de ciclos a realizar. Respecto a la parte

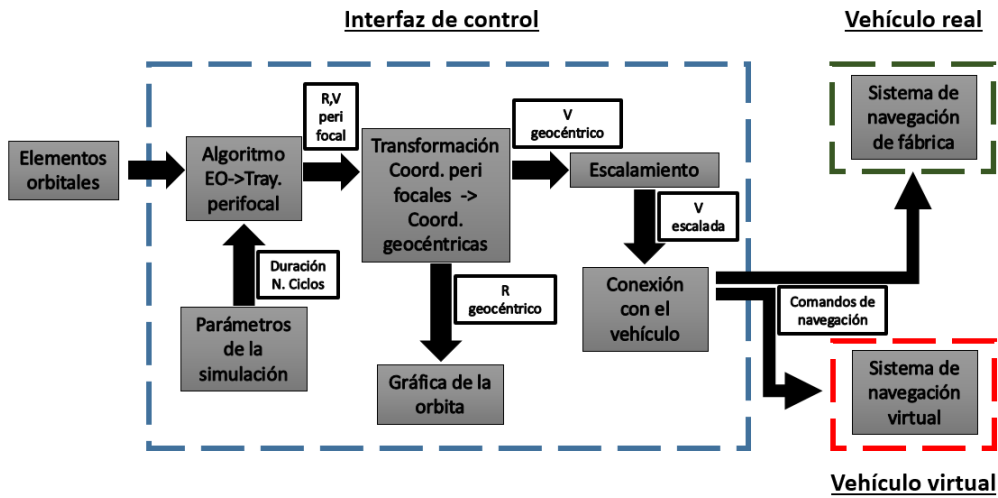


Figura 3.1: Diagrama del funcionamiento general del sistema propuesto, donde R y V son los vectores de posición y velocidad del satélite respectivamente.

de los vehículos, el sistema de navegación registra los datos de sensores y actuadores siempre que se realiza un vuelo, por lo tanto, obtendremos información del desempeño de la simulación orbital analizando dichos datos.

3.2. Interfaz Gráfica de Usuario

La interfaz gráfica de usuario para el control de la simulación de trayectoria orbital se desarrolló en 'python' por ser un software de código libre y se utilizan las librerías de soporte 'dronekit' que implementan los protocolos de comunicación 'MAVLINK' para llevar a cabo la comunicación con el vehículo. A continuación se muestra una imagen de la interfaz siendo ejecutada:

La interfaz consta de 3 partes principales, en la figura 3.2 en la parte derecha se encuentra la sección para la introducción de los elementos orbitales de un satélite deseado y un botón para comenzar el calculo de la trayectoria con el algoritmo descrito en la sección anterior, la forma de la órbita resultante así como su orientación con respecto a la tierra pueden ser apreciadas en la ventana central.

En la parte superior central se encuentra el control para la duración del periodo en segundos de la órbita simulada y por último en la parte izquierda podemos observar los controles de la conexión con el vehículo, para poder ejecutar las tareas de despegue, ejecución de la trayectoria y aterrizaje. Adicionalmente se muestra la información del espacio requerido para llevar a cabo la

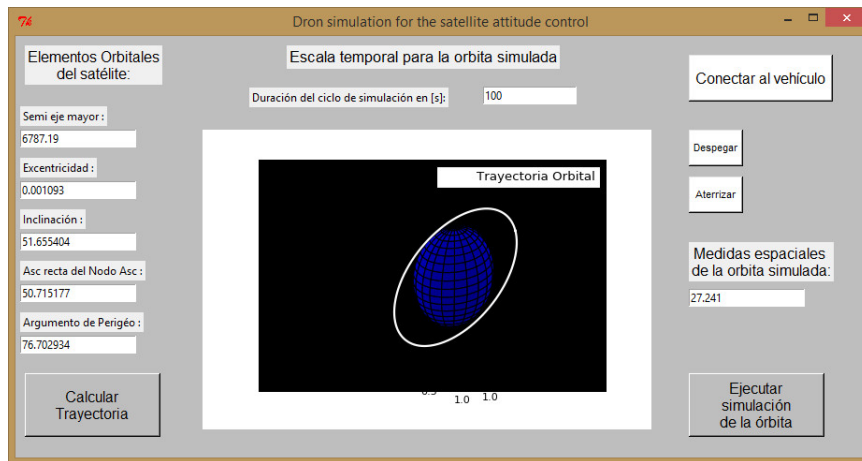


Figura 3.2: Interfaz gráfica de usuario para el control del simulador de trayectoria orbital.

trayectoria simulada con el vehículo cuadrotor.

Los datos generados por el algoritmo se envían en forma de velocidades para cada eje de un marco de referencia inercial, los datos son enviados cada segundo durante la simulación.

3.2.1. Especificaciones técnicas del desarrollo de la interfaz

- Sistema operativo de desarrollo y función: Windows 8.1 64 bits - Linux/Ubuntu 14.04 64 bits
- Lenguaje de programación: Python 2.7 [?]
- Interfaz Gráfica de Desarrollo(IDE): Spyder 3.0.0. [?]
- Detalles de librería de gráficos utilizada: Tkinter [?]

3.3. Elementos orbitales a trayectoria orbital escalada

Como se mencionó en secciones anteriores para la descripción del movimiento de un cuerpo rígido se necesitan 6 parámetros que definan su estado por completo. Cuando se trata de la descripción de trayectorias orbitales, es posible realizar su descripción mediante los elementos orbitales que definen la forma y orientación de una órbita, así como la posición del objeto en forma única.

En esta sección se describe el sistema encargado de generar una trayectoria local a partir de una trayectoria orbital real, esto se realiza mediante el muestreo de información de la posición y

velocidad a lo largo de la órbita real referida a un marco de referencia perifocal. La información de la posición y velocidad puede ser obtenida muestreando la trayectoria de dos formas distintas: mediante el muestreo sobre el rango de la variable de la anomalía verdadera o mediante el periodo.

En este trabajo se eligió realizar el muestreo de la órbita sobre el rango del periodo, ya que de hacerlo sobre el rango de la anomalía verdadera, se obtendría un sobre-muestreo en la parte del perigeo y un sub-muestreo en la parte del apogeo ya que la relación de las distancias entre estos dos puntos es muy grande, por lo tanto su proyección en la órbita no es uniforme.

El algoritmo para la obtención de la trayectoria orbital de un satélite a partir de sus elementos orbitales se encuentra descrito en la siguiente figura:

Primero partimos de 5 de 6 los elementos orbitales del satélite: semi-eje mayor, excentricidad, inclinación, ascensión recta del nodo ascendente y argumento de perigeo $[a, e, i, \Omega, \omega]$. Se omite la introducción de la anomalía verdadera ya que esta variable denota la posición exacta del satélite y debido a que se requiere conocer la trayectoria completa, esta variable sera utilizada en todo su rango. Las órbitas son consideradas alrededor de la tierra por lo tanto la constante gravitacional es: $\mu = 398600$.

A partir de lo parámetros antes mencionados podemos calcular el periodo de la órbita resultante con la siguiente expresión:

$$T = \frac{2\pi}{\mu^2} \left(\frac{h}{\sqrt{1-e^2}} \right)^3 \quad (3.1)$$

A continuación calculamos el paso de muestreo, dividiendo el periodo entre el tiempo en segundos seleccionado para la duración de la simulación y con esta información generamos un vector con los tiempos de muestreo de la órbita real. Ahora calculamos la anomalía verdadera en función de los tiempos obtenidos en el paso anterior, comenzamos calculando la anomalía media Me para cada instante de tiempo:

$$Me = 2\pi \frac{t}{T} \quad (3.2)$$

La anomalía media se define como la posición en radianes de un objeto ficticio moviéndose a un velocidad promedio constante. Con este valor procedemos a calcular la anomalía excéntrica, es un parámetro auxiliar que denota una posición angular considerando la trayectoria sobre un circulo concentrico con la órbita y de radio igual al semi-eje mayor de la misma. Este valor de obtiene

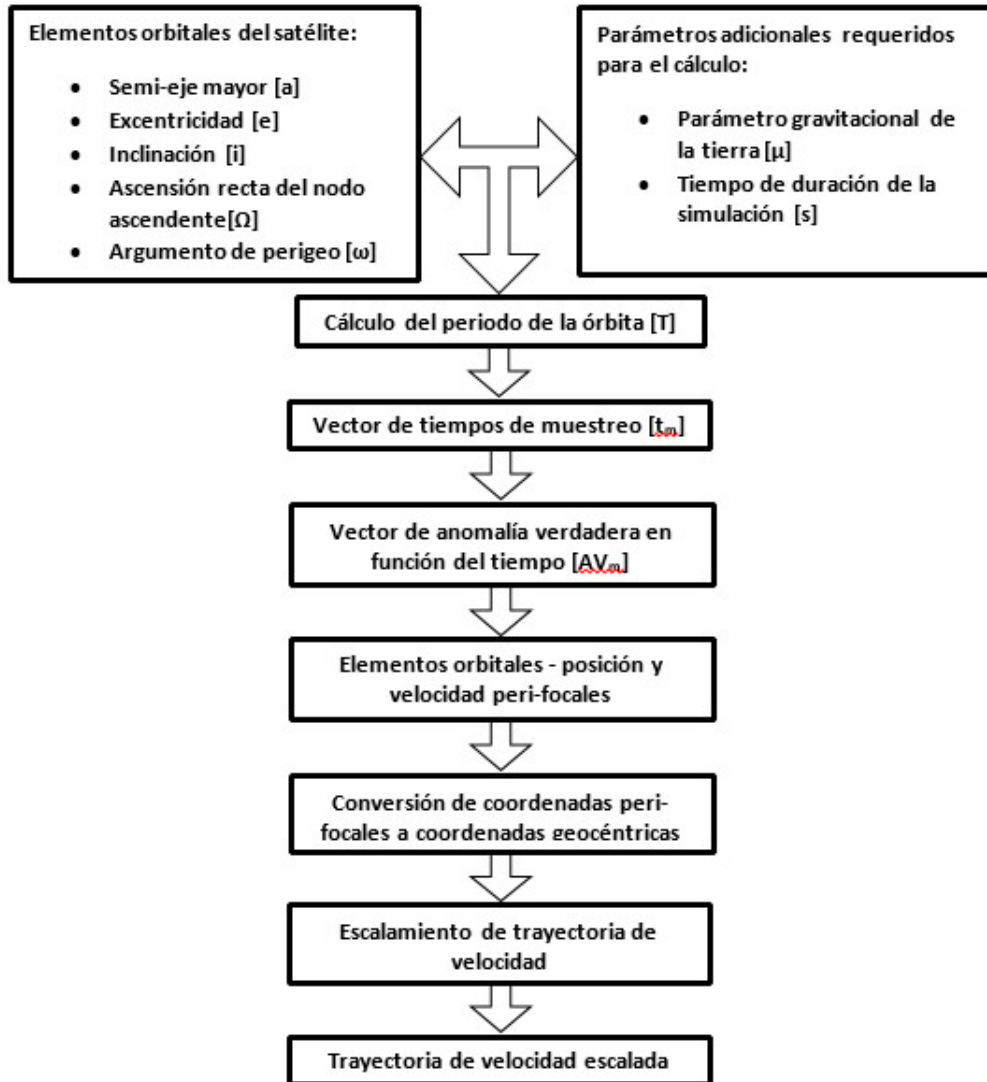


Figura 3.3: Diagrama para convertir elementos orbitales a trayectoria local escalada.

resolviendo la siguiente función implícita:

$$E - e \sin(E) = Me \quad (3.3)$$

Ahora utilizamos la siguiente expresión para despejar el valor de la anomalía verdadera:

$$\tan \frac{\theta}{2} = \sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \quad (3.4)$$

Con los elementos orbitales dados y el valor de anomalía verdadera calculado podemos obtener la información de la posición y la velocidad en coordenadas perifocales, utilizando las siguientes expresiones:

$$\{\mathbf{r}_p\} = \frac{h^2}{\mu} \frac{1}{1 + e \cos \theta} \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \quad (3.5)$$

$$\{\mathbf{v}_p\} = \frac{\mu}{h} \begin{pmatrix} -\sin \theta \\ e + \cos \theta \\ 0 \end{pmatrix} \quad (3.6)$$

Estas trayectorias velocidad y posición son transformadas del marco de referencia perifocal a un marco de referencia geocéntrico ecuatorial con la siguiente matriz de transformación determinada por los parámetros orbitales:

$$Q_{pg} = \begin{bmatrix} -\sin \Omega \cos i \sin \omega + \cos \Omega \cos \omega & -\sin \Omega \cos i \cos \omega - \cos \Omega \sin \omega & \sin \Omega \sin i \\ \cos \Omega \cos i \sin \omega + \sin \Omega \cos \omega & \cos \Omega \cos i \cos \omega - \sin \Omega \sin \omega & -\cos \Omega \sin i \\ \sin i \sin \omega & \sin i \cos \omega & \cos i \end{bmatrix} \quad (3.7)$$

Para finalizar el proceso escalamos la trayectoria dividiendo entre el máximo de los componentes de las velocidades registrados. Esto nos deja como resultado un vector con posiciones y velocidades de la órbita real escaladas y listas para poder ser enviada como referencia al cuadrotor.

El detalle de la obtención de las expresiones (3.1) a (3.7) puede ser consultado en [?].

3.3.1. Especificaciones técnicas del cálculo de la trayectoria

- Trayectoria sin perturbaciones
- Precisión de los datos: Flotante doble
- Método numérico utilizado para resolver la ecuación (3.3): Newton-Rapson
- Detalles del algoritmo y su implementación en 'Python' pueden ser consultados en: [?]

3.4. Características del vehículo cuadrotor comercial

El cuadrotor con una estructura física como la que se muestra en la figura 1.5, además el vehículo cuenta con un módulo GPS y una tarjeta controladora de vuelo 'Pixhawk'.

La técnica de control programada en este vehículo consta de un control PID para cada uno de los 4 grados de libertad con que cuenta el modelo utilizado: 3 correspondientes a la orientación y uno correspondiente a la elevación. En la figura 3.4 se muestra el esquema de control utilizado.

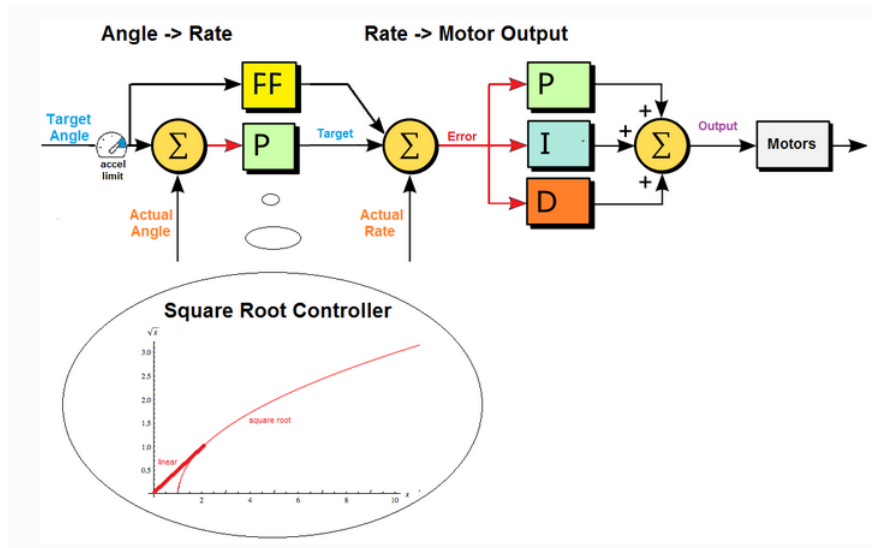


Figura 3.4: Esquema de control implementado en la tarjeta 'Pixhawk' para cada grado de libertad del vehículo.[?]

3.4.1. Especificaciones técnicas del vehículo

- Frecuencia del ciclo de control: 400hz
- Resolución del modulo GPS: 3 metros
- Resolución de datos del registro durante el vuelo: 8 dígitos
- Parámetros de los controladores programados en el sistema de navegación de fábrica:

Tabla 3.1: Valores de los parámetros de los controladores bajo el esquema presentado en la figura 3.4 para cada grado de libertad del vehículo

	Roll	Pitch	Yaw	Z
P_{src}	4.5	4.5	4.5	5
P	.150	.150	.200	.500
I	.100	.100	.020	1.000
D	.004	.004	0	0

3.5. Funcionamiento del simulador con un vehículo real o simulado.

La forma de funcionamiento se describe a continuación:

- Debemos asegurarnos que el vehículo este encendido y listo para realizar un vuelo. Los pasos para una instalación correcta de sistema del cuadrotor se encuentran en [?]. En caso de utilizar un vehículo virtual, llevar a cabo los pasos para su correcto funcionamiento disponibles en [?]
- Instalar sobre 'Windows' o 'Linux' una version 2.7 de python [?], para ejecutar el código de la interfaz.
- Una vez que se esté ejecutando la interfaz gráfica de usuario [ver figura 3.2] se introducen los elementos orbitales del satélite, la duración deseada del ciclo y la cantidad de ciclos deseados y se presiona el botón calcular trayectoria. Ésto genera un vector de velocidades de referencia adecuados para ser enviados al vehículo durante la simulación, en la ventana central se muestra una gráfica con la forma y orientación de la órbita seleccionada.

- Continuamos con la conexión al vehículo con el botón conectar, del panel de la derecha (el radio de telemetría debe estar conectado a la computadora donde se este ejecutando la interfaz del simulador). Cuando se haya establecido la conexión los colores de los botones cambiarán como se muestra en la figura 3.5.

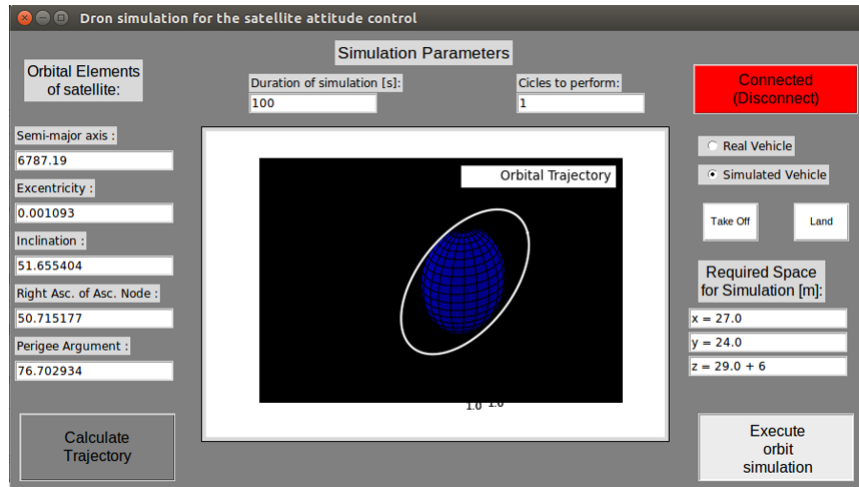


Figura 3.5: Interfaz gráfica de control durante la conexión con el sistema de navegación de un vehículo virtual.

- Cuando el vehículo ha sido correctamente conectado presionamos el botón de despegar para que el vehículo ascienda a una altura preestablecida de 6 metros sobre el suelo. Este es el punto inicial de la simulación y consideramos que el satélite comienza su recorrido en el perigeo, el cuadrotor inicia la trayectoria desde el vuelo estacionario.
- Una vez realizados los pasos anteriores el sistema se encuentra listo para ejecutar la trayectoria, esto se lleva a cabo presionando el botón: 'Ejecutar simulación de la órbita'. El vehículo comenzara a realizar la trayectoria orbital escalada. El seguimiento de la trayectoria se realiza mediante comandos de velocidad para $\{x, y, z\}$.
- Cuando el vehículo haya regresado a la posición de inicio o perigeo de la órbita simulada se detendrá y mantendrá un vuelo estacionario hasta recibir la orden de aterrizar o de ejecutar de nuevo la trayectoria orbital. La trayectoria programada no puede ser modificada mientras en vehículo se encuentra ejecutando una simulación.
- Si el vehículo se encuentra en vuelo estacionario, solo es necesario presionar el botón aterrizar para que el vehículo regrese al punto donde despegó.

3.6. Elementos orbitales a partir de la trayectoria orbital simulada

Para la obtención de los elementos orbitales de la trayectoria realizada por el vehículo cuadrotor virtual se utilizó el algoritmo 4.2 que se encuentra en [?][p.197], donde se calculan los elementos orbitales de un objeto a partir de su posición y velocidad en un instante dado.

Los pasos para calcular los elementos orbitales se muestran a continuación:

- Se calculan los elementos orbitales para cada punto de la trayectoria donde fue enviado un comando de velocidad al vehículo.
- Los elementos orbitales se calculan a partir de la posición ideal del satélite y la velocidad registrada por el cuadrotor escalada por el factor inverso utilizado para escalar a una trayectoria local.
- Se calcula el valor promedio de cada elemento orbital durante la trayectoria simulada y se obtienen los elementos orbitales correspondientes a la trayectoria simulada.

3.7. Consideraciones de variación en las características físicas del vehículo y la carga útil

El sistema simulador de trayectoria orbital planteado a lo largo de este capítulo tiene la capacidad de realizar la trayectoria orbital simulada, lo que es útil para experimentar con sistemas que tengan bajos requerimientos de peso y medida, además de que las modificaciones que se pueden hacer al vehículo o a la carga útil son muy reducidas.

Para evitar este problema y darle versatilidad al sistema de simulación, se propone construcción de una plataforma de implementación de controladores que permita hacer modificaciones al sistema de vuelo desde el ajuste de parámetros hasta el cambio en la estrategia de control. Al mismo tiempo es necesario considerar que el sistema de navegación de un dron puede llegar a ser muy complejo (figura 1.6), por lo tanto se comienza construyendo una plataforma de que tenga los elementos necesarios para realizar el vuelo desde el radio control y una estrategia de control que se considere adecuada.

Capítulo 4

Plataforma de implementación de controladores y control difuso

Para aumentar la robustez del funcionamiento del simulador de trayectoria orbital como plataforma de experimentación de sub-sistemas satelitales, se requiere del desarrollo de un sistema de control de vuelo que sirva como plataforma de implementación de controladores de forma que se pueda adecuar el sistema al cambiar las cargas útiles de experimentación sin afectar el correcto seguimiento de la trayectoria.

Basado en el trabajo de [?] observamos que las técnicas de control como el LQ son ineficientes para realizar un vuelo autónomo con un cuadrotor cuando la tarjeta controladora cuenta con características de procesamiento reducidas. Debido a esto existen trabajos que buscan la solución a este problema mediante el diseño de controladores difusos que simplifican el proceso de control a nivel de requerimientos de memoria y procesamiento como podemos observar en [?].

Así mismo en el trabajo de [?] se propone un método para la obtención de controladores difusos para cuadrotos y con base en el procedimiento planteado, se implementan los controladores utilizando la plataforma que consta de un sistema de vuelo adecuado para probar distintos tipos de controladores.

Sistema de vuelo para un cuadrotor con tarjeta "Pixhawk" utilizando controladores difusos en tiempo real.

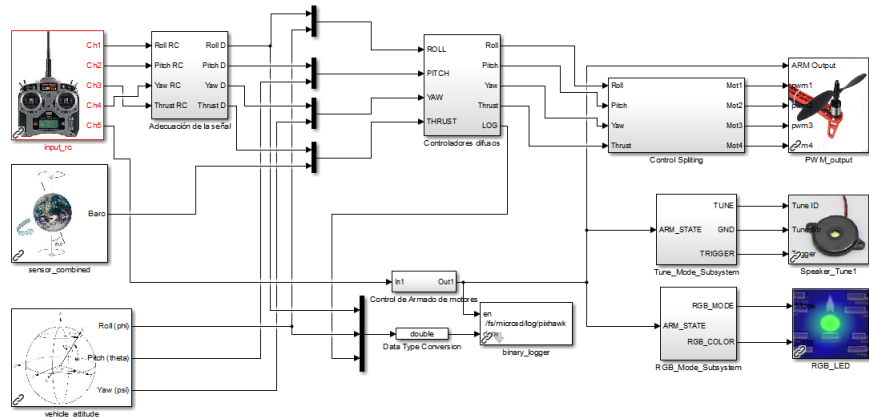


Figura 4.1: Sistema de vuelo experimental desarrollado en 'Simulink' para la implementación rápida de algoritmos de control en la tarjeta controladora 'Pixhawk'

4.1. Plataforma de implementación de controladores para la tarjeta 'Pixhawk'

El sistema de vuelo que servirá para implementar controladores de vuelo de una manera sencilla, tendrá la capacidad de realizar el vuelo mediante comandos provenientes del radio control, también utiliza de los estimadores de orientación y altura como entradas a los controladores, para generar una señal PWM hacia los motores como se muestra en la figura 4.1.

El bloque para el manejo del radio control (input_rc) nos permite obtener y manejar las señales provenientes del radio control. El bloque de sensores (sensor_combined) nos permite obtener información de los distintos sensores con los que cuenta la tarjeta, en este caso, nos interesa la señal del barómetro para medir la altura. El bloque de la orientación del vehículo (vehicle_attitude) nos proporciona información proveniente de estimadores. La señal que contiene el valor PWM de salida para el motor se conecta al bloque (PWM_output) y el bloque para guardar las señales generadas por el sistema es el (binary_logger), que almacena los datos deseados en la tarjeta de memoria para su posterior análisis. Los bloques (Tune_Mode_Subsystem) junto con el (Speaker_tune_1) controlan los sonidos y los bloques (RGB_Mode_Subsystem) y (RGB_LED) controlan las señales de los LEDs indicadores de el vehículo. Para una descripción detallada del funcionamiento de cada bloque revisar la documentación disponible en [?].

Los 3 bloques restantes fueron creados para a la sección del controlador. El bloque (Ade-

cuación de la señal) como su nombre indica, adecua la señal del RC mediante una normalización. El bloque (Controladores difusos) contiene las estructuras del control a implementar, finalmente tenemos el bloque (control_splitting) para transformar los comandos de orientación y empuje provenientes de los controladores, en un valor PWM para cada uno de los motores. La función que tiene programada este bloque es:

$$\begin{aligned}m_1 &= -Roll - Pitch + Yaw + Thrust \\m_2 &= Roll + Pitch + Yaw + Thrust \\m_3 &= Roll - Pitch - Yaw + Thrust \\m_4 &= -Roll + Pitch - Yaw + Thrust\end{aligned}\tag{4.1}$$

Gracias a esto podemos controlar el vehículo conociendo la señal del roll, pitch, yaw y thrust de un solo motor y replicarlo en los motores restante, esta forma de calcular la salida de cada motor denota el funcionamiento simétrico del cuadrotor.

4.2. Controlador difuso en tiempo real

4.2.1. Datos de entrenamiento

De un vuelo de prueba extraemos el archivo de registro de datos (Log) en formato binario, que se encuentra en la tarjeta de memoria del vehículo y generamos un arreglo con los datos en forma de vectores. Para ver el proceso de conversión a binario utilizando Matlab, ver la documentación encontrada en [?].

Para lograr un adecuado entrenamiento de los controladores vamos a hacer un pre-procesamiento a los datos PWM de los motores, que de manera práctica representa la señal enviada a los controladores de velocidad, además el requerimiento del radio control y la orientación y la altura proveniente de los estimadores. A continuación se muestra el proceso para obtener los datos de entrenamiento:

- Promediamos el la fuerza o valor PWM de los 4 motores. Debido a que el funcionamiento de los motores respecto a la orientación es simétrico, se eliminaran los aportes respectivos a la estabilización de la orientación por lo tanto obtendremos la fuerza debido al empuje, ver figura 4.2.
- Elegimos indistintamente la señal de uno de los motores y restamos la fuerza debida al empuje calculada con el promedio de todos los motores en el paso anterior. El resultado son los aportes

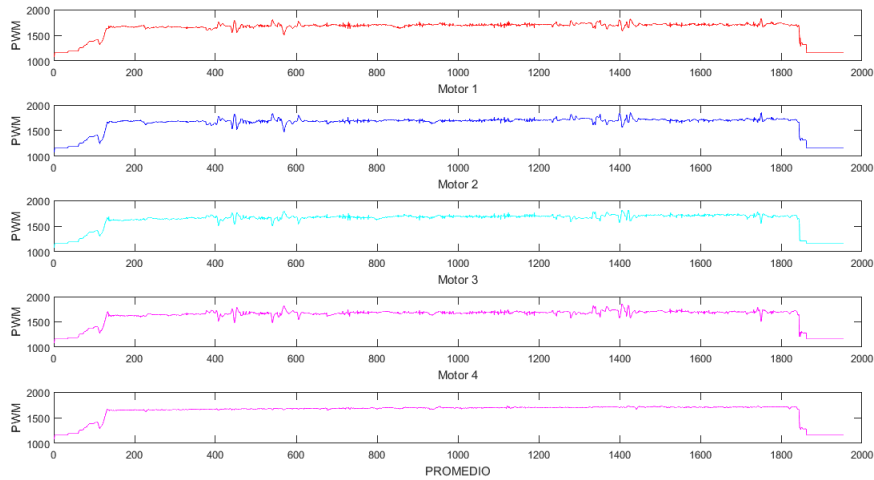


Figura 4.2: Señal de cada uno de los motores y el promedio. El eje vertical representa el valor PWM de cada uno de los motores, el eje horizontal representa los ciclos del procesador

combinados o la fuerza debida a la orientación, ver figura 4.3.

- La señal proveniente del radio control se normaliza entre los siguiente valores: roll $[-1,1]$, pitch $[-1,1]$, yaw $[-1,1]$ y thrust $[0,1]$, las señales se muestran en la figura 4.4
- Calculamos un coeficiente de requerimiento para cada grado de libertad multiplicando el requerimiento del radio control por el aporte combinado de orientación y dividimos entre la suma total de los requerimientos provenientes del radio control. El resultado nos da un coeficiente de requerimiento por cada grado de libertad: roll, pitch y yaw, ver figura 4.5.
- Multiplicamos el coeficiente calculado en el paso anterior por la fuerza combinada debida a la orientación y obtenemos la fuerza producida o PWM debido a la orientación. Estos datos son los que vamos a intentar replicar con los controladores a partir de la posición del vehículo y el requerimiento del radio control, ver figura 4.6.

A partir de los datos obtenidos vamos a elegir las secciones mas convenientes para el entrenamiento de cada grado de libertad. Es conveniente elegir para cada grado de libertad una sección reducida donde el requerimiento correspondiente del radio control sea elevado para distinguir de mejor manera los cambios en el sistema, esto se debe a que la redes neuronales aproximan funciones en un rango reducido.

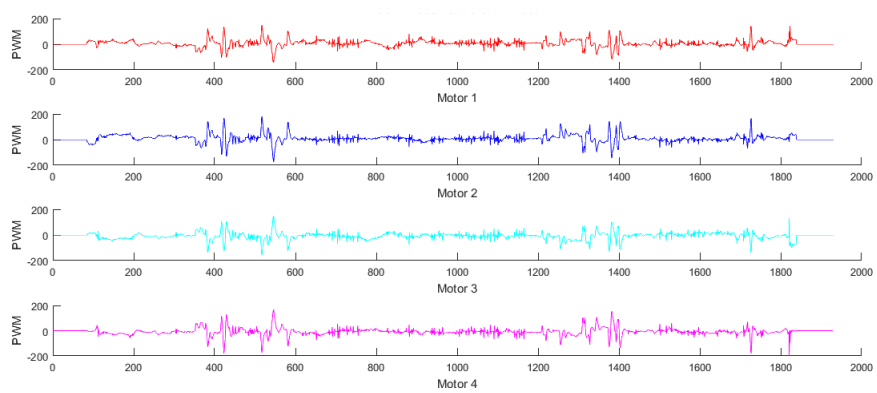


Figura 4.3: Valor PWM de los motores debida a la estabilización de la orientación, el eje vertical representa el valor PWM de cada uno de los motores, el eje horizontal representa los ciclos del procesador.

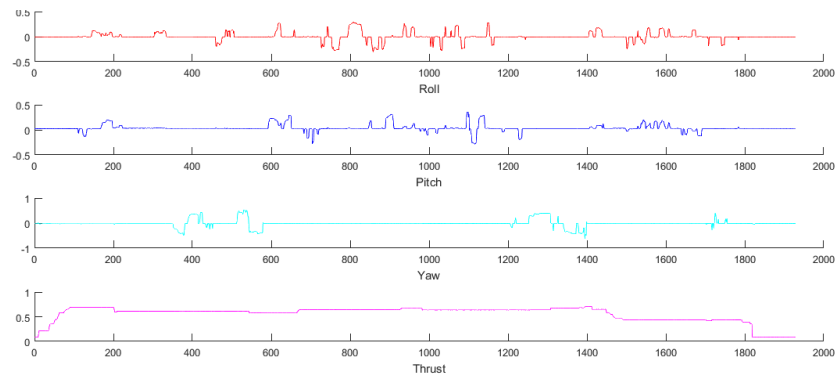


Figura 4.4: Señal proveniente del radio control normalizada, el eje vertical indica la posición de los joysticks y el eje horizontal representa los ciclos del procesador.

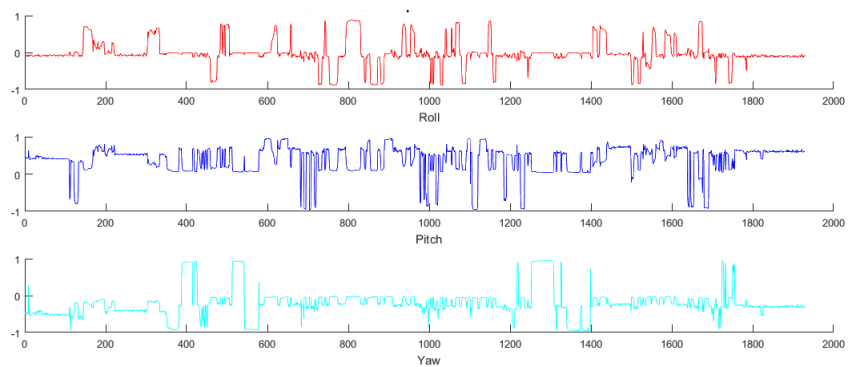


Figura 4.5: Coeficiente de requerimiento de fuerza de cada grado de libertad, el eje vertical indica la cantidad de fuerza requerida y el eje horizontal representa las muestras de cada ciclo del procesador.

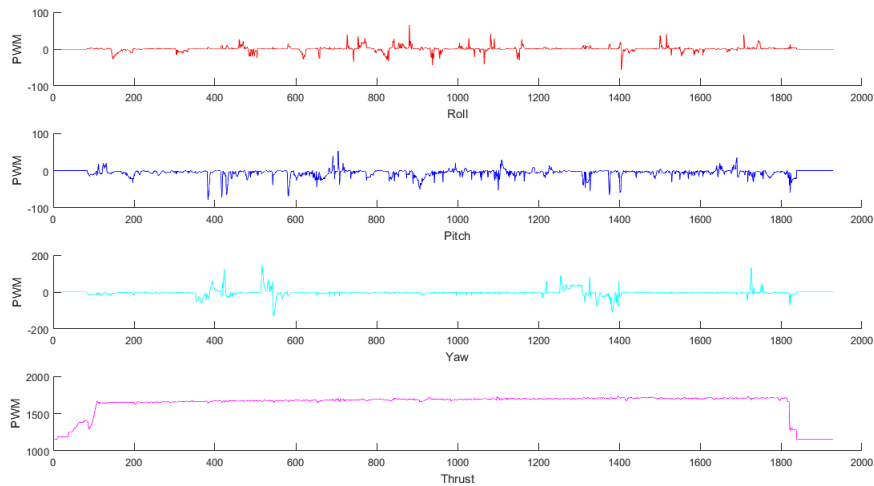


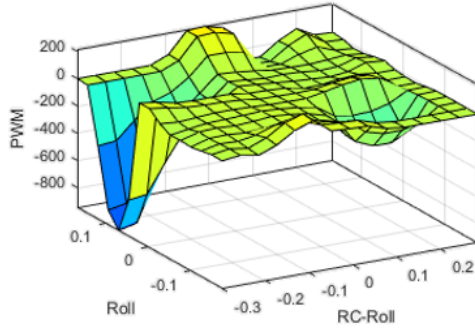
Figura 4.6: Contribución de fuerza para cada grado de libertad debida a las estabilizaciones del vuelo, el eje horizontal representa el valor en PWM y el eje horizontal representa las muestras de cada ciclo del procesador.

4.2.2. Parámetros de entrenamiento de ANFIS

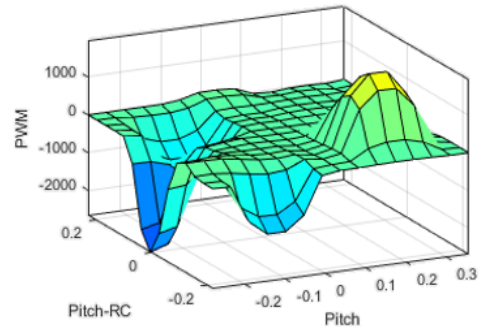
Para el entrenamiento del controlador difuso es necesario acomodar los vectores del requerimiento del radio control, la posición y por último el vector con los datos que deseamos reproducir, calculados en la sección anterior. Además debemos establecer algunos parámetros de sistema ANFIS, como se muestra a continuación:

- Tipo FIS: Sugeno
- Entradas: 2
- Salidas: 1
- Número de reglas: 25
- Tipo de funciones de pertenencia: Función Gaussiana de 2 parámetros
- Intersección: Producto
- Unión: Max
- Desfusificación: Promedio ponderado

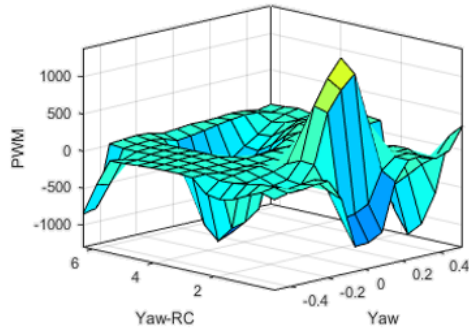
Las superficies de control obtenidas después del entrenamiento se muestran en la figura 4.7, donde los ejes horizontales muestran la posición del grado de libertad en radianes y el requerimiento del radio control. El eje vertical indica la salida PWM generada.



(a) Roll.



(b) Pitch.



(c) Yaw.

Figura 4.7: Superficies de control resultantes para cada grado de libertad, los ejes horizontales muestran la posición del grado de libertad en radianes y el requerimiento del radio control. El eje vertical indica la salida PWM generada.

Capítulo 5

Experimentos y resultados

En este capítulo se muestran los experimentos y resultados obtenidos en las distintas etapas de funcionamiento del simulador de trayectoria orbital desarrollado en este trabajo, así como los datos de los vuelos de prueba realizados para comprobar el funcionamiento correcto del sistema.

5.1. Especificaciones de funcionamiento de la Interfaz de control del simulador

La parte principal del sistema desarrollado en este trabajo es la interfaz gráfica de control que integra los subsistemas requeridos para el funcionamiento adecuado de la simulación. Esta cuenta con los elementos requeridos para:

- Introducir los parámetros de la simulación.
- Introducir los elementos orbitales del satélite deseado.
- Visualizar la forma de la órbita alrededor de la tierra mediante un gráfico.
- Conectar con el sistema de navegación de un vehículo real o uno virtual, que utilice el protocolo de comunicación 'MAVLINK'.
- Ejecutar comandos en el vehículo de: despegue, aterrizaje y ejecución de la trayectoria.

Los parámetros de la simulación son la duración del ciclo de la trayectoria local y la cantidad de ciclos a realizar. Esta interfaz gráfica desarrollada en 'Python', fue desarrollada y probada bajo

los sistemas operativos 'Windows' y 'linux/ubuntu'. A continuación se muestran los experimentos y resultados que validan el correcto funcionamiento de los subsistemas mencionados y su adecuado funcionamiento en conjunto.

5.1.1. Algoritmo para la obtención de la trayectoria orbital a partir de los elementos orbitales un satélite

La fase para la obtención de la trayectoria orbital de un satélite a partir de sus elementos orbitales, genera un vector con las velocidades correspondientes a puntos a lo largo de la órbita. Los puntos para los que calculará la velocidad se eligen mediante el muestreo uniforme a lo largo del periodo real. El primer comando que es enviado es la velocidad del satélite correspondiente al perigeo, el algoritmo considera un modelo sin perturbaciones en el calculo de las trayectorias. En la figura 5.1 se muestra la interfaz con los elementos orbitales correspondientes a un satélite de la constelación de GPS(24876), la gráfica se construye a partir de los datos de posición geocéntrica.

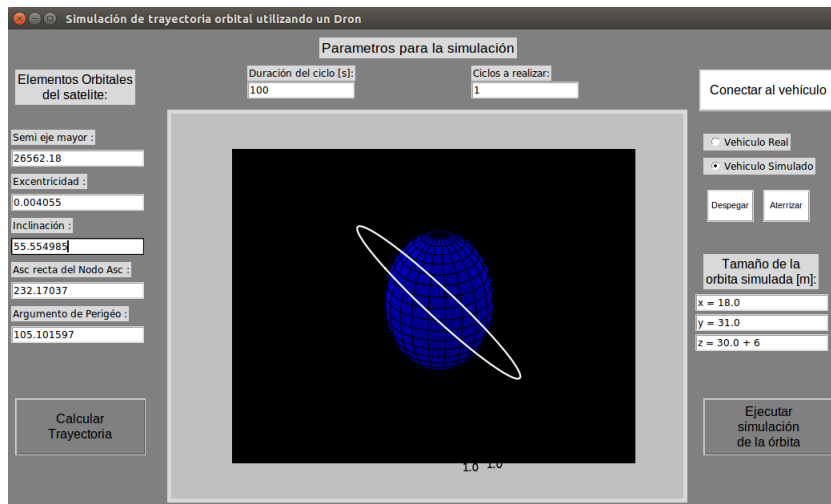


Figura 5.1: Interfaz de usuario para el control de la simulación mostrando la gráfica de la órbita del satélite GPS-24876.

La simulación de la trayectoria tiene como punto inicial el perigeo de la órbita del satélite y parte del vuelo estacionario.

5.1.2. Comunicación con el vehículo

La comunicación con el vehículo se realizó mediante los protocolos de comunicación 'MAVLINK' implementados dentro de la interfaz con el fin de controlar el funcionamiento del vehículo con el sistema de navegación de fábrica. Los comandos implementados tienen la función de:

- Conectar con el sistema del vehículo real o simulado.
- Despegar a una altura predeterminada de 6 metros.
- Ejecutar la trayectoria orbital.
- Aterrizar en el punto de despegue.

Los datos que envía el sistema del simulador para realizar la trayectoria son velocidades para los ejes X , Y y Z calculados en la etapa anterior. La frecuencia con la que son enviados los datos es una vez cada segundo, considerando que este valor puede ser variable, se elige este valor por causas de estabilidad.

5.2. Simulación orbital

El simulador de trayectoria orbital fue probado utilizando las dos posibilidades de funcionamiento que ofrece: utilizar un vehículo virtual o un vehículo real. En ambas opciones el sistema del vehículo nos permite recopilar los datos del vuelo durante la simulación para su posterior extracción y análisis con los software 'Mission Planner' y 'Matlab'.

5.2.1. Simulación orbital con un vehículo virtual

Para la prueba de funcionamiento del simulador con un vehículo virtual se requirió de una computadora con sistema operativo Linux/Ubuntu, con un vehículo cuadrotor virtual con un firmware 'quadcopter v3.3.1' en el software SITL para ser ejecutado, la posición del vehículo virtual es independiente de la ejecución de la órbita y toma en cuenta un marco de referencia NED (North-East-Down).

Para la prueba, se seleccionó dentro de la interfaz del simulador la conexión con un vehículo virtual, en la figura 5.2 se muestra el simulador de trayectoria orbital funcionando con un vehículo virtual.

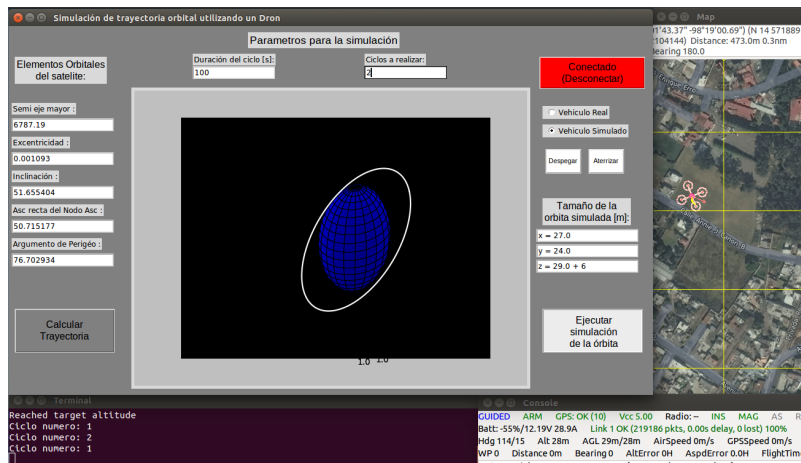


Figura 5.2: Interfaz gráfica de control funcionando en conjunto con un vehículo virtual, el vehículo virtual se ejecuta sobre el software 'SITL' bajo el sistema operativo 'Linux - Ubuntu'

El software del vehículo virtual ofrece una visualización de la trayectoria del cuadrotor en un mapa setelital, así como el estado de sus sensores durante los vuelos virtuales. La trayectoria orbital fue correctamente realizada por el vehículo, su forma final se puede observar en la figura 5.3.



Figura 5.3: Trayectoria simulada de la órbita de la estación espacial internacional realizada por un vehículo virtual. La visualización de la trayectoria vista desde arriba, se realiza en un mapa geo-referenciado .

Posición del vehículo virtual durante la trayectoria

La figura 5.8 muestra la posición del vehículo registrada durante la simulación.

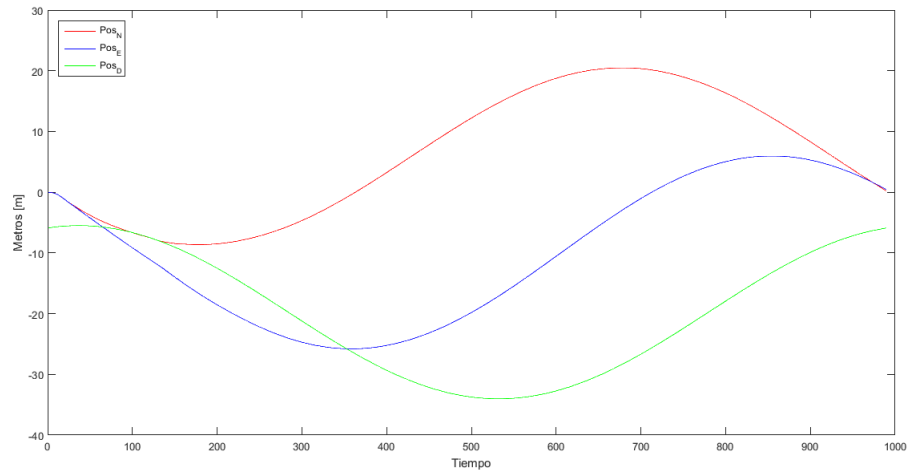
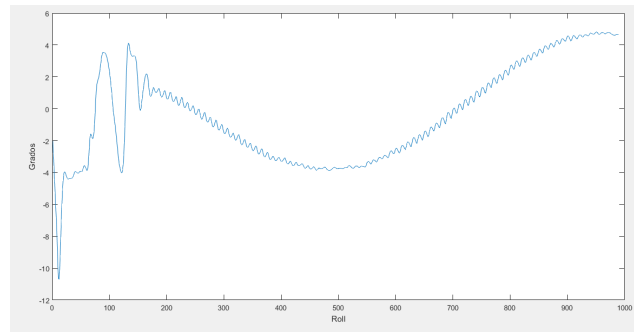


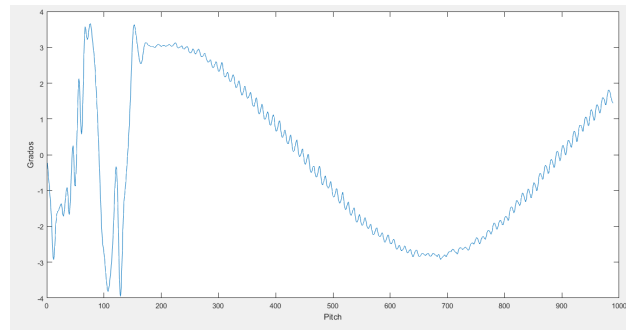
Figura 5.4: Posición del vehículo virtual bajo el marco de referencia NED durante la simulación orbital. El eje vertical indica la posición en metros y el eje vertical representa cada muestra del ciclo del procesador.

Orientación del vehículo virtual durante la trayectoria

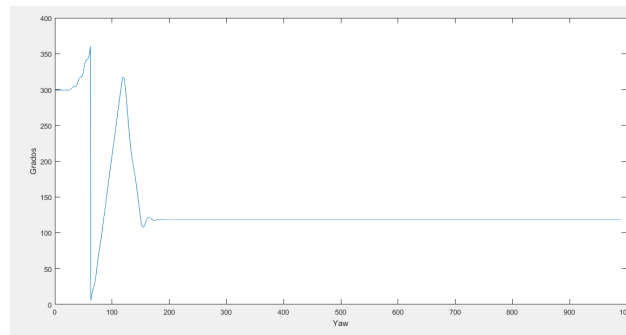
Los datos de orientación se muestran en la figura 5.5.



(a) Ángulo roll del vehículo virtual bajo el marco de referencia NED.



(b) Ángulo pitch del vehículo virtual bajo el marco de referencia NED.

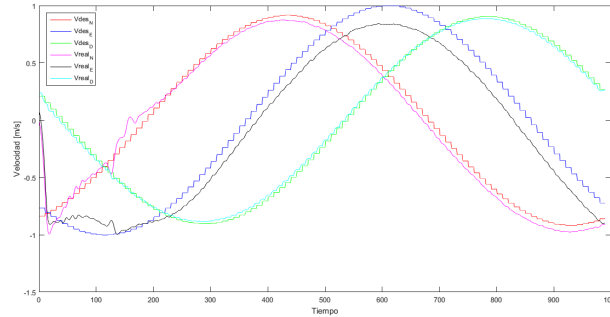


(c) Ángulo yaw del vehículo virtual bajo el marco de referencia NED.

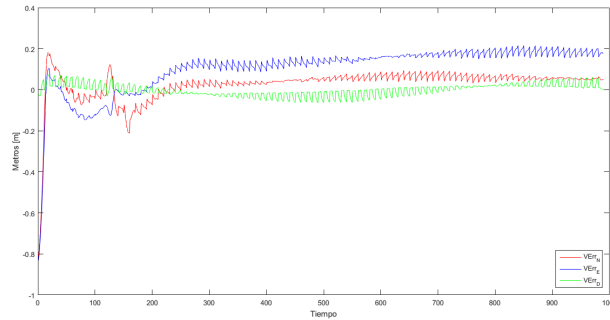
Figura 5.5: Orientación del vehículo virtual durante la simulación orbital. El eje vertical indica la posición angular de cada eje medido en grados y el eje vertical representa cada muestra del ciclo del procesador.

Velocidad del vehículo virtual durante la trayectoria

Respecto al desempeño de la trayectoria se comparan las velocidades de referencia generadas por la interfaz y las velocidades registradas durante el vuelo, estos datos y el error entre ambas velocidades son mostrados en la figura 5.6. Los resultados se detallan en la tabla 5.1.



(a) Velocidades deseadas y registradas por el vehículo virtual.



(b) Error entre la velocidad deseada y la registrada durante el vuelo virtual.

Figura 5.6: Gráficas de desempeño de la trayectoria orbital simulada con un vehículo virtual.

Tabla 5.1: Datos de la trayectoria orbital simulada realizada por un vehículo virtual

	Pos_{max+}	Pos_{max-}	Vel_{max+}	Vel_{max-}	$Vdes_{max-}$	$Vdes_{max+}$	$ Err_{max} $	Err_{rms}	$Err\%$
N	20.4862	-8.6347	0.8764	-0.9951	0.9154	-0.9168	0.8173	0.0602	6.02%
E	5.9622	-25.7907	0.8430	-0.9970	0.9997	-1.0017	0.8312	0.1383	13.83%
D	-5.4964	-34.0207	0.8877	-0.8857	0.9039	-0.9046	0.0755	0.0228	2.28%

Elementos orbitales de la trayectoria simulada realizada por un vehículo virtual

Los elementos orbitales de la trayectoria ideal y de la realizada por el vehículo virtual se presentan en la tabla 5.2.

Tabla 5.2: Elementos orbitales de la trayectoria realizada por el vehículo virtual. Semi-eje mayor, excentricidad, inclinación, argumento de perigeo, ascensión recta del nodo ascendente. Los datos corespondientes a angulos están dados en grados.

<i>ElemOrb</i>	<i>a</i>	<i>e</i>	<i>i</i>	ω	Ω
<i>Ideal</i>	6787.19	0.0010	51.65	76.70	50.71
<i>Simulada</i>	5700.39	0.9671	118.02	153.85	186.55

5.2.2. Simulación orbital con un vehículo real

Así mismo se realizó una prueba del simulador con un vehículo cuadrotor de la marca '3DR Robotics' con modulo GPS con una resolución de 3m. y tarjeta controladora 'Pixhawk', la cual toma en cuenta un marco de referencia local NED para su sistema de navegación. Se utiliza una computadora con sistema operativo 'Windows' o 'Linux/Ubuntu' con la interfaz del simulador ejecutandose, además de un modulo de telemetría conectado a la computadora. El vehículo previamente calibrado se conectó con el simulador y se ejecuto la trayectoria orbital, los datos de vuelo durante la simulación son almacenados y analizados. La figura 5.7 muestra la posición de la órbita simulada en un mapa geo-referenciado.



Figura 5.7: Trayectoria de la órbita simulada realizada por el vehículo real visualizada en un mapa georeferenciado.

Posición del vehículo real durante la trayectoria

En la figura 5.8 se muestra la gráfica con la posición del vehículo durante la simulación de la trayectoria orbital.

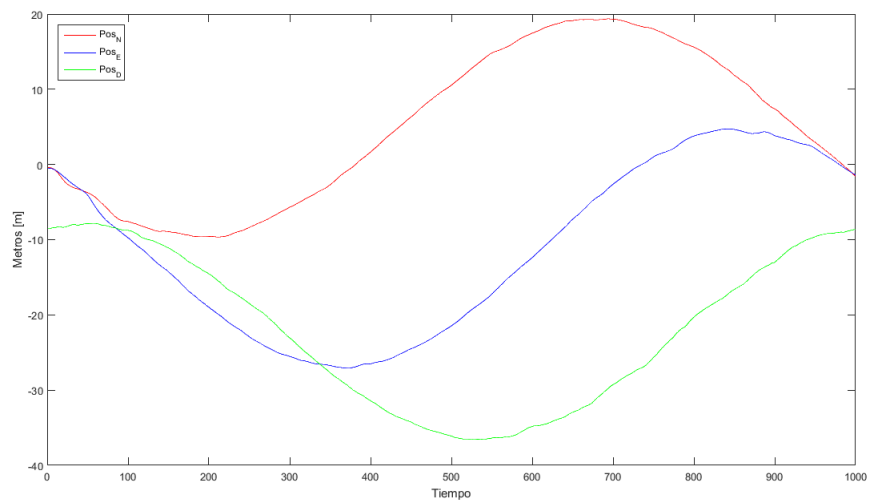
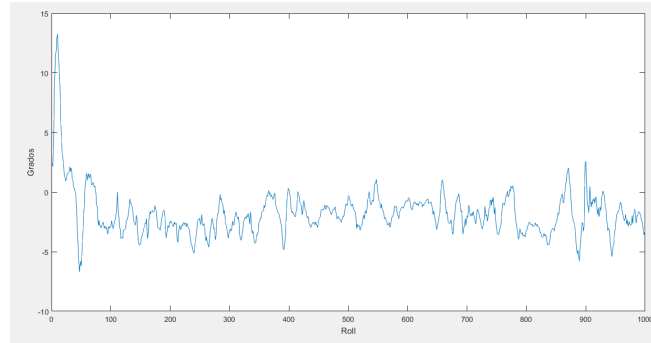


Figura 5.8: Posición del vehículo real bajo el marco de referencia NED durante la simulación orbital. El eje vertical indica la posición en metros y el eje horizontal representa cada muestra del ciclo del procesador.

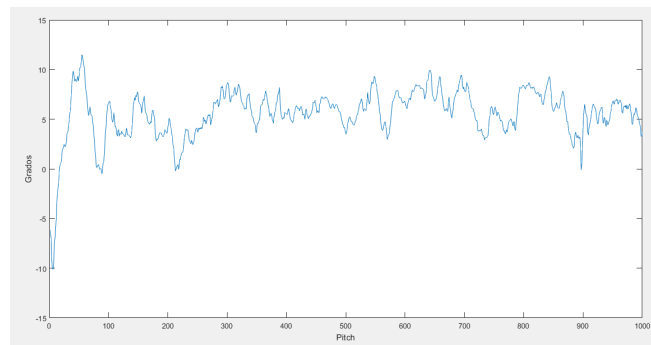
Podemos observar que la forma de la órbita es suave y sin cambios bruscos.

Orientación del vehículo real durante la trayectoria

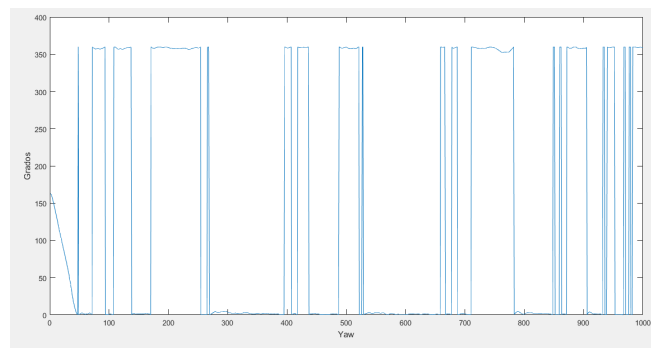
Los datos del vuelo respectivos a la orientación del vehículo se muestran en la figura 5.9.



(a) Ángulo roll del vehículo real bajo el marco de referencia NED.



(b) Ángulo pitch del vehículo real bajo el marco de referencia NED.

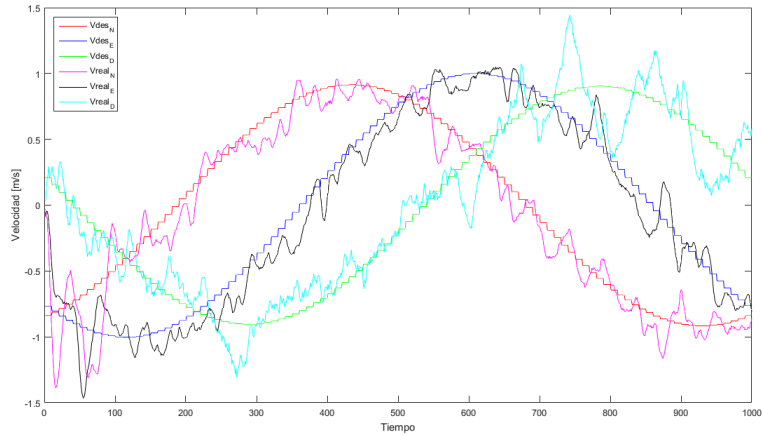


(c) Ángulo yaw del vehículo real bajo el marco de referencia NED.

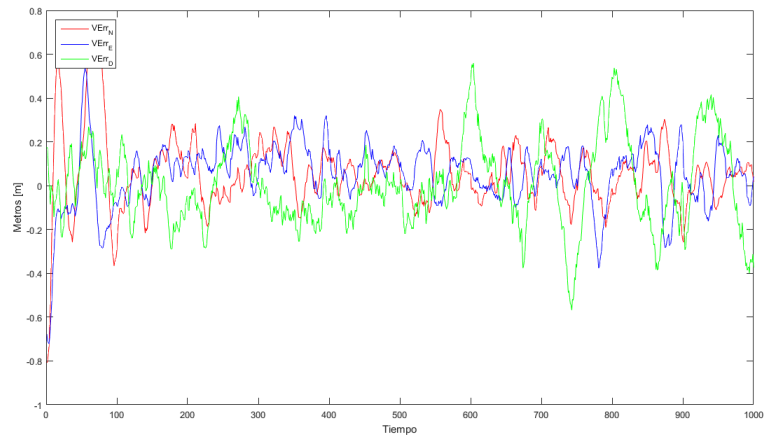
Figura 5.9: Orientación del vehículo real durante la simulación orbital. El eje vertical indica la posición angular de cada eje medido en grados y el eje vertical representa cada muestra del ciclo del procesador.

Velocidad del vehículo real durante la trayectoria

Se compararon las velocidades reales del vuelo y las velocidades deseadas enviadas al vehículo, en la figura 5.10 se pueden observar las velocidades para cada eje y el error entre ellas.



(a) Velocidades deseadas y registradas por el vehículo real.



(b) Error entre la velocidad deseada y la registrada durante el vuelo real.

Figura 5.10: Gráficas de desempeño de la trayectoria orbital simulada con un vehículo real.

El error en el seguimiento la trayectoria proviene principalmente del sistema de navegación del vehículo utilizado y de la frecuencia con que se envía la señal deseada.

Tabla 5.3: Datos de la trayectoria orbital simulada realizada por un vehículo real

	Pos_{max+}	Pos_{max-}	Vel_{max+}	Vel_{max-}	$Vdes_{max-}$	$Vdes_{max+}$	$ Err_{max} $	Err_{rms}	$Err\%$
<i>N</i>	19.3662	-9.6443	0.9581	-1.3891	0.9154	-0.9168	0.8095	0.1194	11.94 %
<i>E</i>	4.7221	-27.0906	1.0513	-1.4634	0.9997	-1.0017	0.7209	0.1191	11.91 %
<i>D</i>	-7.8187	-36.5726	1.4441	-1.3056	0.9039	-0.9046	0.5678	0.1494	14.94 %

Elementos orbitales de la trayectoria simulada por un vehículo real

Los elementos orbitales de la trayectoria ideal y de la realizada por el vehículo real se presentan en la tabla 5.4.

Tabla 5.4: Elementos orbitales de la trayectoria realizada por el vehículo real. Semi-eje mayor, excentricidad, inclinación, argumento de perigeo, ascensión recta del nodo ascendente. Los datos corespondientes a angulos están dados en grados.

<i>ElemOrb</i>	<i>a</i>	<i>e</i>	<i>i</i>	ω	Ω
<i>Ideal</i>	6787.19	0.0010	51.65	76.70	50.71
<i>Simulada</i>	8324.00	0.2722	50.55	118.03	50.22

5.2.3. Comparativa entre simulación con vehículo real y virtual

En la tabla 5.5 podemos observar el error relativo entre los errores de la trayectoria virtual y real.

Tabla 5.5: Comparativa entre los errores de las simulaciones con vehículo real y virtual.

	$ErrMax_{virtual}$	$ErrMax_{real}$	$ErrRel_{rms}$
<i>N</i>	0.8173	0.8095	0.1174
<i>E</i>	0.8312	0.7209	0.1171
<i>D</i>	0.0755	0.5678	0.1439

5.3. Plataforma de implementación de algoritmos de control

La plataforma de implementación de algoritmos de control que pretende brindar versatilidad y robustez al sistema del simulador, en una etapa inicial, habilita al vehículo para estabilizar el vuelo con comandos provenientes del radio control. La estructura permite cambiar la estrategia de control de manera sencilla y programarlo directamente en la tarjeta controladora 'Pixhawk', utilizando la toolbox 'Pixhawk-Simulink'.

Para probar el correcto funcionamiento de esta herramienta, se desarrolló un sistema de control difuso como técnica para la estabilización del vuelo y se codificó correctamente en la tarjeta de control. El registro de datos generado por el sistema implementado se puede observar en la figura 5.11, las señales corresponden a la configuración dada en la figura 4.1.

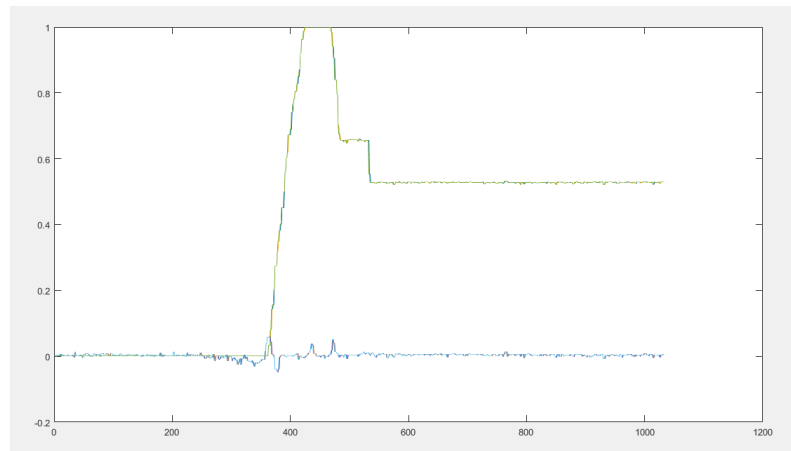
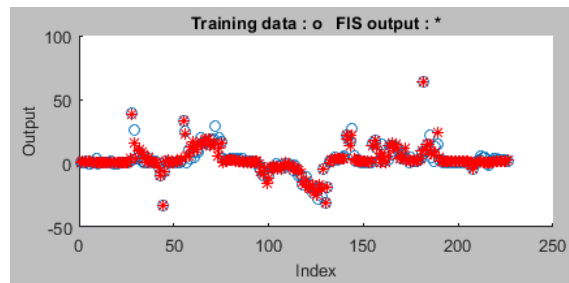


Figura 5.11: Registro de datos del sistema de vuelo implementado.

5.3.1. Controlador difuso en tiempo real

Las salidas del controlador difuso entrenado para este trabajo se comparan con los datos de entrenamiento y se muestran en la figura 5.12.

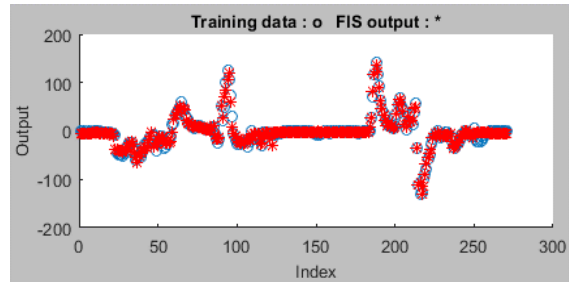
Estos resultados son comparados con los datos de entrenamiento. El sistema de vuelo utilizando los controladores difusos entrenados, fue codificado exitosamente en la tarjeta controladora del vehículo con el proposito de experimentar el correcto funcionamiento.



(a) Salida del controlador, Roll.



(b) Salida de controlador, Pitch.



(c) Salida de controlador, Yaw.

Figura 5.12: Superficies de control resultantes para cada grado de libertad

Capítulo 6

Conclusiones y trabajo futuro

El sistema propuesto para utilizar un dron comercial como simulador de la trayectoria orbital de un satélite descrita a partir de sus elementos orbitales demostró un funcionamiento satisfactorio durante las pruebas realizadas. El simulador fue diseñado para funcionar en dos modalidades: en conjunto con un vehículo virtual o con un vehículo real. Esta forma de trabajo le da al simulador un bajo costo ya que brinda la posibilidad de utilizar el simulador incluso sin un vehículo real.

La interfaz de control fue exitosamente programada utilizando software de código libre, integrando exitosamente los sub-sistemas necesarios para la simulación orbital propuesta. El diseño de funcionamiento la hace fácilmente adaptable y escalable en funciones además de funcionar en los sistemas operativos más populares 'Windows' y 'Linux'.

La comunicación para el control del sistema de navegación del vehículo también mostró un correcto funcionamiento al desarrollar la simulación tanto en un vehículo virtual y en uno real. Los comandos implementados aseguran un funcionamiento seguro del vehículo. Debido a la utilización del protocolo 'MAVLink' este simulador es compatible con una amplia gama de vehículos cuadrotores comerciales.

La interfaz también es capaz de calcular y mostrar correctamente la trayectoria obtenida a partir de los elementos orbitales de un satélite, lo que facilita la comprensión de la relación de estos parámetros con la forma y tamaño de la órbita. La utilización de los elementos orbitales nos da la ventaja de obtener la información de la órbita de un satélite de manera fácil y práctica ya que estos los encontramos fácilmente en la red.

Respecto a la trayectoria orbital simulada podemos observar que el error obtenido entre los

datos de trayectoria generados y los datos registrados durante el vuelo no afecta de manera significativa el recorrido, lo que se comprueba de manera visual en las figura 5.7 y 5.3 correspondientes a las trayectorias realizada por los vehículos real y virtual respectivamente. También se puede observar que el error en velocidad obtenido es menor a 15cm/s lo que se considera aceptable tomando en cuenta que se utiliza un modulo GPS de bajo costo.

Debido a los resultados este simulador es considerado adecuado para el proposito de experimentar, integrar y validar subsistemas satelitales cuyos requerimientos espaciales y de peso sean reducidos. La presición con la que la trayectoria es realizada puede ser aumentada en caso de ser requerido y el registro de datos facilita el analisis de los resultados, lo que es esencial al momento de realizar una plataforma de experimentación.

Para extender el funcionamiento del simulador como plataforma de experimentación se construyó un de implementación de algoritmos de control, como una primera aproximación a un sistema de vuelo adecuado en su totalidad para la realización de trayectorias orbitales más precisas además del registro y el fácil analisis de datos requeridos durante el experimento.

Se logró la correcta codificación del sistema de vuelo con el controlador difuso en tiempo real, aunque muestra buen desempeño en la simulación, es necesaria la aplicación de una técnica más precisa para la correcta estabilización del vehículo en un vuelo real. Esta forma de implementación reduce el problema de la dificultad de acceder y modificar el sistema de vuelo para futuras modificaciones o adaptaciones tanto del vehículo, del sistema de control o del simulador.

Como trabajo futuro se pretende la implementación de la simulación de constelaciones satelitales realizadas por drones bajo el mismo concepto presentado en este trabajo, lo que amplia la posibilidad para la experimentación de sistemas de comunicaciones y vuelo en formación.

Apéndice A

Interfaz Gráfica para el control de la simulación

El código correspondiente a la interfaz se encuentra en el DVD adjunto en este trabajo de tesis.

