



Instituto Politécnico Nacional
CENTRO DE INVESTIGACIÓN EN CIENCIA APLICADA
Y TECNOLOGÍA AVANZADA
UNIDAD QUERÉTARO



DOCTORADO EN TECNOLOGÍA AVANZADA

**Centro de Investigación en Ciencia Aplicada y Tecnología
Avanzada
Unidad Querétaro**

Línea de investigación: análisis de imágenes.

TESIS DOCTORAL

**Control de morfologías cambiantes
para manipulación aérea.**

Autor: Omar Rodríguez Abreo.

Director: Dr. Francisco Javier Ornelas Rodríguez.

Codirector: Dr. Alfonso Ramírez Pedraza

Querétaro, diciembre 2021



INSTITUTO POLITÉCNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México, a 11 de septiembre del 2020

El Colegio de Profesores de Posgrado de **CICATA Unidad Querétaro** en su Sesión **ordinaria** No. **200703** celebrada el día **3** del mes **julio** de **2020**, conoció la solicitud presentada por el alumno:

| | | | | | |
|-------------------|-----------|-------------------|-------|-------------|------|
| Apellido Paterno: | Rodríguez | Apellido Materno: | Abreo | Nombre (s): | Omar |
|-------------------|-----------|-------------------|-------|-------------|------|

Número de registro:

| | | | | | | |
|---|---|---|---|---|---|---|
| A | 2 | 0 | 0 | 6 | 7 | 4 |
|---|---|---|---|---|---|---|

del Programa Académico de Posgrado:

| |
|----------------------------------|
| Doctorado en Tecnología Avanzada |
|----------------------------------|

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

| |
|---|
| Control de morfologías cambiantes para manipulación aérea |
|---|

Objetivo general del trabajo de tesis:

Diseñar y construir un sistema capaz de manipular objetos en torres de telecomunicaciones.
Objetivos específicos: Obtener el modelo dinámico del sistema
Obtener la ley de control para el sistema
Desarrollar un sistema de control que permita el desplazamiento autónomo del manipulador
Diseñar y programar un algoritmo de aproximación visual
Desarrollar sistema de control de fuerza para el actuador final del manipulador el cual sujetará objetos de diversos pesos.

2.- Se designa como Directores de Tesis a los profesores:

Director:

| |
|--|
| Dr. Francisco Javier Ornelas Rodríguez |
|--|

 2° Director

| |
|-----------------------------------|
| Dr. Pedro Alfonso Ramírez Pedraza |
|-----------------------------------|

No aplica:

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

| |
|--|
| CICATA-Unidad Queretaro y (CIO campus Monterrey) |
|--|

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director de Tesis

Dr. Francisco Javier Ornelas Rodríguez

2° Director de Tesis

Dr. Pedro Alfonso Ramírez Pedraza

Aspirante

Omar Rodríguez Abreo

Presidente del Colegio

Dr. Juan Bautista Hurtado Ramos
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de siendo las horas del día del mes de del se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: para examinar la tesis titulada:

del (la) alumno (a):

| | | | | | |
|-------------------|-----------|-------------------|-------|-------------|------|
| Apellido Paterno: | Rodríguez | Apellido Materno: | Abreo | Nombre (s): | Omar |
|-------------------|-----------|-------------------|-------|-------------|------|

Número de registro:

Aspirante del Programa Académico de Posgrado:

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 19% de similitud. **Se adjunta reporte de software utilizado.**

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo SI NO SE CONSTITUYE UN POSIBLE PLAGIO.

JUSTIFICACIÓN DE LA CONCLUSIÓN: Las coincidencias radican en las referencias, palabras comunes y citas de texto de algunas referencias usadas, en ninguna de las fuentes se supera el 1% de plagio por lo que no se tomó ninguna idea sustancial de otro trabajo.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** **SUSPENDER** **NO APROBAR** la tesis por **UNANIMIDAD** o **MAYORÍA** en virtud de los motivos siguientes:

Originalidad del trabajo, publicación de resultados en revista científica de alto impacto, así como resultados aplicables a la industria.

COMISIÓN REVISORA DE TESIS

Director de Tesis
Dr. Francisco Javier Ornelas Rodríguez

2º Director de Tesis
Dr. Pedro Alfonso Ramírez Pedraza

Dr. Raúl Alejandro Ávalos Zúñiga

Dr. José Joel González Barbosa

Dr. Eduardo Castillo Castañeda

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN Y POSGRADO

Dr. Juan Bautista Portado Ramos
PRESIDENTE DEL COLEGIO DE PROFESORES



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día **07** del mes de **febrero** del año **2023**, el que suscribe **Omar Rodríguez Abreo** alumno del programa de **Doctorado en Tecnología Avanzada** con número de registro **A200694**, adscrito al **Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, CICATA-UNIDAD QUERÉTARO**, manifiesta que es autor intelectual del presente trabajo de tesis bajo la dirección de **Dr. Francisco Javier Ornelas Rodríguez** y el **Dr. Pedro Alfonso Ramírez Pedraza** y cede los derechos del trabajo intitulado **Control De Morfologías Cambiantes Para Manipulación Aérea**, al Instituto Politécnico Nacional, para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expresado del autor y/o director(es). Este puede ser obtenido escribiendo a las siguiente(s) direcciones de correo ornelasr@ipn.mx, pedro.ramirez@cio.mx, omar.rodriguez@upq.edu.mx. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.

M. en I. Omar Rodríguez Abreo.

AGRADECIMIENTOS

Quiero dedicar este trabajo a toda la gente que siempre ha creído en mí y me apoyado en este camino. Especialmente a mi esposa, Mayra Cruz Fernández por su incondicional apoyo en este viaje.

A mis hijas, padres y hermanas por ser la familia que siempre me ha apoyado.

Finalmente, agradezco a INDESYTH, a mi asesor Francisco Ornelas Rodríguez y a mis sinodales por haber apoyado en el desarrollo de este trabajo bajo su tutela.

RESUMEN

El objetivo fundamental de la presente tesis doctoral es el diseño e implementación de un sistema que permita la manipulación precavida de objetos mediante un manipulador acoplado a un vehículo aéreo no tripulado, denominados UAV por sus siglas en inglés (*Unmanned Aerial Vehicle*). El vehículo deberá acercarse a infraestructuras de telecomunicaciones y realizar de manera segura maniobras de manipulación. Para ello el movimiento del vehículo y el del manipulador serán ejecutados de manera autónoma apoyados por algoritmos de control que aportarán el movimiento necesario para que el sistema pueda alcanzar los objetos que desea manipular. Por otra parte, el sistema de retroalimentación de posición suele fallar cuando hay interferencia con el cielo abierto por ello se propone un sistema de control complementario basado en visión artificial.

Los trabajos con UAVs han despertado gran interés en diversos grupos de investigación en los últimos años, logrando destacados avances, sin embargo, la propia mecánica del sistema contiene severas limitaciones, con lo cual la mayoría de las aplicaciones se basan en sistemas de detección y supervisión, para superar estas limitantes se ha acoplado al sistema un manipulador robótico de dos grados de libertad, con el cual podrá realizar una gama más amplia de aplicaciones. El brazo está acoplado en la parte superior lo que permite al dron seguir utilizando la cámara en la parte inferior.

La navegación autónoma se apoyará en múltiples sensores que permitirán al vehículo conocer su posición absoluta y su orientación en el espacio, debido a que para el desarrollo de este trabajo se utilizaron UAVs de exteriores, los principales sensores de los vehículos son sistemas de posicionamiento global GPS (Global Positioning System), unidades de medición inercial denominadas IMUs (Inertial Measurement Unit), apoyados de sensores secundarios como son barómetro y brújula. La información que debe recibir el sistema de control debe incluir el estado actual en el que se encuentra el manipulador para permitir una navegación eficiente en espacios abiertos libres de interferencias, sin embargo, el GPS pierde precisión cuando se acerca a infraestructuras, es por ello que se para la aproximación final se apoyara en un sistema de retroalimentación visual, para ello la cámara ubicará un punto conocido en la infraestructura (estos puntos estarán indicados con marcadores especiales) y posteriormente determinará la posición del dron respecto de ese punto. Conociendo la posición del marcador y la posición del UAV al marcador se puede encontrar una posición real del UAV.

Para analizar los movimientos del conjunto UAV-manipulador se desarrolló el modelado matemático del sistema con el cual podemos analizar las fuerzas que actúan sobre cada subsistema, así como ejecutar simulaciones del comportamiento del conjunto ante distintos escenarios antes de implementarse físicamente.

Una vez concluido el desarrollo matemático del modelo se procedió a desarrollar la estrategia de control que generará las señales para que el conjunto del sistema UAV-manipulador ejecute la

tarea deseada de manera autónoma, para ello se emplea la técnica denominada Backstepping la cual está basada en la teoría de estabilidad de Lyapunov y permite controlar sistemas dinámicos no lineales como el presentado en este trabajo, entre sus principales características presenta un balance adecuado entre bajo coste computacional y robustez ante incertidumbres paramétricas. Para comparar los resultados obtenidos se implementara de igual forma un controlador del tipo PID, el cual es uno de los controladores más utilizados debido tanto a su rapidez como a su facilidad a la hora de implementarse en sistemas reales, como primera etapa se realizaron las pertinentes simulaciones que demuestran el comportamiento del sistema bajo estos paradigmas de control así como su respuesta ante diversos cambios como perturbaciones o incertidumbres paramétricas, se analizó las respuesta de ambos controladores para trayectorias predefinidas y finalmente se compararan la precisión de cada técnica empleada.

Posteriormente se desarrolló la implementación de los protocolos en un sistema real lo que nos permitan observar el comportamiento auténtico de las leyes de control usadas en situaciones completamente reales, se realizan los ajustes correspondientes para optimizar el control y finalmente se obtiene el comportamiento deseado del sistema, con errores aceptable a lo largo de la ejecución de la tarea, cumpliendo con los diferentes requisitos y garantizando la correcta aplicación en un ambiente real. Concluida la etapa de la navegación se desarrolla el sistema de retroalimentación visual el cual permitirá al dron acercarse de manera gradual al objetivo.

Posteriormente se muestran a detalle los resultados obtenidos de las pruebas experimentales los cuales son presentados a través de graficas que permiten observar el desempeño del sistema, así como su comportamiento ante distintas trayectorias.

Por último, se describen las conclusiones a las cuales se ha llegado a partir de los resultados obtenidos en simulaciones y en pruebas experimentales, y se proponen los trabajos que podrán mejorar el desempeño de la manipulación con drones.

Abstract

The main objective of this doctoral thesis is the design and implementation of a system that allows the safe manipulation of objects employing a manipulator coupled to an unmanned aerial vehicle, called UAVs (Unmanned Aerial Vehicle). The vehicle must approach telecommunications infrastructures and safely carry out handling maneuvers. For this, the movement of the vehicle and that of the manipulator will be executed autonomously supported by control algorithms that will provide the necessary movement so that the system can reach the objects it wishes to manipulate. On the other hand, the position feedback system usually fails when interference with the open sky. Therefore, a complementary control system based on artificial vision is proposed.

In recent years, work with UAVs has aroused great interest in various research groups, achieving notable advances. However, the mechanic systems contain severe limitations, with which most of the applications are based on detection and supervision systems. To overcome these limitations, a two-degree-of-freedom robotic manipulator has been coupled to the system. It will be able to carry out a wider range of applications. The arm is attached at the top, allowing the drone to continue using the camera at the bottom.

Autonomous navigation will be supported by multiple sensors that will allow the vehicle to know its absolute position and orientation in space. Since outdoor UAVs were used to develop this work, the primary sensors of the vehicles are GPS global positioning systems. (Global Positioning System), inertial measurement units called IMUs (Inertial Measurement Unit), supported by secondary sensors such as barometer and compass. The information that the control system must receive must include the current state of the manipulator to allow efficient navigation in open spaces free of interference. However, GPS loses precision when approaching infrastructures, which is why it will be supported by a visual feedback system for the final approach. For this, the camera will locate a known point in the infrastructure (these points will be indicated with special markers), and later it will determine the position of the drone with respect to that point. A UAV's real position can be found knowing the position of the marker and the position of the UAV to the marker.

The mathematical modeling of the system was developed to analyze the movements of the UAV-manipulator assembly. The model was used to analyze the forces that act on each subsystem and run behavior simulations of the assembly in different scenarios before being physically implemented.

Once the mathematical development of the model is concluded, the control strategy that generates the signals for the whole UAV-manipulator system to execute the desired task autonomously was developed. For this, the Backstepping technique is used, which is based on Lyapunov's stability theory and allows controlling non-linear dynamic systems such as the one presented in this work. Among its main characteristics, it shows an adequate balance between low computational cost and robustness in the face of parametric uncertainties. A PID-type controller was being implemented in the same way to compare the results obtained, which is one of the

most used controllers due to both its speed and its ease of implementation in real systems. As the first stage, the pertinent ones have carried out simulations that demonstrate the system behavior under these control paradigms and its response to various changes such as disturbances or parametric uncertainties. The responses of both controllers for predefined trajectories were analyzed, and finally, the precision of each technique used was compared.

Subsequently, the implementation of the protocols in a real system was developed. The implementation allows it to observe the authentic behavior of the control laws used in genuine situations. The corresponding adjustments were made to optimize the control, and finally, the desired behavior of the system is obtained, with acceptable errors throughout the execution of the task, complying with the different requirements and guaranteeing the correct application in a real environment. After the navigation stage, the visual feedback system is developed to allow the drone to approach the objective gradually.

Subsequently, the results obtained from the experimental tests are shown in detail, which is presented through graphs that allow observing the performance of the system, as well as its behavior before different trajectories.

Finally, the conclusions reached from the results obtained in simulations and experimental tests are described, and the works that may improve the performance of drone manipulation are proposed.

Índice.

| | |
|---|----|
| 1 INTRODUCCIÓN | 15 |
| 1.1 OBJETIVOS | 17 |
| 1.1.1 OBJETIVO GENERAL | 17 |
| 1.1.2 OBJETIVOS ESPECÍFICOS | 17 |
| 1.2 ORGANIZACIÓN DEL TRABAJO | 18 |
| 2 ESTADO DEL ARTE | 20 |
| 2.1 LOS UAVS | 20 |
| 2.2 CONTROL DE UAVS | 21 |
| 2.3 BACKSTEPPING | 21 |
| 2.4 PID | 23 |
| 2.5 SISTEMAS UAVS MANIPULADORES | 23 |
| 2.6 SISTEMAS DE RETROALIMENTACION VISUAL | 24 |
| 3. DESARROLLO | 26 |
| 3.1 DESCRIPCION DEL VEHICULO | 26 |
| 3.2 MODELADOS DEL VEHÍCULO | 26 |
| 3.2.1 MODELO CINEMÁTICO | 26 |
| 3.3 MODELO DINÁMICO | 33 |
| 3.3.1 MODELADO DINÁMICO DEL UAV | 34 |
| 3.4 CONTROL DEL UAV | 41 |
| 3.4.1 CONTROL BACKSTTEPING | 41 |
| 3.4.1.1 CONTROL BACKSTEPPING DEL UAV | 42 |
| 3.4.1.2 CONTROL DEL UAV CON PARES GIROSCOPICOS. | 48 |
| 3.4.2 CONTROL PID | 52 |
| 3.4.2.1 CONTROL PID DEL UAV | 53 |
| 3.4.2.2 CONTROL PID DE TRASLACIÓN | 53 |
| 3.4.2.3 CONTROL PID DE ORIENTACIÓN | 54 |
| 3.5 MODELADO DE MAIPULADORES | 55 |
| 3.5.1 MODELO CINEMÁTICO DEL MANIPULADOR | 55 |
| 3.5.2 MODELO DINÁMICO DEL MANIPULADOR | 56 |
| 3.6 CONTROL DEL MANIPUALDOR | 57 |

| | |
|--|-----|
| 3.6.1 CONTROLADOR BACSTEPPING PARA MANIPULADOR | 57 |
| 3.6.2 CONTROL PD+G PARA MANIPULADOR | 59 |
| 3.7 MODELADO SISTEMA UAV MANIPULADOR | 60 |
| 3.8 CONTROL SISTEMA UAV MANIPULADOR | 62 |
| 3.8.1 CONTROL BACKSTEPPING SISTEMA UAV MANIPULADOR | 62 |
| 3.8.2 CONTROL PID SISTEMA UAV MANIPULADOR | 67 |
| 3.9 CONTROL DE FUERZA EN MAIPULADOR | 68 |
| 3.10 Sistema de retroalimentación visual. | 70 |
| 3.10.1 Calibración de la cámara para medición. | 70 |
| 3.10.2 Red neuronal profunda para estimación de rangos. | 71 |
| 4. RESULTADOS | 73 |
| 4.1 Resultados de simulación para el control del UAV. | 73 |
| 4.2 Resultados de simulación para el control del sistema MM-UAV. | 81 |
| 4.3 Sintonización del sistema MM-AUV por el algoritmo de la búsqueda del cuco. | 84 |
| 4.4 Resultados de simulación para el sistema de retroalimentación visual. | 93 |
| 4.5 Resultados experimentales del control para el sistema MM-UAV. | 96 |
| 4.6 Resultados experimentales del sistema de navegación visual. | 100 |
| 5. CONCLUSIONES | 103 |
| 6. BIBLIOGRAFIA | 104 |
| ANEXOS | 112 |
| A.1 SEÑALES DE CONTROL U_3 , U_4 y U_1 . | 112 |
| A.2 CODIGOS IMPLEMENTADOS EN LA SIMULACIÓN | 115 |
| A.3 CODIGOS IMPLEMENTADOS EN EL HARDWARE | 126 |

Índice de Figuras.

| | |
|--|----|
| Figura 1. Vehículo multirroto utilizado. | 26 |
| Figura 2. Diagrama de cuerpo libre de un multirroto. | 27 |
| Figura 3. Ángulos de Tait-Bryan | 27 |
| Figura 4. Rotación básica, en sentido antihorario, del ángulo ϕ sobre el eje "x" | 28 |
| Figura 5. Rotación básica, en sentido antihorario, del ángulo θ sobre el eje "y" | 29 |
| Figura 6. Rotación básica, en sentido antihorario, del ángulo ψ sobre el eje "z" | 29 |
| Figura 7. Sistemas de referencia dependientes en el espacio tridimensional. | 30 |
| Figura 8. Sistema clásico de control PID. | 52 |
| Figura 9. Manipulador planar de 2 grados de libertad. | 55 |
| Figura 10. Sistema UAV-Manipulador. | 60 |
| Figura 11. Proceso de calibración de la cámara. | 70 |
| Figura 12. Simulación de control Backstepping para UAV. (a) Modelo de Simulink; (b) Subsistema de control de rotación de UAV; (c) Subsistema de rotación de UAV. | 73 |
| Figura 13. Trayectoria real del UAV (línea sólida azul) vs. Trayectoria 1 (línea roja punteada). | 74 |
| Figura 14. X real del UAV (línea sólida azul) vs. X deseada (línea roja punteada). | 75 |
| Figura 15. Y real del UAV (línea sólida azul) vs. Y deseada (línea roja punteada). | 75 |
| Figura 16. Z real del UAV (línea sólida azul) vs. Z deseada (línea roja punteada). | 76 |
| Figura 17. Errores de desplazamiento de la trayectoria 1 del UAV. | 76 |
| Figura 18. Errores de orientación de la trayectoria 1 del UAV. | 77 |
| Figura 19. Trayectoria helicoidal con perturbaciones. Posición Real del UAV (línea sólida azul) vs. Posición deseada (línea roja punteada). | 78 |
| Figura 20. X real del UAV en trayectoria helicoidal (línea sólida azul) vs. X deseada (línea roja punteada). | 78 |
| Figura 21. Y real del UAV en trayectoria helicoidal (línea sólida azul) vs. Y deseada (línea roja punteada). | 79 |
| Figura 22. Z real del UAV en trayectoria helicoidal (línea sólida azul) vs. Z deseada (línea roja punteada). | 79 |
| Figura 23. Errores de desplazamiento del UAV en trayectoria helicoidal. | 80 |
| Figura 24. Errores de rotación del UAV en trayectoria helicoidal. | 80 |
| Figura 25. Simulación del sistema MM-UAV. (a) Modelo de simulink; (b) subsistema de control de rotación; (c) subsistema de rotación. | 81 |
| Figura 26. Sistema físico: (a) UAV hexarroto; (b) Manipulador planar. | 82 |
| Figura 27. MM-UAV Posición cartesiana del sistema MM-UAV para la trayectoria simulada 1. (a) Posición del UAV; (b) Posición del manipulador. | 83 |
| Figura 28. MM-UAV Posición por ejes del sistema MM-UAV para la trayectoria simulada 1. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador. | 84 |
| Figura 29. Diagrama de flujo del algoritmo de búsqueda del cuco utilizado. | 85 |

| | |
|--|-----|
| Figura 30. Respuesta al escalón con las ganancias obtenidas por el algoritmo de la búsqueda del cuco. (a) Respuesta en el eje X; (b) Respuesta en el eje Y; (c) Respuesta en el eje Z; (d) Respuesta en el manipulador. | 87 |
| Figura 31. Posición cartesiana del MM-UAV en el seguimiento de la trayectoria para la tarea 1. (a) Posición del UAV; (b) Posición del manipulador. | 89 |
| Figura 32. Seguimiento de la trayectoria de la tarea 1 en cada subsistema. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador. | 90 |
| Figura 33. Posición cartesiana del MM-UAV en el seguimiento de la trayectoria con perturbaciones para la tarea 2. (a) Posición del UAV; (b) Posición del manipulador. | 90 |
| Figura 34. Seguimiento de la trayectoria de la tarea 2 en cada subsistema. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador. | 91 |
| Figura 35. Respuesta de paso del controlador PID. (a) Respuesta en el eje X; (b) Respuesta en el eje Y; (c) Respuesta en el eje Z; (d) Respuesta en el manipulador. | 92 |
| Figura 36. Imágenes utilizadas en el proceso de calibración. | 93 |
| Figura 37. Calibración de la cámara mediante el uso de un patrón conocido | 94 |
| Figura 38. Cálculo de la distancia al patrón mediante la cámara calibrada. | 94 |
| Figura 39. Simulación virtual de la aproximación visual para la manipulación aérea. | 95 |
| Figura 40. Point of View solidario a la cámara del UAV. | 96 |
| Figura 41. Sistema MM-UAV en control de posición y orientación. | 98 |
| Figura 42. Seguimiento de la trayectoria en cada eje del sistema físico del MM-UAV. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador. | 99 |
| Figura 43. Señales de control del sistema físico MM-UAV. (a) Señal de control U1; (b) Señal de control U2; (c) Señal de control U3; (d) Señal de control U4; (e) Señal de control τ_1 ; (f) Señal de control τ_2 . | 100 |
| Figura 44. Señales de retroalimentación. | 101 |
| Figura 45. Errores de seguimiento para las señales de retroalimentación (a) Señal del GPS; (b) Señal del sistema de retroalimentación. | 101 |

1

INTRODUCCIÓN

1 INTRODUCCIÓN

Los trabajos sobre navegación autónoma en robots móviles se enfocan a resolver problemas específicos con los que se encuentra un robot móvil durante la ejecución de una tarea, recientemente se han solventado algunas complicaciones de la navegación autónoma por lo que los esfuerzos se han focalizado en aplicar a tareas específicas la navegación de los robots [1].

Dentro de la robótica móvil los vehículos aéreos no tripulados (llamados UAV's por sus siglas en inglés *-Unmanned Aerial Vehicules*) han logrado un amplio desarrollo en la última década, debido principalmente al interés que supone obtener un vehículo aéreo capaz de completar diversas tareas sin intervención humana. Múltiples aplicaciones militares y civiles muestran el amplio alcance los UAVS, entre las que destacan: exploración [2], recolección de datos [3], búsqueda en edificios colapsados[4], detección y localización de incendios [5], inspección de estructuras [6], cartografía y fotografía aérea [7].

Los resultados recientes obtenidos en robótica, mecatrónica y la microelectrónica han permitido superar las limitaciones que se tenían debido al hardware y en la actualidad es posible realizar el procesamiento en tiempo real y navegar de manera completamente autónoma[1], aunado a esto la construcción de sensores y actuadores en escala reducida ofrecen un enorme potencial para el desarrollo de vehículos aéreos no tripulados [8].

A pesar de que existen múltiples tipos de aeronaves la elección del vehículo aéreo apropiado está en función de los requisitos de la tarea que se desea ejecutar, algunas son muy exigentes en términos de tamaño, peso o maniobrabilidad. Un tipo de aeronaves referenciadas debido a su agilidad son los vehículos multirrotor, que en comparación con los aeroplanos posee una mayor maniobrabilidad [9]. Un multirrotor es un vehículo aéreo que consta de una unidad central donde, de manera general, se realiza el procesamiento, y tal y como su nombre lo indica múltiples rotores (al menos 3, aunque generalmente son de 4 o múltiplos de dos) distribuidos de manera concéntrica al vehículo.

Los vehículos multirrotor se encuentra dentro de un subconjunto de vehículos aéreos capaces de despegar y aterrizar de manera vertical, llamados VTOL por sus siglas en inglés (*Vertical takeoff and landing*). El desarrollo a gran escala de multirrotores había despertado poco interés a finales del siglo pasado, a pesar de ello el primer vuelo tripulado corto fue en un vehículo de cuatro rotores en, 1907 [10]. Los multirrotores presentan dos ventajas principales sobre otros vehículos VTOL, como los helicópteros. En primer lugar, tienen una mecánica muy sencilla ya que el control de su posición está determinado por los cambios de velocidad en sus motores. En segundo lugar, el uso de cuatro rotores o más garantiza que cada uno de estos motores es más pequeño que el rotor principal que requeriría un helicóptero [11] distribuyendo de manera equitativa el esfuerzo en cada rotor.

Los multirrotores son utilizados cuando se requiere una alta maniobrabilidad, ya que son capaces de moverse en cualquier dirección o volar a bajas velocidades [11], es por ello que se han implementado múltiples técnicas de control, de las cuales se han obtenido resultados favorables, sin embargo a pesar de los avances, aun se requiere hacer frente a algunas dificultades, en particular, la presencia de perturbaciones al sistema, como pueden ser las turbulencias atmosféricas, las cuales puede conducir a la inestabilidad del sistema, además el problema de navegación es complejo y requiere de sensores y dispositivos específicos; algunos autores han solucionado el problema de navegación mediante la visión artificial [12–14], no obstante esto se logra en ambientes altamente controlados.

Varias empresas comerciales y organizaciones de investigación como: Microdrones, AirRobot, Draganfly, AscTec, DJI entre otras, han desarrollado múltiples vehículos aéreos no tripulados y los han distribuido en el mercado civil y militar, con gran éxito. Algunos grupos de investigación se han enfocado en el diseño de vehículos más pequeños, ligeros y ágiles, por lo que es posible encontrar diversos diseños de multirrotores [1,15], debido a que poseen entre sus características principales la posibilidad de ejecutar un vuelo estacionario estable y preciso gracias al equilibrio en las fuerzas de propulsión ejercidas por sus hélices.

El objetivo más importante de la investigación para la navegación autónoma es eliminar o en su defecto, reducir al mínimo la intervención del humano, de modo que el sistema sea capaz de ejecutar la tarea deseada sin ayuda [1,8,11], con lo cual se lograrían múltiples ventajas, como una mayor eficiencia, un menor riesgo para el hombre y la reducción en el tiempo de ejecución de la tarea.

El desarrollo de leyes de control para los vehículos aéreos y en específico para los multirrotores no es un problema trivial, debido principalmente a la compleja dinámica, propia de este tipo de vehículos. Los vehículos aéreos son sistemas, no lineales, sub-actuados y multivariables; esto implica que las estructuras de control lineales o monovariables tienen resultados limitados y pueden ocurrir inestabilidades cuando el sistema se aleja del equilibrio[16]. De igual forma las metodologías diseñadas para sistemas completamente actuados no se aplican directamente a los UAV's, ya que son sistemas mecánicos no lineales y sub-actuados.

En trabajos anteriores, el interés de los principales grupos de investigación se centró en resolver los problemas del control y de navegación [1,11]. Es por esto por lo que diversos autores han propuesto diversas teorías que solventen estos inconvenientes y que permitan mayor exactitud o un sistema más robusto. En los trabajos previos se puede observar que a que no hay estrategia ideal para el control de un UAV, ya que los resultados varían de acuerdo con las condiciones del vuelo y la tarea que se dese ejecutar.

Debido a su gran simplicidad mecánica, un multirroter también cuenta con limitaciones mecánicas y de actuación, lo cual los ha llevado a ejecutar tareas principalmente de supervisión y detección, puesto que típicamente llevan incluido como equipo principal una cámara con características específicas acorde a la aplicación que se desea ejecutar, a pesar de ser de gran utilidad, hay aplicaciones en las que se requiere un mayor grado de implicación por parte del vehículos por lo

cual, algunos grupos de investigación han redirigido sus esfuerzos al estudio y al control de un sistema integrado por un UAV y un manipulador robótico, lo cual representa un sistema aún más complejo debido a que el movimiento de un subsistema, ya sea el brazo o el vehículo, afectara directamente el estado del otro subsistema. Por otra parte, en cuanto el manipulador sostenga un objeto la masa del sistema cambiara lo cual debe ser considerado y compensado por la metodología que se implante para el control de un sistema UAV-manipulador.

Otra consideración es la precisión del sistema de control y retroalimentación, si bien el GPS proporciona una navegación aceptable en exteriores, para manipulación se requieren errores de centímetros mientras que un GPS, un GLONASS (Global'naya Navigatsionnaya Sputnikovaya Sistema - Sistema de navegación global por satélite) o los GALILEOS comerciales ofrecen precisiones de 1.5 a 2.5 metros.

La manipulación del objeto depende de la precisión de la retroalimentación, por lo que el GPS aun a pleno rendimiento no es una opción para la manipulación, aunado a esto, su precisión depende del número de satélites con los que el sistema puede comunicarse, a mayor número de satélites mayor es la precisión, sin embargo, cuando el UAV se acerca a un edificio se pierde recepción con la mayoría de los satélites esto puede ocasionar que una misión de acercamiento a una infraestructura se vuelva catastrófica. Estos mismos problemas los sufren el sistema de posicionamiento ruso y el europeo.

Por lo tanto, esta tesis centra su planteamiento en dar solución al control del sistema AUV-manipulador, dentro de un entorno tridimensional en exteriores, mediante el cual se pueda llevar a cabo la manipulación de objetos en infraestructuras, para lo cual se requiere un control preciso y un sistema de retroalimentación complementario basado en visión artificial.

1.1 OBJETIVOS

1.1.1 OBJETIVO GENERAL

Diseñar y construir un sistema capaz de manipular objetos en torres de telecomunicación

1.1.2 OBJETIVOS ESPECÍFICOS

- Obtener el modelo dinámico del sistema.
- Obtener la ley de control para el sistema.
- Desarrollar un sistema de control que permita el desplazamiento autónomo del Manipulador.
- Diseñar y programar un algoritmo de aproximación visual.
- Desarrollar sistema de control de fuerza para el actuador final del manipulador el cual sujetara objetos de diversos pesos.

1.2 ORGANIZACIÓN DEL TRABAJO

El presente trabajo presenta un sistema de manipulación aérea UAV-manipulador que permite el transporte de objetos de distinto peso. Para ello se presenta el desarrollo de dos estrategias de control para este vehículo y posteriormente la adaptación de estas estrategias al sistema UAV-manipulador, además de la implantación de un sistema de control de fuerza para el manipulador todo ello estructurado de la siguiente forma:

En el primer apartado de este trabajo se presenta la parte introductoria de este trabajo, en la cual se encuentra el planteamiento de la tesis, se describe de manera breve el contexto de desarrollo y se exponen los objetivos y los requisitos que se deben cumplir.

El segundo apartado muestra el denominado estado del arte, el cual expone sistemáticamente los avances cualitativos en el tema de navegación con UAV y más específicamente en el tema de UAVs con un manipulador acoplado, de igual forma se encuentran los avances en el campo de la robótica sensorial y la imitación del tacto en robots.

El tercer apartado muestra el análisis realizado para encontrar los modelos matemáticos que describen al vehículo aéreo, así como su comportamiento, para poder observar de manera objetiva las fuerzas movimientos u otros factores que afecten directamente al vehículo, para ello se calcularan tanto el modelo cinemático como el modelo dinámico del vehículo, del manipulador y del conjunto. Se presenta además el desarrollo de las leyes de control para el vehículo multirrotores, para el manipulador y para el conjunto.

El cuarto apartado muestra los resultados obtenidos en la simulación de los algoritmos de control y el procedimiento para implementar el control y se exponen los resultados de las pruebas experimentales del sistema tanto del desempeño de las leyes de control como la manipulación de distintos objetos.

El quinto apartado muestra las conclusiones obtenidas durante la realización de la presente tesis, además se expondrá brevemente las posibles líneas de investigación que permitirían ampliar y/o mejorar el trabajo aquí realizado.

Finalmente, en última sección de esta tesis se añaden los anexos los detalles del desarrollo de los cálculos realizados en la tesis.

2

ESTADO DEL ARTE

2 ESTADO DEL ARTE

En este capítulo se mostrará a detalle el estado del arte relacionado con la navegación autónoma de UAVs, y el trabajo realizado con plataformas compuestas por UAV y manipulador, con esto se pretende mostrar de manera resumida los avances logrados en el área lo cual permitirá exponer de una manera más clara los aportes realizados por este trabajo. Para ello se hará un breve recorrido por la evolución de los estudios en la navegación de robots aéreos y se mostrarán algunos de los éxitos de estos en determinadas aplicaciones.

También se ahondará en algunas de las técnicas de control habitualmente usadas y en sus beneficios y desventajas reportadas por los autores de diferentes estudios, y se pone especial énfasis en las técnicas utilizadas en esta tesis. De igual forma se expondrán los enfoques empleados por distintos autores para el análisis del sistema UAV-manipulador, sus limitaciones y sus ventajas.

Finalmente se mostrará el desarrollo en el campo de los sistemas sensoriales particularmente aquellos avances en sistemas especializados en visión artificial para analizar las técnicas de guiado basad en visión.

2.1 LOS UAVS

El trabajo con vehículos aéreos no tripulados tiene varias décadas realizándose en primera instancia a nivel militar, sin embargo ya en la década de los ochenta encontrabas algunas simulaciones de la navegación autónoma actual de vehículos aéreos, posteriormente al inicio de la década de los noventa se demostraba que el enlace entre un vehículo aéreo no tripulado y un satélite era viable^[16], para finales de esa década se encontraba una mayor cantidad de recursos a nivel hardware y software disponibles para la navegación autónoma [17].

Para inicios del nuevo siglo ya se encontraban trabajos con formaciones de UAV's [18], aunque muchos trabajos seguían enfocados al área militar y la tasa de éxito en misiones autónomas era relativamente baja [17].

Sin embargo a partir de mediados de la década anterior la reducción de los costos en hardware, su reducción tanto en tamaño y peso como en consumo de energía y la ampliación de capacidades a nivel software permitieron una explosión en el desarrollo de diversas aplicaciones civiles [19].

En la década actual los UAVs civiles son una realidad, múltiples tipos de aeronaves han sido ampliamente estudiados e implementado en aplicaciones con una alta tasa de éxito logrando aplicaciones industriales y comerciales de altos requerimientos técnicos [19]. Con un grado de avance tan elevado, la tendencia actual no es en el estudio de los UAVs en sí mismos si no en un conjunto del UAV con algún dispositivo altamente especializado que se enfoque a la ejecución de una tarea específica; otra vertiente de la investigación de UAV's es la utilización de varios vehículos que trabajen de manera colaborativa para lograr un objetivo en común [20] o a la construcción de sistemas tolerantes a fallos dentro de una misión autónoma [21].

2.2 CONTROL DE UAVS

En este apartado se muestran de manera específica las leyes de control implementadas en la literatura actual, se analizarán sus beneficios y sus desventajas. Entre las leyes de control recurrentes en la literatura se encuentran: linealización exacta por realimentación dinámica la cual tiene un control aceptable si no existen grandes perturbaciones al sistema [22]; Backstepping en la cual se permite obtener una ley de control basada en la teoría de Lyapunov a partir de un modelo con la aproximación de ángulo pequeño [23], H_∞ que trata de minimizar la relación entre la energía de la señal de error y la energía de la señal de perturbación [24]; Control por redes neuronales [25], métodos probabilísticos para la estimación de la posición [26], modificaciones de las técnicas anteriores [27], entre otras.

Debido a la complejidad del problema de manera general se suele simplificar o considerar nulas algunas características no lineales del sistema como las fricciones debido a los momentos aerodinámicos, las fuerzas de arrastre; o los efectos giroscópicos, con algunas de estas simplificaciones se ha demostrado que un controlador PID independiente correctamente sintonizado puede estabilizar los UAV's [28], algunos autores proponen modificaciones a la implementación del PID logrando buenos resultados [29] o incluso la combinación con otra técnica como en el caso de los PID difusos [30],

Recientemente se ha logrado implementar diversos métodos adaptativos inteligentes ya sean basados en lógica difusa [31], en redes neuronales [32] o en otras técnicas de inteligencia artificial [33]. Otros autores han enfocado su trabajo en metodologías de control bioinspiradas [34] en las cuales control el UAV tomando como referencia alguna función y/o diseño biológico.

El coste computacional de algunos de los métodos anteriormente mencionados es muy alto y a pesar de que hoy en día tenemos capacidades de procesamiento muy elevadas no siempre es viable ejecutarlas en tiempo real, por lo cual algunos de los trabajos quedan demostrados solo a nivel simulación, un subconjunto de estas mismas leyes de control son aplicadas con éxito en un ambiente de trabajo altamente controlado, lo cual no siempre es posible, es por ello que los actuales Vehículos Aéreos no Tripulados de uso comercial usan controladores altamente probados y con un costo computacional bajo (generalmente PID's), lo que permite reducir costos, tomando en cuenta todas estas consideraciones en este trabajo se implementó una ley de control con una buena relación entre calidad/costo computacional, la denominada técnica de control Backstepping, de igual forma para contrastar los resultados se implementará un PID el cual es el controlador más usado a nivel industrial y comercial debido a su gran desempeño de calidad/sencillez.

2.3 BACKSTEPPING

Existen múltiples metodologías para desarrollar y verificar una ley de control, la más frecuente en la literatura es obtener el modelo matemático considerando al vehículo aéreo como un cuerpo rígido en el espacio y se suman los términos aerodinámicos al modelo, por ejemplo el cálculo de la

matriz de inercia, momentos giroscópicos y Coriolis [35], una vez que se tiene el modelo que representa al vehículo es posible ejecutar simulaciones que demuestren el desempeño de la ley de control empleada y ajustar los parámetros para obtener una navegación fiable antes de implementarlo en campo. Generalmente, el desarrollo de una ley de control para estabilizar el vehículo se torna altamente complejo de procesar por lo que se consideran simplificaciones al modelo, las cuales consisten en linealizarlo o desprestigiar algunos términos aerodinámicos por considerarlos muy pequeños para afectar al sistema, esto ocasiona variaciones entre la simulación y la implementación en un vehículo real.

En este apartado hablaremos de la literatura existente respecto a la técnica de control denominada Backstepping aplicada a los vehículos aéreos no tripulados y a los manipuladores ambos objetos de estudio de esta tesis.

El seguimiento de trayectorias se ha desarrollado con éxito en un hexarrotor e incluso se han propuesto técnicas que proporcionen un control robusto mediante esta técnica [23], se ha demostrado también su alto desempeño en otro tipo de configuraciones como UAV u octorotores [36], de igual forma esta técnica es empleada para la navegación de formaciones de UAV's [37], demostrando un gran desempeño tanto en respuesta transitoria como en estacionaria.

Con el objetivo de presentar controladores más sofisticados que aprovechen las ventajas del Backstepping aunadas a las de alguna técnica adicional algunos autores han propuesto una combinación de controladores utilizando por ejemplo el Backstepping con control por modos deslizantes [38], con lógica difusa [39], algunos autores han explorado su equivalencia con otros sistemas de control [40]. Otros autores se han enfocado en mejorar el desempeño del controlador Backstepping haciéndolo más robusto, implementando técnicas adaptativas al controlador [23] o acoplándolo con sistemas tolerantes a fallos [41].

De igual forma esta técnica se ha aplicado por diversos autores en los manipuladores robóticos siguiendo una metodología similar se obtiene el modelo del manipulador y se diseña la estrategia de control basada en Backstepping [42,43] sin importar el tipo de manipulador o la clase de articulaciones que tenga, esta técnica es aplicable [44,45].

En la misma línea algunos autores han implementado este método en manipuladores industriales con técnicas de inteligencia artificial logrando grandes resultados [46,47]. También se han desarrollado controladores Backstepping del tipo adaptativo e inteligente [48] logrando un buen desempeño aun asumiendo como desconocidos los parámetros iniciales del robot [49]; así mismo se han diseñado controladores basados en Backstepping para obtener la respuesta optima[50], controladores Backstepping tolerantes a fallos [51] o incluso controladores Backstepping diseñados para que un conjunto de manipuladores trabajen de manera conjunta [52].

Con este apartado se deja de manifiesto la alta flexibilidad que presenta la técnica de control Backstepping y sus amplias capacidades al emplearla para el diseño de un controlador robótico sin importar su morfología.

2.4 PID

El PID es, sin lugar a duda, uno de los controladores más estudiados, su uso en este trabajo es para fines comparativos ya que al desarrollarlo se obtiene una referencia fiable y conocida para muchos investigadores, en este apartado se mencionarán algunos de los avances del PID aplicados solamente a los sistemas objeto de estudio de esta tesis.

A pesar de ser un controlador ampliamente estudiado, en bibliografías recientes se ha publicado diversos artículos con PID en UAVs de distinto tipo, logrando un alto desempeño al controlar el vehículo aéreo [53], o algún grado de libertad del mismo [54], algunos autores han controlado vehículos multirrototres de interiores en su totalidad con un PID [55]. Al igual que en otras técnicas diversos autores han propuesto la fusión con técnicas de inteligencia artificial [30,56,57] u otras técnicas como LQR [58] lo que permite obtener controladores PIDs de altas prestaciones o se han enfocado en sistemas controlados con PIDs tolerantes a fallos [59].

Las líneas de investigación continúan abiertas y múltiples trabajos se enfocan a la optimización del controlador PID para logra el seguimiento de una trayectoria de manera más precisa [60], algunos investigadores presentan trabajos en los cuales, un filtro de Kalman junto al PID proporciona un mejor desempeño en el control de los UAV's [61], Otros trabajos de PIDs actualmente están enfocados en el seguimiento de trayectorias en un marco de referencia no inercial [62].

Respecto a los controladores PID implementados en brazos robóticos, a nivel industrial son el elemento más utilizado para el control de manipuladores[63,64], es por ello que actualmente se sigue investigando sobre los beneficios y mejoras que puede aportar un PID al control de los brazos robóticos[65], demostrando en algunos estudios comparativos los puntos a mejorar respecto a otras técnicas[66].

Por ejemplo, algunos estudios se basan en la generación de controladores PID que contrarresten la vibración mecánica [67], otros centran su atención en optimizar el seguimiento de trayectorias [68], algunos más se enfocan en realizar un control en ambientes dinámicos [69], para ello nuevamente se basa en la fusión de dos o más técnicas de control [69,70].

2.5 SISTEMAS UAVS MANIPULADORES

Los sistemas integrados por un vehículo aéreo no tripulado y un manipulador han comenzado a ser estudiados en los últimos años, debido al interés que supone que un UAV pueda trasladar o cambiar algún objeto de su entorno. En estudios como "Design and simulation of an aerial robotic arm" [71] se investiga la estabilidad y adaptabilidad del sistema mientras el sistema se encuentra en maniobras de manipulación, explorando además algunas alternativas para un diseño de brazo ligero; En "Modeling and control aspects of a UAV with an attached manipulator" [72] los autores se centran en el desarrollo del modelo dinámico del conjunto, en el cual se tiene en cuenta la fuerza y el momento transferidos desde la base del manipulador al UAV, en un estudio similar se explica la interacción que tiene el brazo sobre el movimiento del UAV y viceversa [73]; En

“Modeling and controller design of hydraulic rotorcraft aerial manipulator” [74] se describe el proceso para el control de un pequeño brazo hidráulico acoplado a un vehículo de ala rotatoria; Otros autores se centran en el control de robots paralelos [75] o controles robustos a nivel simulación [76]; algunos estudios desarrollan un control satisfactorio mediante técnicas basadas en retroalimentación por visión[77,78], control de impedancia [78], control de tiempo-finito [79], etc.

Por otra parte, algunos estudios centran su atención en la tele operación de estos sistemas[80], en el diseño óptimo del brazo para su uso en multirrotores comerciales [81], o en el aterrizaje del sistema apoyado mediante el manipulador[1,82,83]. Sin embargo, estos estudios abarcan factores más alejados al objeto de estudio de esta tesis, entre los trabajos más similares en la literatura actual podemos encontrar: “Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms” [84] en el cual se presentan dos brazos acoplados a un multirrotor con los cuales se puede interactuar físicamente con el ambiente. “Development of a robust framework for an outdoor mobile manipulation UAV” [85] en el cual se trabaja sobre el diseño y control de un multirrotor de exteriores con un brazo acoplado para la recolección de objetos en lugares desconocidos y “Contact Force Control of an Aerial Manipulator in Pressing an Emergency Switch Process” [86] en el cual se muestra un hexarotor con un manipulador de un grado de libertad.

2.6 SISTEMAS DE RETROALIMENTACION VISUAL

Los sistemas de navegación basados en visión son utilizados en múltiples estudios [14,78,87] sin embargo la mayoría de los sistemas se basan en múltiples cámaras, por medio de múltiples ángulo se estima con precisión la posición del vehículo, sin embargo se trata de un ambiente altamente controlado y en interiores. En “Design, control, and visual navigation of the DelftaCopter VTOL tail-sitter UAV” [88] se navega en exterior por medio de visión artificial utilizando para ello una cámara estereoscópica con deformación de pez.

El sistema de navegación integrado tradicional GPS / IMU tiene muchas limitaciones. Entre las ventajas de la navegación visual es que se cuenta con una gran cantidad de información y precisión, entre las desventajas que se encuentran es la sensibilidad al ruido, en este caso sensibilidad a la luz, para este trabajo se utilizaran Hotspot, este enfoque se ha analizado en el trabajo de revisión “A Survey of UAV Visual Navigation Based on Monocular SLAM” [89] . Adicionalmente se pueden encontrar trabajo para navegación en exteriores donde el uso de GPS no es posible, este tema es abordado en el trabajo de revisión “Survey on UAV navigation in GPS denied environments” [90]

3

DESARROLLO

3. DESARROLLO

El trabajo será dividido en varias etapas: siendo la primera de ellas el trabajo con el UAV, la segunda etapa consta del trabajo con el manipulador, Como tercera etapa de trabajo tenemos el análisis del sistema UAV-Manipulador y como última etapa de trabajo se expone el desarrollo de la etapa de retroalimentación visual para aproximación a infraestructuras.

3.1 DESCRIPCION DEL VEHICULO

El modelo de multirrotor elegido es un hexarrotor, el cual es un vehículo aéreo que dispone de 6 hélices en distribución de X o X6, se basa en el modelo comercial Tarot FY690S, sin embargo, las medidas fueron ajustadas para implementar el equipo personalizado, el vehículo puede observarse en la Figura 1.

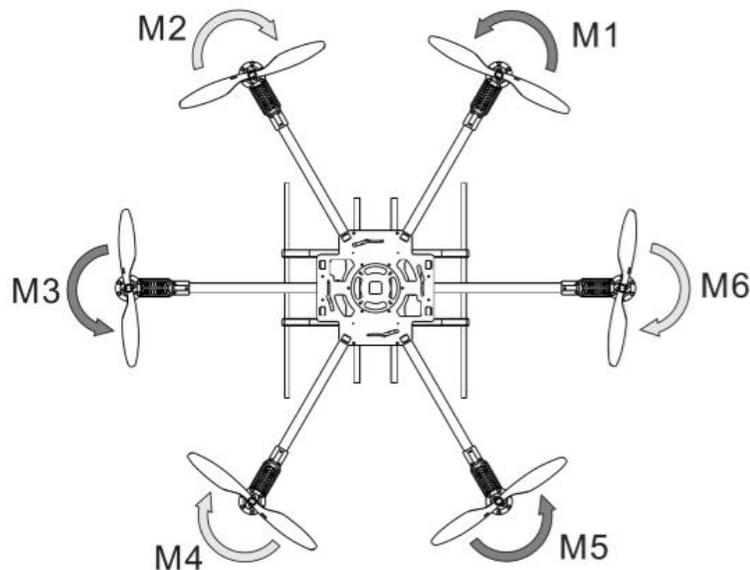


Figura 1. Vehículo multirrotor utilizado.

3.2 MODELADOS DEL VEHÍCULO

Una vez descrito el vehículo a utilizar, es necesario encontrar las ecuaciones que describan su movimiento, sin importar las fuerzas que lo provocan (Modelo cinemático), así como aquellas que describan la interacción de las fuerzas en el movimiento del vehículo (Modelo dinámico), una vez obtenidas serán utilizados para diseñar y validar las leyes de control propuestas.

3.2.1 MODELO CINEMÁTICO

Para la descripción global de los movimientos principales de un multirrotor, así como para permitir el análisis de las fuerzas que actúan sobre el vehículo, se presenta un diagrama (Figura 2) de cuerpo libre donde han de mostrarse sus características principales:

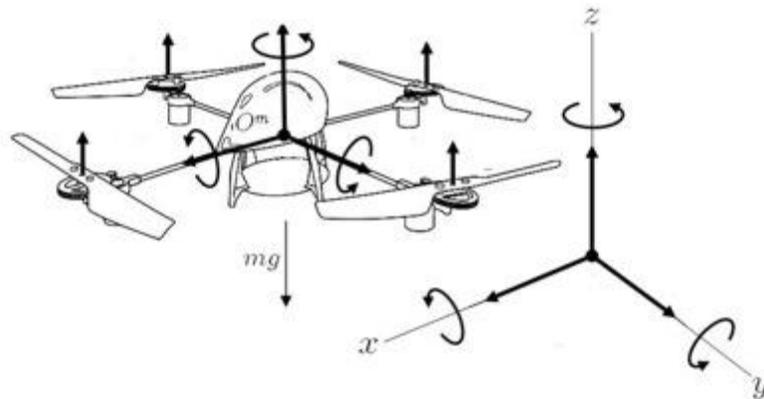


Figura 2. Diagrama de cuerpo libre de un multirrotor.

En el movimiento de un cuerpo rígido que posee seis grados de libertad: tres de ellos definen la posición de un punto de referencia en el cuerpo (generalmente el centro de masas), y los otros tres definen la orientación del cuerpo, por lo que, para obtener el modelo cinemático del vehículo, se supone a este como un cuerpo rígido en el espacio, sujeto a una fuerza principal y tres momentos, los cuales en conjunto representan las entradas de control del sistema. Este trabajo usará los ángulos de navegación (ψ , θ , ϕ) definidos por Tait-Bryan (Figura 3) para describir la rotación del sistema en el espacio tridimensional, dichos ángulos se basan en tres rotaciones descritas por una matriz de transformación que representa los giros realizados sobre los tres ejes de rotación del sistema x , y , z .

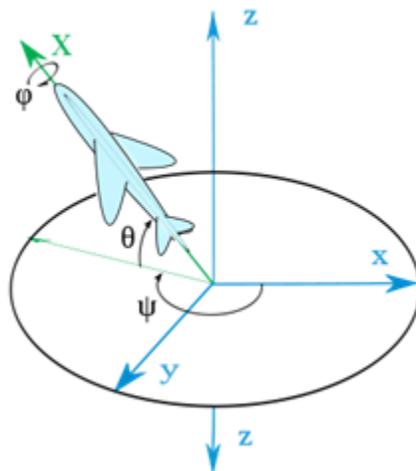


Figura 3. Ángulos de Tait-Bryan

La rotación de un sólido rígido en el espacio puede ser representada por una matriz de rotación descrita mediante la expresión iR_h . Dicha matriz puede ser calculada representando los ejes principales de un sistema libre F_h en términos de un sistema fijo F_i , es decir:

$${}^i R_h = \begin{bmatrix} \cos(\alpha_{x_h x_i}) & \cos(\alpha_{y_h x_i}) & \cos(\alpha_{z_h x_i}) \\ \cos(\alpha_{x_h y_i}) & \cos(\alpha_{y_h y_i}) & \cos(\alpha_{z_h y_i}) \\ \cos(\alpha_{x_h z_i}) & \cos(\alpha_{y_h z_i}) & \cos(\alpha_{z_h z_i}) \end{bmatrix} \quad (1)$$

Utilizando la expresión anterior como referencia, es posible obtener las matrices de transformación para cada eje principal (R_x, R_y, R_z) mostrado en la Figura 4. Por tanto, se tiene que las matrices de rotación alrededor de los ejes principales son:

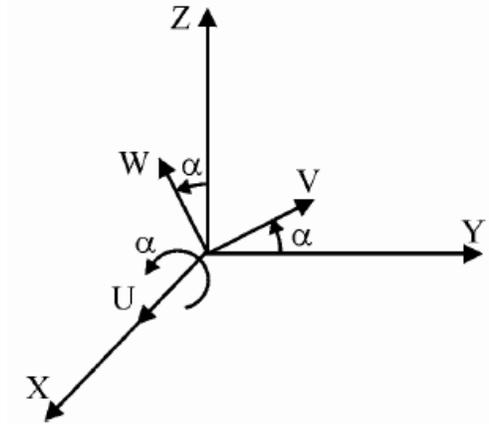


Figura 4. Rotación básica, en sentido antihorario, del ángulo ϕ sobre el eje "x"

$$R_x^{+\phi} = \begin{bmatrix} \cos(0) & \cos(90^\circ) & \cos(90^\circ) \\ \cos(90^\circ) & \cos(\phi) & \cos(90^\circ - \phi) \\ \cos(90^\circ) & \cos(90^\circ + \phi) & \cos(\phi) \end{bmatrix} \quad (2)$$

Por lo que al realizar las operaciones correspondientes se obtiene:

$$R_x^{+\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3)$$

De igual forma para $R_y^{+\theta}$ (Figura 5) se tiene:

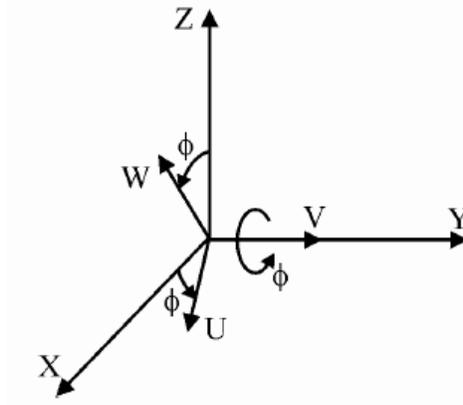


Figura 5. Rotación básica, en sentido antihorario, del ángulo θ sobre el eje "y"

$$R_y^{+\theta} = \begin{bmatrix} \cos(\theta) & \cos(90^\circ) & \cos(90^\circ + \theta) \\ \cos(90^\circ) & \cos(0) & \cos(90^\circ) \\ \cos(90^\circ - \theta) & \cos(90^\circ) & \cos(\theta) \end{bmatrix} \quad (4)$$

Simplificando la expresión anterior obtenemos:

$$R_y^{+\theta} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (5)$$

Finalmente, para el cálculo de la rotación sobre el eje z (Figura 6), se tiene:

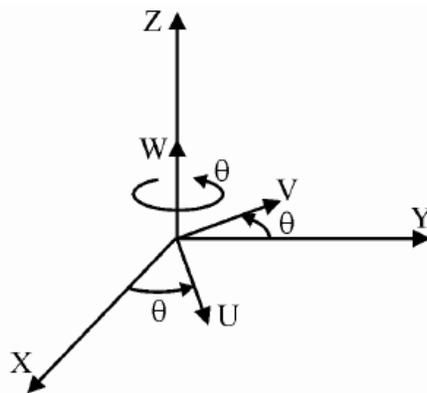


Figura 6. Rotación básica, en sentido antihorario, del ángulo ψ sobre el eje "z"

$$R_z^{+\psi} = \begin{bmatrix} \cos(\psi) & \cos(90^\circ - \psi) & \cos(90^\circ) \\ \cos(90^\circ + \psi) & \cos(\psi) & \cos(90^\circ) \\ \cos(90^\circ) & \cos(90^\circ) & \cos(0) \end{bmatrix} \quad (6)$$

Simplificando:

$$R_z^{+\psi} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Tal como se ha explicado en el apartado anterior y tomando en consideración las expresiones anteriores, cualquier rotación en el espacio tridimensional puede ser descrita por una matriz R_I de la forma:

$$R_I = \begin{bmatrix} \cos \Psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (8)$$

A partir de esta expresión es posible determinar las relaciones existentes entre las velocidades del sistema, para dos marcos de referencia dependientes (Figura 7):

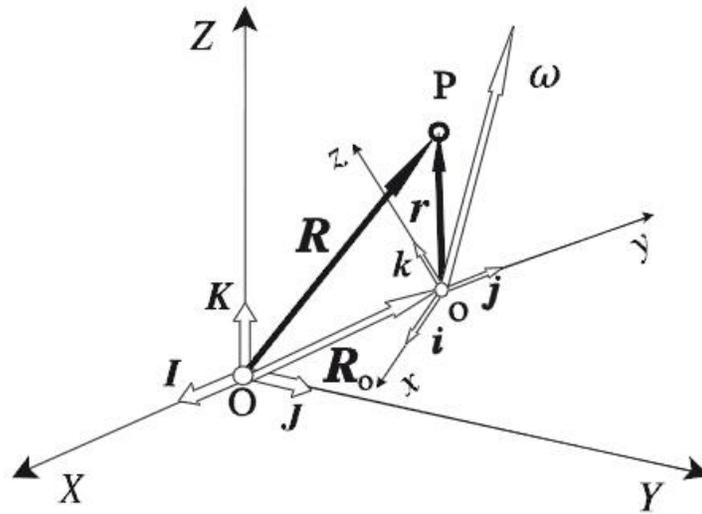


Figura 7. Sistemas de referencia dependientes en el espacio tridimensional.

Con el fin de obtener una expresión matemática que sea capaz de representar la relación existente entre velocidades angulares y lineales se tomará un vector de posición $A_{|I}$ de un punto P, definido en un marco de referencia I, a partir de lo cual es posible transformar este vector al marco de referencia "a", tomando como base las ecuaciones de rotación obtenidas en el apartado anterior, planteando así el siguiente supuesto:

$$A_{|a} = R_{Ia} A_{|I} \quad (9)$$

Donde M_{Ia} denota una matriz de transformación y es similar a: $R_{Ia} = R_x^{+\phi} R_y^{+\theta} R_z^{+\psi}$, y derivando respecto al tiempo, tenemos que $\dot{A}_{|a}$ es:

$$\dot{A}_{|a} = R_{Ia} \dot{A}_{|I} + \dot{R}_{Ia} A_{|I} \quad (10)$$

Despejando de la expresión anterior, tenemos que: $R_{Ia}\dot{A}_{|I} = \dot{A}_{|a} - \dot{R}_{Ia}A_{|I}$, por lo tanto podemos decir que:

$$A_{|I} = R_{Ia}^T A_{|a} = R_{aI} A_{|a} \quad (11)$$

Una vez realizado lo anterior podemos encontrar que, si sustituimos el valor de $A_{|I}$, se tiene que: $R_{Ia}\dot{A}_{|I} = \dot{A}_{|a} - \dot{R}_{Ia}R_{aI}A_{|a}$, y tomando en consideración las propiedades de matrices, se tiene:

$$R_{Ia}\dot{A}_{|I} = \dot{A}_{|a} + R_{Ia}\dot{R}_{aI}A_{|a} \quad (12)$$

De las propiedades de matrices se sabe que $R_{Ia}\dot{R}_{aI}$ dan como resultado una matriz simétrica, por lo que a partir de las operaciones entre matrices sabemos que $R_{Ia}\dot{R}_{aI}A_{|a}$ es igual a $\Omega_{|a}XA_{|a}$, por lo que podemos obtener finalmente:

$$R_{Ia}\dot{A}_{|I} = \dot{A}_{|a} + \Omega_{|a}XA_{|a} = V_{|a} \quad (13)$$

Es decir:

$$V_{|a} = R_{Ia}\dot{A}_{|I} \quad (14)$$

Una vez obtenido lo anterior es esencial describir una expresión que relacione tanto velocidades angulares como sus respectivas posiciones, por lo que si suponemos un vector de posición P definido en un marco de referencia I (R_I), y se pretende rotar otro sistema respecto a este, en este caso un sistema con coordenadas X_b, Y_b, Z_b , la expresión que relaciona esta dependencia se define de la siguiente forma:

$$P_{|b} = R_{Ib}P_{|I} \quad (15)$$

Partiendo de las ecuaciones anteriores es posible reescribir la expresión anterior tal como sigue:

$$P_{|b} = R_x^{+\phi} R_y^{+\theta} R_z^{+\psi} P_{|I} \quad (16)$$

Despejando se obtiene:

$$P_{|I} = R_{Ib}^T P_{|b} \quad (17)$$

Ahora si dejamos que $P_{|b}$ sea un vector de posición, con:

$$\frac{dP_{|b}}{dt} = 0 \quad (18)$$

Entonces las mismas condiciones son aplicables para $V_{|I}$, por tanto: $V_{|I} = R_{Ib}^T V_{|b}$, obteniendo así:

$$V_{|I} = \frac{dP_{|I}}{dt} = \frac{d(R_{Ib}^T P_{|b})}{dt} \quad (19)$$

Tal como se ha comentado durante el transcurso de esta sección, si se considera $P_{|b}$ como constante se puede decir:

$$V_{|l} = \frac{dP_{|l}}{dt} = \frac{d(R_{|b}^T P_{|b})}{dt} = \frac{d(R_{|b}^T)}{dt} P_{|b} = R_{|b}^T V_{|b} \quad (20)$$

Finalmente se obtiene:

$$V_{|l} = R_{|b} \frac{d(R_{|b}^T)}{dt} P_{|b} = (R_x^{+\phi} R_y^{+\theta} R_z^{+\psi} R_z^{\dot{\psi}} R_y^{-\theta} R_x^{-\phi} + R_x^{+\phi} R_y^{+\theta} R_y^{\dot{\theta}} R_x^{-\phi} + R_x^{-\phi} R_x^{\dot{\phi}}) R_{|b} \quad (21)$$

Para la obtención de la expresión que relacione velocidades angulares y partiendo de las operaciones con matrices, es posible establecer que un producto descrito por: $R_{|b} R_{|b}^T$, genere una matriz anti-simétrica de la forma:

$$R_{|b} R_{|b}^T = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (22)$$

De igual forma se sabe que la multiplicación de una matriz antisimétrica con un vector es igual al producto cruz entre dicho vector y el formado por los componentes principales de la matriz, es decir:

$$\begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (23)$$

Una vez establecidas las relaciones anteriores se realiza una simplificación de los términos de la ecuación (23), obteniendo así la ecuación (24):

$$\begin{bmatrix} 0 & -\dot{\psi} \cos(\theta) \cos(\phi) + \dot{\theta} \sin(\theta) & \dot{\psi} \cos(\theta) \sin(\phi) + \dot{\theta} \cos(\theta) \\ \dot{\psi} \cos(\theta) \cos(\phi) - \dot{\theta} \sin(\theta) & 0 & \dot{\psi} \sin(\theta) - \dot{\phi} \\ -\dot{\psi} \cos(\theta) \sin(\phi) - \dot{\theta} \cos(\theta) & \dot{\psi} \sin(\theta) + \dot{\phi} & 0 \end{bmatrix} \quad (24)$$

Tal como se puede apreciar se trata de una matriz antisimétrica, por lo que finalmente con las expresiones (23) y (24) se obtiene la expresión (25):

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi} \sin(\theta) \\ q &= \dot{\theta} \cos(\phi) + \dot{\psi} \cos(\theta) \sin(\phi) \\ r &= -\dot{\theta} \sin(\phi) + \dot{\psi} \cos(\theta) \cos(\phi) \end{aligned} \quad (25)$$

Expresando las ecuaciones anteriores en forma Matricial:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (26)$$

De dónde (p, q, r) son las velocidades angulares en el sistema de coordenadas fijo en el espacio tridimensional. Lo que permite relacionar las velocidades angulares entre marcos de referencia dependientes.

Es sabido que el movimiento de un cuerpo rígido en el espacio se describe a través de sus velocidades lineales y angulares, por lo que, si escribimos estas velocidades con respecto al marco de referencia fijo P_h , tenemos que, para las velocidades lineales, la expresión obtenida será:

$$V = [{}^h v_x, {}^h v_y, {}^h v_z]' \quad (27)$$

La relación existente entre las velocidades lineales respecto al marco de referencia fijo del cuerpo (V) y el marco de referencia del mundo ($\dot{\xi}$), está dada por la siguiente expresión:

$$\dot{\xi} = R_I V \quad (28)$$

Donde $R_I \in \mathbb{R}^3$ representa la matriz de transformación que describe la rotación del sistema mediante el giro sobre los tres ejes principales, es decir:

$$R_I = R_{z,\psi} R_{y,\theta} R_{x,\phi} \quad (29)$$

Por lo que, utilizando la convención de la mano derecha, se obtiene:

$$R_{z,\psi} R_{y,\theta} R_{x,\phi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (30)$$

Realizando las operaciones entre matrices descritas por la expresión anterior obtenemos:

$$R_I = \begin{bmatrix} \cos \Psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (31)$$

3.3 MODELO DINÁMICO

El modelo dinámico determina la fuerza y los torques que se requieren para generar movimiento en el dron. Definidos los siguientes vectores:

$$\xi = [x, y, z]' \in \mathbb{R}^3; \eta = [\phi, \theta, \psi]' \in \mathbb{R}^3$$

Donde η :

$$\phi = Roll \left(-\frac{\pi}{2} < \phi < \frac{\pi}{2} \right), \quad \theta = Pitch \left(-\frac{\pi}{2} < \theta < \frac{\pi}{2} \right), \quad \psi = yaw \left(-\pi < \psi < \pi \right)$$

3.3.1 MODELADO DINÁMICO DEL UAV

El modelo dinámico del UAV establece cuáles son las fuerzas y los momentos actuantes respecto al sistema de coordenadas establecido. Por lo que obtener las expresiones que nos brinden tal información es de gran importancia para poder realizar una estrategia de control que nos permita obtener una manipulación completa del vehículo. Las ecuaciones que describen la dinámica de un sólido rígido en el espacio se pueden resumir en la siguiente expresión:

$$F + F_d = m\dot{v} + \omega X(mv) \quad (32)$$

$$\tau + \tau_d = J\dot{\omega} + \omega X(J\omega) \quad (33)$$

De las ecuaciones anteriores se tiene que: V es el vector de velocidad traslacional respecto al respecto al marco de referencia fijo, ω es la velocidad angular, m es la masa total del helicóptero, F representa el vector de fuerzas que actúan sobre el vehículo, τ representa los torques aplicados al sistema y $J \in \mathbb{R}^{3 \times 3}$ es la matriz de inercia, la cual es utilizada como una medida de la inercia rotacional de un cuerpo. Sabiendo que en el caso más general se utiliza el tensor de inercia:

$$J = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix} \quad (34)$$

Sin embargo, cuando un cuerpo rígido gira en torno a uno de los ejes principales, la inercia rotacional puede ser representada como una magnitud escalar llamada momento de inercia, elementos que compondrán la matriz de inercia correspondiente:

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (35)$$

De esta manera la expresión anterior determina el valor de las inercias que actúan sobre el vehículo, y son indispensables para la obtención del modelo matemático general del sistema.

Una vez consideradas las expresiones anteriores y tomando como base el diagrama de cuerpo libre de la Figura 7, es posible establecer las siguientes relaciones:

$$\varepsilon = [x \ y \ z] \quad (36)$$

Tal como se ha dicho con anterioridad esta expresión representa la posición del vehículo, por lo que su derivada temporal vendrá dada por:

$$\dot{\varepsilon} = [\dot{x} \ \dot{y} \ \dot{z}] \quad (37)$$

Del vector de estados sabemos que: $v = [\dot{x} \ \dot{y} \ \dot{z}]$, por lo tanto es posible reescribir esta expresión como: $v = [v_x \ v_y \ v_z]$, partiendo de esto se puede decir que:

$$\dot{\varepsilon} = v \quad (38)$$

Ahora se sabe que la relación existente entre las velocidades descritas por un cuerpo rígido, el cual se encuentra referenciado mediante marcos de referencia dependientes está dada por las siguientes expresiones:

$$\dot{\epsilon} = R_I V \quad (39)$$

Donde R_I representa la matriz de transformación entre marcos de referencia dependientes. Paso seguido y sabiendo de la formulación de Newton que $F = ma$, es posible reescribir el conjunto de expresiones anteriores de la siguiente manera:

$$m\dot{v} = R_I F_b \quad (40)$$

Donde F_b representa las fuerzas externas aplicadas al cuerpo del vehículo, por lo que podemos decir que las fuerzas externas aplicadas al vehículo vienen representadas mediante la expresión:

$$F_{ext} = R_I F_b \quad (41)$$

por lo que es posible descomponer esta expresión, ya que las fuerzas externas pueden ser representadas de manera individual, por lo que reescribiendo se tiene:

$$F_{ext} = F_g + F_{emp} + A_T \quad (42)$$

Dónde F_g es la fuerza que se presenta debido a la gravedad: $F_g = mg$, de donde se sabe que g representa la constante de gravedad ($g = 9.81$), y A_T se define como la fuerza aerodinámica:

$$A_T = K_t V \quad (43)$$

Con K_t como una matriz diagonal que expresa los parámetros de fricción. De la misma forma sabemos que F_{emp} es la fuerza producida por el empuje de las hélices y puede ser representada por el vector:

$$F_{emp} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^6 |f_i| \end{bmatrix} \quad (44)$$

Donde f_i representa el empuje generado por cada motor y puede ser expresado de la siguiente manera: $f = \rho C_T A R^2 \Omega^2$, donde ρ es la densidad del aire, C_T es el coeficiente de empuje, A es el área del disco del rotor, y R el radio de la pala. Si se sabe que $f = b\Omega^2$, es posible reescribir la expresión que representa las fuerzas de empuje de la siguiente manera:

$$F_{emp} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^6 |b\Omega_i^2| \end{bmatrix} \quad (45)$$

Al tratarse de una estructura compleja y totalmente codependiente, es necesario conocer la fuerza de empuje en cada eje para lograr un control eficiente del sistema, por lo que para obtener el resultado deseado y sabiendo que se trata de un cuerpo rígido representado por sistemas de referencia dependientes, es necesario utilizar la matriz de transformación R_I , por lo que se tiene:

$$F_{emp} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^6 |b\Omega_i^2| \end{bmatrix} R_I \quad (46)$$

Dado lo anterior es posible reescribir la expresión que denota las fuerzas externas de la siguiente forma:

$$R_I F_b = F_g + F_{emp} + A_T \quad (47)$$

Sustituyendo los valores correspondientes a los términos del lado derecho de la ecuación tenemos que:

$$R_I F_b = -mg + A_T + R_I \sum_{i=1}^6 b\Omega_i^2 \quad (48)$$

Por lo que, para definir la correcta operación vectorial en la ecuación anterior, es necesario multiplicar el primer término por un vector $[0 \ 0 \ 1]^T = E_3$, por lo tanto:

$$R_I F_b = -mg \cdot E_3 + R_I \sum_{i=1}^6 b\Omega_i^2 + A_T \quad (49)$$

Como parte fundamental de este desarrollo es necesario determinar los torques externos aplicadas al vehículo, por lo que partiendo de la expresión (33) es posible obtener:

$$\tau_b = J\dot{\omega} + \omega X J \omega \quad (50)$$

Si de la expresión anterior se sabe que τ_b representa las torques externas aplicadas al cuerpo y que puede ser representado de la forma:

$$\tau_b = \tau_a + A_R \quad (51)$$

Donde A_R representa los torques aerodinámicos y viene determinado por la expresión: $A_R = K_r \omega$.

De lo anterior y partiendo de la consideración que establece a: $\omega = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]$, aunado a que K_r representa los coeficientes de fricción de los pares producidos por los motores y que τ_a es una matriz con los torques generados en cada eje, por lo que podemos determinar el valor de τ_a como:

$$\tau_a = \begin{bmatrix} l(f_5 - f_2 + \sin 30)(f_6 - f_1 + f_4 - f_3) \\ l(f_1 + f_6 - f_3 - f_4) \\ \sum_{i=1}^6 (-1)^{i+1} Q_i \end{bmatrix} \quad (37)$$

Donde Q_i representa los torques generados por cada uno de los cuatro motores, y viene representado por la siguiente expresión: $Q_i = \rho C_q A R^3 \omega^2$. Tal como se mencionó en apartados anteriores τ_a representa los torques generados por las fuerzas de empuje y es un vector con las torques generadas en cada eje.

Una vez determinadas las ecuaciones anteriores, es posible representar el comportamiento dinámico del sistema a través del siguiente conjunto de ecuaciones:

$$\dot{\varepsilon} = v \quad (52)$$

$$m\dot{v} = R_I F_b \quad (53)$$

$$\omega = R_r \dot{\eta} \quad (54)$$

$$J\dot{\omega} = -\omega X J \omega + \tau_b \quad (55)$$

Considerando el valor de los términos de las ecuaciones anteriores es posible reescribirlas de la siguiente manera:

$$\dot{\varepsilon} = v \quad (56)$$

$$m\dot{v} = -mg \cdot E_3 + R_I \sum_{i=1}^4 b\Omega_i^2 + A_T \quad (57)$$

$$\omega = R_r \quad (58)$$

$$J\dot{\omega} = -\omega X J \omega + \tau_a + A_R \omega = R_r \dot{\eta} \quad (59)$$

De esta manera se obtienen las expresiones principales que han de describir el comportamiento dinámico del sistema. Es importante señalar que, al trabajar con la formulación de Newton, es indispensable obtener el valor de las aceleraciones lineales por lo que de la ecuación (57) se despejara \dot{v} obteniendo así el valor de la aceleración la cual viene representada de la forma:

$$\dot{v} = -g \cdot E_3 + \frac{1}{m} R_I \sum_{i=1}^4 b\Omega_i^2 + \frac{1}{m} A_T \quad (60)$$

La sumatoria de fuerzas traslacionales que actúan sobre el vehículo, tal y como se mencionó anteriormente está compuesto por el empuje total generado por la suma de los cuatro rotores, la fuerza gravitacional y por la fuerza aerodinámica. Dado lo anterior es posible describir la fuerza principal U_1 que representa la entrada de control aplicada al vehículo, es decir:

$$U_1 = \left(\sum_{i=1}^6 f_i \right) = \left(\sum_{i=1}^6 b\Omega_i^2 \right) \quad (61)$$

Donde f_i es la fuerza de empuje generada por cada rotor. Al igual que la entrada de control anterior si se consideran las entradas de control U_2 , U_3 y U_4 las cuales son las fuerzas que actúan para realizar el giro en ϕ , θ y ψ respectivamente, es posible obtener las expresiones matemáticas que describan su comportamiento, es decir:

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} lb(\Omega_3^2 + \Omega_4^2 - \Omega_5^2 - \Omega_6^2) \\ \frac{1}{2} lb(\Omega_3^2 + \Omega_4^2 - \Omega_5^2 - \Omega_6^2 - 2\Omega_2^2 + 2\Omega_1^2) \\ K_T \sum_{i=1}^6 (-1)^{i+1} \Omega_i^2 \end{bmatrix} \quad (62)$$

Así, el par de control que representa la actuación general en los tres ejes de movimiento del sistema viene dado por:

$$\tau_a = \begin{bmatrix} \frac{\sqrt{3}}{2} l(f_3 + f_4 - f_5 - f_6) \\ \frac{1}{2} l(f_3 + f_4 - f_5 - f_6 - 2f_2 + 2f_1) \\ \sum_{i=1}^6 (-1)^{i+1} Q_i \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} lb(\Omega_3^2 + \Omega_4^2 - \Omega_5^2 - \Omega_6^2) \\ \frac{1}{2} lb(\Omega_3^2 + \Omega_4^2 - \Omega_5^2 - \Omega_6^2 - 2\Omega_2^2 + 2\Omega_1^2) \\ K_T \sum_{i=1}^6 (-1)^{i+1} \Omega_i^2 \end{bmatrix} \quad (63)$$

Dadas las expresiones anteriores y sabiendo que $\dot{\varepsilon} = v$, donde $v = [u_o \ v_o \ w_o]$ y: $\dot{x} = u_o$, $\dot{y} = v_o$, $\dot{z} = w_o$, es posible reescribir la expresión (60) en forma matricial obteniendo así:

$$\dot{v} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} \cos \Psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} [U_1] + \frac{A_T}{m} \quad (64)$$

Realizando las operaciones, de la expresión anterior se obtiene:

$$\dot{v} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} [U_1] + \begin{bmatrix} \frac{A_x}{m} \\ \frac{A_y}{m} \\ \frac{A_z}{m} \end{bmatrix} \quad (65)$$

Puesto que $\dot{v} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$ y considerando la expresión anterior obtenemos finalmente:

$$\begin{aligned} \ddot{x} &= (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \frac{U_1}{m} + \frac{A_x}{m} \\ \ddot{y} &= (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \frac{U_1}{m} + \frac{A_y}{m} \\ \ddot{z} &= -g + (\cos \theta \cos \phi) \frac{U_1}{m} + \frac{A_z}{m} \end{aligned} \quad (66)$$

Tal como se ha realizado el procedimiento para la obtención de las expresiones que representan las aceleraciones lineales, para obtener la relación de ecuaciones que describan las aceleraciones angulares, se comienza despejándolas de la ecuación 45 teniendo así:

$$\dot{\omega} = \begin{bmatrix} \frac{A_p}{I_{xx}} + \frac{lU_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} \\ \frac{A_q}{I_{yy}} + \frac{l}{I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} \\ \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\theta} \dot{\phi} \end{bmatrix} \quad (67)$$

Dada la expresión anterior y puesto que $\dot{\omega} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}$ es posible obtener finalmente las expresiones que determinan el comportamiento de las velocidades angulares del sistema, es decir:

$$\begin{aligned} \ddot{\phi} &= \frac{A_p}{I_{xx}} + \frac{lU_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} \\ \ddot{\theta} &= \frac{A_q}{I_{yy}} + \frac{l}{I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} \\ \ddot{\psi} &= \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\theta} \dot{\phi} \end{aligned} \quad (68)$$

Como se puede observar en las expresiones anteriores se ha excluido la consideración de los efectos giroscópicos sobre el sistema, por lo que para obtener un conjunto que relaciones estos efectos, y de esta manera obtener un modelo matemático más detallado que permita un mejor análisis del comportamiento del vehículo, es necesario realizar algunos cambios en estas expresiones. Para introducir los efectos generados por los pares giroscópicos en las ecuaciones del sistema dinámico es necesario modificar la expresión:

$$\tau_b = J\dot{\omega} + \omega X J \omega \quad (69)$$

Es decir, a través de la ecuación anterior se harán notar los efectos giroscópicos puesto que estos generan momentos que actúan directamente sobre el vehículo. Dado lo anterior y tomando como base la representación de los pares aplicados sobre el UAV, es posible reescribir la expresión anterior de la siguiente forma:

$$\tau_b = \tau_a + \tau_{Ga} + A_R \quad (70)$$

Donde τ_{Ga} representa el momento giroscópico, el cual se genera debido a que el rotor que gira con una velocidad Ω , es forzado a girar en el eje de referencia, con una velocidad angular ω , es decir:

$$\tau_{Ga} = - \sum_{i=1}^6 J_R(\omega X E_3) \cdot \Omega_i \quad (71)$$

Donde J_R es el momento de inercia rotacional del rotor alrededor de su eje. Dadas las expresiones anteriores, las ecuaciones que representan la dinámica del UAV se puede reescribir de la siguiente manera:

$$\dot{\epsilon} = v \quad (72)$$

$$\dot{v} = -g \cdot E_3 + \frac{1}{m} R_I \sum_{i=1}^6 b \Omega_i^2 + \frac{1}{m} A_T \quad (73)$$

$$\omega = R_r \quad (74)$$

$$J \dot{\omega} = -\omega X J \omega - \sum_{i=1}^6 J_R(\omega X E_3) \cdot \Omega_i + \tau_a + A_R \quad (75)$$

Dado que los pares giroscópicos afectan solamente a las expresiones que denotan el comportamiento de las aceleraciones angulares del sistema, las ecuaciones (72) a (74) no sufren cambio alguno. Por otra parte, las nuevas expresiones que describen las aceleraciones angulares se pueden obtener despejando de la ecuación (75) teniendo así:

$$\dot{\omega} = \begin{bmatrix} \frac{A_p}{I_{xx}} + \frac{l U_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \psi - \frac{J_R}{I_{xx}} \Omega \dot{\theta} \\ \frac{A_q}{I_{yy}} + \frac{l}{I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \psi + \frac{J_R}{I_{yy}} \Omega \dot{\phi} \\ \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\theta} \dot{\phi} \end{bmatrix} \quad (76)$$

Donde Ω es la suma de las velocidades de los rotores, es decir: $\Omega = \omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 + \omega_6$, partiendo de esta expresión y sabiendo que $\dot{\omega} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}$ se obtiene finalmente:

$$\ddot{\phi} = \frac{A_p}{I_{xx}} + \frac{lU_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} - \frac{J_R}{I_{xx}} \Omega \dot{\theta} \quad (77)$$

$$\ddot{\theta} = \frac{A_q}{I_{yy}} + \frac{l}{I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} + \frac{J_R}{I_{yy}} \Omega \dot{\phi} \quad (78)$$

$$\ddot{\psi} = \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\theta} \dot{\phi} \quad (79)$$

3.4 CONTROL DEL UAV

Concluido el modelado del UAV se procede a desarrollar las leyes de control que permitan el seguimiento de una trayectoria dada, para ello se consideran la técnica denominada Backstepping la cual presenta una buena relación en cuanto robustez/costo computacional, es procedimiento recursivo y permite dividir un sistema complejo en subsistemas más simples, para objeto de comparación se desarrolla también un sistema de control PID con el mismo objetivo.

3.4.1 CONTROL BACKSTTEPING

Backstepping es un procedimiento de control recursivo, el cual está basado en la teoría de Lyapunov y es útil en el diseño de control por realimentación, permite controlar sistemas dinámicos no lineales y presenta un alto desempeño ante incertidumbres paramétricas y posibles errores en el modelo del sistema, es por ello por lo que se ha seleccionado esta técnica para el desarrollo de un controlador que permita al UAV realizar el seguimiento de una trayectoria de referencia.

La estrategia de control Backstepping se caracteriza por descomponer el sistema original en una secuencia de subsistemas de orden reducido (que pueden llegar a ser escalares). Esta notable característica permite utilizar la flexibilidad adicional que existe con sistemas de bajo orden y escalares, lo que ayuda a disminuir la complejidad de los modelos utilizados para la obtención del control del sistema. A través de la metodología Backstepping es posible resolver problemas de estabilización, seguimiento y control robusto bajo condiciones menos restrictivas que las encontradas en otros métodos, lo que la convierte en una técnica atractiva para su implementación en el desarrollo de este trabajo. Es importante señalar que la estructura principal de esta técnica, se fundamenta principalmente en el método directo de Lyapunov, el cual plantea el uso de una función candidata que debe cumplir ciertos parámetros para garantizar la estabilidad del sistema, por lo que si se pretende reducir el error del sistema, dicha función debe

definirse sobre los errores de seguimiento de la salida del mismo, lo que permitirá el desarrollo de una estrategia de control que garantice la estabilidad continua del sistema.

La metodología de control Backstepping representa un poderoso método recursivo que permite obtener controladores robustos, los cuales garantizan un alto desempeño y un bajo índice de error bajo incertidumbres de modelado. El hecho de obtener tales características en estos controladores se debe a que esta metodología, combina la elección de una función de Lyapunov con el diseño de un control en realimentación, por lo que es posible resolver problemas de estabilización y seguimiento de trayectorias, por lo que la metodología de control Backstepping se convierte en una opción adecuada para el desarrollo de este trabajo.

3.4.1.1 CONTROL BACKSTEPPING DEL UAV

Para el desarrollo del control del sistema se parte de las siguientes ecuaciones obtenidas en la sección de modelado y que describen la dinámica del vehículo:

$$\ddot{x} = (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \frac{U_1}{m} + \frac{A_x}{m}$$

$$\ddot{y} = (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \frac{U_1}{m} + \frac{A_y}{m}$$

$$\ddot{z} = -g + (\cos \theta \cos \phi) \frac{U_1}{m} + \frac{A_z}{m}$$

$$\ddot{\phi} = \frac{A_p}{I_{xx}} + \frac{\sqrt{3}lU_2}{2I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta}\dot{\psi}$$

$$\ddot{\theta} = \frac{A_q}{I_{yy}} + \frac{l}{2I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi}\dot{\psi}$$

$$\ddot{\psi} = \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\theta}\dot{\psi}$$

Para diseñar el controlador se reescribe el sistema anterior en la forma de espacio de estados $\dot{X} = f(X, U)$. En ingeniería de control, una representación de espacios de estados es un modelo matemático de un sistema físico descrito mediante un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales de primer orden que se combinan en una ecuación diferencial matricial de primer orden. Para prescindir del número de entradas, salidas y estados, las variables son expresadas como vectores y las ecuaciones algebraicas se escriben en forma matricial (esto último sólo puede hacerse cuando el sistema dinámico es lineal e invariante en el tiempo). La representación de espacios de estado (también conocida como aproximación en el dominio del tiempo) provee un modo compacto y conveniente de modelar y analizar sistemas con múltiples entradas y salidas. El espacio de estado se refiere al espacio de n dimensiones cuyos ejes coordenados están formados por variables de estados. Por tanto, el estado del sistema puede ser representado como un vector dentro de ese espacio.

Consideremos entonces el vector $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})$ como el vector de estados del sistema. Dónde:

$$\begin{aligned} x_1 &= \phi, x_2 = \dot{x}_1 = \dot{\phi}, & x_3 &= \theta, x_4 = \dot{x}_3 = \dot{\theta} \\ x_5 &= \psi, x_6 = \dot{x}_5 = \dot{\psi}, & x_7 &= z, x_8 = \dot{x}_7 = \dot{z} \\ x_9 &= x, x_{10} = \dot{x}_9 = \dot{x}, & x_{11} &= y, x_{12} = \dot{x}_{11} = \dot{y} \end{aligned}$$

Una vez descritas las relaciones anteriores el vector de estado puede reescribirse de la siguiente manera:

$$\dot{X} = \begin{bmatrix} x_2 \\ \frac{lU_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} x_4 x_6 \\ x_4 \\ \frac{l}{I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} x_2 x_6 \\ x_6 \\ \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \theta \dot{\psi} \\ x_8 \\ -g + (\cos x_3 \cos x_1) \frac{U_1}{m} + \frac{A_z}{m} \\ x_{10} \\ u_x \frac{U_1}{m} \\ x_{12} \\ u_y \frac{U_1}{m} \end{bmatrix} \quad (80)$$

Para lograr la simplificación de los últimos términos se ha definido:

$$\begin{aligned} u_x &= (\sin x_5 \sin x_3 \cos x_1 - \cos x_5 \sin x_1) \\ u_y &= (\cos x_5 \sin x_3 \cos x_1 + \sin x_5 \sin x_1) \end{aligned} \quad (81)$$

Es importante señalar que las expresiones anteriores definen el movimiento del vehículo sobre el plano x-y, lo que sugiere la necesidad de diseñar una señal de control específica para cada una de las expresiones anteriores, para poder lograr un control eficiente sobre el plano bidimensional x-y. Para el control del sistema global se crean entonces las señales de control que permitan modificar la Altura (U_1), Roll, Pitch y Yaw (U_2 , U_3 y U_4 respectivamente), además de las señales de control virtual U_x y U_y que nos permitirán calcular los ángulos Roll y Pitch.

Control para U_2 : El fundamento de la estrategia de control Backstepping, al igual que la mayoría de las técnicas de control, se basan en reducir el error del sistema por lo tanto para proponer la señal de control U_2 , se partirá del error del sistema definido para este caso por:

$$z_1 = x_{1d} - x_1$$

Según el teorema de Lyapunov se debe considerar una función candidata a Lyapunov $V(z)$ definida positiva como la presentada a continuación y su derivada respecto al tiempo debe ser semidefinida negativa, es decir:

$$V(z_1) = \frac{1}{2} z_1^2$$

Cuya derivada es: $\dot{V}(z_1) = z_1 \dot{z}_1$, dónde:

$$z_1 = x_{1d} - x_1$$

Lo que implica que su deriva temporal viene determinada por la siguiente expresión:

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$$

Dada la expresión anterior y sabiendo que: $\dot{x}_1 = x_2$, se puede escribir la derivada de la función de Lyapunov como:

$$\dot{V}(z_1) = z_1 [\dot{x}_{1d} - x_2]$$

Por lo que la estabilización de Z_1 puede ser obtenida introduciendo la siguiente entrada de control virtual x_2 :

$$x_2 = \dot{x}_{1d} - \alpha_1 z_1$$

El nuevo sistema debe minimizar un error, por lo que el nuevo sistema con el que se trabaja viene representado por el siguiente conjunto de ecuaciones:

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$$

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1$$

Dónde: $\alpha_1 < 0$. Tomando el valor de x_2 de la ecuación anterior podemos reescribir la derivada de la función de Lyapunov.

$$\dot{V}(z_1) = z_1 [\dot{x}_{1d} - \dot{x}_{1d} - \alpha_1 z_1]$$

$$\dot{V}(z_1) = -\alpha_1 z_1^2$$

Con lo cual garantizamos que la derivada de la función de Lyapunov propuesta es semidefinida negativa. A través de las expresiones anteriores y sabiendo que: $z_1 = x_{1d} - x_1$, $\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$, y puesto que $\dot{x}_1 = x_2$, con $\dot{z}_1 = \dot{x}_{1d} - x_2$, es posible sustituir en la ecuación anterior el valor de $\dot{x}_{1d} = x_2 - \alpha_1 z_1$ obteniendo así:

$$\dot{z}_1 = x_2 - \alpha_1 z_1 - x_2$$

Una vez obtenida la expresión anterior y considerando el valor del control propuesto $x_2 = \dot{x}_{1d} + \alpha_1 z_1$ y sustituyéndolo en la ecuación anterior tenemos:

$$\dot{z}_1 = x_2 - \alpha_1 z_1 - \dot{x}_{1d} + \alpha_1 z_1$$

Tal como se puede apreciar en la ecuación anterior, la expresión del error inicial del sistema ha sido aumentada, por lo que, para estabilizar los nuevos términos sin modificar el sistema anterior, se realiza el siguiente cambio de variable:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1$$

Para anular este nuevo error se propone una función de Lyapunov aumentada respecto a la función propuesta para la expresión que describía el error inicial del sistema obteniendo así:

$$V(z_1, z_2) = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2$$

Al igual que ocurre con la función del subsistema anterior es importante determinar su derivada respecto al tiempo para corroborar que cumple con los criterios de Lyapunov, por tanto, se tiene:

$$\dot{V}(z_1, z_2) = z_1 \dot{z}_1 + z_2 \dot{z}_2$$

Para determinar la señal de control U_2 se partirá de las siguientes ecuaciones:

$$\begin{aligned} z_1 &= x_{1d} - x_1 & z_2 &= x_2 - \dot{x}_{1d} - \alpha_1 z_1 & x_2 &= \dot{x}_{1d} + \alpha_1 z_1 \\ \dot{z}_1 &= \dot{x}_{1d} - \dot{x}_1 & \dot{z}_2 &= \dot{x}_2 - \ddot{x}_{1d} - \alpha_1 \dot{z}_1 & \dot{x}_1 &= x_2 \end{aligned}$$

Sustituyendo el valor de \dot{z}_1 y \dot{z}_2 en la derivada de la función de Lyapunov se puede reescribir dicha expresión de la siguiente manera:

$$\dot{v}(z_1, z_2) = z_1(\dot{x}_{1d} - \dot{x}_1) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(\dot{x}_{1d} - \dot{x}_1))$$

Obtenida la expresión anterior y despejando el valor de $\dot{x}_{1d} = x_2 - \alpha_1 z_1 - z_2$ de la ecuación de z_2 obtenemos:

$$\dot{v}(z_1, z_2) = z_1(x_2 - \alpha_1 z_1 - z_2 - \dot{x}_1) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(x_2 - \alpha_1 z_1 - z_2 - \dot{x}_1))$$

Sabiendo que $\dot{x}_1 = x_2$ es posible reescribir la función anterior de tal forma que se tiene:

$$\dot{v}(z_1, z_2) = z_1(x_2 - \alpha_1 z_1 - z_2 - x_2) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(x_2 - \alpha_1 z_1 - z_2 - x_2))$$

Simplificando la expresión anterior y tomando en consideración las ecuaciones anteriores obtenemos finalmente:

$$\dot{v}(z_1, z_2) = z_1(-\alpha_1 z_1 - z_2) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(-\alpha_1 z_1 - z_2))$$

Tal como se explicó en el apartado del modelo dinámico del sistema, para determinar el comportamiento del sistema se parte de la formulación de Newton, por lo que considerando las ecuaciones del modelo obtenemos el valor de $\ddot{\phi}$:

$$\ddot{\phi} = \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} + \frac{l}{I_{xx}} U_2 + \frac{A_p}{I_{xx}}$$

Puesto que las operaciones restantes requieren el uso constante de la expresión anterior se considerarán las siguientes simplificaciones:

$$a_1 = \frac{(I_{yy} - I_{zz})}{I_{xx}} \quad b_1 = \frac{l}{I_{xx}}$$

Establecidas las expresiones anteriores y sabiendo que:

$$\dot{x}_2 = \ddot{x}_1 = \ddot{\phi} \quad x_4 = \dot{\theta} \quad x_6 = \dot{\psi}$$

Es posible reescribir el valor de $\ddot{\phi}$ de la siguiente manera:

$$\ddot{\phi} = a_1 x_4 x_6 + b_1 U_2$$

Por lo que reescribiendo la ecuación de la derivada de la función de Lyapunov se obtiene:

$$\dot{v}(z_1, z_2) = z_1(-\alpha_1 z_1 - z_2) + z_2(a_1 x_4 x_6 + b_1 U_2 - \dot{x}_{1d} - \alpha_1(-\alpha_1 z_1 - z_2))$$

Si se considera la referencia en aceleración como un valor nulo, pues ha de suponerse que el vehículo parte de un estado de reposo, se obtiene:

$$\dot{v}(z_1, z_2) = z_1(-\alpha_1 z_1 - z_2) + z_2(a_1 x_4 x_6 + b_1 U_2 - \alpha_1(-\alpha_1 z_1 - z_2))$$

De la ecuación anterior y para satisfacer que $\dot{v}(z_1, z_2) \leq 0$ se propone una señal de control que lo garantice, obteniendo de esta manera:

$$U_2 = \frac{1}{b_1} (z_1 - a_1 x_4 x_6 - \alpha_1(\alpha_1 z_1 + z_2) - \alpha_2 z_2) \quad (82)$$

Por lo que sustituyendo la señal de control y con el fin de corroborar que se cumplen con los criterios establecidos por la teoría de Lyapunov, se tiene:

$$\dot{v}(z_1, z_2) = -\alpha_1 z_1^2 - \alpha_2 z_2^2$$

Lo que garantiza el cumplimiento de los criterios de Lyapunov, hecho que establece como apropiada a la expresión obtenida para la señal de control U_2

Al igual que con la señal de control anterior, se realiza el mismo procedimiento para obtener las señales de control U_3 y U_4 , por tanto, las expresiones que definen dichas señales de control son:

$$U_3 = \frac{1}{b_2} (z_3 - a_3 x_2 x_6 - \alpha_3 (\alpha_3 z_3 + z_4) - \alpha_4 z_4) \quad (83)$$

$$U_4 = \frac{1}{b_3} (z_5 - a_5 x_2 x_4 - \alpha_5 (\alpha_5 z_5 + z_6) - \alpha_6 z_6) \quad (84)$$

Dónde:

$$z_3 = x_{3d} - x_3$$

$$z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3$$

$$z_5 = x_{5d} - x_5$$

$$z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5$$

Las expresiones anteriores se encargan del control de movimiento sobre los ejes x e y, mediante la variación de las velocidades de los motores, así como de los ángulos de ataque lo que genera las fuerzas de empuje necesarias para el desplazamiento del vehículo. Para completar el control de movimientos del UAV, es indispensable contar con una señal de control que regule el comportamiento de la altura; tal y como se puede apreciar en las expresiones del espacio de estados, la señal encargada de controlar la altura está representada por la variable U_1 , por lo que realizando el mismo procedimiento que en las señales de control anterior se obtiene:

$$U_1 = \frac{m}{\cos x_3 \cos x_1} (g + z_7 - \alpha_7 (\alpha_7 z_7 + z_8) - \alpha_7 z_7) \quad (85)$$

Dónde:

$$z_7 = x_{7d} - x_7$$

$$z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7$$

CONTROL DEL MOVIMIENTO SOBRE EL PLANO X-Y.

El modelo del sistema representado en la Figura 1 refleja que el movimiento a través de los ejes x e y depende de la entrada de control U_1 . Por lo que, si se considera U_x y U_y las responsables del movimiento a través de los ejes x e y, es indispensable obtener las ecuaciones de control para dichas variables, pues de este modo ha de controlarse el movimiento lineal sobre los ejes X e Y, cabe señalar que al igual que con las demás señales de control, las señales U_x y U_y , deben satisfacer la condición $V(z_9, z_{10}) < 0$ y $V(z_{11}, z_{12}) < 0$ respectivamente para garantizar un control fiable del vehículo. Dado lo anterior y siguiendo la misma metodología que se siguió para la obtención de la señal de control U_2 , se obtiene:

$$U_x = \frac{m}{U_1} (z_9 - \alpha_9 (\alpha_9 z_9 + z_{10}) - \alpha_{10} z_{10}) \quad (86)$$

$$U_y = \frac{m}{U_1} (z_{11} - \alpha_{11}(\alpha_{11}z_{11} + z_{12}) - \alpha_{12}z_{12}) \quad (87)$$

Dónde:

$$\begin{aligned} z_9 &= x_{9d} - x_9 & z_{10} &= x_{10} - \dot{x}_{9d} - \alpha_9 z_9 \\ z_{11} &= x_{11d} - x_{11} & z_{12} &= x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11} \end{aligned}$$

3.4.1.2 CONTROL DEL UAV CON PARES GIROSCOPICOS.

En este apartado se observará que, al igual que en la sección de modelado si se desea incluir el efecto de los pares giroscópicos, es necesario realizar algunos ajustes sobre las expresiones que determinan el comportamiento dinámico del sistema, específicamente en las ecuaciones que definen las aceleraciones angulares, por lo que en este apartado las ecuaciones que se tomaran como base para la realización de la estrategia de control son:

$$\begin{aligned} \ddot{x} &= (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \frac{U_1}{m} + \frac{A_x}{m} \\ \ddot{y} &= (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \frac{U_1}{m} + \frac{A_y}{m} \\ \ddot{z} &= -g + (\cos \theta \cos \phi) \frac{U_1}{m} + \frac{A_z}{m} \\ \ddot{\phi} &= \frac{A_p}{I_{xx}} + \frac{lU_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} - \frac{J_R}{I_{xx}} \Omega \dot{\theta} \\ \ddot{\theta} &= \frac{A_q}{I_{yy}} + \frac{l}{I_{yy}} U_3 + \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} + \frac{J_R}{I_{yy}} \Omega \dot{\phi} \\ \ddot{\psi} &= \frac{A_r}{I_{zz}} + \frac{1}{I_{zz}} U_4 + \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\theta} \dot{\phi} \end{aligned}$$

Tal como se puede observar en las expresiones anteriores, el efecto de los pares giroscópicos se ve reflejada sobre las aceleraciones de los componentes ϕ y θ por lo que las señales de control que se ven afectadas directamente son: U_2 y U_3 , es por esto por lo que en este apartado se realizaron únicamente los caculos correspondientes a estas entradas de control, pues las señales restantes permanecen iguales.

Control para U_2 : Una vez establecidos las expresiones que determinan el comportamiento del vehículo en presencia de pares giroscópicos, se procederá a aplicar la misma metodología implementada en el apartado anterior para la obtención de las señales de control. Según el teorema de Lyapunov, y considerando que la función candidata a Lyapunov $V(z)$ presentada a continuación es definida positiva y su derivada respecto al tiempo es semidefinida negativa, es posible decir:

$$V(z_1) = \frac{1}{2}z_1^2$$

Cuya derivada viene dada por: $\dot{V}(z_1) = z_1\dot{z}_1$, dónde: $z_1 = x_{1d} - x_1$, lo que implica que:

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$$

Partiendo de las expresiones anteriores y sabiendo que: $\dot{x}_1 = x_2$, se puede escribir la derivada de la función de Lyapunov tal como sigue:

$$\dot{V}(z_1) = z_1[\dot{x}_{1d} - x_2]$$

Por lo que la estabilización de Z_1 puede ser obtenida introduciendo la siguiente entrada de control virtual x_2 , es decir: $x_1 = x_2(x_1)$, la cual para satisfacer las condiciones de la teoría de Lyapunov debe cumplir: $x_2(0) = 0$. El nuevo sistema debe minimizar el error generado por el nuevo subsistema definido a través de la estrategia Backstepping, por lo que el nuevo sistema con el que se trabaja es:

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$$

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1$$

Con $\alpha_1 < 0$. Tomando el valor de x_2 de la ecuación anterior podemos reescribir la derivada de la función de Lyapunov.

$$\dot{V}(z_1) = z_1[\dot{x}_{1d} - \dot{x}_{1d} - \alpha_1 z_1]$$

$$\dot{V}(z_1) = -\alpha_1 z_1^2$$

Con lo cual garantizamos que la derivada de la función de Lyapunov propuesta es semidefinida negativa. Con lo anterior tenemos: $z_1 = x_{1d} - x_1$, $\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1$, obteniendo así:

$$\dot{z}_1 = \dot{x}_{1d} - x_2$$

Sustituyendo en la ecuación anterior el valor de $\dot{x}_{1d} = x_2 - \alpha_1 z_1$ obtenemos:

$$\dot{z}_1 = x_2 - \alpha_1 z_1 - x_2$$

Considerando el valor del control propuesto $x_2 = \dot{x}_{1d} + \alpha_1 z_1$ y sustituyéndolo en la ecuación anterior tenemos es posible reescribir tal expresión de la siguiente forma:

$$\dot{z}_1 = x_2 - \alpha_1 z_1 - \dot{x}_{1d} + \alpha_1 z_1$$

Se realiza el siguiente cambio de variable: $z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1$, para anular este nuevo error se propone una función de Lyapunov aumentada respecto a la función propuesta anteriormente:

$$V(z_1, z_2) = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2$$

Y su derivada respecto al tiempo es: $\dot{V}(z_1, z_2) = z_1 \dot{z}_1 + z_2 \dot{z}_2$, por lo que partiendo de la expresión anterior y dado las ecuaciones anteriores se sabe que:

$$\begin{aligned} z_1 &= x_{1d} - x_1 & z_2 &= x_2 - \dot{x}_{1d} - \alpha_1 z_1 & x_2 &= \dot{x}_{1d} + \alpha_1 z_1 \\ \dot{z}_1 &= \dot{x}_{1d} - \dot{x}_1 & \dot{z}_2 &= \dot{x}_2 - \ddot{x}_{1d} - \alpha_1 \dot{z}_1 & \dot{x}_1 &= x_2 \end{aligned}$$

Sustituyendo el valor de \dot{z}_1 y \dot{z}_2 en la derivada de la función de Lyapunov obtenemos:

$$\dot{v}(z_1, z_2) = z_1(\dot{x}_{1d} - \dot{x}_1) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(\dot{x}_{1d} - \dot{x}_1))$$

Despejando el valor de $\dot{x}_{1d} = x_2 - \alpha_1 z_1 - z_2$ de la ecuación de z_2 obtenemos y sabiendo que $\dot{x}_1 = x_2$, se puede reescribir la derivada de la función de Lyapunov de la siguiente forma:

$$\dot{v}(z_1, z_2) = z_1(x_2 - \alpha_1 z_1 - z_2 - x_2) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(x_2 - \alpha_1 z_1 - z_2 - x_2))$$

Simplificando la expresión anterior obtenemos finalmente:

$$\dot{v}(z_1, z_2) = z_1(-\alpha_1 z_1 - z_2) + z_2(\dot{x}_2 - \ddot{x}_{1d} - \alpha_1(-\alpha_1 z_1 - z_2))$$

Considerando las ecuaciones del modelo obtenemos el valor de $\ddot{\phi}$:

$$\ddot{\phi} = \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} - \frac{J_R}{I_{xx}} \Omega \dot{\theta} + \frac{l}{I_{xx}} U_2 + \frac{A_p}{I_{xx}}$$

Si consideramos las siguientes constantes:

$$a_1 = \frac{(I_{yy} - I_{zz})}{I_{xx}} \quad a_2 = -\frac{J_R}{I_{xx}} \quad b_1 = \frac{l}{I_{xx}}$$

Y sabiendo que $\dot{x}_2 = \dot{x}_1 = \ddot{\phi}$, $x_4 = \dot{\theta}$, $x_6 = \dot{\psi}$, es posible reescribir a $\ddot{\phi}$ como:

$$\ddot{\phi} = a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2$$

Sustituyendo la expresión anterior en la derivada de la función de Lyapunov se puede obtener:

$$\dot{v}(z_1, z_2) = z_1(-\alpha_1 z_1 - z_2) + z_2(a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2 - \ddot{x}_{1d} - \alpha_1(-\alpha_1 z_1 - z_2))$$

Si de la expresión anterior se considera que la aceleración de referencia del sistema es igual a cero, se puede reescribir la expresión anterior como:

$$\dot{v}(z_1, z_2) = z_1(-\alpha_1 z_1 - z_2) + z_2(a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2 - \alpha_1(-\alpha_1 z_1 - z_2))$$

Para satisfacer que $\dot{v}(z_1, z_2) \leq 0$ se propone una señal de control que lo garantice, es decir:

$$U_2 = \frac{1}{b_1} (z_1 - a_1 x_4 x_6 - a_2 x_4 \Omega - \alpha_1(\alpha_1 z_1 + z_2) - \alpha_2 z_2) \quad (88)$$

Por lo que sustituyendo la señal de control en la función que describe la derivada de la función de Lyapunov se obtiene:

$$\dot{v}(z_1, z_2) = -\alpha_1 z_1^2 - \alpha_2 z_2^2 \quad (89)$$

A través de la expresión anterior se puede observar que se cumplen los criterios establecidos por la teoría de Lyapunov, lo cual garantiza que la señal de control obtenida es apropiada para su aplicación sobre el sistema analizado.

Dado lo anterior es posible determinar las señales de control para el sistema general, tomando en cuenta que tanto el subsistema de traslación, así como la señal U_3 y U_4 de rotación se calculan de manera similar (el desarrollo completo se encuentra en los anexos), obteniendo finalmente:

$$U_1 = \frac{m}{\cos x_3 \cos x_1} (g + z_7 - \alpha_7(\alpha_7 z_7 + z_8) - \alpha_7 z_7) \quad (90)$$

$$U_2 = \frac{1}{b_1} (z_1 - a_1 x_4 x_6 - a_2 x_4 \Omega - \alpha_1(\alpha_1 z_1 + z_2) - \alpha_2 z_2) \quad (91)$$

$$U_3 = \frac{1}{b_2} (z_3 - a_3 x_2 x_6 - a_4 x_2 \Omega - \alpha_3(\alpha_3 z_3 + z_4) - \alpha_4 z_4) \quad (92)$$

$$U_4 = \frac{1}{b_3} (z_5 - a_5 x_2 x_4 - \alpha_5(\alpha_5 z_5 + z_6) - \alpha_6 z_6) \quad (93)$$

Dónde:

$$z_1 = x_{1d} - x_1 \quad z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1$$

$$z_3 = x_{3d} - x_3 \quad z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3$$

$$z_5 = x_{5d} - x_5 \quad z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5$$

$$z_7 = x_{7d} - x_7 \quad z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7$$

De igual forma para el sistema de traslación sobre el plano X, Y se tiene:

$$U_x = \frac{m}{U_1} (z_9 - \alpha_9(\alpha_9 z_9 + z_{10}) - \alpha_{10} z_{10}) \quad (94)$$

$$U_y = \frac{m}{U_1} (z_{11} - \alpha_{11}(\alpha_{11} z_{11} + z_{12}) - \alpha_{12} z_{12}) \quad (95)$$

Dónde:

$$z_9 = x_{9d} - x_9 \quad z_{10} = x_{10} - \dot{x}_{9d} - \alpha_9 z_9$$

$$z_{11} = x_{11d} - x_{11} \quad z_{12} = x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11}$$

Las expresiones anteriores representan el sistema de control global del UAV, generado a través de la técnica de control Backstepping y que además consideran los efectos giroscópicos que se presentan sobre el vehículo, generando así un control más preciso y que ha de garantizar un buen desempeño durante su implementación.

3.4.2 CONTROL PID

El Controlador Proporcional, Integral y Derivativo comúnmente denominado PID, a través del cual se obtendrán las señales de control necesarias para que el UAV, se capaz de seguir las trayectorias de referencia y de esta manera obtener un referente que permita demostrar la eficiencia del control Backstepping, propuesto en la sección anterior. La estructura básica de un controlador clásico PID puede observarse en la Figura 8:

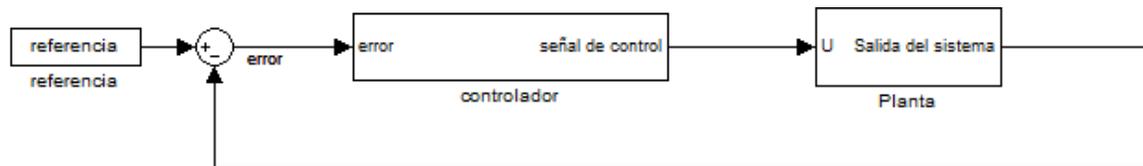


Figura 8. Sistema clásico de control PID.

Como se aprecia en la imagen anterior este tipo de controlador trabaja de manera directa con el error $e(t)$ producido entre la variable de salida del sistema y el valor de referencia, es decir:

$$e(t) = z_r - z \quad (96)$$

El objetivo del PID al igual que el de otros controladores es eliminar o reducir el error, para ello se obtendrá la señal de control a partir de la suma de sus tres componentes fundamentales: la parte proporcional, la integral, y la derivativa. El valor Proporcional (P) del PID está formado por el producto del error con una constante de proporcionalidad generalmente llamada k_p , es decir:

$$P = k_p e(t) \quad (97)$$

Por lo que la parte proporcional del sistema responderá de manera directa en función del error, es decir, proporcionará una salida mayor cuando el error aumente y disminuirá cuando el error sea pequeño. La parte proporcional por sí sola no considera las variaciones del error respecto al tiempo, por lo que para un mayor control en el sistema se suele utilizar las otras componentes del PID.

Otra componente del controlador PID es la parte Integral, la cual otorgará una respuesta proporcional a la integral del error, con lo que se eliminara o minimizara el error en estado estacionario, el cual estaría presente si solo se usase la parte proporcional del controlador. El valor integral (I) del PID estará determinado por la integral del error multiplicado por una constante de proporcionalidad, generalmente llamada k_i , es decir:

$$I = k_i \int e(t) dt \quad (98)$$

Por otra parte, la parte derivativa actúa de manera proporcional a la variación del error respecto al tiempo, por lo que si el error es constante esta parte no afectara al valor de salida del PID. La acción derivativa minimiza el error ya que lo corrige de manera proporcional a la velocidad con que se produce por lo que es fundamental sintonizar de manera adecuada esta componente ya una mayor acción derivativa permitirá al controlador responder de manera adecuada ante cambios rápidos del sistema. El valor de la parte derivativa (D) del controlador PID está determinado por la derivada del error respecto al tiempo, multiplicado por una ganancia de proporcionalidad generalmente llamada k_d , tal y como lo describe la siguiente expresión:

$$D = k_d \frac{de(t)}{dt} \quad (99)$$

La señal de control $U(t)$ representa la salida final del controlador PID, la cual está determinada por la suma del valor de los componentes proporcional, integral, y el derivativo, es decir:

$$U(t) = P + I + D = e(t)k_p + k_i \int e(t)dt + k_d \frac{de(t)}{dt} \quad (100)$$

La implementación del PID suele ser sencilla y es común en muchos sistemas simples a pesar de que esta señal no siempre garantiza un control óptimo del sistema. El valor de las constantes k_p , k_i y k_d determina la eficiencia del PID y una mala selección de estas, pueden volver inestable a un sistema, por lo que se requiere de un análisis de la respuesta que ofrece el controlador al variar dichas constantes, seleccionado posteriormente la combinación que otorgue el mejor resultado sin comprometer la estabilizada del sistema.

3.4.2.1 CONTROL PID DEL UAV

Para el control del UAV se han implementado seis controladores del tipo PID, tres de los cuales permitirán el control de la traslación del vehículo mientras que los tres restantes serán los encargados del control de la orientación del vehículo. Estos controles actuaran sobre el siguiente conjunto de ecuaciones, obtenidas durante el modelado dinámico del UAV.

3.4.2.2 CONTROL PID DE TRASLACIÓN

EL control de la traslación del UAV requiere de 4 controladores PID, siendo el primero de ellos el controlador de altura, el cual es el encargado de regular la posición del vehículo en el eje Z, para ello determinara el valor de la señal de control U_1 que representa la potencia que se debe aplicar a los cuatro rotores para corregir la diferencia entre la altura actual del sistema y la altura deseada.

Por otra parte, el control en los ejes X e Y permite corregir la posición en dichos ejes, ya que por medio de estos controles virtuales se podrán obtener los ángulos de alabeo y cabeceo necesarios para que el vehículo alcance la posición deseada. Los valores de las constantes de control utilizados para los tres controladores PID de traslación pueden observarse en la Tabla 1. Con el uso de estas constantes se observó un desempeño aceptable y no se compromete la estabilidad del sistema.

| | Controlador z | Controlador x | Controlador Y |
|----|---------------|---------------|---------------|
| Kp | 25 | 22 | 14 |
| Ki | 1 | 3 | 0.9 |
| Kd | 9 | 14 | 1.1 |

Tabla 1. Ganancias de control para el subsistema de traslación.

Los tres controladores descritos corresponden al control de la traslación del UAV, los cuales han sido implementados con las ganancias mostradas en la tabla anterior, cabe destacar que las señales de control obtenidas por el controlador X y por el controlador de Y son señales de control virtuales, debido a que no se usaran directamente en el sistema si no que se utilizaran para el cálculo de los ángulos necesarios para mantener el control en el plano XY.

3.4.2.3 CONTROL PID DE ORIENTACIÓN

Para el subsistema de rotación se implementaron tres controladores PID, los cuales determinaran las señales de control U_2 , U_3 y U_4 que actuaran sobre los ángulos de alabeo, cabeceo y guiñada respectivamente. El primer controlador diseñado es el concerniente al ángulo de cabeceo ϕ , dicho controlador permitirá eliminar o minimizar el error entre el ángulo ϕ del sistema y el ángulo ϕ_r , determinando el giro apropiado en el eje X para que el vehículo cumpla con la trayectoria deseada.

El segundo controlador del subsistema de rotación es el referente al ángulo de alabeo del vehículo. Este controlador actúa de la misma manera que el controlador anterior, pero proporciona el valor de la señal de control U_3 , la cual determinara el giro que el vehículo debe efectuar en el eje Y. de igual manera que los otros controladores el objetivo de este PID es reducir el error entre el ángulo del sistema θ y el ángulo de referencia θ_r (previamente calculado), todo ello sin comprometer la estabilidad del sistema.

Por último, se requiere de un tercer controlador PID para completar el sistema de control de rotación del UAV. Este controlador permitirá obtener la señal de control U_4 . La cual es responsable de controlar el giro del vehículo alrededor del eje Z, lo que indica que este controlador trabajara directamente con el error producido entre el ángulo de referencia ψ_r y el ángulo real del sistema ψ . Los valores utilizados para las constantes k_p , k_i y k_d de los controladores del subsistema de rotación pueden observarse en la Tabla 2.

| | Controlador ϕ | Controlador θ | Controlador ψ |
|----|--------------------|----------------------|--------------------|
| Kp | 15 | 15 | 3 |
| Ki | 0.01 | 0.01 | 0.01 |

| | | | |
|----|-----|-----|---|
| Kd | 0.9 | 0.9 | 3 |
|----|-----|-----|---|

Tabla 2. Ganancias de control para el subsistema de rotación.

Las ganancias mostradas para los controles de traslación, así como para el sistema de rotación han mostrado en simulaciones un buen desempeño en conjunto, además de no comprometer la estabilidad del sistema, el cual fue probado ante distintas trayectorias de referencia. Tanto el control del sistema de traslación como el de orientación representan el control PID global del UAV, mediante el cual el sistema ha sido capaz de resolver de manera satisfactoria el problema de seguimiento de trayectorias.

3.5 MODELADO DE MAIPULADORES

Para poder realizar un control eficiente es necesario en primera instancia desarrollar los modelos matemáticos correspondientes, esto permite diseñar un controlador que permita el movimiento preciso del manipulador. Los modelos cinemático y dinámico del manipulador son necesarios para cumplir con los objetivos planteados. El robot manipulador que se estudia es el robot planar de dos grados de libertad como el mostrado en la Figura 9.

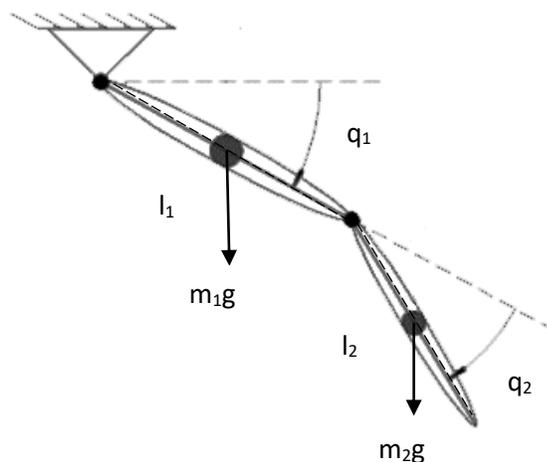


Figura 9. Manipulador planar de 2 grados de libertad.

Este tipo de robot en una base fija nos permitirá realizar movimientos en solo un plano, sin embargo, al estar acoplado a un UAV se superará esta limitación, por otra parte, más eslabones indican un mayor peso por lo que un brazo con componentes mínimos permitirá a su vez acoplar un brazo ligero.

3.5.1 MODELO CINEMÁTICO DEL MANIPULADOR

El modelo cinemático del robot manipulador nos permite conocer la relación existente del movimiento cartesiano con el movimiento articular del propio robot. Este modelo no se consideran las fuerzas que intervienen en el movimiento, de manera general el modelo cinemático de los manipuladores se expresa como:

$$X = f(q) \quad (101)$$

Donde X es la posición y orientación cartesiana del efector final y q representa los valores angulares de las coordenadas articulares. Existen diversos métodos para el cálculo del modelo cinemático no obstante dado el manipulador elegido se utiliza el método geométrico. Por lo cual el modelo cinemático del robot estará determinado por:

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) \end{bmatrix} \quad (102)$$

Donde y, z son las coordenadas cartesianas del efector final, q1 y q2 representan el valor de la primera y segunda articulación respectivamente y l1 y l2 representan las longitudes del primer y segundo eslabón respectivamente.

Por otra parte, para obtener la relación que nos permita saber qué valor debe tomar cada articulación para que el efector final llegue a un punto cartesiano determinado se debe calcular el modelo cinemático inverso el cual es determinado por el método algebraico y está determinado por:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{y}{-z}\right) + \tan^{-1}\left(\frac{l_2 \sin(q_2)}{l_1 + l_2 \cos(q_2)}\right) \\ \tan^{-1}\left(\frac{y^2 + z^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \end{bmatrix} \quad (103)$$

Con lo cual se da por concluido el análisis cinemático del robot.

3.5.2 MODELO DINÁMICO DEL MANIPULADOR

El modelo dinámico del manipulador nos permitirá conocer con mayor detalle el comportamiento de ese robot ante distintas fuerzas de entrada. La dinámica de un manipulador se puede obtener mediante la formulación de LaGrange y de manera general puede escribirse de la siguiente forma:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right] - \frac{\partial L(q, \dot{q})}{\partial q_i} = \tau_i \quad i = 1, \dots, n \quad (104)$$

Donde τ_i son las fuerzas o pares aplicados por los motores a cada articulación y L es el Lagrangiano del robot el cual es definido como la diferencia entre la energía cinética del robot K y la energía potencial U, es decir:

$$L(q(t), \dot{q}(t)) = K(q(t), \dot{q}(t)) - U(q(t))$$

Donde la energía potencial depende de la fuerza de la gravedad. Obtenidas cada ecuación de movimiento de Lagrange para cada articulación, se puede ordenar de manera matricial:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) = \tau \quad (105)$$

Donde:

q: Coordenadas Generalizadas.

$M(q)$: es la matriz de inercia.

$C(q, \dot{q})$: es la matriz de Coriolis y fuerzas centrífugas.

$G(q)$: es el vector de términos gravitatorios.

$F(\dot{q})$: es el vector de fuerzas de rozamiento.

τ : es el vector con los torques.

Si suponemos una fricción despreciable, masas invariables (m_1, m_2) así como eslabones indeformables de longitudes (l_1 y l_2), y con el centro de masa en el centro geométrico, podemos simplificar la ecuación anterior obteniendo:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (106)$$

Donde

$$M_{11} = m_1 \left(\frac{l_1}{2}\right)^2 + m_2 \left(l_1^2 + \left(\frac{l_2}{2}\right)^2 + l_1 l_2 (1 + \cos q_2)\right) + I_1 + I_2, M_{12} = M_{21} = m_2 \left(\frac{l_2}{2}\right)^2 + \frac{l_1 l_2}{2} \cos q_2 + I_2$$

$$M_{22} = m_2 \left(\frac{l_2}{2}\right)^2 + I_2, C_{11} = -m_2 l_1 l_2 \sin q_2 \dot{q}_2, C_{12} = -m_2 l_1 l_2 \sin q_2 (\dot{q}_1 + \dot{q}_2), C_{21} = m_2 l_1 l_2 \sin q_2 \dot{q}_1$$

$$C_{22} = 0, g_1 = \left(m_1 \frac{l_1}{2} + m_2 l_1\right) g \cos q_1 + m_2 \frac{l_2}{2} \cos(q_1 + q_2), g_2 = m_2 \frac{l_2}{2} \cos(q_1 + q_2)$$

3.6 CONTROL DEL MANIPULADOR

El diseño del controlador para el manipulador tomará como base el modelo anteriormente desarrollado, en primera instancia se desarrollará el control mediante Backstepping y posteriormente se desarrollará un controlador PID para fines de comparación. Los controladores diseñados permitirán al manipulador llegar a una posición deseada o realizar el seguimiento de una trayectoria.

3.6.1 CONTROLADOR BACSTEPPING PARA MANIPULADOR

Para el control del manipulador partimos de la ecuación obtenida en el modelado:

$$\dot{x}_{14} = M(q) (\tau - C\dot{x}_{13} + G) \quad (107)$$

Se definen los errores como:

$$z_{13} = [q - q_d] = [x_{13} - x_{13d}] \quad (108)$$

Siendo sus derivadas:

$$\dot{z}_{13} = \dot{z}_{14} = [\dot{q} - \dot{q}_d] = [\dot{x}_{13} - \dot{x}_{13d}] \quad (109)$$

$$\ddot{z}_{13} = \ddot{z}_{14} = [\ddot{q} - \ddot{q}_d] = [\ddot{x}_{13} - \ddot{x}_{13d}] \quad (110)$$

La entrada de control virtual propuesta y su correspondiente derivada es:

$$z_{14} = z_{13} + kz_{13} \quad (111)$$

$$\dot{z}_{14} = \dot{z}_{13} + k\dot{z}_{13} \quad (112)$$

Donde k es una matriz diagonal definida positiva, despejando de la ecuación (111):

$$z_{13} = z_{14} - kz_{13} \quad (113)$$

Por otra parte, usando la ecuación (112) en la ecuación (112) se obtiene:

$$\dot{z}_{14} = [\ddot{x}_{13d} - \ddot{x}_{13}] + k\dot{z}_{13} \quad (114)$$

Sabiendo el valor de \ddot{x}_{13} proveniente del modelo dinámico y usando la ecuación 90 tenemos:

$$\dot{z}_{14} = [(M^{-1}(\tau - C\dot{x}_{13} - G)) - \ddot{x}_{13d}] + k[z_{14} - kz_{13}] \quad (115)$$

Se propone una función de Lyapunov que sea capaz de controlar ambos errores (z_{13} y z_{14}) con la cual también se debe garantizar la estabilidad del sistema:

$$V(z_{13}, z_{14}) = \frac{1}{2}z_{13}^T k z_{13} + \frac{1}{2}z_{14}^T M z_{14} \quad (116)$$

Derivando la función de Lyapunov tenemos:

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T k \dot{z}_{13} + \frac{1}{2}z_{14}^T \dot{M} z_{14} + z_{14}^T M \dot{z}_{14} \quad (117)$$

Fusionando la ecuación 83, 86 y la ecuación 89 con la ecuación (117) logramos:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] + \frac{1}{2}z_{14}^T \dot{M} z_{14} \\ &+ z_{14}^T ([\tau - C\dot{x}_{13} - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \end{aligned} \quad (118)$$

Despejando de 85 el valor de \dot{x}_{13} y de 89 el valor de \dot{z}_{13} y sustituyéndolos en la ecuación anterior obtenemos:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] + \frac{1}{2}z_{14}^T \dot{M} z_{14} \\ &+ z_{14}^T ([\tau - C[x_{13d} + (z_{14} - kz_{13})] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \end{aligned} \quad (119)$$

Según el trabajo expuesto en [104] sabemos que:

$$z_{14}^T (\dot{M} - 2C) z_{14} = 0 \quad (120)$$

Aplicando esta propiedad simplificamos a:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] \\ &+ z_{14}^T \left([\tau - C[x_{13d} + (z_{14} - kz_{13})] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}] + \frac{1}{2}\dot{M} z_{14} \right) \end{aligned} \quad (121)$$

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] \\ &+ z_{14}^T ([\tau - C[x_{13d} - kz_{13}] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \end{aligned} \quad (122)$$

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) = & z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} \\ & + z_{14}^T (\tau - C[x_{13d} - k z_{13}] - G - M\ddot{x}_{13d} + M k z_{14} - M k^2 z_{13}) \end{aligned} \quad (123)$$

Finalmente se propone una señal de control que satisfaga las condiciones de estabilidad del sistema, en este caso se propone:

$$\tau = M\ddot{x}_{13d} + C x_{13d} - M k z_{14} + G - k z_{13} \quad (124)$$

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} + z_{14}^T (-C k z_{13} - M k^2 z_{13} - k z_{13}) \quad (125)$$

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} - z_{14}^T C k z_{13} - z_{14}^T M k^2 z_{13} - z_{14}^T k z_{13} \quad (126)$$

Para comprobar que la derivada sea definida negativa se utiliza las propiedades de la norma matricial:

$$\dot{V}(z_{13}, z_{14}) = \|z_{13}^T k z_{14}\| - z_{13}^T k^2 z_{13} - z_{14}^T C k z_{13} - z_{14}^T M k^2 z_{13} \quad \| -z_{14}^T k z_{13} \| \leq 0 \quad (127)$$

$$-z_{13}^T k^2 z_{13} - z_{14}^T C k z_{13} - z_{14}^T M k^2 z_{13} \leq 0 \quad (128)$$

$$-(Ck + Mk^2) \|z_{14}^T\| \|z_{13}^T\| - k^2 \|z_{13}\|^2 < 0 \quad (129)$$

Por lo tanto, la derivada de la función de Lyapunov es definida negativa lo cual garantiza la estabilidad y da por válida la ley de control propuesta.

3.6.2 CONTROL PD+G PARA MANIPULADOR

La ley de control de un PID simple se encuentra definida por la misma ecuación sin importar en qué tipo de estructura sea aplicado:

$$U(t) = P + I + D = e(t)k_p + k_i \int e(t)dt + kd \frac{de(t)}{dt} \quad (130)$$

Donde Kd, Kp, Ki son las ganancias proporcionales, derivativas e integrales respectivamente, sin embargo, se ha decidido emplear una ligera variación en la que se utiliza un PD más un término que compense los efectos de los pares gravitatorios en el manipulador, es decir:

$$U(t) = P + D + G = e(t)k_p + kd \frac{de(t)}{dt} + g(q_1, q_2) \quad (131)$$

Por lo que una vez obtenidos los modelos del manipulador podemos obtener los siguientes términos:

$$\begin{aligned} U &= \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ e &= \begin{bmatrix} q_{d1} - q_1 \\ q_{d2} - q_2 \end{bmatrix} \\ K_p &= \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix} \end{aligned}$$

$$K_d = \begin{bmatrix} k_{d1} & 0 \\ 0 & k_{d2} \end{bmatrix}$$

$$G = \begin{bmatrix} \left(m_1 \frac{l_1}{2} + m_2 l_1 \right) g \cos q_1 + m_2 \frac{l_2}{2} \cos(q_1 + q_2) \\ m_2 \frac{l_2}{2} \cos(q_1 + q_2) \end{bmatrix}$$

Donde e representa el error de seguimiento del manipulador.

3.7 MODELADO SISTEMA UAV MANIPULADOR

En este apartado se desarrolla el modelo del conjunto UAV-Manipulador para analizar el movimiento de manera más detallada y poder desarrollar una ley de control adecuada para el conjunto. Para la descripción global de los movimientos del sistema UAV-manipulador, así como para permitir el análisis de las fuerzas que actúan sobre el vehículo, en la Figura 10 se presenta un diagrama del sistema con los ejes de referencia de cada subsistema:

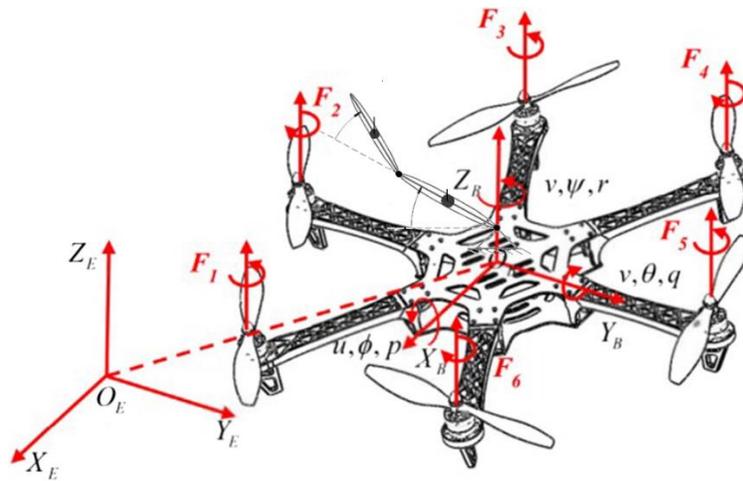


Figura 10. Sistema UAV-Manipulador.

Por otra parte dada la complejidad del modelo dinámico del multirroto, y como se mencionó anteriormente, se han considerado algunas simplificaciones para desarrollar el modelado, si aunado a ellas se desprecian el efecto suelo y los efectos de los momentos ocasionados por el cuerpo rígido sobre las dinámicas trasnacionales, se puede aplicar una matriz diagonal de inercias del modo:

$$I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \quad (132)$$

Sin embargo, la inercia del sistema completo UAV-manipulador es diferente puesto que la inercia del manipulador afecta directamente al multirroto por lo tanto es necesario calcular el efecto de

la inercia del manipulador en el sistema del hexarrotor, para lo cual se utiliza el teorema de los ejes paralelos, obteniendo así la matriz de inercia del sistema UAV-Manipulador:

$$I = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (133)$$

Donde:

$$I_{xx} = I_x + \frac{m_1 l_1^2 \sin^3 q_1}{3} + \frac{m_2 l_2 \sin^3 q_2}{3} + m_2 l_1^2 \sin^2 q_1 \quad (134)$$

$$I_{yy} = I_y + \frac{m_1 l_1^2}{3} + \frac{m_2 l_2^2}{3} + m_2 l_1^2 \quad (135)$$

$$I_{zz} = I_z + \frac{m_1 l_1^2 \cos^3 q_1}{3} + \frac{m_2 l_2 \cos^3 q_2}{3} + m_2 l_1^2 \cos^2 q_1 \quad (136)$$

Del mismo modo se calcula el nuevo centro de masa para el sistema completo tomando como referencia el sistema empotrado al multirrotor, es decir el centro del sistema de coordenadas F_B .

$$c_x = \frac{\left(l_1 \cos q_1 \left(\frac{m_1}{2} + m_2 \right) + m_2 \frac{l_2}{2} \cos(q_1 - q_2) \right)}{m + m_1 + m_2} \quad (137)$$

$$c_y = 0 \quad (138)$$

$$c_z = \frac{\left(l_1 \sin q_1 \left(\frac{m_1}{2} + m_2 \right) + m_2 \frac{l_2}{2} \sin(q_1 - q_2) \right)}{m + m_1 + m_2} \quad (139)$$

Finalmente se calcula la acción que tiene los torques del manipulador sobre el sistema completo siguiendo la metodología presentada en [91]tenemos que:

$$\tau_r = \tau_1 + m_2 g l_{1c} \cos q_1 + m_2 g l_{2c} \cos(q_1 - q_2) + \tau_2 \quad (140)$$

Donde τ_r es el torque total que afecta al multirrotor por efecto de los torques del manipulador, y l_{c1} y l_{c2} son las distancias al centro de masas de cada eslabón. Por lo tanto, el sistema que define al modelo dinámico de un multirrotor con un brazo acoplado es:

$$\left\{ \begin{array}{l} \ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}} (\dot{\theta}\dot{\psi}) - \frac{U_2 - \tau_r}{I_{xx}} \\ \ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{xx}} (\dot{\phi}\dot{\psi}) + \frac{g(m_1 + m_2)\Delta_y \sin\theta}{I_{yy}} - \frac{U_3 - \tau_r}{I_{yy}} \\ \ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}} (\dot{\phi}\dot{\psi}) - \frac{U_4}{I_{zz}} \\ \ddot{z} = (-g + (\cos\theta \cos\phi)) \frac{U_1}{m_T} \\ \ddot{y} = (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi) \frac{U_1}{m_T} \\ \ddot{x} = (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi) \frac{U_1}{m_T} \\ \tau = M(q) \ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \end{array} \right. \quad (141)$$

Donde m_T es la masa total del sistema y está determinada por $m + m_1 + m_2$

3.8 CONTROL SISTEMA UAV MANIPULADOR

En esta sección se desarrollarán los controladores para el seguimiento de trayectorias del sistema UAV-manipulador en primera instancia se diseñará el controlador basado en la técnica Backstepping y posteriormente y para fines comparativos se desarrollará un controlador PID para el sistema en conjunto.

3.8.1 CONTROL BACKSTEPPING SISTEMA UAV MANIPULADOR

Para el desarrollo del control del sistema se parte de las ecuaciones obtenidas en la sección de modelado y que describen la dinámica del vehículo-brazo. Para diseñar el controlador se reescribe dicho modelo considerando el siguiente vector:

$$X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16})^T$$

Donde:

$$\begin{aligned} x_1 &= \phi, & x_2 &= \dot{x}_1 = \dot{\phi}, & x_3 &= \theta, & x_4 &= \dot{x}_3 = \dot{\theta}, & x_5 &= \psi, & x_6 &= \dot{x}_5 = \dot{\psi} \\ x_7 &= z, & x_8 &= \dot{x}_7 = \dot{z}, & x_9 &= x, & x_{10} &= \dot{x}_9 = \dot{x}, & x_{11} &= y, & x_{12} &= \dot{x}_{11} = \dot{y} \\ x_{13} &= q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, & x_{14} &= \dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \end{aligned}$$

Considerando lo anterior podemos reescribir el modelo como:

$$\left\{ \begin{array}{l} \dot{x}_2 = \frac{I_{yy} - I_{zz}}{I_{xx}} (x_4 x_6) - \frac{U_2 - \tau_r}{I_{xx}} \\ \dot{x}_4 = \frac{I_{zz} - I_{xx}}{I_{xx}} (x_2 x_6) + \frac{g(m_1 + m_2) \Delta_y \sin x_3}{I_{yy}} - \frac{U_3 - \tau_r}{I_{yy}} \\ \dot{x}_6 = \frac{I_{xx} - I_{yy}}{I_{zz}} (x_1 x_4) - \frac{U_4}{I_{zz}} \\ \dot{x}_8 = (-g + (\cos x_3 \cos x_1)) \frac{U_1}{m_T} \\ \dot{x}_{10} = (\sin x_5 \sin x_1 + \cos x_5 \sin x_3 \cos x_1) \frac{U_1}{m_T} \\ \dot{x}_{12} = (-\cos x_5 \sin x_1 + \sin x_5 \sin x_3 \cos x_1) \frac{U_1}{m_T} \\ \dot{x}_{14} = M^{-1}(q) (\tau - C(q, \dot{q})x_{14} - G(q)) \end{array} \right. \quad (142)$$

Señal de control U_2 : El objetivo de la ley de control que se diseña es reducir al máximo el error en el sistema por lo cual el primer paso es definir el error z y su derivada:

$$z_1 = x_{1d} - x_1 \quad (143)$$

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1 \quad (144)$$

Para garantizar la estabilidad se debe proponer una función candidata a Lyapunov que sea definida positiva y que su derivada sea semidefinida negativa [92], por lo tanto, se propone la siguiente función:

$$V(z_1) = \frac{1}{2} z_1^2 \quad (145)$$

La cual es definida positiva, siendo su derivada la siguiente expresión:

$$\dot{V}(z_1) = z_1 \dot{z}_1 \quad (146)$$

Sustituimos la ecuación (144) en la ecuación (146):

$$\dot{V}(z_1) = z_1 [\dot{x}_{1d} - \dot{x}_1] \quad (147)$$

Para garantizar la estabilidad se debe garantizar que la derivada de la función de Lyapunov sea semidefinida negativa, para ello proponemos una entrada virtual de control x_2 que lo garantice:

$$x_2 = \dot{x}_{1d} + \alpha z_1 \quad (148)$$

Donde α_1 es una ganancia de control >0 , por lo tanto:

$$\begin{aligned} \dot{V}(z_1) &= z_1 [\dot{x}_{1d} - \dot{x}_{1d} - \alpha z_1] \\ \dot{V}(z_1) &= -\alpha_1 z_1^2 \end{aligned} \quad (149)$$

En la ecuación (149) se puede observar que la derivada de la función candidata a Lyapunov es semidefinida negativa. Utilizando el valor de \dot{x}_{1d} de la ecuación (148) y sustituyéndolo en la ecuación (144) tenemos:

$$\dot{z}_1 = x_2 - \alpha_1 z_1 - \dot{x}_{1d}$$

Utilizando el valor del control virtual x_2 y sustituyéndolo en la ecuación anterior se tiene que:

$$\dot{z}_1 = x_2 - \alpha_1 z_1 - \dot{x}_{1d} - \alpha_1 z_1$$

Posteriormente se realiza un cambio de variable:

$$z_2 = x_2 - \alpha_1 z_1 - \dot{x}_{1d}$$

Para anular este nuevo error se propone una función de Lyapunov aumentada:

$$V(z_1, z_2) = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2$$

Y cuya derivada es

$$\dot{V}(z_1, z_2) = z_1 \dot{z}_1 + z_2 \dot{z}_2$$

Sabiendo que:

$$\begin{aligned} z_1 &= x_{1d} - x_1, & \dot{z}_1 &= \dot{x}_{1d} - \dot{x}_1, & \dot{x}_1 &= x_2, & z_2 &= x_2 - \alpha_1 z_1 - \dot{x}_{1d}, \\ z_2 &= \dot{x}_2 - \alpha_1 \dot{z}_1 - \ddot{x}_{1d}, & x_2 &= \dot{x}_{1d} + \alpha_1 z_1 \end{aligned}$$

Obtenemos:

$$\begin{aligned} \dot{V}(z_1, z_2) &= z_1[\dot{x}_{1d} - \dot{x}_1] + z_2[\dot{x}_2 - \ddot{x}_{1d} - \alpha_1[\dot{x}_{1d} - \dot{x}_1]] \\ \dot{V}(z_1, z_2) &= z_1[x_2 - \alpha_1 z_1 - z_2 - \dot{x}_1] + z_2[\dot{x}_2 - \ddot{x}_{1d} - \alpha_1[x_2 - \alpha_1 z_1 - z_2 - \dot{x}_1]] \\ \dot{V}(z_1, z_2) &= z_1[x_2 - \alpha_1 z_1 - z_2 - x_2] + z_2[\dot{x}_2 - \ddot{x}_{1d} - \alpha_1[x_2 - \alpha_1 z_1 - z_2 - x_2]] \\ \dot{V}(z_1, z_2) &= z_1[-\alpha_1 z_1 - z_2] + z_2[\dot{x}_2 - \ddot{x}_{1d} - \alpha_1[-\alpha_1 z_1 - z_2]] \end{aligned}$$

Conocemos el valor de \dot{x}_2 de las ecuaciones del modelo, por lo que sustituyendo obtenemos:

$$\dot{V}(z_1, z_2) = z_1[-\alpha_1 z_1 - z_2] + z_2[a_1 x_4 x_6 - b_1 - b_1 \tau_r - \ddot{x}_{1d} - \alpha_1[-\alpha_1 z_1 - z_2]]$$

Donde:

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}, b_1 = \frac{1}{I_{xx}}$$

Simplificando la expresión anterior:

$$\dot{V}(z_1, z_2) = -\alpha_1 z_1^2 - z_1 z_2 + z_2[a_1 x_4 x_6 - b_1 - b_1 \tau_r - \ddot{x}_{1d} - \alpha_1[-\alpha_1 z_1 - z_2]]$$

Considerando una aceleración de cero en la referencia $\ddot{x}_{1d} = 0$ obtenemos:

$$\dot{V}(z_1, z_2) = -\alpha_1 z_1^2 - z_2[z_1 - a_1 x_4 x_6 + b_1 U_2 + b_1 \tau_r - \alpha_1[\alpha_1 z_1 + z_2]] \quad (150)$$

Por lo que se propone una señal de control que garantice la estabilidad del sistema:

$$U_2 = \frac{1}{b_1}(-z_1 + a_1 x_4 x_6 - b_1 \tau_r + \alpha_1[\alpha_1 z_1 + z_2] + \alpha_2 z_2) \quad (151)$$

Donde $0 < \alpha_2$ Sustituyéndolo en la ecuación 103 se observa que la derivada de la función de Lyapunov con esta señal de control es semidefinida negativa:

$$\dot{V}(z_1, z_2) = -\alpha_1 z_1^2 - \alpha_2 z_2^2$$

Los procesos para obtener U_3 y U_4 son similares a los realizados en el desarrollo de U_2

$$U_3 = \frac{1}{b_1}(-z_1 + a_1 x_4 x_6 - b_1 \tau_r + \alpha_1[\alpha_1 z_1 + z_2] + \alpha_2 z_2) \quad (152)$$

$$U_4 = \frac{1}{b_1}(-z_1 + a_1 x_4 x_6 - b_1 \tau_r + \alpha_1[\alpha_1 z_1 + z_2] + \alpha_2 z_2) \quad (153)$$

Control del subsistema de traslación:

La entrada de control de altura U_1 se calcula con un procedimiento similar al efectuado en la sección anterior y se obtiene la siguiente señal de control:

$$U_1 = \frac{1}{b_1}(-z_1 + a_1 x_4 x_6 - b_1 \tau_r + \alpha_1[\alpha_1 z_1 + z_2] + \alpha_2 z_2) \quad (154)$$

Control de traslación plano X-Y.

Para el control en este plano se definen las siguientes variables:

$$u_x = (\sin x_5 \sin x_1 + \cos x_5 \sin x_3 \cos x_1)$$

$$u_y = (-\cos x_5 \sin x_1 + \sin x_5 \sin x_3 \cos x_1)$$

Al sustituir las variables U_x y U_y en el modelo del plano X-Y tenemos:

$$x_{10} = u_x \frac{U_1}{m_T}$$

$$x_{12} = u_y \frac{U_1}{m_T}$$

Como se puede observar en las ecuaciones del modelo dinámico la traslación en el plano X-Y depende de una entrada de control (U_1). Al ser un sistema subactuado la traslación en este plano depende del empuje total y del ángulo de balanceo y cabeceo del vehículo por lo cual al utilizar U_x y U_y como la orientación de U_1 podemos calcular los ángulos de cabeceo y balanceo necesarios para el desplazamiento deseado en el eje X e Y. siguiendo la metodología de control anterior obtenemos las siguientes entradas de control:

$$U_x = \frac{m_T}{U_1}(z_9 - \alpha_9(\alpha_9 z_9 + z_{10}) - \alpha_{10} z_{10}) \quad (155)$$

$$U_y = \frac{m_T}{U_1} (z_9 - \alpha_9(\alpha_9 z_9 + z_{10}) - \alpha_{10} z_{10}) \quad (156)$$

CONTROL DEL SUBSISTEMA DEL MANIPULADOR.

Para el control del subsistema del manipulador partimos de la ecuación obtenida en el modelado:

$$x_{14} = M(q) (\tau - C\dot{x}_{13} + G) \quad (157)$$

Se definen los errores como:

$$z_{13} = [q - q_d] = [x_{13} - x_{13d}] \quad (158)$$

Siendo sus derivadas:

$$z_{13} = z_{14} = [\dot{q} - \dot{q}_d] = [\dot{x}_{13} - \dot{x}_{13d}] \quad (159)$$

$$z_{13} = z_{14} = [\ddot{q} - \ddot{q}_d] = [\ddot{x}_{13} - \ddot{x}_{13d}] \quad (160)$$

La entrada de control virtual propuesta y su correspondiente derivada es:

$$z_{14} = z_{13} + kz_{13} \quad (161)$$

$$\dot{z}_{14} = \dot{z}_{13} + k\dot{z}_{13} \quad (162)$$

Donde k es una matriz diagonal definida positiva, despejando de la ecuación (161):

$$\dot{z}_{13} = z_{14} - kz_{13} \quad (163)$$

Usando la ecuación (162) en la ecuación (163) obtenemos:

$$\dot{z}_{14} = [\ddot{x}_{13d} - \ddot{x}_{13}] + k\dot{z}_{13} \quad (164)$$

Sabiendo el valor de \ddot{x}_{13} proveniente del modelo dinámico y usando la ecuación (164) tenemos:

$$\dot{z}_{14} = [(M^{-1}(\tau - C\dot{x}_{13} - G)) - \ddot{x}_{13d}] + k[z_{14} - kz_{13}] \quad (165)$$

Se propone una función de Lyapunov que sea capaz de controlar ambos errores (z_{13} y z_{14}) con la cual se garantizara además la estabilidad del sistema:

$$V(z_{13}, z_{14}) = \frac{1}{2} z_{13}^T k z_{13} + \frac{1}{2} z_{14}^T M z_{14} \quad (166)$$

Derivando la función de Lyapunov se tiene:

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T k \dot{z}_{13} + \frac{1}{2} z_{14}^T \dot{M} z_{14} + z_{14}^T M \dot{z}_{14} \quad (167)$$

Fusionando la ecuación (167), (161) y la ecuación (162) con la ecuación anterior logramos:

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T k [z_{14} - kz_{13}] + \frac{1}{2} z_{14}^T \dot{M} z_{14} + z_{14}^T ([\tau - C\dot{x}_{13} - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}])$$

Despejando el valor de \dot{x}_{13} y el valor de \dot{z}_{13} y sustituyéndolos en la ecuación anterior obtenemos:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k[z_{14} - kz_{13}] + \frac{1}{2} z_{14}^T \dot{M} z_{14} \\ &\quad + z_{14}^T ([\tau - C[x_{13d} + (z_{14} - kz_{13})] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \end{aligned} \quad (168)$$

Según el trabajo expuesto en [93] se sabe que:

$$z_{14}^T (\dot{M} - 2C) z_{14} = 0$$

Aplicando esta propiedad simplificamos a:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k[z_{14} - kz_{13}] \\ &\quad + z_{14}^T \left([\tau - C[x_{13d} + (z_{14} - kz_{13})] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}] + \frac{1}{2} \dot{M} z_{14} \right) \\ \dot{V}(z_{13}, z_{14}) &= z_{13}^T k[z_{14} - kz_{13}] + z_{14}^T ([\tau - C[x_{13d} - kz_{13}] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \\ \dot{V}(z_{13}, z_{14}) &= z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} \\ &\quad + z_{14}^T (\tau - C[x_{13d} - kz_{13}] - G - M\ddot{x}_{13d} + Mk z_{14} - Mk^2 z_{13}) \end{aligned} \quad (169)$$

Finalmente se propone una señal de control que satisfaga las condiciones de estabilidad del sistema, en este caso se propone:

$$\tau = M\ddot{x}_{13d} + Cx_{13d} - Mkz_{14} + G - kz_{13} \quad (170)$$

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} + z_{14}^T (-Ckz_{13} - Mk^2 z_{13} - kz_{13}) \\ \dot{V}(z_{13}, z_{14}) &= z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} - z_{14}^T Ckz_{13} - z_{14}^T Mk^2 z_{13} - z_{14}^T k z_{13} \end{aligned}$$

Para comprobar que la derivada sea definida negativa se utiliza las propiedades de la norma matricial:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= \|z_{13}^T k z_{14}\| - z_{13}^T k^2 z_{13} - z_{14}^T Ckz_{13} - z_{14}^T Mk^2 z_{13} \quad \|-z_{14}^T k z_{13}\| \\ &\leq -z_{13}^T k^2 z_{13} - z_{14}^T Ckz_{13} - z_{14}^T Mk^2 z_{13} \\ &\leq -(Ck + Mk^2) \|z_{14}^T\| \|z_{13}^T\| - k^2 \|z_{13}\|^2 < 0 \end{aligned}$$

Por lo tanto, la derivada de la función de Lyapunov es definida negativa lo cual garantiza la estabilidad y da por válida la ley de control propuesta.

3.8.2 CONTROL PID SISTEMA UAV MANIPULADOR

Al igual que en los controladores PID anteriores partimos la ecuación general del controlador:

$$U(t) = P + D + G = e(t)kp + kd \frac{de(t)}{dt} + G \quad (171)$$

Donde Kd, Kp, son las ganancias proporcionales, derivativas y el termino G corresponde a la compensación de los pares gravitatorios, U representa la señal de control calculada y e(t) es el error o la diferencia entre la posición real y la posición deseada. Por lo que después de un proceso estocástico, obtenemos las siguientes ganancias de control para cada PID necesario para los sistemas:

3.9 CONTROL DE FUERZA EN MAIPULADOR

Cuando se pretende realizar una tarea en la que el robot deba mantener contacto con el medio o ejercer fuerza sobre este, y que además se considere ejercer un rango de fuerza necesario para la correcta ejecución de la tarea, es indispensable aplicar un control de fuerzas para el manipulador, El cual no es necesario cuando solo se requiere un control en posición. Al implementar un control de este tipo se puede determinar con exactitud la fuerza que se requiere que el manipulador ejerza sobre el entorno.

Puesto que el manipulador sostendrá un objeto de masa variable es necesario añadirla al sistema del manipulador y considerarla en el control del mismo, de esta forma el controlador diseñado para el UAV corregirá de manera automática la potencia necesaria para el seguimiento de la trayectoria deseada. Para lograrlo se considera una incertidumbre paramétrica en el manipulador, debido a la cual se ajustará el diseño del controlador Backstepping, de modo que sea capaz de soportar el cambio de masa en el manipulador y debido a la forma del diseño en cascada el controlador Backstepping del UAV corregirá de manera automática las señales de control al variar la masa del manipulador. Por lo que para el control del sistema UAV-Manipulador con la manipulación de un objeto simplemente se considerara variable la masa 2 correspondiente al manipulador. Considerando la ecuación obtenida en el modelado se tiene:

$$M(q) \ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

No obstante, se sabe que la masa 2, correspondiente al eslabón dos del robot, puede variar por lo tanto podemos definir las matrices totales de la ecuación anterior, como:

$$M_T = M + M\Delta$$

$$C_T = C + C\Delta$$

$$G_T = G + G\Delta$$

Donde $M\Delta$, $C\Delta$ y $G\Delta$ representan las incertidumbres provocadas por el cambio de la masa 2 del manipulador, agrupando estas incertidumbres y realizando el correspondiente despeje obtenemos:

$$M(q) \ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - \Delta$$

Para rediseñar el controlador se parte de la misma señal de error:

$$z_{13} = [q - q_d] = [x_{13} - x_{13d}] \quad (172)$$

Siendo sus derivadas:

$$z_{13}^{\dot{}} = z_{14} = [\dot{q} - \dot{q}_d] = [\dot{x}_{13} - \dot{x}_{13d}] \quad (173)$$

$$z_{13}^{\ddot{}} = z_{14}^{\dot{}} = [\ddot{q} - \ddot{q}_d] = [\ddot{x}_{13} - \ddot{x}_{13d}] \quad (174)$$

La entrada de control virtual propuesta y su correspondiente derivada es:

$$z_{14} = z_{13}^{\dot{}} + kz_{13} \quad (175)$$

$$\dot{z}_{14} = \dot{z}_{13} + k\dot{z}_{13} \quad (176)$$

Donde k es una matriz diagonal definida positiva, despejando de la ecuación (173):

$$\dot{z}_{13} = \dot{z}_{14} - k\dot{z}_{13} \quad (177)$$

Usando la ecuación (174) en la ecuación (176) logramos:

$$\dot{z}_{14} = [\ddot{x}_{13d} - \ddot{x}_{13}] + k\dot{z}_{13} \quad (178)$$

Sabiendo el valor de \ddot{x}_{13} proveniente del modelo dinámico y usando la ecuación 130 tenemos:

$$\dot{z}_{14} = [(M^{-1}(\tau - \Delta - C\dot{x}_{13} - G)) - \ddot{x}_{13d}] + k[z_{14} - kz_{13}] \quad (179)$$

Se propone una función de Lyapunov que sea capaz de controlar ambos errores (z13 y z14) con la cual se garantizara además la estabilidad del sistema:

$$V(z_{13}, z_{14}) = \frac{1}{2} z_{13}^T k z_{13} + \frac{1}{2} z_{14}^T M z_{14} \quad (180)$$

Derivando la función de Lyapunov tenemos:

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T k \dot{z}_{13} + \frac{1}{2} z_{14}^T \dot{M} z_{14} + z_{14}^T M \dot{z}_{14} \quad (181)$$

Fusionando las ecuaciones (175), (176) y (181) con la ecuación anterior se obtiene:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] + \frac{1}{2} z_{14}^T \dot{M} z_{14} \\ &+ z_{14}^T ([\tau - \Delta - C\dot{x}_{13} - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \end{aligned}$$

Despejando el valor de \dot{x}_{13} y valor de \dot{z}_{13} y sustituyéndolos en la ecuación anterior obtenemos:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] + \frac{1}{2} z_{14}^T \dot{M} z_{14} \\ &+ z_{14}^T ([\tau - \Delta - C[x_{13d} + (z_{14} - kz_{13})] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \end{aligned} \quad (182)$$

Considerando que:

$$z_{14}^T (\dot{M} - 2C) z_{14} = 0$$

Aplicando esta propiedad simplificamos a:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] \\ &+ z_{14}^T \left([\tau - \Delta - C[x_{13d} + (z_{14} - kz_{13})] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}] \right. \\ &\quad \left. + \frac{1}{2} \dot{M} z_{14} \right) \\ \dot{V}(z_{13}, z_{14}) &= z_{13}^T k [z_{14} - kz_{13}] + z_{14}^T ([\tau - \Delta - C[x_{13d} - kz_{13}] - G - M\ddot{x}_{13d}] + Mk[z_{14} - kz_{13}]) \\ \dot{V}(z_{13}, z_{14}) &= z_{13}^T k z_{14} - z_{13}^T k^2 z_{13} \\ &+ z_{14}^T (\tau - \Delta - C[x_{13d} - kz_{13}] - G - M\ddot{x}_{13d} + Mk z_{14} - Mk^2 z_{13}) \end{aligned} \quad (183)$$

Finalmente se propone una señal de control que satisfaga las condiciones de estabilidad del sistema, en este caso se propone:

$$\tau = M\ddot{x}_{13d} + Cx_{13d} - Mkz_{14} + G - kz_{13} + \|\Delta\| \quad (184)$$

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T kz_{14} - z_{13}^T k^2 z_{13} + z_{14}^T (-Ckz_{13} - Mk^2 z_{13} - kz_{13})$$

$$\dot{V}(z_{13}, z_{14}) = z_{13}^T kz_{14} - z_{13}^T k^2 z_{13} - z_{14}^T Ckz_{13} - z_{14}^T Mk^2 z_{13} - z_{14}^T kz_{13}$$

Para comprobar que la derivada de la función candidata a Lyapunov sea definida negativa se utiliza la norma matricial ^[95]:

$$\begin{aligned} \dot{V}(z_{13}, z_{14}) &= \|z_{13}^T kz_{14}\| - z_{13}^T k^2 z_{13} - z_{14}^T Ckz_{13} - z_{14}^T Mk^2 z_{13} \quad \|-z_{14}^T kz_{13}\| \\ &\leq -z_{13}^T k^2 z_{13} - z_{14}^T Ckz_{13} - z_{14}^T Mk^2 z_{13} \\ &\leq -(Ck + Mk^2) \|z_{14}^T\| \|z_{13}^T\| - k^2 \|z_{13}\|^2 < 0 \end{aligned}$$

Con lo anterior se comprueba que la señal de control propuesta para la incertidumbre paramétrica del manipulador es correcta.

3.10 Sistema de retroalimentación visual

En esta subsección se presenta el desarrollo del sistema de retroalimentación basado de visión, el procedimiento se encuentra en proceso, la idea general es la identificación de puntos basados en la infraestructura, y posteriormente identificar la distancia del UAV hacia el punto en la infraestructura. La estructura a la que el sistema UAV-Manipulador debe acercarse a un patrón el cual se fijó previamente. La ventaja de colocar un patrón radica en el hecho de que se conoce la ubicación absoluta del patrón, lo cual permite reducir el problema a cuantificar la distancia del patrón al MM-UAV. Para efectuar la medición de esta distancia esto se debe efectuar una calibración de la cámara colocada en el UAV.

3.10.1 Calibración de la cámara para medición

La calibración de una cámara es un tema ampliamente estudiado, consiste en ajustar las lecturas del sensor de la cámara para que coincida con el mundo real. La cámara tiene dos tipos de parámetros, los denominados parámetros intrínsecos describen la óptica y la geometría interna de la cámara y los parámetros extrínsecos que relacionan la posición del sensor con la posición en el mundo real. A través de los cálculos de ambos parámetros se puede ubicar un punto del entrono en 3 dimensiones en una imagen en dos dimensiones tal y como se muestra en la Figura 11.

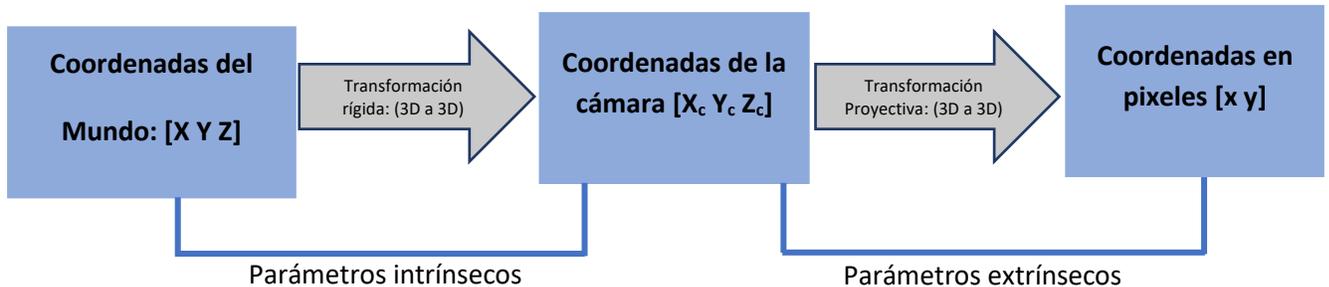


Figura 11. Proceso de calibración de la cámara.

Existen diferentes procedimientos y métodos para la calibración, la forma más común es a través de un patrón de forma y medidas conocidas. A su vez el patrón más utilizado es el tablero de ajedrez. El procedimiento estándar consta en tomar múltiples capturas del patrón en diversas posiciones y orientaciones del mundo real. Posteriormente se detectan las esquinas en el patrón excluyendo los cuadros externos en las orillas, esto se puede realizar de manera manual en una imagen o de manera automática con algún software de calibración. Si la distorsión de la cámara es muy alta existe la posibilidad de que el proceso de calibración falle, en ese determinado caso se puede recurrir a un filtro para corregir la deformación conocida de la cámara. Nuevamente esta es una opción en múltiples softwares de calibración.

Para el trabajo presentado en esta tesis doctoral se utilizaron las herramientas de calibración integradas en Open CV debido a su amplio uso y soporte de la comunidad además de ser software libre. Finalmente, los parámetros obtenidos son válidos dentro de un rango de visión, si la cámara se aleja demasiado del punto de calibración se requiere recalcular los parámetros de la cámara. Esto genera un problema adicional ya que se debe estimar rangos de posición en los que los parámetros son válidos y en cual se debe recalibrar. Se ha determinado mediante pruebas que el rango de operación con los parámetros calculados es de aproximadamente un metro.

3.10.2 Red neuronal profunda para estimación de rangos

El problema de recalibración puede abordarse de distintos enfoques como el abordado en [94] donde los autores trabajan un sistema de recalibración automática en línea. Este enfoque no fue abordado ya que el sistema debe operar a 4 metros y debido al tamaño del patrón se requiere ajustar no solo las matrices sino la región de interés en la fotografía para detectar con éxito el patrón. Considerando esto se utilizó por un enfoque basado en redes neuronales para la implementación de la estimación del rango.

La red neuronal profunda abreviada DNN por sus siglas en inglés (Deep Neural Network) cuenta con múltiples capas ocultas entre la capa de entrada y la capa de salida. En el presente trabajo la red tuvo como objetivo estimar los rangos de distancia para la utilización de diferentes matrices de calibración acorde a la distancia del patrón.

La red neuronal será entrenada para clasificar las imágenes en 3 tipos, lejanas en un rango de 3.5 a 2.5 metros; medianas en un rango de 2.5 a 1.5 metros y finalmente en cercanas en un rango de 1.5 a 0.5 metros. En este rango el manipulador debe ser capaz de efectuar la tarea, no se puede acercar más el UAV debido a la distancia de los brazos y las hélices, por lo que la distancia máxima de acercamiento será de 0.5 metros.

Se utilizó un enfoque de propagación hacia atrás supervisado y el método del descenso de gradiente para optimizar la red y con esto se minimiza la función de pérdida. El código utilizado para las pruebas puede encontrarse en la sección de anexos.

4

RESULTADOS

Los subsistemas de traslación y rotación, así como sus respectivos controles se simulan con una función S. Se utilizaron cinco funciones S, dos para simular la dinámica de cada subsistema (rotación y traslación del UAV), dos para simular el control y una para la trayectoria deseada. La función S proporciona un mecanismo poderoso para usar el código m (código nativo de Matlab) en un bloque de Simulink. Los códigos implementados para cada función S pueden consultarse en los anexos de esta tesis.

Para la simulación de seguimiento de trayectorias se simuló el proceso durante 90 s., además se utilizó el método de paso variable ODE 45. Para probar su funcionamiento se utilizan las siguientes trayectorias de referencia:

$$\begin{aligned} \text{Trayectoria 1: } & x_r = \left(\frac{1}{2}\right) * \cos\left(\frac{1}{2} * t\right) - \frac{1}{2}; \quad y_r = \left(\frac{1}{2}\right) * \sin\left(\frac{1}{2} * t\right); \quad z_r = \left(\frac{t}{30}\right); \quad \psi_r = \frac{\pi}{3}; \\ \text{Trayectoria 2: } & x_r = (3) * \cos(0.35 * t) * \cos(0.35 * t); \quad y_r = (3) * \cos(0.35 * t) * \sin(0.35 * t); \\ & z_r = (0.000875 * t); \quad \psi_r = \frac{\pi}{9}; \quad \text{con perturbaciones aleatorias (para simular los efectos del aire).} \end{aligned}$$

Trayectoria 1. El sistema se probó con una referencia en forma de trébol. En esta prueba, el hexarrotor comienza con un error inicial de 1 m en la posición, como se muestra en la Figura 13, lo que supone un reto dadas las características del vehículo.

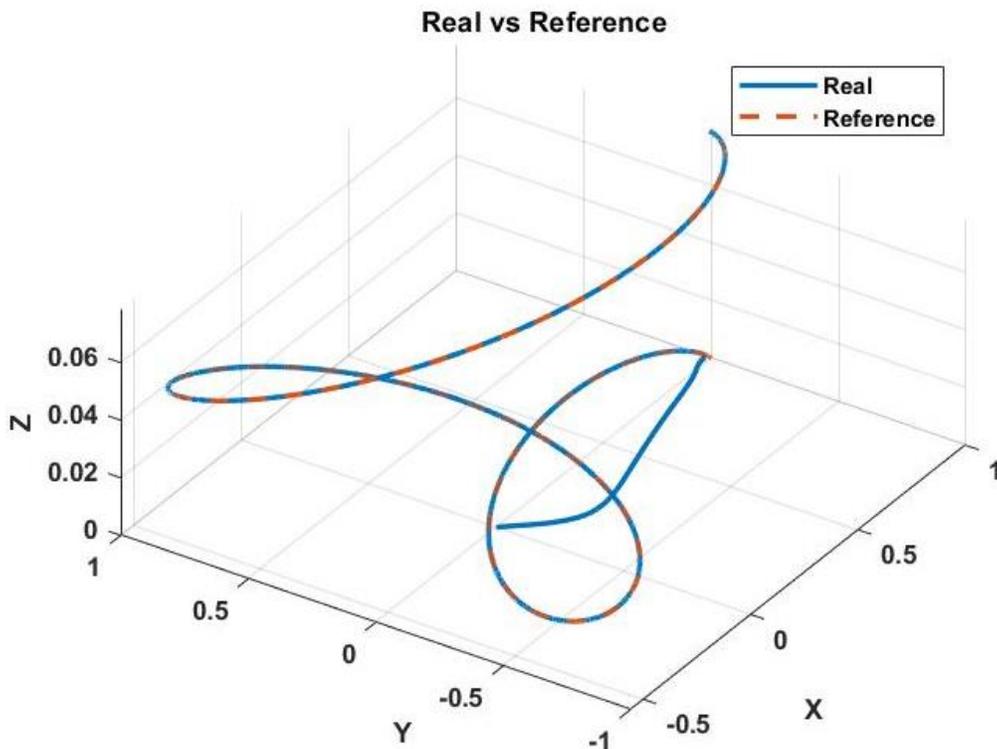


Figura 13 . Trayectoria real del UAV (línea solida azul) vs. Trayectoria 1 (línea roja punteada).

La respuesta del sistema a la trayectoria de referencia 1 muestran que el vehículo puede seguir la referencia aun con errores iniciales. La Figura 14, Figura 15 y Figura 16 muestran el comportamiento del dron en cada eje.

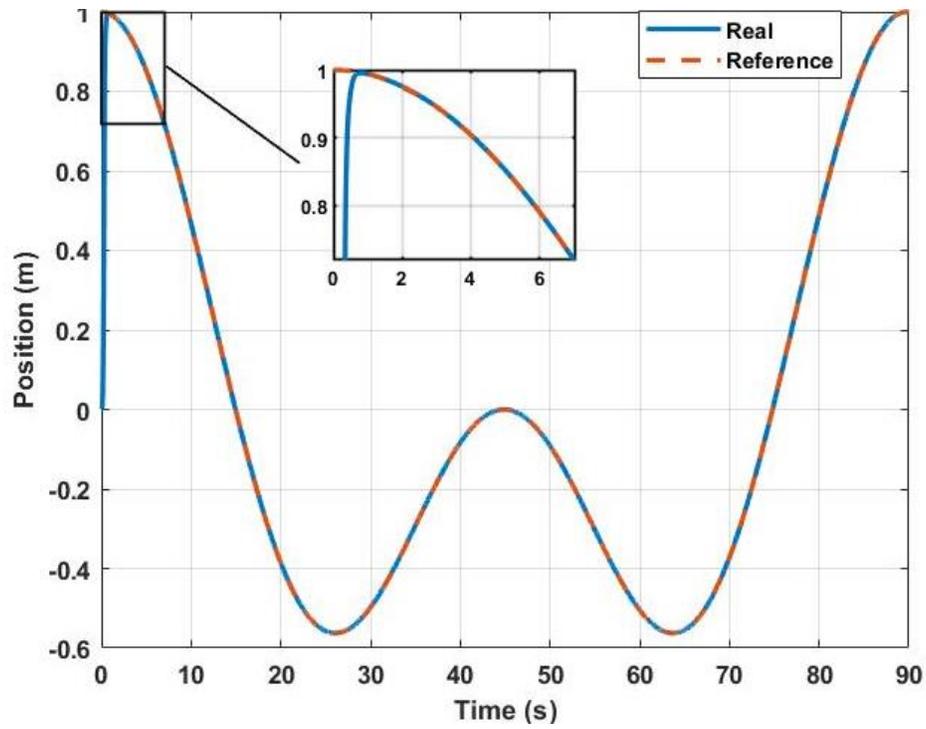


Figura 14. X real del UAV (línea solida azul) vs. X deseada (línea roja punteada).

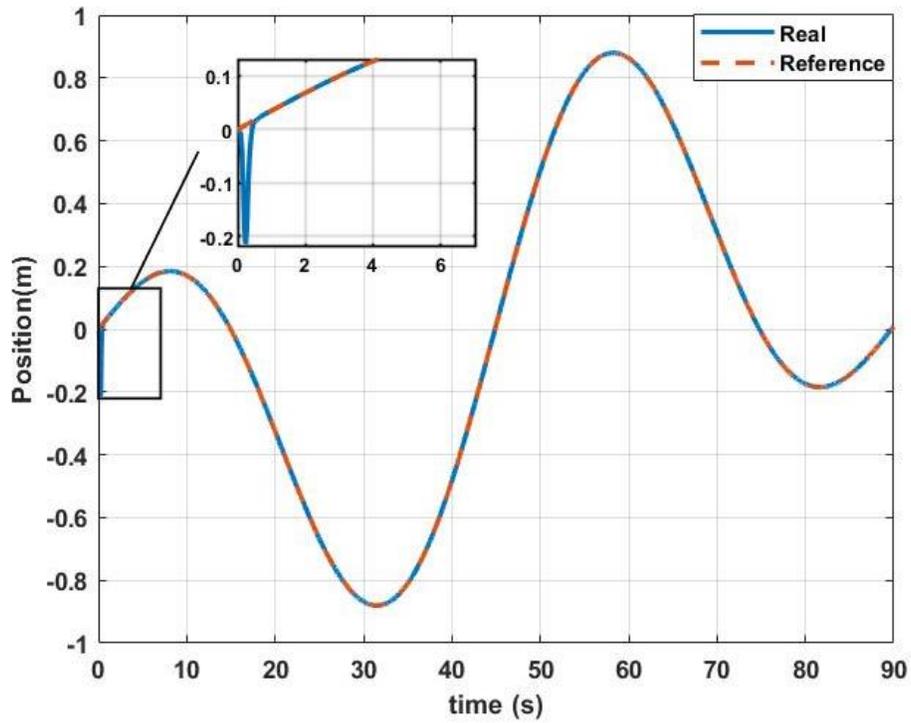


Figura 15. Y real del UAV (línea solida azul) vs. Y deseada (línea roja punteada).

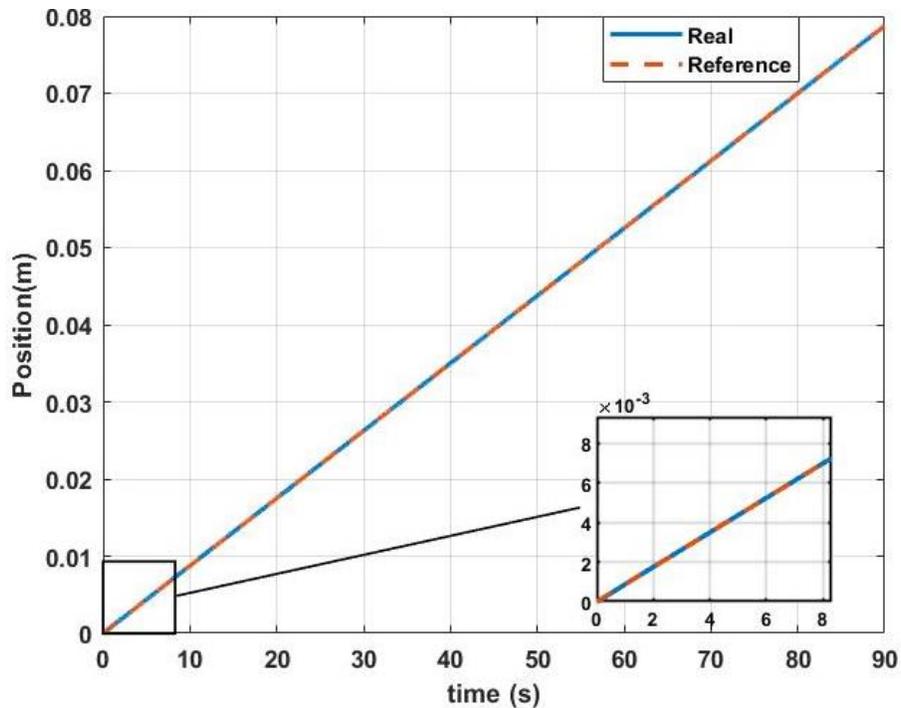


Figura 16. Z real del UAV (línea solida azul) vs. Z deseada (línea roja punteada).

El funcionamiento del sistema se puede analizar observando los errores de seguimiento en posición y orientación, ambos son mostrados en las Figura 17 y Figura 18, respectivamente.

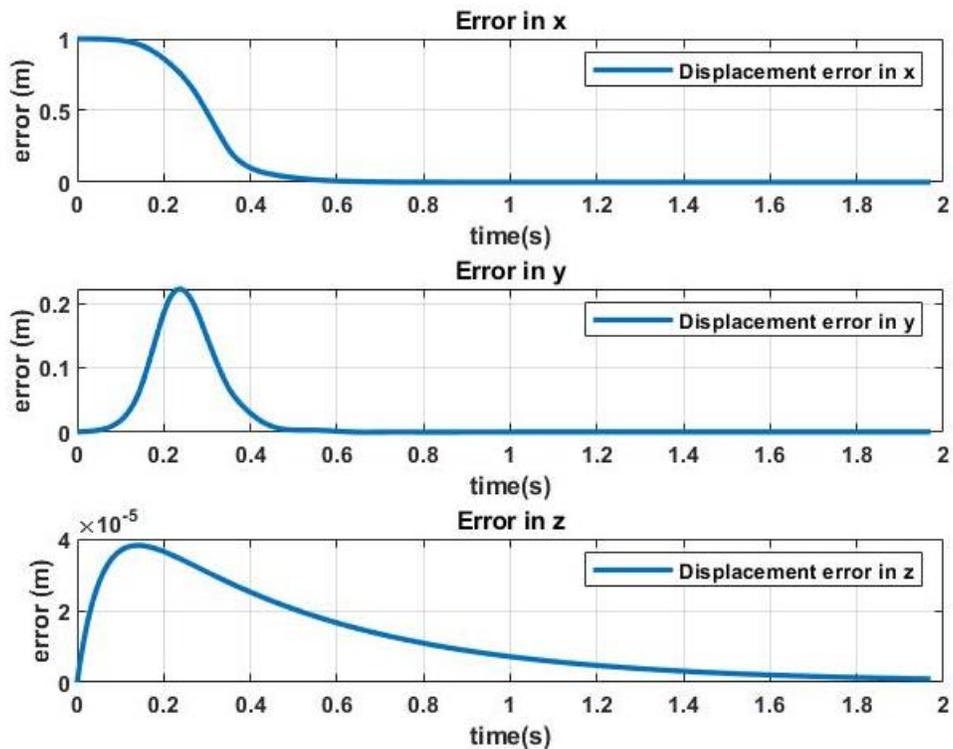


Figura 17. Errores de desplazamiento de la trayectoria 1 del UAV.

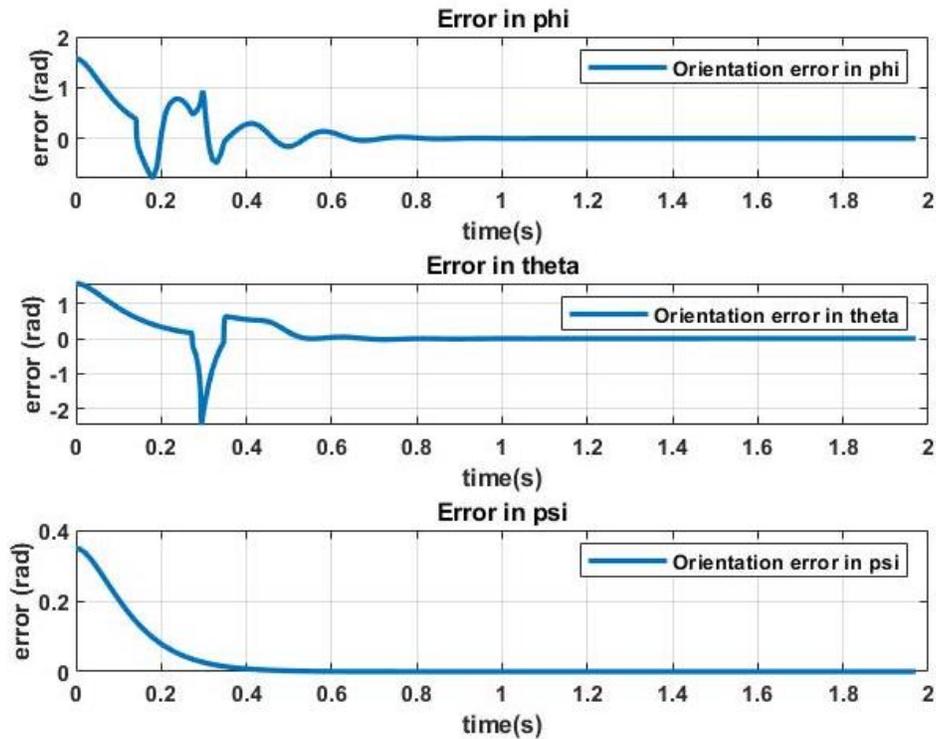


Figura 18. Errores de orientación de la trayectoria 1 del UAV.

Como puede verse, los errores del seguimiento de la posición y la orientación tienden a 0 rápidamente y permanecen en ese valor durante los 90 s de la simulación.

Trayectoria 2. Esta trayectoria es de tipo helicoidal en la que el hexarrotor comienza con un error inicial de 0; además, se agregó una perturbación aleatoria durante la simulación, la cual comenzando a los 4.5 s y termina a los 9 s; la perturbación tiene componentes aleatorios que cambian de valor durante la perturbación. A través de pruebas adicionales, se determinó que el controlador Backstepping diseñado es capaz de rechazar una perturbación con una magnitud máxima de 10.1 m/s, siempre que ningún componente de la perturbación supere los 6.6 m / s. Los resultados del controlador bajo perturbación se muestran desde la Figura 19 a la Figura 22. El efecto de la perturbación se puede observar a partir de los 4,5 s en las gráficas. Puede verse que el controlador es capaz de volver a la trayectoria deseada una vez que desaparece la perturbación.

La trayectoria helicoidal utilizada se puede ver en la Figura 19. La respuesta del controlador en cada uno de los ejes del sistema inercial fijo X, Y, Z se muestra desde la Figura 20 hasta la Figura 22.

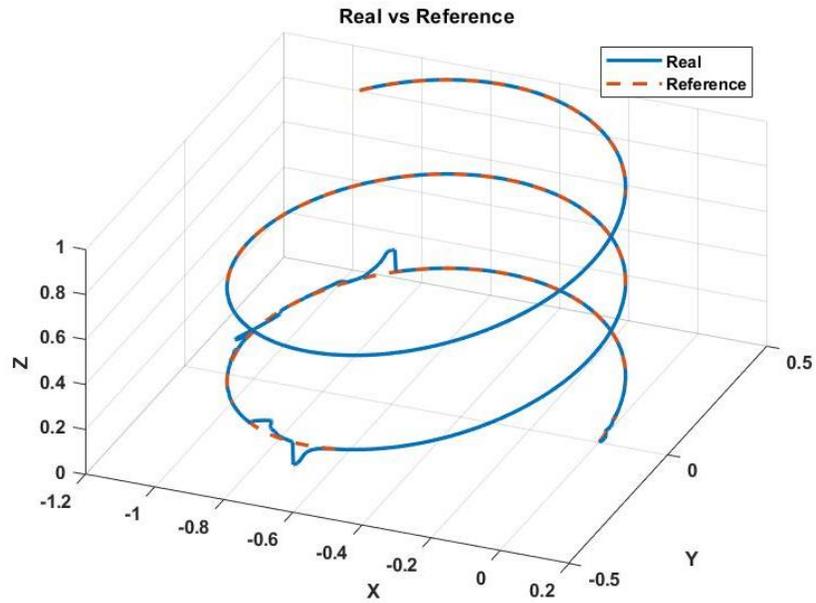


Figura 19. Trayectoria helicoidal con perturbaciones. Posición Real del UAV (línea sólida azul) vs. Posición deseada (línea roja punteada).

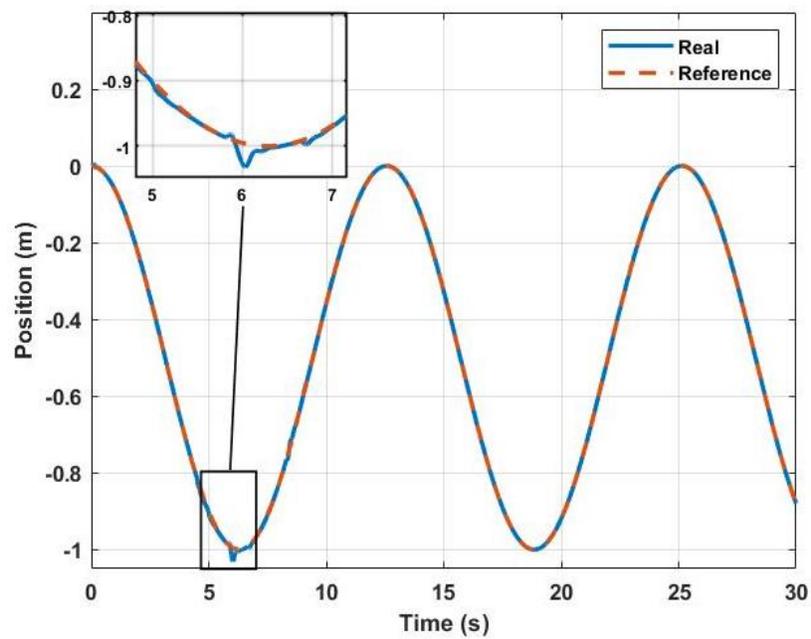


Figura 20. X real del UAV en trayectoria helicoidal (línea sólida azul) vs. X deseada (línea roja punteada).

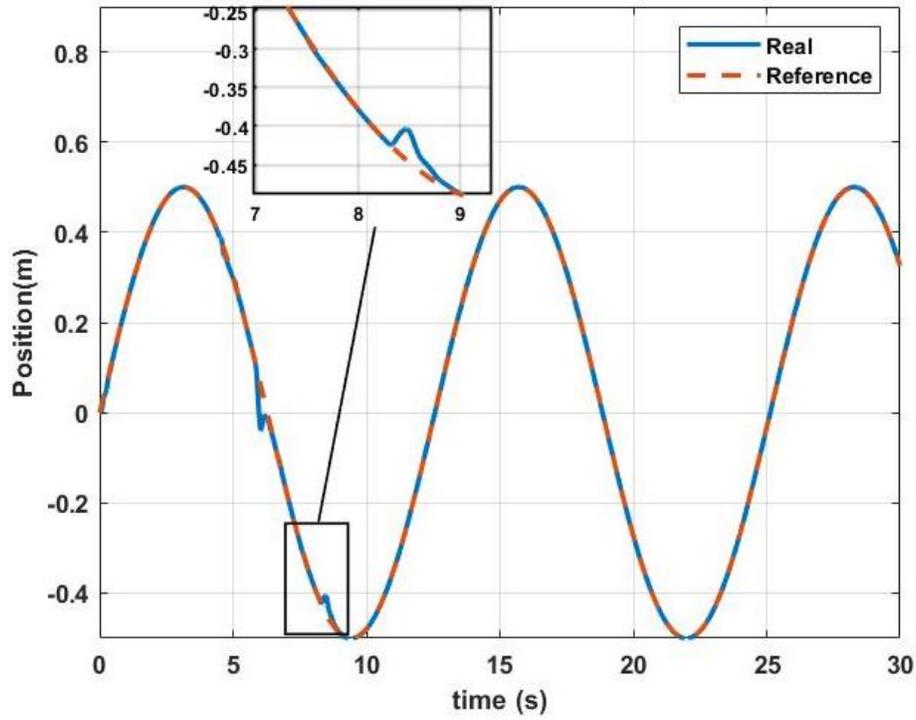


Figura 21. Y real del UAV en trayectoria helicoidal (línea sólida azul) vs. Y deseada (línea roja punteada).

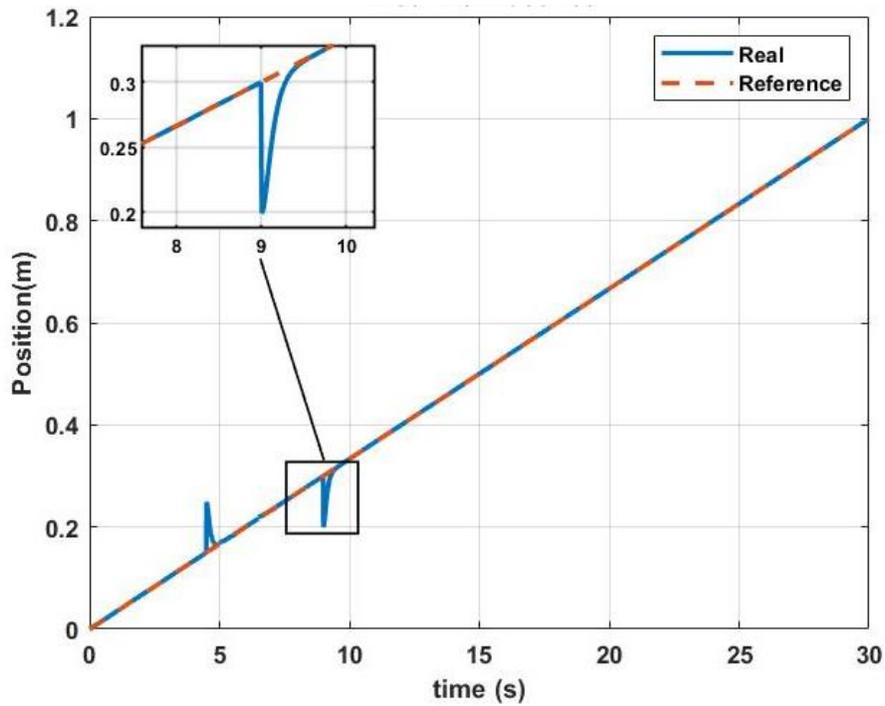


Figura 22. Z real del UAV en trayectoria helicoidal (línea sólida azul) vs. Z deseada (línea roja punteada).

De igual manera en la Figura 23 y Figura 24 se muestran los errores de seguimiento tanto en posición como en rotación respectivamente.

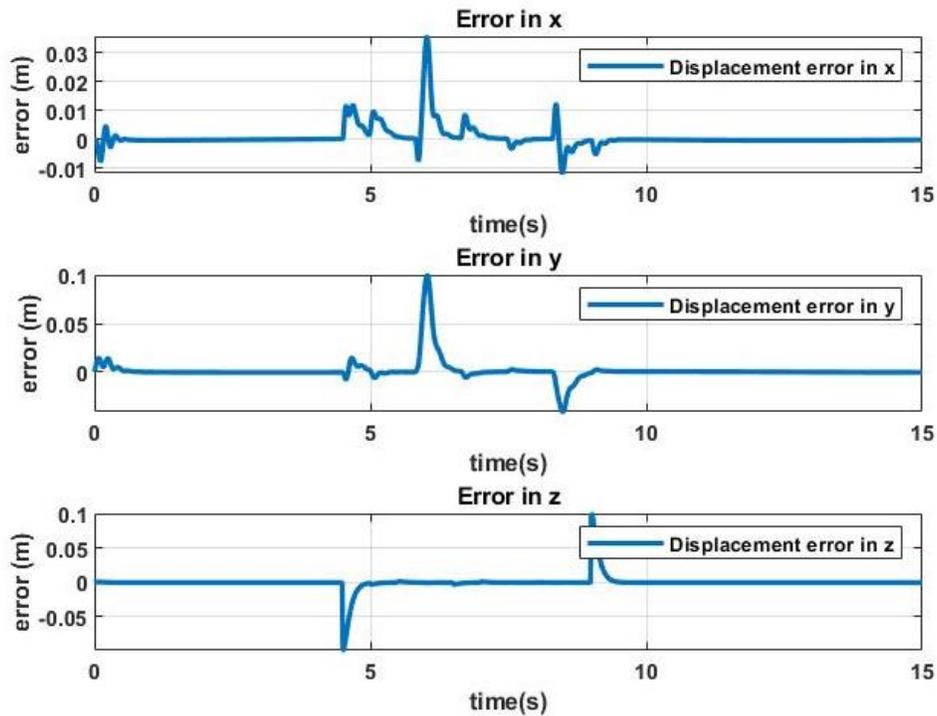


Figura 23. Errores de desplazamiento del UAV en trayectoria helicoidal.

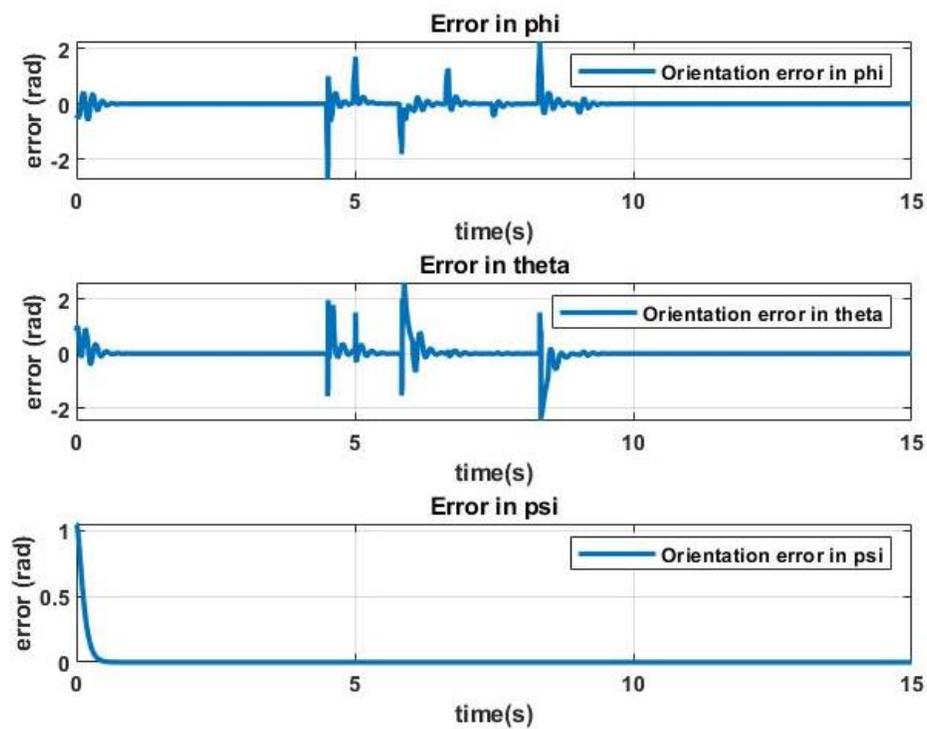


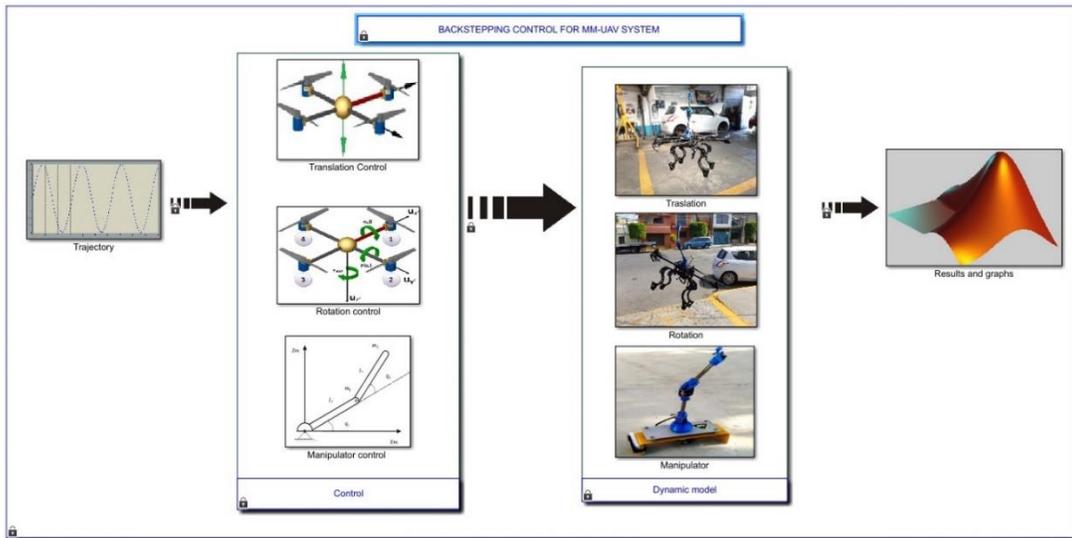
Figura 24. Errores de rotación del UAV en trayectoria helicoidal.

Las simulaciones muestran que el controlador es capaz de seguir trayectorias incluso en presencia de errores iniciales o perturbaciones aleatorias, como las que pueden causar corrientes de aire en un vuelo. El sistema gestiona correctamente el error.

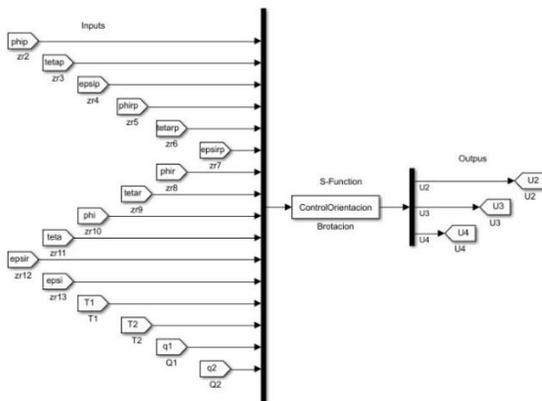
4.2 Resultados de simulación para el control del sistema MM-UAV

En esta sección se describen los resultados de simulación del sistema MM-UAV. Para simular el sistema nuevamente se utiliza un modelo de Simulink (Figura 25a). El Sistema es similar al de la sección 4.1, sin embargo, se ha añadido el Sistema del manipulador por lo cual las ecuaciones que se implementan en esta simulación es el sistema de ecuaciones (141), y las leyes de control (151) a (156). Para este caso se utilizaron 7 funciones S, tres para simular la dinámica (Rotación UAV, Traslación UAV y manipulador), tres para simular sus respectivas leyes de control y la última se utiliza para programar la ecuación de la trayectoria deseada.

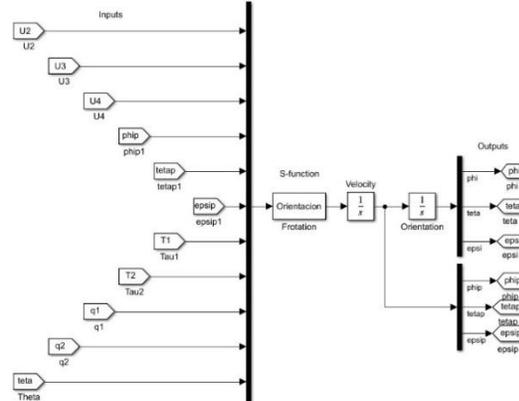
Los parámetros de simulación son los mismos que los mostrados en la sección anterior, 90 segundos de simulación para el seguimiento de trayectorias y se utiliza el método numérico de paso variable ODE 45. En la Figura 25b se puede observar el subsistema de control de orientación y en la Figura 25c se puede observar el subsistema de rotación del UAV.



(a)



(b)



(c)

Figura 25. Simulación del sistema MM-UAV. (a) Modelo de simulink; (b) subsistema de control de rotación; (c) subsistema de rotación.

Para que esta simulación sea lo más precisa posible, se deben utilizar los parámetros físicos del sistema, para ello se deben realizar mediciones sobre la planta del sistema, para el trabajo de esta tesis se utilizarán los elementos mostrados en la Figura 26. Donde es el hexarrotor utilizado se muestra en la Figura 26a y la Figura 26b exhibe el manipulador planar utilizado, ambos sistemas son de fabricación propia, el hexarrotor fue hecho a partir de placas y tubos de carbono, mientras que el manipulador fue hecho a partir de acero inoxidable 304 Cal 16.



Figura 26. Sistema físico: (a) UAV hexarrotor; (b) Manipulador planar.

| Parámetro | Valor | Descripción |
|------------|---|---|
| m | 3.2 Kg. | Masa de UAV (con batería). |
| m1, m2 | 0.2 Kg, 0.18 Kg | Masas de los eslabones del manipulador. |
| lx, ly, lz | $28.7 \times 10^{-3} m^4, 20.7 \times 10^{-3} m^4, 20.7 \times 10^{-3} m^4$ | Inercia del UAV. |
| l1, l2 | $8.6 \times 10^{-3} m^4, 7.8 \times 10^{-3} m^4$ | Inercia de los eslabones del manipulador. |
| l | 0.35 m | Longitud de los brazos del UAV |
| l1, l2 | 0.21 m, 0.21 m | Longitud de los eslabones del manipulador |

Tabla 3. Parámetros físicos del Sistema MM-UAV.

Nuevamente se proponen dos trayectorias de referencia para probar el controlador, cabe señalar que el controlador a este punto ya se encuentra sintonizado, el proceso de sintonización se describe a detalle en la sección 4.3. La trayectoria 1 consta de una trayectoria helicoidal mostrada por la Ecuación (185). Para la trayectoria 2 se propone una trayectoria muy útil desde el punto de vista práctico, ya que es una trayectoria típica de aproximación de un UAV, la ecuación (186) describe esta trayectoria.

$$\begin{aligned}
 \text{Trayectoria 1: } x_r &= (4 * \cos(0.25 * t)) - 4; y_r = 4 * \cos(0.25 * t); z_r = \frac{t}{15}; \psi_r = \frac{\pi}{3}; q_1 \\
 &= l_2 * \cos\left(\frac{t}{3}\right) + l_1 * \cos\left(\frac{\pi}{4}\right); q_2 = l_2 * \sin\left(\frac{t}{3}\right) + l_1 * \sin\left(\frac{\pi}{4}\right);
 \end{aligned}
 \tag{185}$$

$$\begin{aligned}
 \text{Trayectoria 2: } x_r &= y_r = z_r = \psi_r = \text{step}\left(\frac{1}{s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 5s + 1}\right); q_1 \\
 &= \left(\frac{4}{5}l_2\right) * \cos\left(\frac{t}{3}\right) + l_1 * \cos\left(\frac{\pi}{4}\right); q_2 = \left(\frac{4}{5}l_2\right) * \sin\left(\frac{t}{3}\right) + l_1 * \sin\left(\frac{\pi}{4}\right);
 \end{aligned}
 \tag{186}$$

La respuesta del sistema se puede observar en la XXX, donde la XXX muestra la posición del UAV durante la simulación y la XXXX muestra la posición del manipulador.

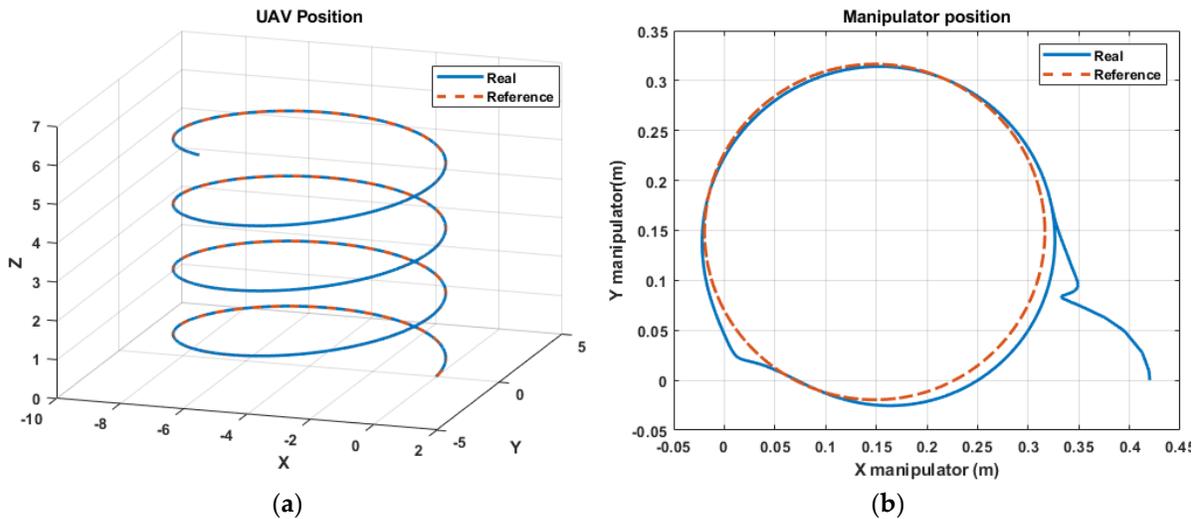
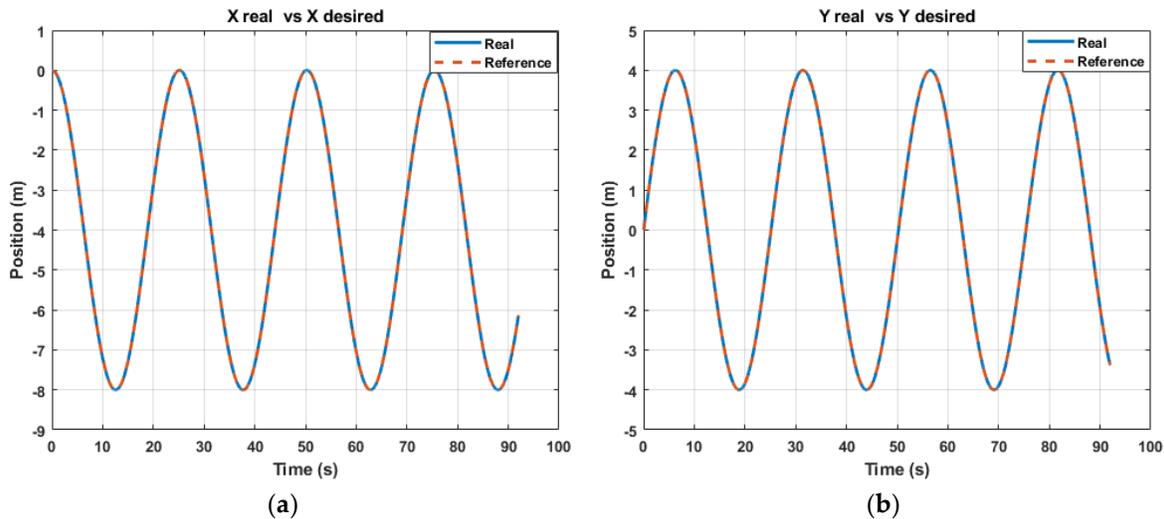


Figura 27. MM-UAV Posición cartesiana del sistema MM-UAV para la trayectoria simulada 1. (a) Posición del UAV; (b) Position del manipulador.

Los errores de seguimiento por eje pueden observarse en la Figura 28. MM-UAV Posición por ejes del sistema MM-UAV para la trayectoria simulada 1. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador.



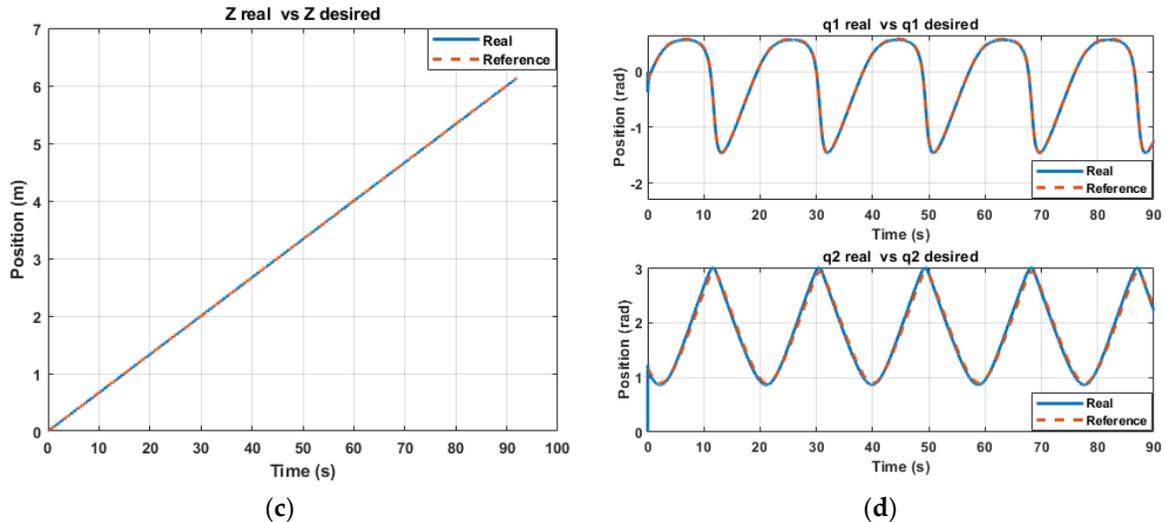


Figura 28. MM-UAV Posición por ejes del sistema MM-UAV para la trayectoria simulada 1. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador.

4.3 Sintonización del sistema MM-AUV por el algoritmo de la búsqueda del cuco.

Los pájaros cucos tienen un patrón de reproducción peculiar. Ponen sus huevos en nidos de otros para ser criados por otras aves. Este comportamiento ha inspirado la realización del algoritmo de búsqueda del cuco, que, junto con el vuelo de Lévy [95,96], un tipo de vuelo al azar en el que los incrementos se distribuyen según una distribución de probabilidad de cola pesada, el algoritmo de búsqueda se mostró eficiente al utilizar este tipo de vuelo [95,96].

Para usar CS como sintonizador, consideramos los siguientes supuestos:

- El número de nidos es fijo e igual al número de huevos para cada iteración.
- Un huevo en un nido representa una solución; solo se permitirá un huevo por nido para este problema.
- Se pasarán los huevos con la mejor aptitud a las próximas generaciones.
- La idoneidad de este problema será más notable cuanto más se acerque a la respuesta al escalón deseada.
- Cada huevo tiene una posibilidad absoluta de ser descubierto. Si se descubre, se descarta esa solución.
- Si un huevo es descubierto se genera nuevas soluciones a través de un vuelo de Levy en torno a las mejores soluciones.

Bajo estos supuestos, en la Figura 29, se puede ver el diagrama de flujo del algoritmo CS aplicado.

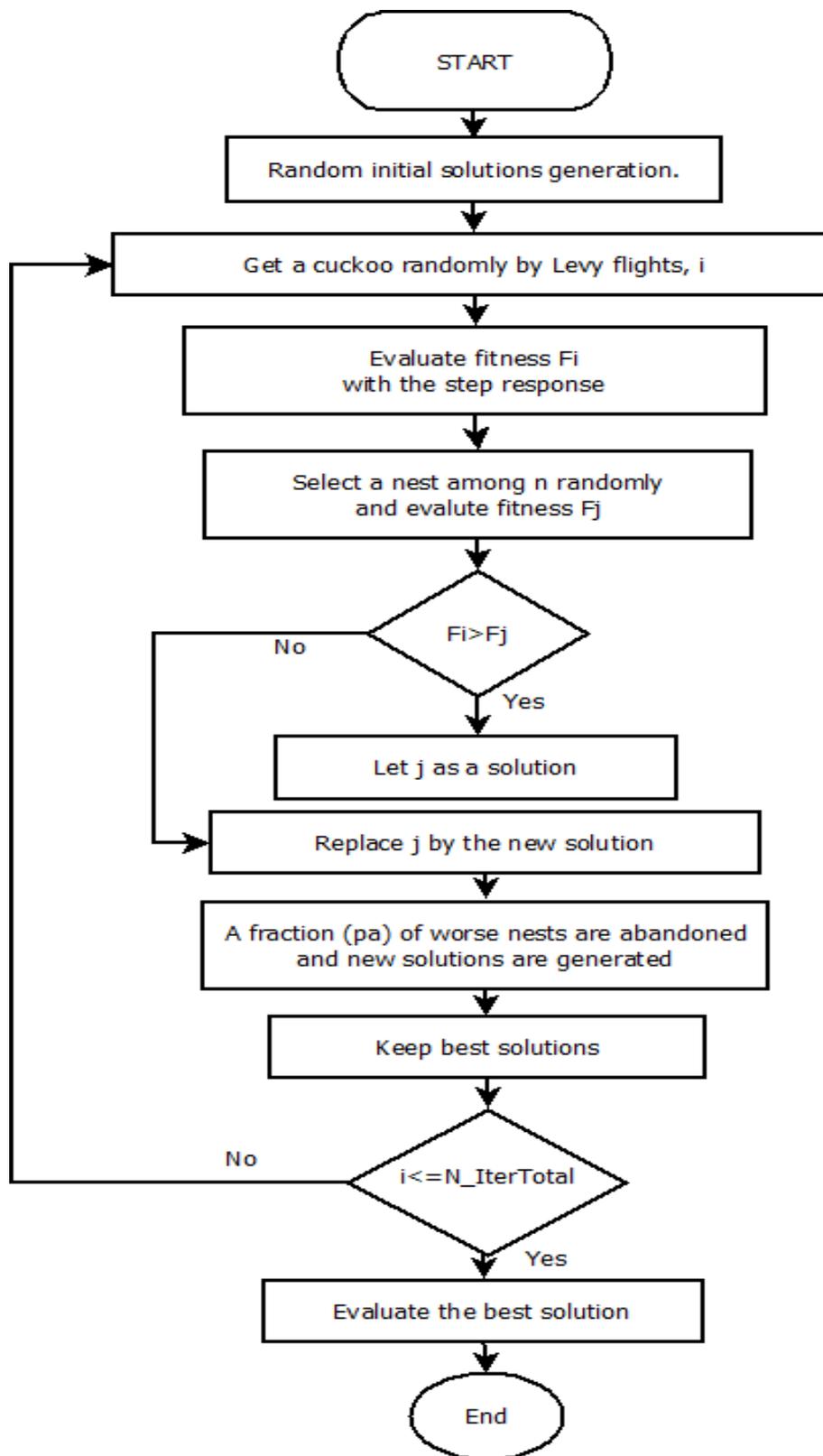


Figura 29. Diagrama de flujo del algoritmo de búsqueda del cuco utilizado.

Los parámetros del algoritmo de búsqueda del cuco determinan su eficiencia de búsqueda; los parámetros pueden obtenerse observando los resultados de múltiples simulaciones [95,96]. El parámetro de probabilidad de descubrimiento (P_a) se ha probado en varias aplicaciones, mostrando un buen rendimiento con un valor fijo del 25% [95,96]. La aptitud para este trabajo se evaluará con la distancia euclidiana, considerando un M_p máximo del 5% y un t_s máximo de 0,8 segundos. Los parámetros del algoritmo se pueden ver en la Tabla 4.

| Parámetros | Valor | Descripción |
|--|--|---|
| Número de nidos | 100 | Número fijo de la población. |
| Huevo | $[\alpha_i \alpha_j \alpha_k \alpha_l]$ | Vector con cuatro ganancias α aleatorias. |
| Iteraciones totales | 200 | El número de iteraciones en las que busco el algoritmo. |
| Tasa de descubrimiento de huevos alienígenas | 25% | Probabilidad de ser descubiertos los huevos alienígenas. |
| Aptitud | $\sqrt{(M_p - M_{pdi})^2 + (t_s - t_s)^2}$ | La aptitud evaluada con la distancia euclidiana es el tiempo de establecimiento $<0,5$ s y M_p es el sobreimpulso máximo $<5\%$. |
| Rango | [0 300] | Valores máximos y mínimos en los que se buscan los valores de cada α . |

Tabla 4. parámetros utilizados para el algoritmo de búsqueda del cuco.

Para observar el desempeño del controlador con la mejor combinación de alfas, se simula en ambiente Matlab-Simulink. Para la simulación utilizada, el modelo dinámico mostrado en las Ecuación (142). El modelo está controlado, con la ley de Backstepping que se muestra en la Ecuaciones (151) a (156) y ajustado por CS. Las ganancias obtenidas se muestran en la Tabla 5.

| Ganancia | Resultado | Ganancia | Resultado | Ganancia | Resultado | Ganancia | Resultado |
|------------|-----------|------------|-----------|---------------|-----------|---------------|-----------|
| α_1 | 45.6117 | α_5 | 4.7576 | α_9 | 20.7755 | α_{13} | 4.2522 |
| α_2 | 53.1245 | α_6 | 200 | α_{10} | 24.8929 | α_{14} | 5.3746 |
| α_3 | 1.69 | α_7 | 200 | α_{11} | 143.9536 | α_{15} | 135.6401 |
| α_4 | 57.51 | α_8 | 5.2648 | α_{12} | 5.4772 | α_{16} | 145.65 |

Tabla 5. Ganancias obtenidas por el algoritmo de la búsqueda del cuco.

Nuevamente se utilizan los parámetros físicos mostrados en la Tabla 3. En el proceso de ajuste, el sistema se divide en cuatro subsistemas, sometidos a una entrada escalón. El subsistema del manipulador tiene las ganancias asociadas ($\alpha_{13}, \alpha_{14}, \alpha_{15}, \alpha_{16}$) y su respuesta a la entrada escalón es mostrada en la Figura 30d. Los tres subsistemas restantes pertenecen al UAV. En cada subsistema UAV, el eje principal se asocia al ángulo que lo afecta directamente, es decir: el eje X está asociado al ángulo θ , ya que un giro en este ángulo provoca un movimiento en el eje X. Por lo tanto, las cuatro ganancias relacionadas con este subsistema se sintonizarán juntas ($\alpha_3, \alpha_4, \alpha_9, \alpha_{10}$), Figura 30a; se encuentra la misma dependencia con el subsistema Y- ϕ ($\alpha_1, \alpha_2, \alpha_{11}, \alpha_{12}$), Figura 30b. Finalmente se agrupo el ángulo ψ con el eje Z ($\alpha_5, \alpha_6, \alpha_7, \alpha_8$), Figura 30c. La Tabla 6 muestra la respuesta numérica al escalón de cada subsistema.

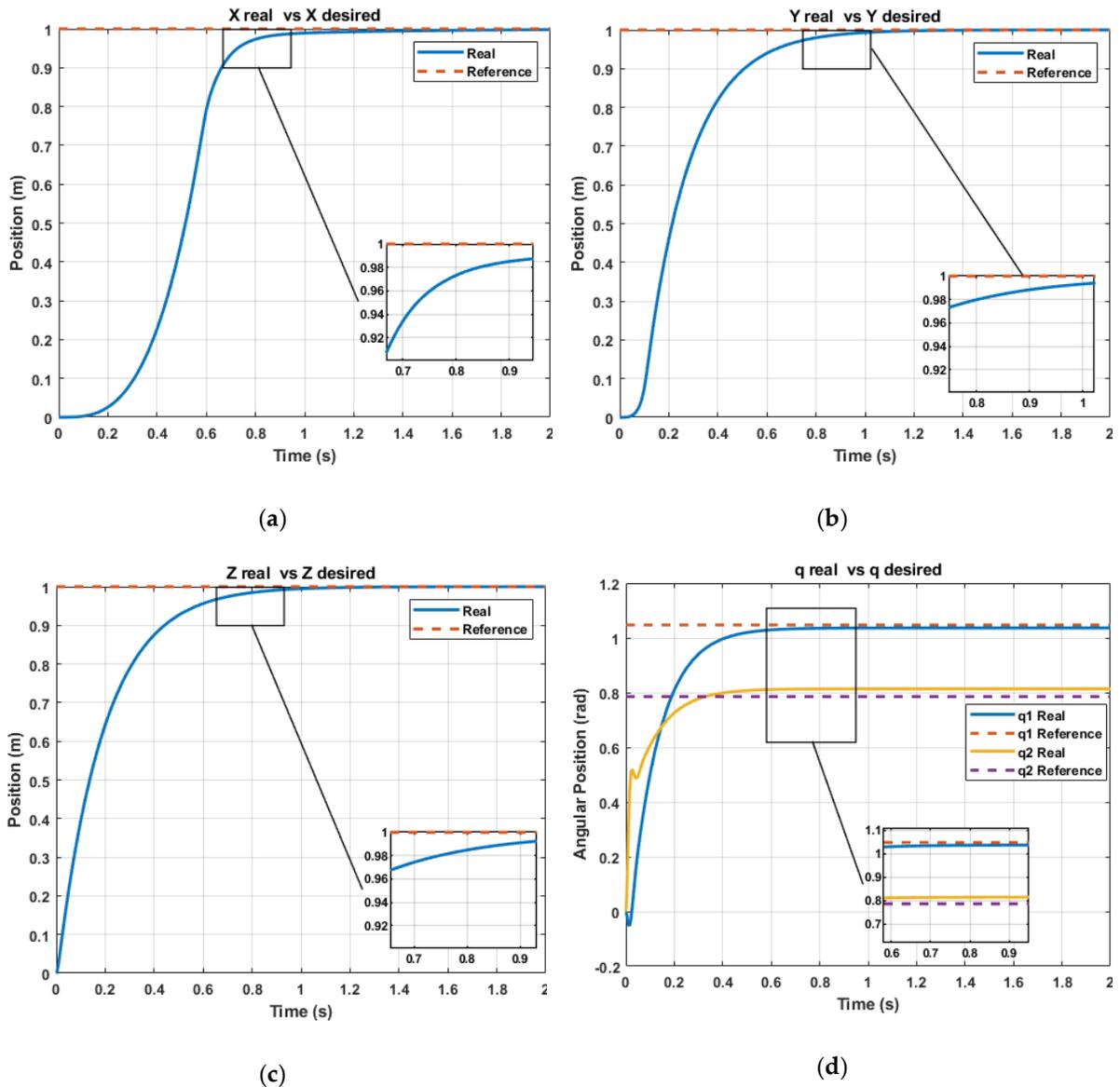


Figura 30. Respuesta al escalón con las ganancias obtenidas por el algoritmo de la búsqueda del cuco. (a) Respuesta en el eje X; (b) Respuesta en el eje Y; (c) Respuesta en el eje Z; (d) Respuesta en el manipulador.

| Axis | T _s | M _p | Reference |
|------|----------------|-------------------|-----------|
| X | Tiempo:0.7232 | Porcentaje: 0 | 1 |
| | Valor:0.9481 | Max Valor: 0.998 | |
| Y | Tiempo: 0.6336 | Porcentaje: 0 | 1 |
| | Valor: 0.9499 | Max Valor: 0.9999 | |
| Z | Tiempo: 0.5683 | Porcentaje: 0 | 1 |
| | Valor: 0.9486 | Max Valor: 0.9999 | |
| q1 | Tiempo: 0.4634 | Porcentaje: 0 | $\pi/3$ |
| | Valor: 1.0133 | Max Valor: 1.036 | |
| q2 | Tiempo: 0.3802 | Porcentaje: 3.6 | $\pi/4$ |
| | Valor: 0.7958 | Max Valor: 0.8138 | |

Tabla 6. Resultados numéricos para la respuesta al escalón.

Considerando la Tabla 6 y la Figura 30, se muestra que cumplió con las condiciones de diseño del controlador sintonizado con las ganancias de CS. Para la validación del rendimiento del controlador, el sistema se probó con dos tareas; la segunda tarea contiene perturbaciones generadas con números aleatorios para emular las ráfagas de aire que podría encontrar el sistema. Las ecuaciones (187) y (188) describen la ruta deseada de ambas tareas.

$$\begin{aligned} \text{Tarea 1: } x_r &= (4 * \cos(0.25 * t)) - 4; y_r = 4 * \cos(0.25 * t); z_r = \frac{t}{15}; \psi_r = \frac{\pi}{3}; q_1 \\ &= l_2 * \cos\left(\frac{t}{3}\right) + l_1 * \cos\left(\frac{\pi i}{4}\right); q_2 = l_2 * \sin\left(\frac{t}{3}\right) + l_1 * \sin\left(\frac{\pi i}{4}\right); \end{aligned} \quad (187)$$

$$\begin{aligned} \text{Tarea 2: } x_r = y_r = z_r = \psi_r &= \text{step}\left(\frac{1}{s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 5s + 1}\right); q_1 \\ &= \left(\frac{4}{5} l_2\right) * \cos\left(\frac{t}{3}\right) + l_1 * \cos\left(\frac{\pi i}{4}\right); q_2 = \left(\frac{4}{5} l_2\right) * \sin\left(\frac{t}{3}\right) + l_1 * \sin\left(\frac{\pi i}{4}\right); \end{aligned} \quad (188)$$

La tarea uno del UAV es una trayectoria helicoidal, Figura 31a, mientras que la tarea 2 muestra la respuesta al paso en cada eje de una función de transferencia de orden seis aplicada en diferentes momentos para cada eje. Para ambos caminos, el manipulador dibujará un círculo Figura 31b; esto nos permitirá evaluar el control de seguimiento de la vía cuando el manipulador está en constante movimiento.

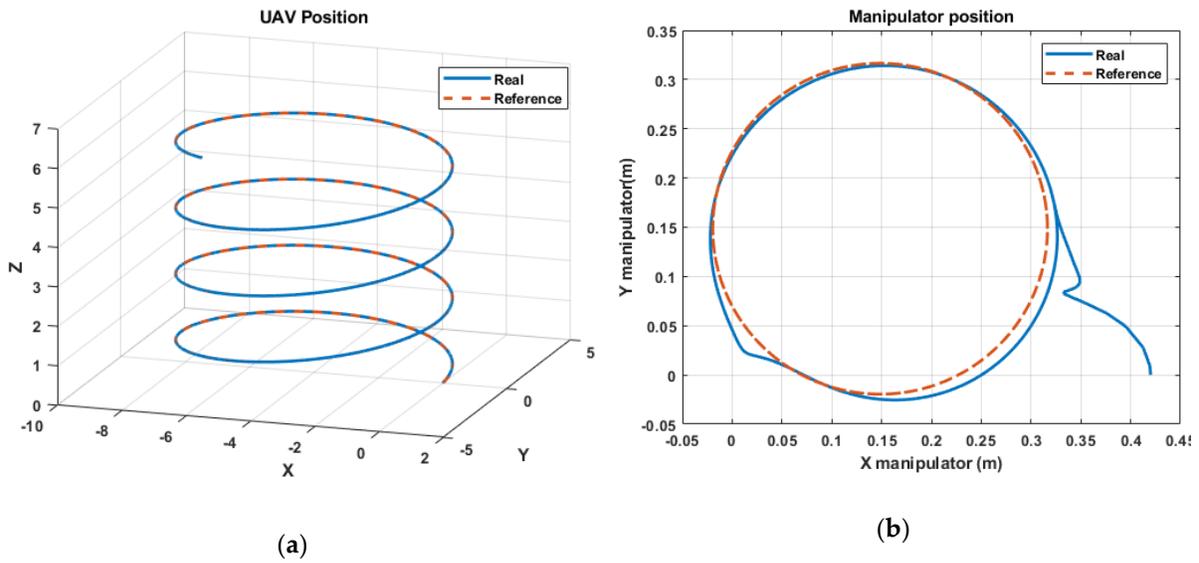
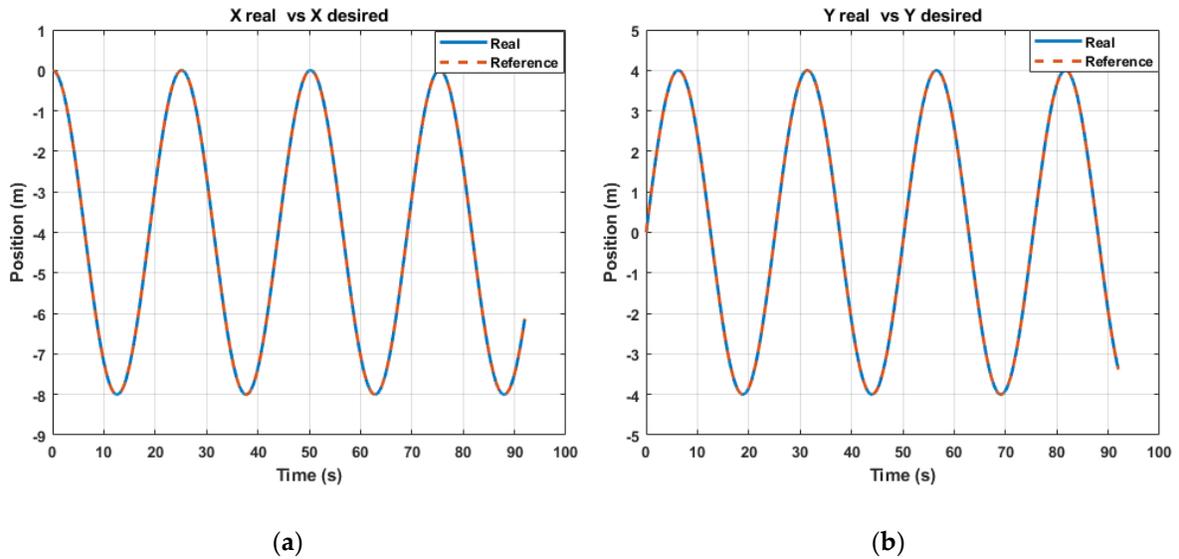


Figura 31. Posición cartesiana del MM-UAV en el seguimiento de la trayectoria para la tarea 1. (a) Posición del UAV; (b) Posición del manipulador.

Los errores de seguimiento en cada eje del UAV y en ambos ángulos del manipulador pueden verse en la Figura 32.



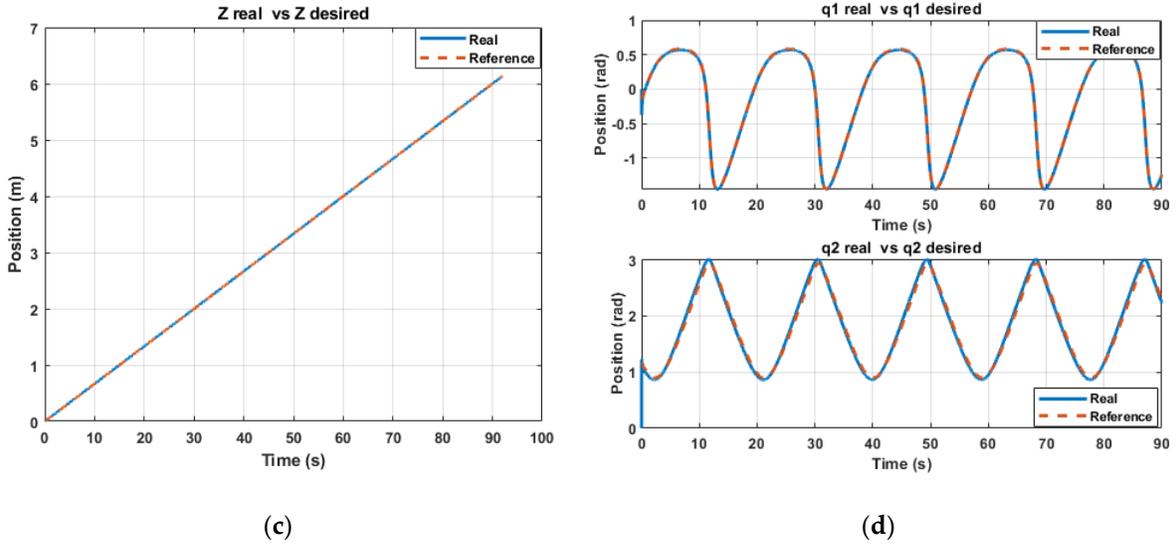


Figura 32. Seguimiento de la trayectoria de la tarea 1 en cada subsistema. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador.

La segunda trayectoria tiene una perturbación para simular los efectos del viento en un vuelo real; esta perturbación aparece en el segundo 10 de la simulación; tienen una magnitud de 10 m/s y una duración de 2 segundos. La Figura 8 muestra un resultado de la trayectoria de seguimiento para la tarea 2.

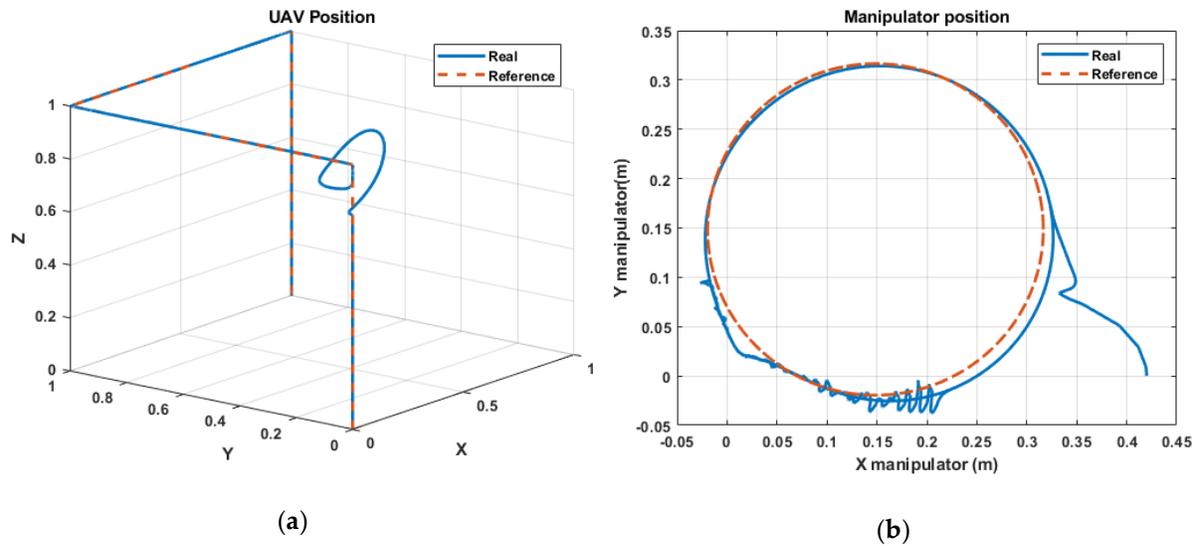


Figura 33. Posición cartesiana del MM-UAV en el seguimiento de la trayectoria con perturbaciones para la tarea 2. (a) Posición del UAV; (b) Posición del manipulador.

La Figura 34 muestra los errores observados en esta tarea además se observa la aparición y efecto de la perturbación.

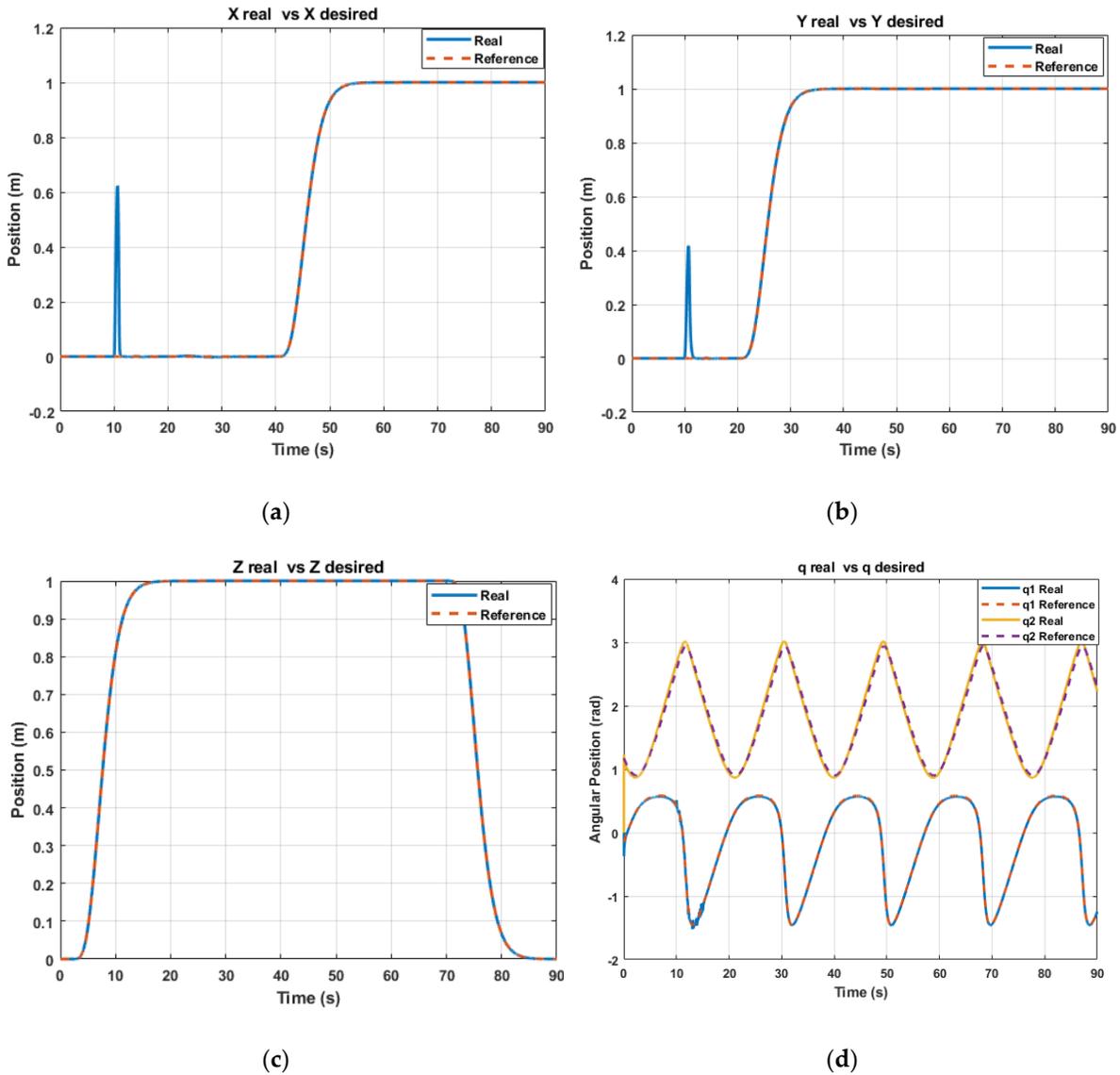


Figura 34. Seguimiento de la trayectoria de la tarea 2 en cada subsistema. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador.

Para fines de comparación, el MM-UAV está sujeto a un controlador PID para el seguimiento de la trayectoria. Este controlador utiliza la ley de control descrita en las ecuaciones (151) a la (156). La respuesta al escalón de cada eje se muestra gráficamente en la Figura 35, y los valores numéricos se muestran en la Tabla 7.

| Eje | PID | Backstepping | Referencia | Parámetros de diseño |
|-----|------------------------|------------------------|------------|-----------------------|
| X | ts: 0.634 s Mp: 24% | ts: 0.7232 s Mp: 0% | 1 | Ts < 0.8 s Mp < 5% |
| Y | ts: 1.384 Mp: 21% | ts: 0.6336 Mp: 0% | 1 | |
| Z | ts: 0.483 Mp: 4% | ts: 0.5683 Mp: 0% | 1 | |

| | | | |
|-----------|-----------|------------|---------|
| q1 | ts: 0.901 | ts: 0.4634 | $\pi/3$ |
| | Mp: 1% | Mp: 0% | |
| q2 | ts: 0.925 | ts: 0.3802 | $\pi/4$ |
| | Mp: 1% | Mp: 3.6% | |

Tabla 7. Comparación numérica PID contra Backstepping.

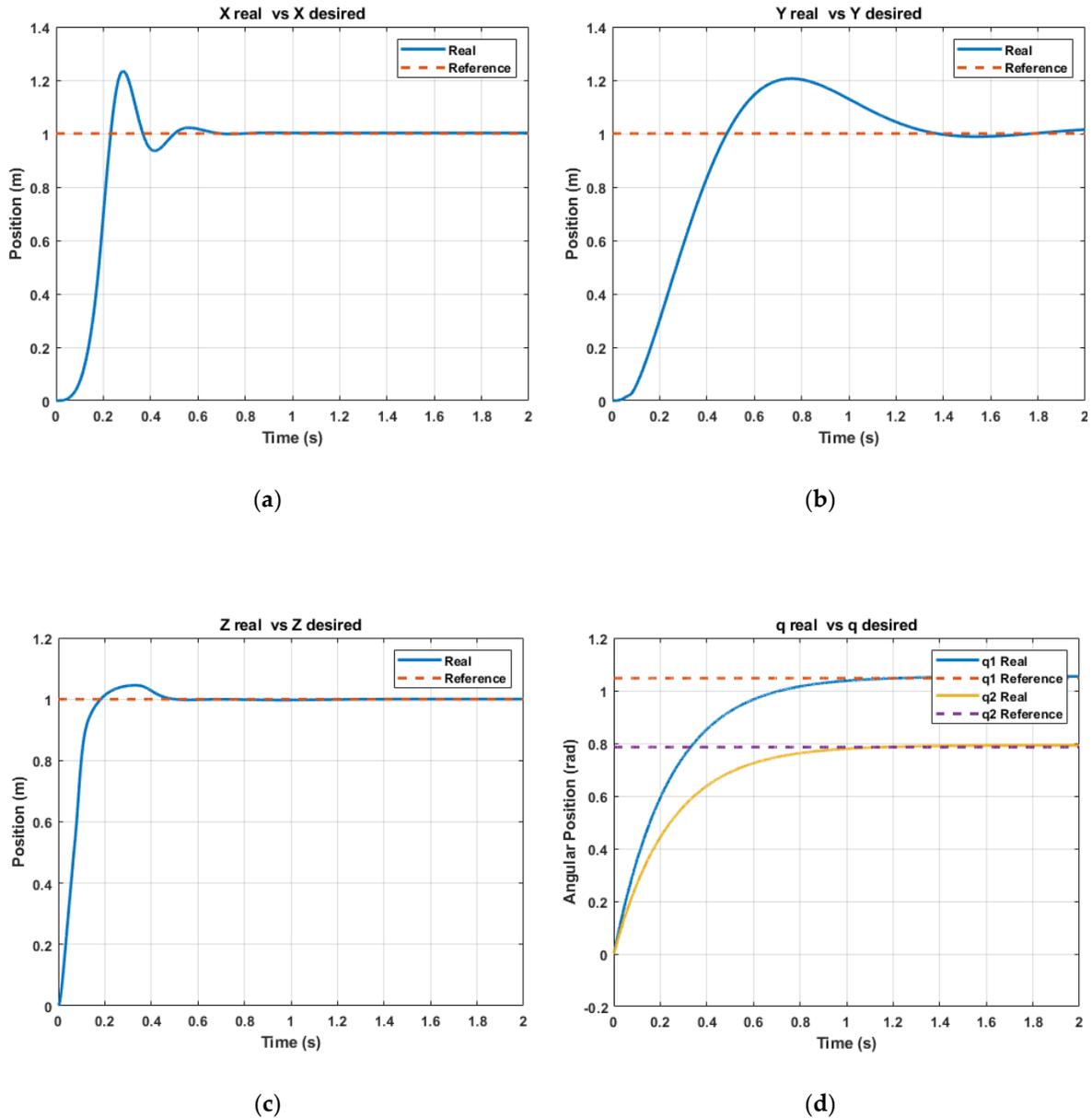
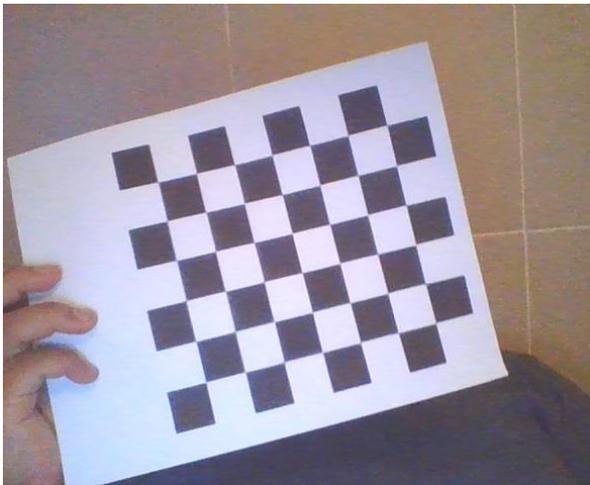


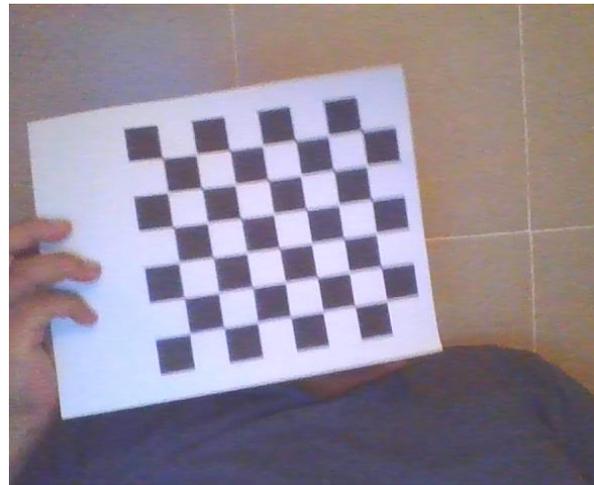
Figura 35. Respuesta de paso del controlador PID. (a) Respuesta en el eje X; (b) Respuesta en el eje Y; (c) Respuesta en el eje Z; (d) Respuesta en el manipulador.

4.4 Resultados de simulación para el sistema de retroalimentación visual.

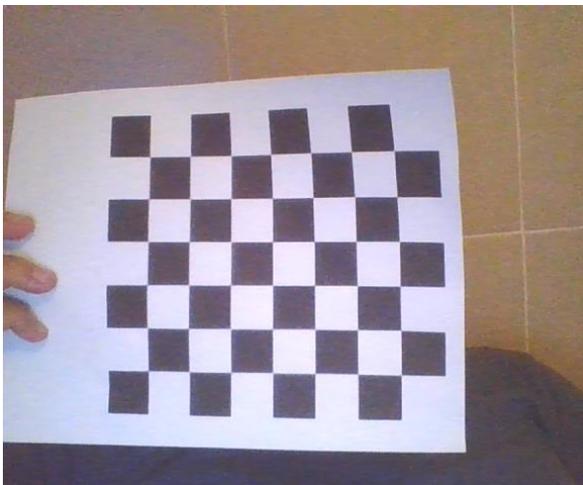
Para la simulación del sistema de retroalimentación visual se comenzó por el sistema de calibración de la cámara para obtener la distancia a la que se encuentra la cámara del sistema. Se utilizó el OpenCV para este proceso. El primer paso es utilizar un patrón impreso conocido en distintas formas, para este proceso se calibró con 20 imágenes para cada subconjunto de distancia, es decir 20 para el rango de 0.5 a 1.5 metros, 20 para la distancia de 1.5 a 2.5 metros y 20 imágenes para una distancia de 2.5 a 3.5 metros. Algunas de las imágenes utilizadas para el proceso de calibración se pueden observar en la Figura 36.



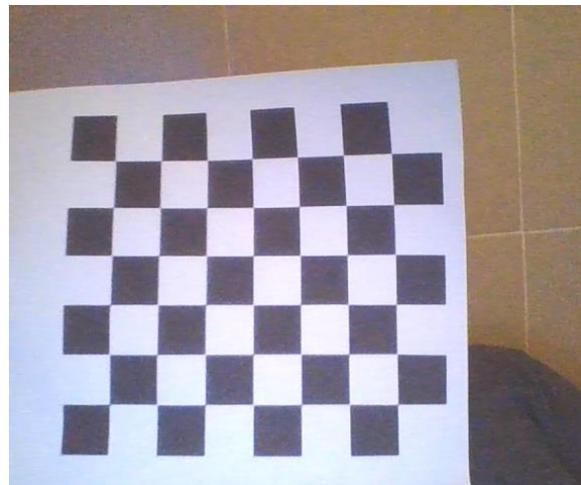
(a)



(b)



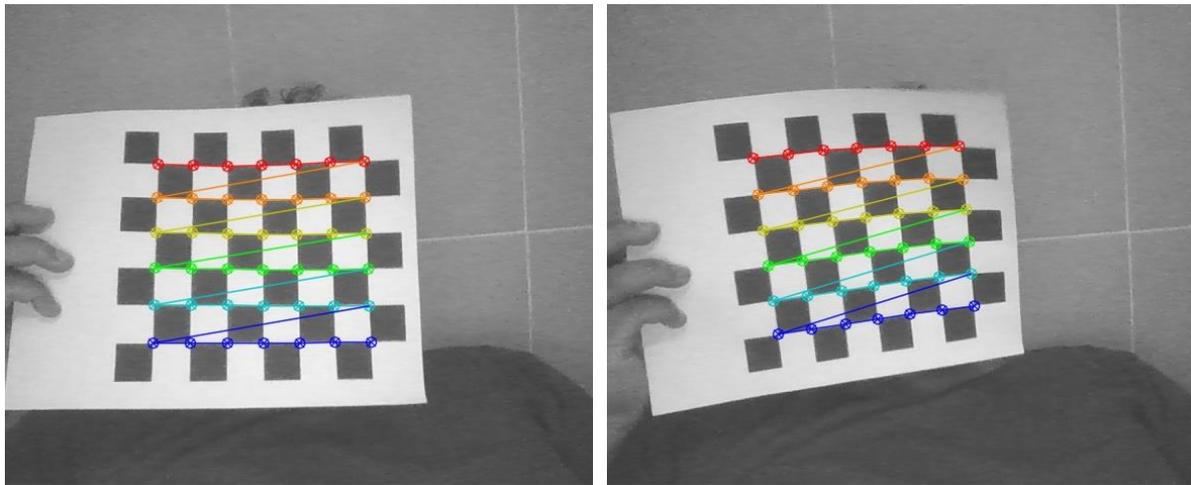
(c)



(d)

Figura 36. Imágenes utilizadas en el proceso de calibración.

Posteriormente se realiza el proceso de detección de esquinas tal y como se muestra en la Figura 37.



(a) (b)
Figura 37. Calibración de la cámara mediante el uso de un patrón conocido

Finalmente se estiman los parámetros de la cámara para que coincidan con la estimación del tamaño del patrón, tal y como se muestra en la Figura 38.

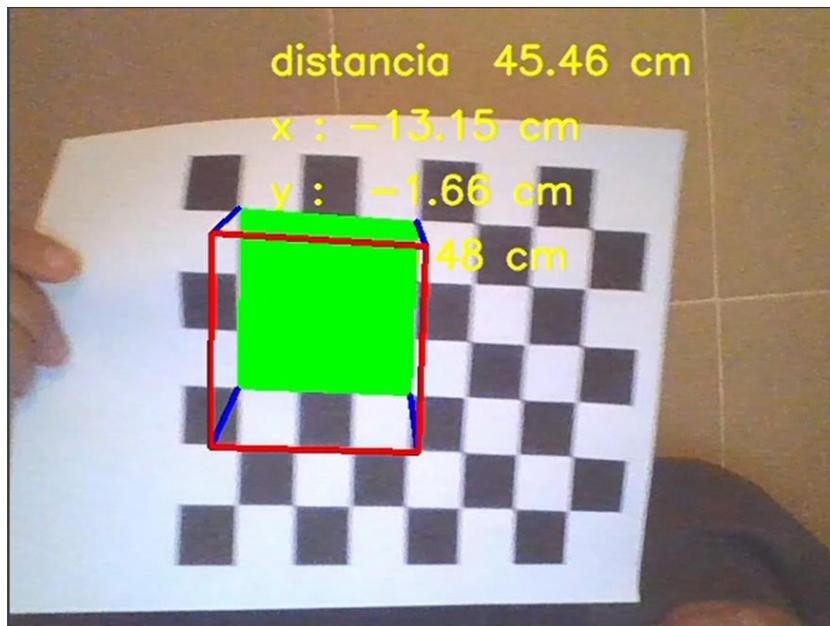


Figura 38. Cálculo de la distancia al patrón mediante la cámara calibrada.

Una vez obtenida la distancia mediante la cámara se debe probar en el sistema MM-UAV de manera segura. Para ello se creó un entorno virtual en VRLM con condiciones de operación similares a las de trabajo tal y como se muestra en la Figura 39.

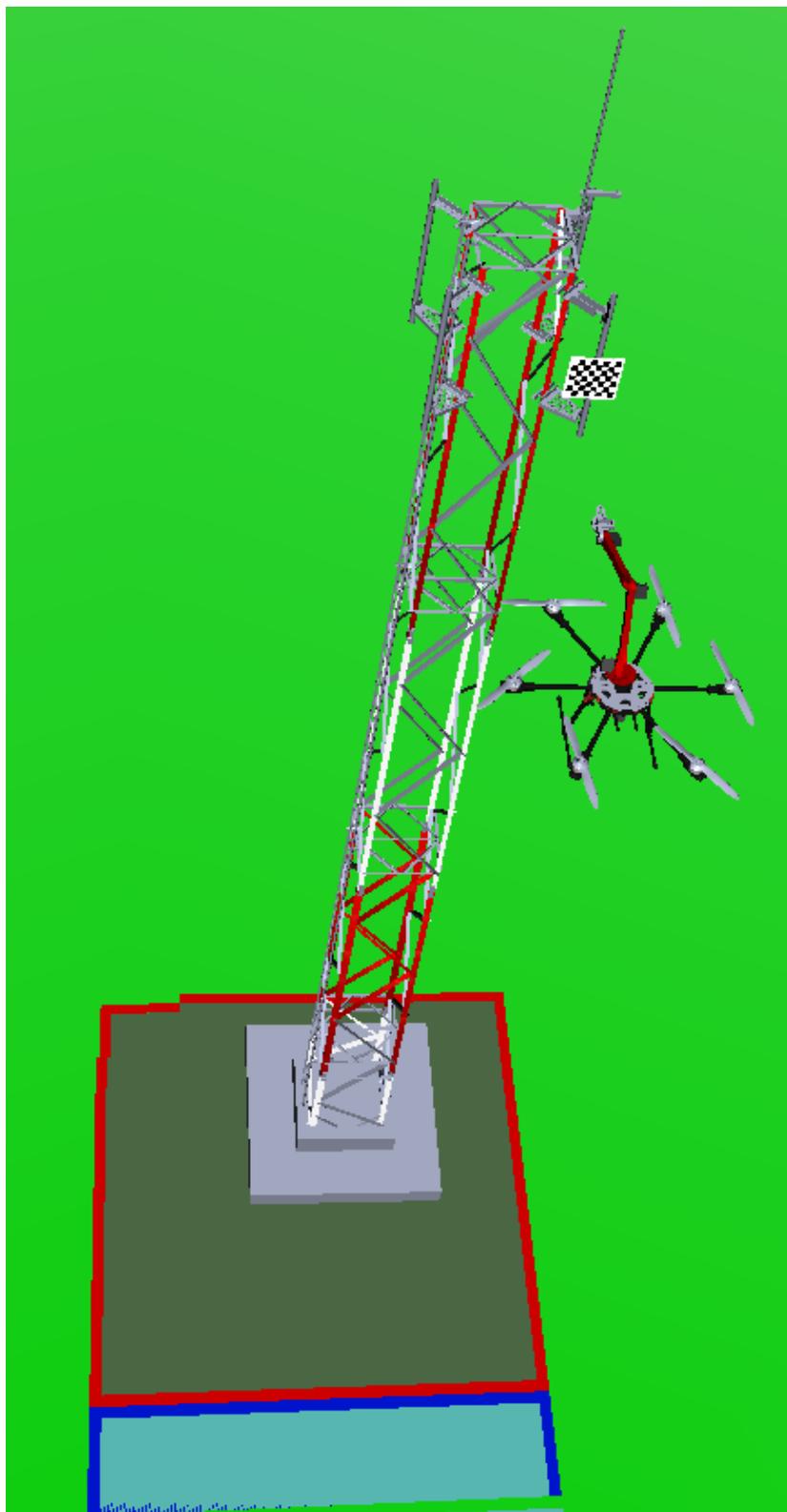


Figura 39. Simulación virtual de la aproximación visual para la manipulación aérea.

En este proceso la dinámica del sistema MM-UAV es determinada por la simulación de Matlab y la cámara en el dron es representada por un point of view solidario al modelo de la cámara del dron. El sistema recrea el acercamiento al patrón para identificar la distancia tal y como se muestra en la Figura 40.

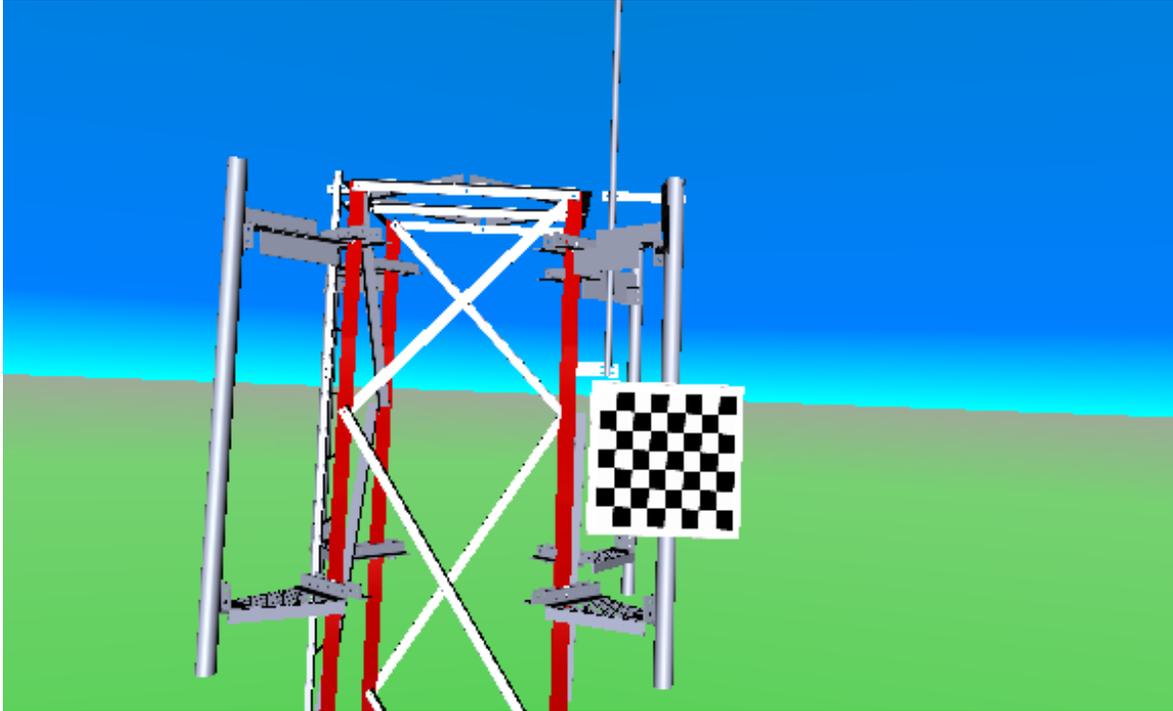


Figura 40. Pont of View solidario a la cámara del UAV.

4.5 Resultados experimentales del control para el sistema MM-UAV.

Esta subsección presenta los resultados de pruebas experimentales que validan la eficacia del controlador Backstepping sintonizado por el algoritmo CS en un entorno real. El hardware utilizado fue un hexacóptero y un brazo de diseño propio. Para el control se utilizó un Pixhawk 4, un autopiloto con fines comerciales y de desarrollo basado en un procesador STM32F765 Arm® Cortex®-M7 de 32 bits a 216 MHz con 2 MB de memoria y 512 KB de RAM. Este piloto automático se completa con los siguientes sensores a bordo: la unidad de medición inercial de 6 ejes ICM-20689 incluye ADC de 16 bits en el chip, filtros digitales programables, un sensor de temperatura integrado e interrupciones programables para la retroalimentación de la orientación y un GPS: GPS/GLONASS U-Blox Neo-M8N para la retroalimentación de posición. El par en cada eslabón se estima a partir del conocimiento de los parámetros dinámicos del motor, midiendo la corriente con un sensor de efecto Hall, y midiendo la velocidad con un codificador de cuadratura absoluto.

Para la implementación del algoritmo Backstepping, se requiere programar las ecuaciones (151) a la (156). Estas ecuaciones requieren parámetros fijos como la masa del UAV y del manipulador. Sin embargo, también implica datos que pueden ser actualizados en cada iteración, como las posiciones y velocidades obtenidas del GPS y la IMU para las variables lineales y angulares, respectivamente. Debido a la precisión del hardware para la estimación de la posición, se utilizó un filtro Chebyshev digital de tipo I de orden 2. Las aplicaciones típicas de transporte aéreo requieren manipular objetos mientras se realiza un vuelo estático, por lo que, para la prueba experimental, se eligió una trayectoria que permite que el manipulador se mueva en una

etapa segura de vuelo. Esta referencia es la misma que la tarea 2 para el UAV; sin embargo, el manipulador tiene dos entradas de tipo escalón como referencias $q_1 = \pi/3$, $q_2 = \pi/4$. La Figura 41 muestra el sistema físico acoplado en control de orientación y posición.

$$x_r = y_r = z_r = \psi_r = \text{step}\left(\frac{1}{s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 5s + 1}\right); q_1 = \frac{\pi}{3}; q_2 = \frac{\pi}{4}; \quad (189)$$

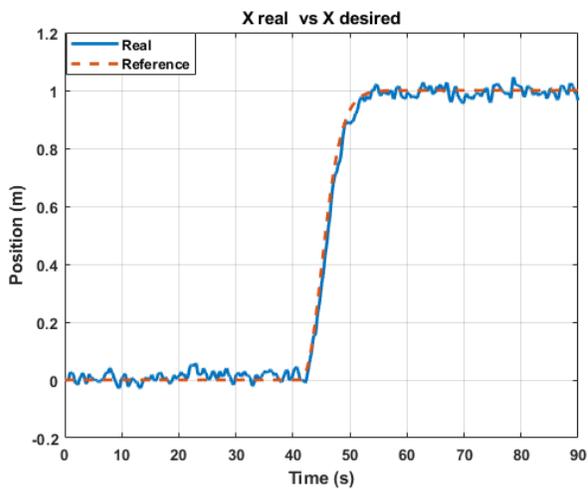
Esta referencia es la misma que la trayectoria 2 del UAV; sin embargo, el manipulador tiene dos entradas de tipo escalón como referencias debido a las restricciones mecánicas. La Figura 41 muestra el sistema físico acoplado en la trayectoria de seguimiento.



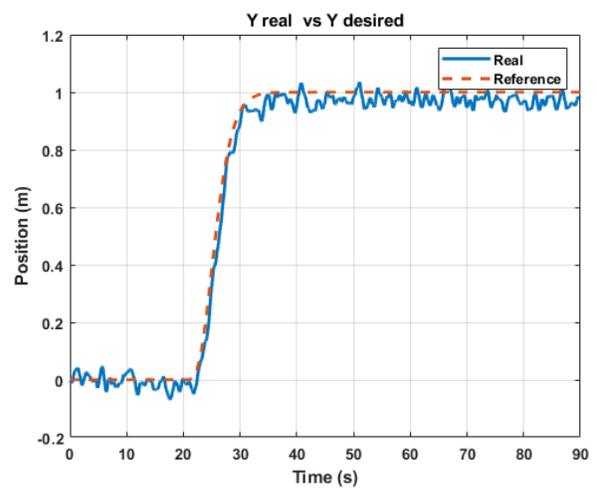


Figura 41. Sistema MM-UAV en control de posición y orientación.

La Figura 42 muestra las posiciones del UAV y del manipulador, y los resultados del seguimiento en cada subsistema. Debido a las limitaciones de hardware y de precisión, existe incertidumbre respecto al sistema de posición real. Sin embargo, el sistema sigue la trayectoria deseada con errores parcialmente asociados al sistema de telemetría.



(a)



(b)

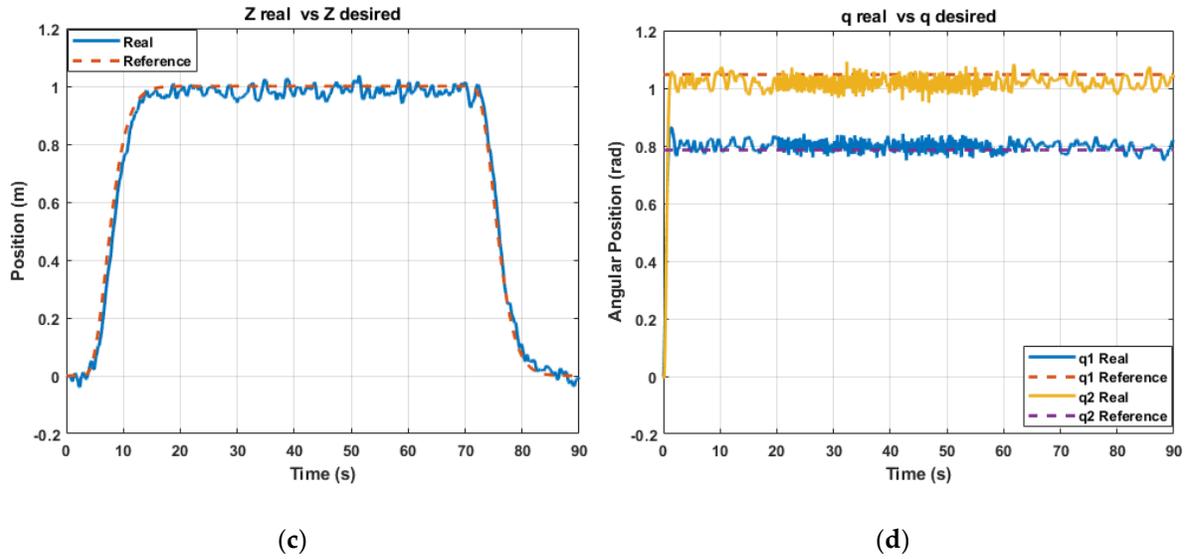
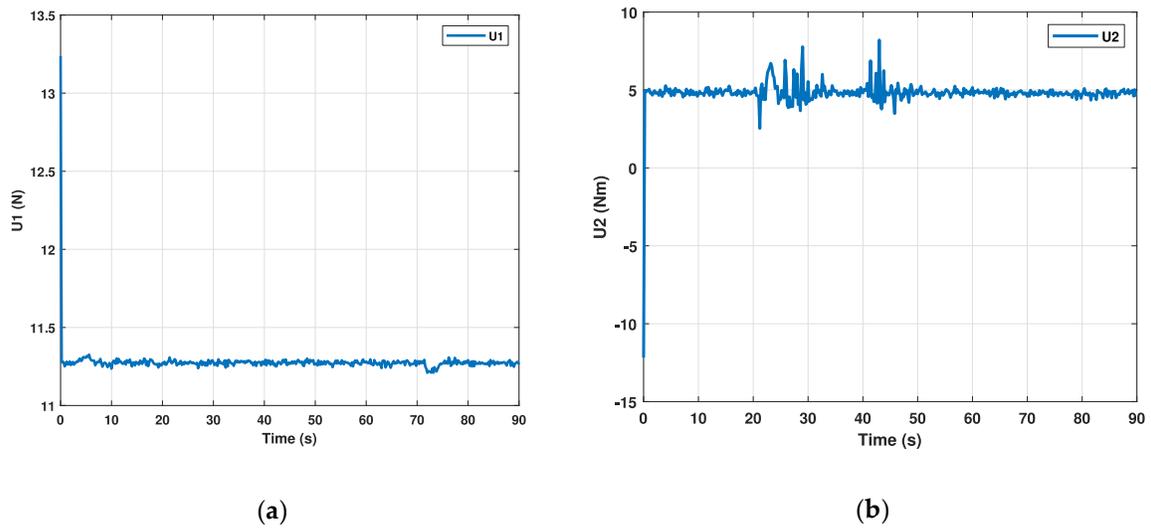


Figura 42. Seguimiento de la trayectoria en cada eje del sistema físico del MM-UAV. (a) Subsistema del eje X; (b) Subsistema del eje Y; (c) Subsistema del eje Z; (d) Subsistema del manipulador.

Finalmente, la Figura 43 muestra las señales de control del sistema real.



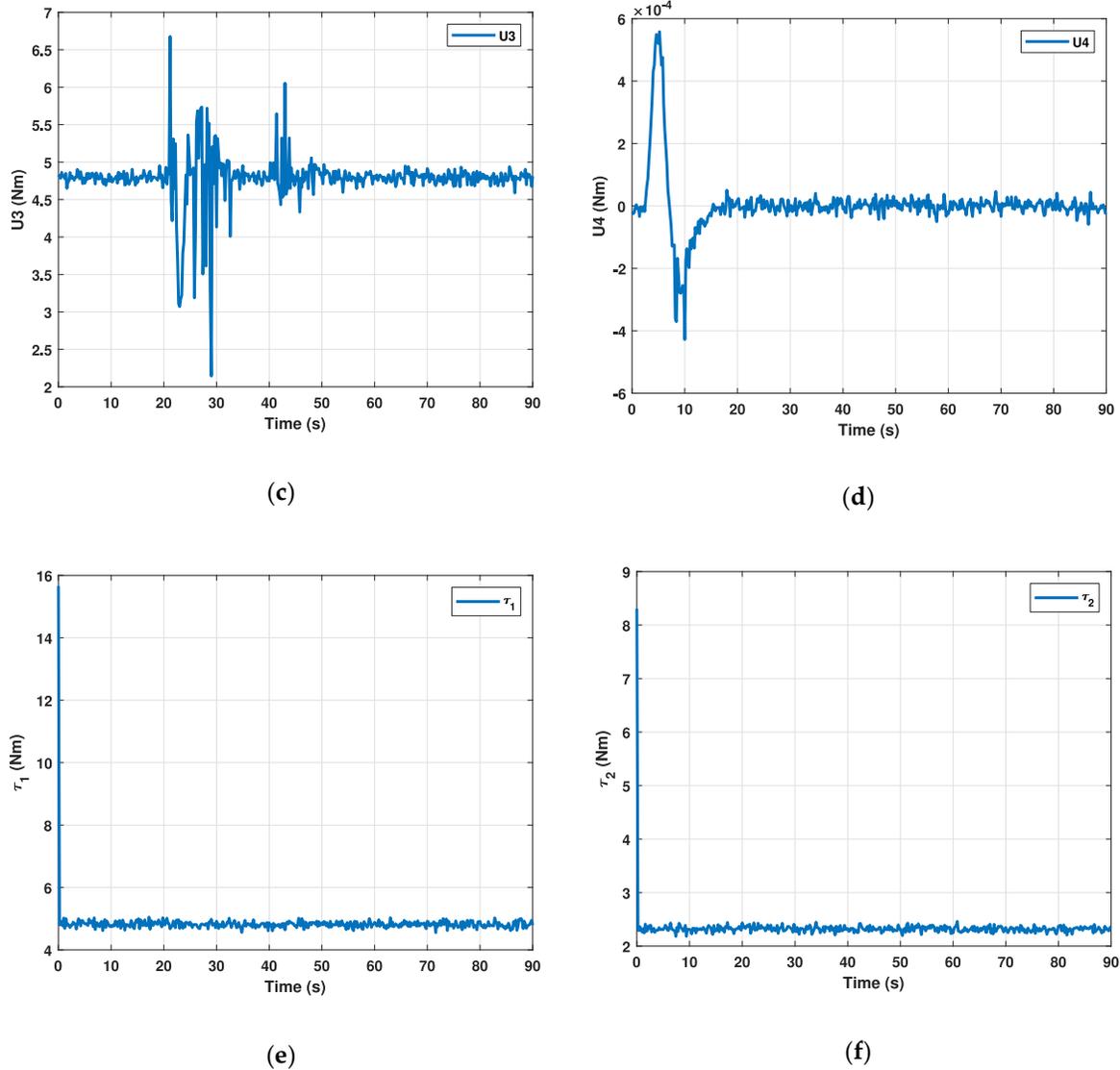


Figura 43. Señales de control del sistema físico MM-UAV. (a) Señal de control U1; (b) Señal de control U2; (c) Señal de control U3; (d) Señal de control U4; (e) Señal de control τ_1 ; (f) Señal de control τ_2 .

4.6 Resultados experimentales del sistema de navegación visual.

Para las pruebas experimentales de retroalimentación visual se utilizó una Jetson nano para procesar el sistema de visión artificial y la red neuronal que clasifica las imágenes en cercana mediana y lejana y el sistema de visión basado en la OpenCV con la cámara calibrada para medir distancia. El sistema solo se utiliza en la aproximación final. Para analizar en términos de viabilidad se debe comparar la señal de retroalimentación del GPS con el sistema de visión propuesto. Adicionalmente se requiere de una señal de posicionamiento real, es decir con la menor incertidumbre posible para poder medir la variabilidad del sistema propuesto y compararlo con la del GPS. Es por ello que se utilizó un sistema RTK GNSS North SmarTK para validar la medición del sistema de retroalimentación basada en visión.

La prueba es un movimiento en el plano X-Y tal y como se esperaría en un acercamiento final para manipulación. Se realizó en un campo despejado para que el GPS obtuviera línea directa y condiciones óptimas de conexión. Los resultados de las pruebas pueden observarse en la Figura 44.

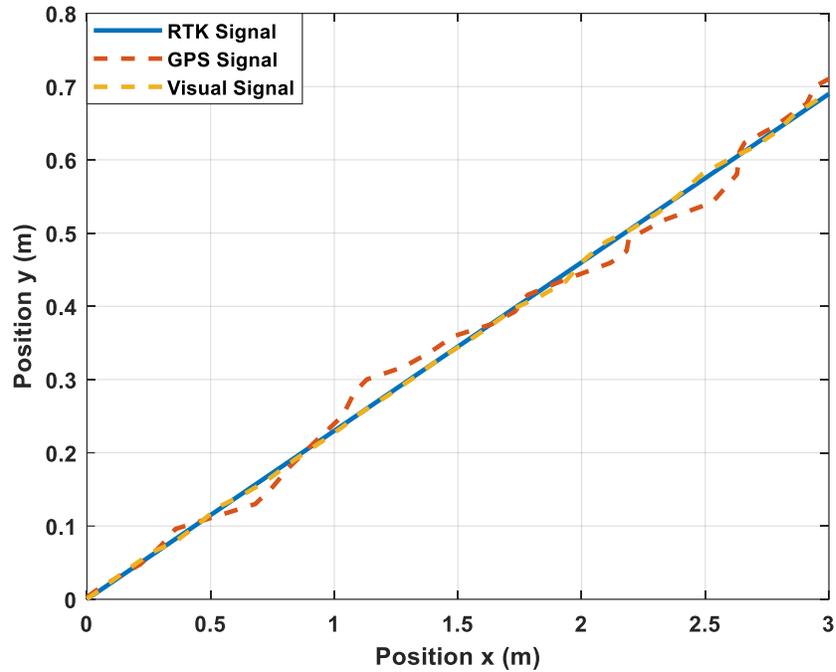


Figura 44. Señales de retroalimentación.

La Figura 45 muestra los errores de medición tanto para el GPS como para el sistema de retroalimentación.

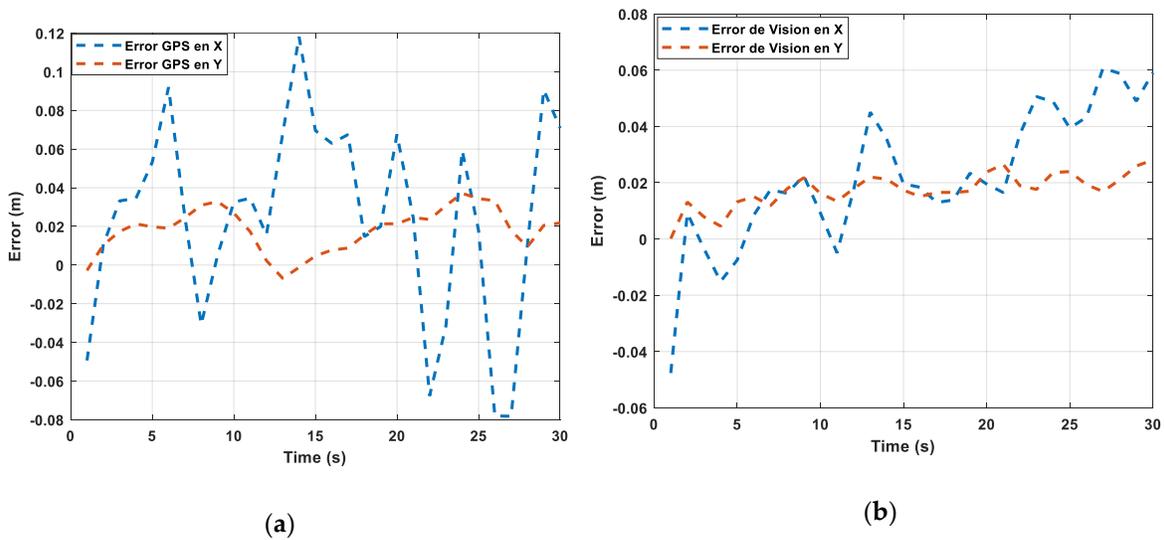


Figura 45. Errores de seguimiento para las señales de retroalimentación (a) Señal del GPS; (b) Señal del sistema de retroalimentación.

5

CONCLUSIONES

5. CONCLUSIONES

En este trabajo se desarrolló un sistema de manipulación aérea con un algoritmo de aproximación final basada en retroalimentación visual. Para ello se presenta el modelo del sistema MM-UAV y se utiliza para diseñar un controlador Backstepping sintonizado mediante el algoritmo Cuckoo Search. Los parámetros utilizados en el diseño son el tiempo de ajuste deseado y el sobre impulso deseado para realizar este ajuste. También se añade un sistema de calibración de una cámara para medir la distancia de la cámara a un patrón conocido y finalmente un sistema basado en redes neuronales para clasificar las imágenes adquiridas mediante la cámara y determinar que parámetros de cámara deben utilizarse.

Las pruebas iniciales no pueden ejecutarse en plataformas reales debido a que un fallo resulta catastrófico para el sistema se implementaron plataformas de simulación para ejecutar el banco de pruebas en simulación antes de pasar a las pruebas en sistemas reales. Los resultados indican que la retroalimentación visual con cámara y un control preciso del sistema permiten aumentar hasta 10 veces la presión de un sistema de control PID basado en retroalimentación de posición GPS.

La simulación del controlador sintonizado por Backstepping muestra que el sistema es capaz de rechazar las perturbaciones además de tener un mejor rendimiento en comparación con un controlador PID sintonizado por GA, indicando menos M_p en todos los ejes y cumpliendo con un t_s inferior a t_{sd} . El sistema PID permite el seguimiento de la Tarea 1 pero no cumple con los requisitos de diseño y no rechaza las perturbaciones simuladas para el estudio 2 del controlador Backstepping, que muestra mejores resultados ya que permite considerar el movimiento del manipulador y corregirlo. El PID no mostró resultados favorables porque es un controlador básico que no considera la dinámica del sistema.

El controlador muestra resultados favorables en las pruebas experimentales, siendo la incertidumbre en la medición de las variables de posición el principal problema. Se recomienda para futuros trabajos utilizar un hardware con mayores prestaciones para observar mejor el rendimiento del controlador. Sin embargo, se observa un correcto seguimiento de las referencias propuestas, un corto tiempo de asentamiento y un bajo sobreimpulso, lo que ayuda a los sistemas con energía muy limitada. Aunque se han desarrollado múltiples controladores Backstepping, no se ha estudiado la sintonía e implementación para un sistema MM-UAV de exteriores. La realización de pruebas experimentales con maniobras más complicadas requiere un hardware con mayor precisión o, en su defecto, técnicas de corrección en la retroalimentación.

Se requieren realizar más pruebas de ajuste y hacer más robusto la técnica de medición ante cambios de luz, también se sugiere implementar un sistema basado en redes neuronales exclusivamente. Evitando de este modo el patrón en las torres, esto permitirá aproximarse de manera más eficiente a diferentes torres. Sin embargo, requiere una base de datos extensa de fotografías las cuales puede ser difíciles de adquirir por el problema de acercamiento descrito en este trabajo. No obstante, la principal conclusión de este trabajo es que la manipulación aérea de

bajo coste con un sistema de aproximación final visual es viable en los términos presentados en esta tesis.

6. BIBLIOGRAFIA

1. Meng, X.; Meng, X.; He, Y.; Han, J.; Han, J. Survey on Aerial Manipulator: System, Modeling, and Control. *Robotica* 2020.
2. Maciel-Pearson, B.G.; Marchegiani, L.; Akçay, S.; Atapour-Abarghouei, A.; Garforth, J.; Breckon, T.P. Online deep reinforcement learning for autonomous UAV navigation and exploration of outdoor environments. *arXiv* 2019.
3. Gong, J.; Chang, T.H.; Shen, C.; Chen, X. Flight time minimization of UAV for data collection over wireless sensor networks. *IEEE J. Sel. Areas Commun.* **2018**, doi:10.1109/JSAC.2018.2864420.
4. Malehmir, A.; Dynesius, L.; Paulusson, K.; Paulusson, A.; Johansson, H.; Bastani, M.; Wedmark, M.; Marsden, P. The potential of rotary-wing UAV-based magnetic surveys for mineral exploration: A case study from central Sweden. *Lead. Edge* **2017**, doi:10.1190/tle36070552.1.
5. Carvajal-Ramírez, F.; da Silva, J.R.M.; Agüera-Vega, F.; Martínez-Carricondo, P.; Serrano, J.; Moral, F.J. Evaluation of fire severity indices based on pre- and post-fire multispectral imagery sensed from UAV. *Remote Sens.* **2019**, doi:10.3390/rs11090993.
6. Hamaza, S.; Georgilas, I.; Richardson, T. An adaptive-compliance manipulator for contact-based aerial applications. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM; 2018.
7. Reinoso, J.F.; Gonçalves, J.E.; Pereira, C.; Bleninger, T. Cartography for Civil Engineering Projects: Photogrammetry Supported by Unmanned Aerial Vehicles. *Iran. J. Sci. Technol. - Trans. Civ. Eng.* **2018**, doi:10.1007/s40996-017-0076-x.
8. Bangkui, F.; Yun, L.; Ruiyu, Z.; Qiqi, F. Review on the technological development and application of uav systems. *Chinese J. Electron.* **2020**, *29*, 199–207, doi:10.1049/cje.2019.12.006.
9. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomatics* 2014.
10. Akash, S.; Sridhar, S.S.; Kabi, B.; Chellammal, N. Developing a flight control strategy for a four-rotor under-actuated UAV. In Proceedings of the International Conference on “Emerging Trends in Robotics and Communication Technologies”, INTERACT-2010; 2010.
11. Chen, J.; Zhou, Y.; Lv, Q.; Deveerasetty, K.K.; Ugochi Dike, H. A review of autonomous obstacle avoidance technology for multi-rotor UAVs. In Proceedings of the 2018 IEEE International Conference on Information and Automation, ICIA 2018; 2018.
12. Thomas, J.; Loiano, G.; Sreenath, K.; Kumar, V. Toward image based visual servoing for

- aerial grasping and perching. In Proceedings of the Proceedings - IEEE International Conference on Robotics and Automation; 2014.
13. Thomas, J.; Loianno, G.; Daniilidis, K.; Kumar, V. Visual Servoing of Quadrotors for Perching by Hanging From Cylindrical Objects. *IEEE Robot. Autom. Lett.* **2016**, doi:10.1109/LRA.2015.2506001.
 14. Jiang, G.; Voyles, R.; Sebesta, K.; Greiner, H. Estimation and optimization of fully-actuated multirotor platform with nonparallel actuation mechanism. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems; 2017.
 15. Intwala, A.; Parikh, Y. A Review on Vertical Take Off and Landing (VTOL) Vehicles. *Int. J. Innov. Res. Adv. Eng.* **2015**.
 16. Beauchamp Báez, G.; Batista, R. Aplicación de Técnicas de Control Óptimo a una plataforma estacionaria cuatrimotor. *Rev. Científica Ing. Electrónica, Automática y Comun.* **2016**, doi:10.1234/rielac.v37i3.413.
 17. Palik, M.; Nagy, M. Brief history of UAV development. *Repüléstudományi Közlemények* **2019**, doi:10.32560/rk.2019.1.13.
 18. Mehra, R.K.; Boskovic, J.D.; Li, S.M. Autonomous formation flying of multiple UCAVs under communication failure. *Rec. - IEEE PLANS, Position Locat. Navig. Symp.* **2000**, doi:10.1109/plans.2000.838327.
 19. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* 2019.
 20. ZHANG, J.; XING, J. Cooperative task assignment of multi-UAV system. *Chinese J. Aeronaut.* **2020**, doi:10.1016/j.cja.2020.02.009.
 21. Iannace, G.; Ciaburro, G.; Trematerra, A. Fault diagnosis for UAV blades using artificial neural network. *Robotics* **2019**, doi:10.3390/robotics8030059.
 22. Eltayeb, A.; Rahmat, M.F.; Basri, M.A.M. Adaptive feedback linearization controller for stabilization of Quadrotor UAV. *Int. J. Integr. Eng.* **2020**, doi:10.30880/ijie.00.00.0000.00.0000.
 23. Zhang, J.; Gu, D.; Deng, C.; Wen, B. Robust and Adaptive Backstepping Control for Hexacopter UAVs. *IEEE Access* **2019**, doi:10.1109/ACCESS.2019.2951282.
 24. Wang, H.; Li, Z.; Xiong, H.; Nian, X. Robust H^∞ attitude tracking control of a quadrotor UAV on $SO(3)$ via variation-based linearization and interval matrix approach. *ISA Trans.* **2019**, doi:10.1016/j.isatra.2018.11.015.
 25. Razmi, H.; Afshinfar, S. Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV. *Aerosp. Sci. Technol.* **2019**, doi:10.1016/j.ast.2019.04.055.
 26. Xie, R.; Wang, X.; Wang, J.; Cao, Y.; Li, T. A probabilistic robust control method for a small UAV with bounded uncertainty. In Proceedings of the Chinese Control Conference, CCC;

- 2016.
27. Yogi, S.C.; Tripathi, V.K.; Kamath, A.K.; Behera, L. Real-time Trajectory Tracking of a Quadrotor using Adaptive Backstepping Controller and RNN based Uncertainty Observer. In Proceedings of the Proceedings of the International Joint Conference on Neural Networks; 2020.
 28. Noordin, A.; Basri, M.A.M.; Mohamed, Z.; Abidin, A.F.Z. Modelling and PSO fine-tuned PID control of quadrotor UAV. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2017**, doi:10.18517/ijaseit.7.4.3141.
 29. Najm, A.A.; Ibraheem, I.K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Eng. Sci. Technol. an Int. J.* **2019**, doi:10.1016/j.jestch.2019.02.005.
 30. Dong, J.; He, B. Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors (Switzerland)* **2019**, doi:10.3390/s19010024.
 31. Jatsun, S.; Emelyanova, O.; Leon, A.S.M.; Stykanyova, S. Control flight of a UAV type tri-copter with fuzzy logic controller. In Proceedings of the 11th International IEEE Scientific and Technical Conference "Dynamics of Systems, Mechanisms and Machines", Dynamics 2017 - Proceedings; 2017.
 32. Jurado, F.; Lopez, S. Continuous-Time Decentralized Neural Control of a Quadrotor UAV. In *Artificial Neural Networks for Engineering Applications*; 2019.
 33. Mo, H.; Farid, G. Nonlinear and Adaptive Intelligent Control Techniques for Quadrotor UAV – A Survey. *Asian J. Control* **2019**, doi:10.1002/asjc.1758.
 34. Yang, T.; Hu, Y.; Yuan, X.; Mathar, R. Genetic Algorithm based UAV Trajectory Design in Wireless Power Transfer Systems. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC; 2019.
 35. FRANCESCO SABATINO Quadrotor control: modeling, nonlinear control design, and simulation, 2015.
 36. ZHANG, G.; Taha, H. Adaptive back-stepping control applied on octocopter under recoil disturbance. *J. Eng. Sci. Mil. Technol.* **2017**, doi:10.21608/ejmtc.2017.401.1004.
 37. Kartal, Y.; Subbarao, K.; Gans, N.R.; Dogan, A.; Lewis, F. Distributed backstepping based control of multiple UAV formation flight subject to time delays. *IET Control Theory Appl.* **2020**, doi:10.1049/iet-cta.2019.1151.
 38. Xu, L.X.; Ma, H.J.; Guo, D.; Xie, A.H.; Song, D.L. Backstepping Sliding-Mode and Cascade Active Disturbance Rejection Control for a Quadrotor UAV. *IEEE/ASME Trans. Mechatronics* **2020**, doi:10.1109/TMECH.2020.2990582.
 39. Zhou, L.; Zhang, J.; Dou, J.; Wen, B. A fuzzy adaptive backstepping control based on mass observer for trajectory tracking of a quadrotor UAV. *Int. J. Adapt. Control Signal Process.* **2018**, doi:10.1002/acs.2937.
 40. Tran, T.T. Feedback linearization and backstepping: an equivalence in control design of strict-feedback form. *IMA J. Math. Control Inf.* **2020**, doi:10.1093/imamci/dnz024.

41. Zhu, Z.; Cao, S. Back-stepping sliding mode control method for quadrotor UAV with actuator failure. *J. Eng.* **2019**, doi:10.1049/joe.2019.1084.
42. Han, S.; Wang, H.; Tian, Y. Integral backstepping based computed torque control for a 6 DOF arm robot. In Proceedings of the Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017; 2017.
43. Van, M.; Mavrovouniotis, M.; Ge, S.S. An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *IEEE Trans. Syst. Man, Cybern. Syst.* **2019**, doi:10.1109/TSMC.2017.2782246.
44. Wongvanich, N.; Boksuwan, S.; Chesof, A. Simplified modelling and backstepping control of the long arm agricultural rover. *Adv. Differ. Equations* **2020**, doi:10.1186/s13662-020-03158-y.
45. Bayrak, A.; Tatlicioglu, E.; Zegeroglu, E. Backstepping control of electro-hydraulic arm. In Proceedings of the 2018 6th International Conference on Control Engineering and Information Technology, CEIT 2018; 2018.
46. Nguyen, T. Van; Thai, N.H.; Pham, H.T.; Phan, T.A.; Nguyen, L.; Le, H.X.; Nguyen, H.D. Adaptive Neural Network-Based Backstepping Sliding Mode Control Approach for Dual-Arm Robots. *J. Control. Autom. Electr. Syst.* **2019**, doi:10.1007/s40313-019-00472-z.
47. Tran, D.T.; Nguyen, M.N.; Ahn, K.K. RBF neural network based backstepping control for an electrohydraulic elastic manipulator. *Appl. Sci.* **2019**, doi:10.3390/app9112237.
48. Chang, W.; Tong, S.; Li, Y. Adaptive fuzzy backstepping output constraint control of flexible manipulator with actuator saturation. *Neural Comput. Appl.* **2017**, doi:10.1007/s00521-016-2425-2.
49. Guo, C.; Zhang, A.; Zhang, H.; Zhang, L. Adaptive backstepping control with online parameter estimator for a plug-and-play parallel converter system in a power switcher. *Energies* **2018**, doi:10.3390/en11123528.
50. El-Sousy, F.F.M.; Amin, M.; Mohammed, O.A. High-precision adaptive backstepping optimal control using RBFN for PMSM-driven linear motion stage. In Proceedings of the 2019 IEEE International Electric Machines and Drives Conference, IEMDC 2019; 2019.
51. Anjum, Z.; Guo, Y. Finite Time Fractional-order Adaptive Backstepping Fault Tolerant Control of Robotic Manipulator. *Int. J. Control. Autom. Syst.* **2021**, doi:10.1007/s12555-019-0648-6.
52. Bai, K.; Gong, X.; Chen, S.; Wang, Y.; Liu, Z. Sliding mode nonlinear disturbance observer-based adaptive back-stepping control of a humanoid robotic dual manipulator. *Robotica* **2018**, doi:10.1017/S026357471800067X.
53. Walid, M.; Slaheddine, N.; Mohamed, A.; Lamjed, B. Modelling, identification and control of a quadrotor UAV. In Proceedings of the 2018 15th International Multi-Conference on Systems, Signals and Devices, SSD 2018; 2018.
54. Wang, L.; Poksawat, P. Automatic tuning of hexacopter attitude control systems with

- experimental validation. In Proceedings of the 2017 21st International Conference on System Theory, Control and Computing, ICSTCC 2017; 2017.
55. Yang, S.; Wang, Z. Quad-Rotor UAV Control Method Based on PID Control Law. In Proceedings of the Proceedings - 2017 International Conference on Computer Network, Electronic and Automation, ICCNEA 2017; 2017.
 56. Rosales, C.; Soria, C.M.; Rossomando, F.G. Identification and adaptive PID Control of a hexacopter UAV based on neural networks. *Int. J. Adapt. Control Signal Process.* **2019**, doi:10.1002/acs.2955.
 57. Si, W.; She, H.; Wang, Z. Fuzzy PID controller for UAV tracking moving target. In Proceedings of the Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017; 2017.
 58. Kumar, R.; Dechering, M.; Pai, A.; Ottaway, A.; Radmanesh, M.; Kumar, M. Differential flatness based hybrid PID/LQR flight controller for complex trajectory tracking in quadcopter UAVs. In Proceedings of the Proceedings of the IEEE National Aerospace Electronics Conference, NAECON; 2017.
 59. Qiao, J.; Liu, Z.; Zhang, Y. Gain Scheduling based PID control approaches for path tracking and fault tolerant control of a Quad-rotor UAV. *Int. J. Mech. Eng. Robot. Res.* **2018**, doi:10.18178/ijmerr.7.4.401-408.
 60. Salamat, B.; Tonello, A.M. Adaptive Nonlinear PID Control for a Quadrotor UAV Using Particle Swarm Optimization. In Proceedings of the IEEE Aerospace Conference Proceedings; 2019.
 61. Pan, F.; Liu, L.; Xue, D. Optimal PID controller design with Kalman filter for Qball-X4 quad-rotor unmanned aerial vehicle. *Trans. Inst. Meas. Control* **2017**, doi:10.1177/0142331216656753.
 62. Alami, F.; Hussian, A.; Ajlouni, N. Design and implementation of a multi-stage PID controller for non-inertial referenced UAV. *Electrica* **2020**, doi:10.5152/ELECTRICA.2020.20004.
 63. Borase, R.P.; Maghade, D.K.; Sondkar, S.Y.; Pawar, S.N. A review of PID control, tuning methods and applications. *Int. J. Dyn. Control* 2020.
 64. Blevins, T.L. PID advances in industrial control. In Proceedings of the IFAC Proceedings Volumes (IFAC-PapersOnline); 2012.
 65. Ardeshiri, R.R.; Khooban, M.H.; Noshadi, A.; Vafamand, N.; Rakhshan, M. Robotic manipulator control based on an optimal fractional-order fuzzy PID approach: SiL real-time simulation. *Soft Comput.* **2020**, doi:10.1007/s00500-019-04152-7.
 66. Batista, J.G.; Souza, D.A.; Dos Reis, L.L.N.; Filgueiras, L.V.O.; Ramos, K.M.; Junior, A.B.S.; Correia, W.B. Performance comparison between the pid and lqr controllers applied to a robotic manipulator joint. In Proceedings of the IECON Proceedings (Industrial Electronics Conference); 2019.
 67. Ju, J.; Zhao, Y.; Zhang, C.; Liu, Y. Vibration suppression of a flexible-joint robot based on parameter identification and fuzzy PID control. *Algorithms* **2018**, doi:10.3390/a11110189.

68. Pan, Y.; Li, X.; Yu, H. Efficient PID Tracking Control of Robotic Manipulators Driven by Compliant Actuators. *IEEE Trans. Control Syst. Technol.* **2019**, doi:10.1109/TCST.2017.2783339.
69. Zhang, H.; Jin, H.; Liu, Z.; Liu, Y.; Zhu, Y.; Zhao, J. Real-time kinematic control for redundant manipulators in a time-varying environment: Multiple-dynamic obstacle avoidance and fast tracking of a moving object. *IEEE Trans. Ind. Informatics* **2020**, doi:10.1109/TII.2019.2917392.
70. Adhikary, N.; Mahanta, C. Hybrid impedance control of robotic manipulator using adaptive backstepping sliding mode controller with PID sliding surface. In Proceedings of the 2017 Indian Control Conference, ICC 2017 - Proceedings; 2017.
71. Wu, X.; Huang, Y.; Kueh, S. Design and simulation of an aerial robotic arm. In Proceedings of the Proceedings of the 2016 IEEE 11th Conference on Industrial Electronics and Applications, ICIEA 2016; 2016.
72. Stergiopoulos, Y.; Kontouras, E.; Gkountas, K.; Giannousakis, K.; Tzes, A. Modeling and control aspects of a UAV with an attached manipulator. In Proceedings of the 24th Mediterranean Conference on Control and Automation, MED 2016; 2016.
73. Bazylev, D.; Margun, A.; Zimenko, K.; Kremlev, A. UAV equipped with a robotic manipulator. In Proceedings of the 2014 22nd Mediterranean Conference on Control and Automation, MED 2014; 2014.
74. Lin, T.; Li, Y.; Qi, J.; Meng, X.; Han, J. Modeling and controller design of hydraulic rotorcraft aerial manipulator. In Proceedings of the Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015; 2015.
75. Shukla, A.; Karki, H. Teleoperation of a UAV by a 6-DOF parallel manipulator with hybrid mapping for inspection of industrial structures. In Proceedings of the International Conference on Control, Automation and Systems; 2017.
76. Gioioso, G.; Mohammadi, M.; Franchi, A.; Prattichizzo, D. A force-based bilateral teleoperation framework for aerial robots in contact with the environment. In Proceedings of the Proceedings - IEEE International Conference on Robotics and Automation; 2015.
77. Miyazaki, R.; Jiang, R.; Paul, H.; Ono, K.; Shimonomura, K. Airborne Docking for Multi-Rotor Aerial Manipulations. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems; 2018.
78. Cataldi, E.; Muscio, G.; Trujillo, M.A.; Rodriguez, Y.; Pierri, F.; Antonelli, G.; Caccavale, F.; Viguria, A.; Chiaverini, S.; Ollero, A. Impedance control of an aerial-manipulator: Preliminary results. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems; 2016.
79. Bazylev, D.; Kremlev, A.; Margun, A.; Zimenko, K. Design of control system for a four-rotor UAV equipped with robotic arm. In Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops; 2016.
80. Lan, G.; Bu, Y.; Liang, J.; Hao, Q. Action synchronization between human and UAV robotic

- arms for remote operation. In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2016; 2016.
81. Fumagalli, M.; Stramigioli, S.; Carloni, R. Mechatronic design of a robotic manipulator for unmanned aerial vehicles. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems; 2016.
 82. Maier, M.; Kondak, K. Landing of VTOL UAVs using a stationary robot manipulator: A new approach for coordinated control. In Proceedings of the Proceedings of the IEEE Conference on Decision and Control; 2015.
 83. Maier, M.; Oeschger, A.; Kondak, K. Robot-Assisted Landing of VTOL UAVs: Design and Comparison of Coupled and Decoupling Linear State-Space Control Approaches. *IEEE Robot. Autom. Lett.* **2016**, doi:10.1109/LRA.2015.2502920.
 84. Caballero, A.; Bejar, M.; Rodriguez-Castaño, A.; Ollero, A. Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms. *Int. J. Adv. Robot. Syst.* **2018**, doi:10.1177/1729881418770525.
 85. Morton, K.; Toro, L.F.G. Development of a robust framework for an outdoor mobile manipulation UAV. In Proceedings of the IEEE Aerospace Conference Proceedings; 2016.
 86. Meng, X.; He, Y.; Li, Q.; Gu, F.; Yang, L.; Yan, T.; Han, J. Contact Force Control of an Aerial Manipulator in Pressing an Emergency Switch Process. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems; 2018.
 87. Paul, H.; Ono, K.; Ladig, R.; Shimonomura, K. A multirotor platform employing a three-axis vertical articulated robotic arm for aerial manipulation tasks. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM; 2018.
 88. De Wagter, C.; Ruijsink, R.; Smeur, E.J.J.; van Hecke, K.G.; van Tienen, F.; van der Horst, E.; Remes, B.D.W. Design, control, and visual navigation of the DelftaCopter VTOL tail-sitter UAV. *J. F. Robot.* **2018**, doi:10.1002/rob.21789.
 89. Wei, W.; Tan, L.; Jin, G.; Lu, L.; Sun, C. A survey of UAV visual navigation based on monocular SLAM. In Proceedings of the Proceedings of 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, ITOEC 2018; 2018.
 90. Balamurugan, G.; Valarmathi, J.; Naidu, V.P.S. Survey on UAV navigation in GPS denied environments. In Proceedings of the International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016 - Proceedings; 2017.
 91. Orsag, M.; Korpela, C.; Oh, P. Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle. *J. Intell. Robot. Syst. Theory Appl.* **2013**, doi:10.1007/s10846-012-9723-4.
 92. Madani, T.; Benallegue, A. Backstepping control for a quadrotor helicopter. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems; 2006.
 93. Mohammed Junaid, R.; Laila Beebi, M.; Ashima, C.R. Backstepping and adaptive backstepping control on robotic ARM. In Proceedings of the 2015 International Conference

on Control, Communication and Computing India, ICCI 2015; 2016.

94. Hsu, C.M.; Wang, H.T.; Tsai, A.; Lee, C.Y. Online Recalibration of a Camera and Lidar System. In Proceedings of the Proceedings - 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018; 2019.
95. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings; 2009.
96. Yang, X.S.; Deb, S. Engineering optimisation by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **2010**, doi:10.1504/IJMMNO.2010.035430.

ANEXOS

A.1 SEÑALES DE CONTROL U_3 , U_4 y U_1 .

CONTROL PARA U_3

Paso seguido y con el fin de completar el desarrollo de las señales de control para el sistema considerando los efectos de los pares giroscópicos, se procede a determinar la expresión que defina a la señal de control U_3 , para lo que se considera una función candidata a Lyapunov $V(z)$ y su derivada respecto al tiempo similares a las escritas para el control anterior, es decir:

$$V(z_3) = \frac{1}{2} z_3^2$$

$$\dot{V}(z_3) = z_3 \dot{z}_3$$

Dónde: $z_3 = x_{3d} - x_3$, y su derivada respecto al tiempo es:

$$\dot{z}_3 = \dot{x}_{3d} - \dot{x}_3$$

Ahora sabiendo que: $\dot{x}_3 = x_4$, la derivada de la función de Lyapunov se puede reescribir como:

$$\dot{V}(z_3) = z_3 [\dot{x}_{3d} - x_4]$$

La estabilización de Z_3 puede ser obtenida con la siguiente entrada de control virtual x_4 :

$$x_4 = \dot{x}_{3d} + \alpha_3 z_3$$

Recordando que esta debe cumplir la condición:

$$x_4(0) = 0$$

Al igual que con el sistema de control general, para este nuevo subsistema se debe minimizar el error, por lo que se propone el control x_4 como:

$$x_4 = \dot{x}_{3d} + \alpha_3 z_3$$

Con $\alpha_3 < 0$. Sustituyendo el valor de x_4 de la ecuación anterior podemos reescribir la derivada de la función de Lyapunov propuesta como:

$$\dot{V}(z_3) = z_3 [\dot{x}_{3d} - \dot{x}_{3d} - \alpha_3 z_3]$$

$$\dot{V}(z_3) = z_3 [-\alpha_3 z_3]$$

De esta manera se muestra que la derivada de la función de Lyapunov cumple con ser semidefinida negativa:

$$\dot{V}(z_3) = -\alpha_3 z_3^2 \leq 0$$

Si se consideran las ecuaciones del error y su derivada, es decir:

$$z_3 = x_{3d} - x_3 \qquad \dot{z}_3 = \dot{x}_{3d} - \dot{x}_3$$

Y tomando en cuenta que $\dot{x}_3 = x_4$, es posible reescribir la derivada del error de la siguiente manera:

$$\dot{z}_3 = \dot{x}_{3d} - x_4$$

Sustituyendo en la ecuación anterior el valor de $\dot{x}_{3d} = x_4 - \alpha_3 z_3$ y el valor de x_4 el cual viene dado por la expresión $x_4 = \dot{x}_{3d} + \alpha_3 z_3$, se obtiene:

$$\dot{z}_3 = x_4 - \alpha_3 z_3 - \dot{x}_{3d} + \alpha_3 z_3$$

Se realiza el siguiente cambio de variable:

$$z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3$$

Para continuar se propone una función de Lyapunov aumentada la cual compense este nuevo error:

$$V(z_3, z_4) = \frac{1}{2} z_3^2 + \frac{1}{2} z_4^2$$

Derivando respecto al tiempo tenemos: $\dot{V}(z_3, z_4) = z_3 \dot{z}_3 + z_4 \dot{z}_4$, con lo anterior y dadas las siguientes ecuaciones:

$$\begin{aligned} z_3 &= x_{3d} - x_3 & z_4 &= x_4 - \dot{x}_{3d} - \alpha_3 z_3 & x_4 &= \dot{x}_{3d} + \alpha_3 z_3 \\ \dot{z}_3 &= \dot{x}_{3d} - \dot{x}_3 & \dot{z}_4 &= \dot{x}_4 - \ddot{x}_{3d} - \alpha_3 \dot{z}_3 & \dot{x}_3 &= x_4 \end{aligned}$$

Se sustituye el valor de \dot{z}_3 y \dot{z}_4 en la derivada de la función de Lyapunov:

$$\dot{V}(z_3, z_4) = z_3(\dot{x}_{3d} - \dot{x}_3) + z_4(\dot{x}_4 - \ddot{x}_{3d} - \alpha_3(\dot{x}_{3d} - \dot{x}_3))$$

Despejando el valor de $\dot{x}_{3d} = x_4 - \alpha_3 z_3 - z_4$ de la ecuación de z_4 y sustituyéndolo en la expresión anterior se obtiene:

$$\dot{V}(z_3, z_4) = z_3(x_4 - \alpha_3 z_3 - z_4 - \dot{x}_3) + z_4(\dot{x}_4 - \ddot{x}_{3d} - \alpha_3(x_4 - \alpha_3 z_3 - z_4 - \dot{x}_3))$$

Sabiendo que $\dot{x}_3 = x_4$, y realizando las simplificaciones posibles en la expresión anterior se obtiene finalmente:

$$\dot{V}(z_3, z_4) = z_3(-\alpha_3 z_3 - z_4) + z_4(\dot{x}_4 - \ddot{x}_{3d} - \alpha_3(-\alpha_3 z_3 - z_4))$$

Considerando las ecuaciones del modelo que considera el efecto de los pares giroscópicos sobre el sistema, obtenemos el valor de $\ddot{\theta}$:

$$\ddot{\theta} = \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} + \frac{J_R}{I_{xx}} \Omega \dot{\theta} + \frac{l}{I_{yy}} U_3 + \frac{A_q}{I_{yy}}$$

Dónde:

$$a_3 = \frac{(I_{zz} - I_{xx})}{I_{yy}} \quad a_4 = \frac{J_R}{I_{xx}} \quad b_2 = \frac{l}{I_{yy}}$$

Y sabiendo que: $\dot{x}_4 = \ddot{x}_3 = \ddot{\theta}$, $x_2 = \dot{x}_1 = \dot{\phi}$, $x_6 = \dot{x}_5 = \dot{\psi}$, se puede reescribir la expresión que representa a $\ddot{\theta}$ de la forma:

$$\ddot{\theta} = a_3 x_2 x_6 + a_4 x_2 \Omega + b_2 U_3$$

Por lo que la ecuación de la derivada de la función de Lyapunov es ahora:

$$\dot{V}(z_3, z_4) = z_3(-\alpha_3 z_3 - z_4) + z_4(a_3 x_2 x_6 + a_4 x_2 \Omega + b_2 U_3 - \dot{x}_{3d} - \alpha_3(-\alpha_3 z_3 - z_4))$$

Considerando la aceleración de referencia del vehículo como nula, ya que ha de suponerse que parte del reposo, se tiene que:

$$\dot{V}(z_3, z_4) = z_3(-\alpha_3 z_3 - z_4) + z_4(a_3 x_2 x_6 + a_4 x_2 \Omega + b_2 U_3 - \alpha_3(-\alpha_3 z_3 - z_4))$$

Para satisfacer que $\dot{V}(z_3, z_4) \leq 0$ se propone una señal de control que lo garantice, es decir:

$$U_3 = \frac{1}{b_2} (z_3 - a_3 x_2 x_6 - a_4 x_2 \Omega - \alpha_3(\alpha_3 z_3 + z_4) - \alpha_4 z_4)$$

Por lo que sustituyendo la señal de control anterior en la derivada de la función que define el error general de esta señal de control U_3 se obtiene que:

$$\dot{V}(z_3, z_4) = -\alpha_3 z_3^2 - \alpha_4 z_4^2$$

Lo cual cumple con los criterios establecidos por la teoría de Lyapunov para la correcta proposición de una señal de control apropiada, lo que garantiza un desempeño óptimo en la señal de control definida en este apartado.

A.2 CODIGOS IMPLEMENTADOS EN LA SIMULACIÓN

En esta sección se mostrarán los códigos que se implementaron en este trabajo para las simulaciones dinámicas del sistema, así como para los controladores diseñados. Estos códigos fueron escritos en S-function, la cual ofrece una gran versatilidad al permitir la utilización de código m (Código de Matlab) en los modelos desarrollados en Simulink. Para la correcta implementación de las S-function es necesaria una cabecera la cual construirá la estructura básica de la función y permitirá

En primer lugar, se muestran los códigos empleados para simular la dinámica del multirroto, para ello se han empleado dos S-function, la primera de ellas representa la dinámica traslacional del vehículo, mientras que la segunda es utilizada para representar la orientación del vehículo. Cabe mencionar que estos códigos fueron implementados en el diseño de ambos controladores.

En la Figura A1 se puede observar la S-function implementada para representar la traslación del vehículo aéreo, en esta figura se puede observar de igual manera la estructura general de una S-function.

```
function [sys,x0,str,ts] = Posicion(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0;    sizes.NumDiscStates = 0;
        sizes.NumOutputs = 3;    sizes.NumInputs = 4;
        sizes.DirFeedthrough = 1;    sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];    str = [];    ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
function sys=mdlOutputs(u)
%% Parametros del UAV.
m=0.55; g=9.81;
%% Entradas de la funcion.
U1=u(1);    phi=u(2);    teta=u(3);    epsi=u(4);
%% Ecuaciones dinamicas del UAV.
xpp=(1/m)*((cos(epsi))*(sin(teta))*(cos(phi))+((sin(epsi))*(sin(phi))))*(U1);
ypp=(1/m)*((sin(epsi))*(sin(teta))*(cos(phi))-((cos(epsi))*(sin(phi))))*(U1);
zpp=-g+((1/m)*((cos(teta))*(cos(phi))))*U1;
%% Aceleraciones Angulares.
sys = [xpp,ypp,zpp];
```

Figura A1. Código empleado para la representación de la traslación del vehículo.

Para completar la representación del vehículo se implemente el código mostrado en la Figura A2, el cual representa la orientación del UAV, dicho código fue implementado mediante una S-

function y devuelve las aceleraciones angulares, las cuales serán posteriormente integradas para obtener las aceleraciones angulares que a su vez serán integradas para finalmente obtener la orientación del UAV.

```
function [sys,x0,str,ts] = Orientacion(~,~,u,flag)

switch flag,
  case 0,
    sizes = simsizes;
    sizes.NumContStates = 0;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 3;
    sizes.NumInputs = 6;
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0 = []; str = []; ts = [0 0];
    case {1,2,4,9}
      sys=[];
    case 3,
      sys=mdlOutputs(u);
    otherwise
      DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function sys=mdlOutputs(u)
%% Parametros del UAV.
Ixx=8.7e-3;      Iyy=8.7e-3;
Izz=8.7e-3;      l=0.205;
a1=((Iyy-Izz)/(Ixx));  a3=((Izz-Ixx)/(Iyy));
a5=((Ixx-Iyy)/(Izz));  b1=(l/Ixx);
b2=(l/Iyy);          b3=((1)/(Izz));
%% Entradas de la funcion.
U2=u(1);  U3=u(2);  U4=u(3);
phip=u(4); tetap=u(5); epsip=u(6);
%% Ecuaciones dinamicas del UAV.
phipp=((a1)*(tetap)*(epsip))+((b1)*(U2));
tetapp=((a3)*(epsip)*(phip))+((b2)*(U3));
epsipp=((a5)*(tetap)*(phip))+((b3)*(U4));
%% Aceleraciones Lineales.
sys = [phipp,tetapp,epsipp];
```

Figura A2. Código empleado para la representación de la orientación del vehículo

El código mostrado en la Figura A1 representa junto con el código mostrado en la Figura A2 la representación dinámica del UAV.

El controlador Backstepping fue implementado mediante dos S-function, la primera de ellas representa el control de la traslación del vehículo y puede observarse en la Figura A3. Esta función entregara como salida la señal de control U_1 , además de los ángulos de alabeo y cabeceo de referencia.

```

function [sys,x0,str,ts] = ControlTraslacion(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0;   sizes.NumDiscStates = 0;
        sizes.NumOutputs = 3;     sizes.NumInputs = 15;
        sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];   str = [];   ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function sys=mdlOutputs(u)
%% Parametros del UAV y del controlador.
alpha7=50; alpha8=3;   alpha9=15; alpha10=3;
alpha11=15; alpha12=3; g=9.81;   m=0.55;
%% Entradas de la funcion.
epsir=u(1); phi=u(2);   teta=u(3);   x=u(4);   y=u(5);
z=u(6);   xp=u(7);   yp=u(8);   zp=u(9);   xrp=u(10);
yrp=u(11);   zrp=u(12);   xr=u(13);   yr=u(14);   zr=u(15);
%% Declaracion de las variables de error.
z7=zr-z;   z8=zp-zrp-((alpha7)*(z7));
z9=xr-x;   z10=xp-xrp-((alpha9)*(z9));
z11=yr-y;   z12=yp-yrp-((alpha11)*(z11));
%% Señales de control
U1=(m/((cos(phi))*(cos(teta))))*(z7+g-(alpha7*(z8+(alpha7*z7)))-(alpha8*z8));
Ux=(m/U1)*(z9 - ((alpha9)*(z10 + (alpha9*z9))) - (alpha10*z10));
Uy=(m/U1)*(z11 - ((alpha11)*(z12 + (alpha11*z11))) - (alpha12*z12));
%% Angulo de alabeo deseado
a=((sin(epsir))*(Ux))-((Uy)*(cos(epsir)));
if a<-1
    a=-1;
end
if a>1
    a=1;
end
phir=asin(a);
%% Angulo de cabeceo deseado
b=((Ux)-((sin(epsir))*(sin(phir))))/((cos(epsir))*(cos(phir)));
if b<-1
    b=-1;
end
if b>1
    b=1;
end
tetar=asin(b);
%% Salidas de la funcion.
sys = [U1,phir,tetar];

```

Figura A3. Código empleado para el control de la traslación mediante Backstepping.

Para finalizar el controlador Backstepping se muestra en la Figura A4 el código empleado para el control de la orientación del Dron, Dicho código será implementado mediante una S-function la cual entregará como salida las señales de control U_2 , U_3 y U_4 necesarias para el correcto seguimiento de la trayectoria.

```

function [sys,x0,str,ts] = ControlOrientacion(~,~,u,flag)

switch flag,
  case 0,
    sizes = simsizes;
  sizes.NumContStates = 0;   sizes.NumDiscStates = 0;
  sizes.NumOutputs    = 3;   sizes.NumInputs    = 12;
  sizes.DirFeedthrough = 1;   sizes.NumSampleTimes = 1;
  sys = simsizes(sizes);
  x0 = [];   str = [];   ts = [0 0];
  case {1,2,4,9}
    sys=[];
  case 3,
    sys=mdlOutputs(u);
  otherwise
    DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function sys=mdlOutputs(u)
%% Parametros del UAV y del controlador.
Ixx=8.7e-3;   Iyy=8.7e-3;   Izz=8.7e-3;   l=0.205;   alpha1=50;
alpha2=1;     alpha3=50;   alpha4=1;   alpha5=50;   alpha6=1;
b1=(1/Ixx);   b2=(1/Iyy);   b3=(1/Izz);   a1=((Iyy-Izz)/Ixx);
a3=((Izz-Ixx)/Iyy);   a5=((Ixx-Iyy)/Izz);

%% Entradas de la funcion.
phip=u(1);   tetap=u(2);   epsip=u(3);   phirp=u(4);
tetarp=u(5);   epsirp=u(6);   phir=u(7);   tetar=u(8);
phi=u(9);     teta=u(10);   epsir=u(11);   epsi=u(12);
x4=tetap;     x6=epsip;     x2=phip;

%% Declaracion de las variables de error.
z1=(phir-phi);   z2=phip-phirp-(alpha1*z1);
z3=tetar-teta;   z4=tetap-tetarp-(alpha3*z3);
z5=epsir-epsi;   z6=epsip-epsirp-(alpha5*z5);

%% Señales de control
U2=(1/b1)*((z1)-(a1*x4*x6)-((alpha1)*(z2+(alpha1*z1)))-(alpha2*z2));
U3=(1/b2)*(z3 - (a3*x2*x6) - (alpha3*(z4 + (alpha3*z3))) - (alpha4*z4));
U4=(1/b3)*(z5 - (a5*x2*x4) - (alpha5*(z6 + (alpha5*z5))) - (alpha6*z6));

%% Salidas de la funcion.
sys = [U2,U3,U4];

```

Figura A4. Código empleado para el control de la orientación mediante Backstepping.

El código mostrado en la Figura A3 y el mostrado en la Figura A4 representan el controlador Backstepping implementado.

En el desarrollo del controlador PID se implementó una S-function la cual se encarga de calcular los ángulos de alabeo y cabeceo de referencia para que el sistema sea capaz de cumplir con la trayectoria impuesta, el código implementado en esta S-function se muestra en la Figura A5:

```

function [sys,x0,str,ts] = Angulos(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
sizes.NumContStates = 0;    sizes.NumDiscStates = 0;
sizes.NumOutputs    = 2;    sizes.NumInputs    = 3;
sizes.DirFeedthrough = 1;    sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];    str = [];    ts = [0 0];
    case {1,2,4,9}
        sys=[];
    case 3,
        sys=mdlOutputs(u);
    otherwise
        DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function sys=mdlOutputs(u)
%% Entradas de la funcion.
epsir=u(1); Ux=u(2);    Uy=u(3);
%% Angulo de alabeo deseado
a=((sin(epsir))*(Ux))-((Uy)*(cos(epsir)));
if a<-1
    a=-1;
end
if a>1
    a=1;
end
phir=asin(a);
%% Angulo de cabeceo deseado
b=((Ux)-((sin(epsir))*(sin(phir))))/((cos(epsir))*(cos(phir)));
if b<-1
    b=-1;
end
if b>1
    b=1;
end
tetar=asin(b);
sys = [phir,tetar];

```

Figura A5. Código para el cálculo de los ángulos de alabeo y cabeceo en el controlador PID

Por otra parte, para la simulación del brazo se muestra En la Figura A6 la S-function implementada para representar la dinámica del manipulador.

```

function [sys,x0,str,ts] = PosicionM(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0;    sizes.NumDiscStates = 0;
        sizes.NumOutputs = 4;    sizes.NumInputs = 6;
        sizes.DirFeedthrough = 1;    sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];    str = [];    ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function sys=mdlOutputs(u)
%% Parametros del Manipulador.
m1=0.02; g=9.81; l1=0.065; l2=0.085; m2=0.03; lc1=l1/2; lc2=l2/2;
I1=6.83726; I2=8.2937;

%% Entradas de la funcion.
T1=u(1); T2=u(2); q1p=u(3); q2p=u(4); q1=u(5); q2=u(6);

%% Matrices Manipulador
M=[(m1*lc1^2)+(m2*(l1^2+l2^2+2*l1*lc2*cos(q2)))+I1+I2,
(m2*(lc2^2+(l1*lc2*cos(q2))))+I2; (m2*lc2^2)+(m2*l1*lc2*cos(q2))+I2, (m2*l1
c2^2)+I2];
C=[(-m2*l1*lc2*sin(q2)*q2p), -
m2*l1*lc2*sin(q2)*(q1p+q2p);m2*l1*lc2*sin(q2)*q1p, 0];
G=[(m1*g*lc1*sin(q1))+(m2*g*sin(q1))+(m2*g*lc2*sin(q1+q2));m2*g*lc2*sin(
q1+q2)];
T=[T1;T2]; qp=[q1p;q2p];
%% Ecuacion dinamica del Manipulador.

qp=inv(M)*((T)-(C*qp)-(G));
y=(l1*sin(q1))+(l2*sin(q1+q2));
z=(-l1*cos(q1))-(l2*cos(q1+q2));

%% Aceleraciones Angulares.
sys = [qp(1),qp(2),y,z];

```

Figura A6. Código para representar la dinámica del manipulador.

Para la simulación del control Backstepping del manipulador se utilizó el código que puede observarse en la figura A7.

```

function [sys,x0,str,ts] = ControlTraslacionM(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0; sizes.NumDiscStates = 0;
        sizes.NumOutputs = 2; sizes.NumInputs = 10;
        sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
        sys = simsizes(sizes); x0 = [];str = []; ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function sys=mdlOutputs(u)
%% Parametros del Manipulador.
m1=0.02; g=9.81; l1=0.065; l2=0.085; m2=0.03; lc1=l1/2; lc2=l2/2;
I1=6.83726; I2=8.2937;
%% Entradas de la funcion.
q1ppd=u(1); q1pd=u(2); q1d=u(3); q2ppd=u(4); q2pd=u(5); q2d=u(6);
q1=u(7); q1p=u(8); q2=u(9); q2p=u(10);

%% Matrices Manipulador
M=[(m1*lc1^2)+(m2*(l1^2+lc2^2+2*l1*lc2*cos(q2)))+I1+I2,
(m2*(lc2^2+(l1*lc2*cos(q2))))+I2; (m2*lc2^2)+(m2*l1*lc2*cos(q2))+I2, (m2*lc2^2)+I2];
C=[(-m2*l1*lc2*sin(q2)*q2p), -
m2*l1*lc2*sin(q2)*(q1p+q2p); m2*l1*lc2*sin(q2)*q1p, 0];
G=[(m1*g*lc1*sin(q1))+(m2*g*sin(q1))+(m2*g*lc2*sin(q1+q2)); m2*g*lc2*sin(q1+q2)];
q=[q1;q2]; qpd=[q1pd;q2pd];
qppd=[q1ppd;q2ppd]; qd=[q1d;q2d]; qdp=[q1pd;q2pd]; qp=[q1p;q2p];

%% Ganancias de control.
k1=200;
k2=100;
K=[k1, 0; 0, k2];
Z13=q-qd;
Z14=(qp-qdp)+(K*Z13);

%% Calculo Señales de control
T=(M*qppd)+(C*qpd)-(M*K*Z14)+G-(K*Z13);
T1=T(1); T2=T(2);

%% Salidas de la funcion.
sys = [T1,T2];
y=(l1*sin(q1))+(l2*sin(q1+q2));
z=(-l1*cos(q1))-(l2*cos(q1+q2));

```

Figura A7. Código del controlador Backstepping del manipulador.

A continuación, se muestran los códigos pertenecientes a las simulaciones del control del sistema en Matlab. En la Figura A8 se muestra la dinámica correspondiente al Vehículo con el brazo acoplado

```
function [sys,x0,str,ts] = Posicion(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0;    sizes.NumDiscStates = 0;
        sizes.NumOutputs = 3;    sizes.NumInputs = 4;
        sizes.DirFeedthrough = 1;    sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];    str = [];    ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
    end

function sys=mdlOutputs(u)
%% Parametros del Manipulador.
m1=0.02; g=9.81; l1=0.065; l2=0.085; m2=0.03; lc1=l1/2; lc2=l2/2;
%m1=0.5; m2=0.05;
%% Parametros del Quadrotor.

m=0.65; masa=m+m1+m2;
%% Entradas de la funcion.

U1=u(1);    phi=u(2);    teta=u(3);    epsi=u(4);
%% Ecuaciones dinamicas del Quadrotor.

xpp=(1/masa)*((cos(epsi))*(sin(teta))*(cos(phi))+((sin(epsi))*(sin(phi)
)))*U1);
ypp=(1/masa)*((sin(epsi))*(sin(teta))*(cos(phi))-
((cos(epsi))*(sin(phi))))*U1);
zpp=-g+((1/masa)*((cos(teta))*(cos(phi))))*U1);

%% Aceleraciones Angulares.
sys = [xpp,ypp,zpp];
```

Figura A8. Código empleado para representar la dinámica trasnacional del vehículo con el manipulador acoplado.

Del mismo modo en la Figura A8 se muestra el código utilizado para representar la dinámica rotacional del vehículo con el manipulador acoplado.

```

function [sys,x0,str,ts] = Orientacion(~,~,u,flag)

switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0;    sizes.NumDiscStates = 0;
        sizes.NumOutputs = 3;    sizes.NumInputs = 12;
        sizes.DirFeedthrough = 1;    sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];    str = [];    ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
function sys=mdlOutputs(u)
%% Entradas de la funcion.
U2=u(1);    U3=u(2);    U4=u(3);    phip=u(4);    tetap=u(5);    epsip=u(6);
q1=u(7);    q2=u(8);    T1=u(9);    T2=u(10);    yp=u(11);    teta=u(12);
%% Parametros del Manipulador.
m1=0.02;    g=9.81;    l1=0.065;    l2=0.085;    m2=0.03;    lc1=l1/2;    lc2=l2/2;
%% Inercia del Quadrotor.
Ix=8.7e-2;    Iy=8.7e-2;    Iz=8.7e-2;    m=0.65;
%% Inercia del sistema completo
Ixx=Ix+((m1*(l1^2)*(sin(q1))^3)/3)+((m2*l2*(sin(q2))^3)/3)+((m2*l2*(sin(q1)^2)));
Iyy=Iy+((m1*(l1^2))/3)+((m2*(l2^2))/3)+((m2*(l1^2)));
Izz=Iz+((m1*(l1^2)*(cos(q1))^3)/3)+((m2*l2*(cos(q2))^3)/3)+((m2*l2*(cos(q1)^2)));
%% Centros de masa del sistema
Cx=((l1*cos(q1)*(m1/2)+m2)+(m2*lc2*cos(q1-q2)))/(m+m1+m2);
Cy=0;
Cz=((l1*sin(q1)*(m1/2)+m2)+(m2*lc2*sin(q1-q2)))/(m+m1+m2);
%% Efecto del torque del manipulador en el UAV
Tr=T1+(m2*g*lc1*cos(q1)+(m2*g*lc2*cos(q1-q2))+T2;
%% parametros multirrotor
l=0.205;    a1=((Iyy-Izz)/(Ixx));
a3=((Izz-Ixx)/(Iyy));    a5=((Ixx-Iyy)/(Izz));    b1=(1/Ixx);
b2=(1/Iyy);    b3=(1/Izz);
%% Ecuaciones dinamicas del Quadrotor.
phipp=((a1)*(tetap)*(epsip))+((b1)*(U2-Tr));
tetapp=((a3)*(epsip)*(phip))+((b2)*(U3))+(((g*(m1+m2))*(yp*sin(teta)))/(Iyy));
epsipp=((a5)*(tetap)*(phip))+((b3)*(U4));
%% Aceleraciones Angulares.
sys = [phipp,tetapp,epsipp];

```

Figura A9. Código empleado para representar la dinámica Rotacional del vehículo con el manipulador acoplado.

En la Figura A10 se muestra el código empleado para el control del subsistema de traslación del sistema UAV-Manipulador.

```

function [sys,x0,str,ts] = ControlTraslacion(~,~,u,flag)
switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0; sizes.NumDiscStates = 0;
        sizes.NumOutputs = 3; sizes.NumInputs = 15;
        sizes.DirFeedthrough = 1; sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];str = []; ts = [0 0];
    case {1,2,4,9}
        sys=[];
    case 3,
        sys=mdlOutputs(u);
    otherwise
        DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
function sys=mdlOutputs(u)
%% Parametros del Manipulador.
m1=0.02; g=9.81; l1=0.065; l2=0.085; m2=0.03; lc1=l1/2; lc2=l2/2;
%% Parametros del Quadrotor y del controlador.
alpha7=500; alpha8=300; alpha9=10; alpha10=1;
alpha11=10; alpha12=1; g=9.81; m=0.65; masa=m+m1+m2;
%% Entradas de la funcion.
epsir=u(1);phi=u(2);teta=u(3);x=u(4);y=u(5);z=u(6); xp=u(7);yp=u(8);
zp=u(9); xrp=u(10);yrp=u(11);zrp=u(12);xr=u(13);yr=u(14);zr=u(15);
%% Declaracion de las variables de error.
z7=zr-z; z8=zp-zrp-((alpha7)*(z7)); z9=xr-x;
z10=xp-xrp-((alpha9)*(z9)); z11=yr-y; z12=yp-yrp-((alpha11)*(z11));
%% Señales de control
U1=(masa/((cos(phi))*(cos(teta))))*(z7+g-(alpha7*(z8+(alpha7*z7)))-
(alpha8*z8));
Ux=(masa/U1)*(z9 - ((alpha9)*(z10 + (alpha9*z9))) - (alpha10*z10));
Uy=(masa/U1)*(z11 - ((alpha11)*(z12 + (alpha11*z11))) - (alpha12*z12));
%% Angulo de alabeo deseado
a=((sin(epsir))*(Ux))-((Uy)*(cos(epsir)));
if a<-1
    a=-1;
end
if a>1
    a=1;
end
phir=asin(a);
%% Angulo de cabeceo deseado
b=((Ux)-((sin(epsir))*(sin(phir))))/((cos(epsir))*(cos(phir)));
if b<-1
    b=-1;
end
if b>1
    b=1;
end
tetar=asin(b);
%% Salidas de la funcion.
sys = [U1,phir,tetar];

```

Figura A10. Código empleado para representar la dinámica Rotacional del vehículo con el manipulador acoplado.

En la Figura A11 se muestra el código para el control del subsistema de rotación del sistema UAV-Manipulador.

```
function [sys,x0,str,ts] = ControlOrientacion(~,~,u,flag)
switch flag,
    case 0,
        sizes = simsizes;
        sizes.NumContStates = 0;    sizes.NumDiscStates = 0;
        sizes.NumOutputs = 3;    sizes.NumInputs = 17;
        sizes.DirFeedthrough = 1;    sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];    str = [];    ts = [0 0];
        case {1,2,4,9}
            sys=[];
        case 3,
            sys=mdlOutputs(u);
        otherwise
            DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
function sys=mdlOutputs(u)
    phip=u(1);    tetap=u(2);    epsip=u(3);    phirp=u(4);    tetarp=u(5);
    epsirp=u(6);    phir=u(7);    tetar=u(8);    phi=u(9);    teta=u(10);
    epsir=u(11);    epsi=u(12);    q1=u(13);    q2=u(14);    T1=u(15);    T2=u(16);
    yp=u(17);    x4=tetap;    x6=epsip;    x2=phip;
    %% Parametros del Manipulador.
    m1=0.1;    g=9.81;    l1=0.45;    l2=0.45;    m2=0.05;    lc1=l1/2;    lc2=l2/2;
    %m1=0.5;    m2=0.05;
    %% Parametros del Quadrotor y del controlador.
    Ix=8.7e-2;    Iy=8.7e-2;    Iz=8.7e-2;    m=0.55;    l=0.205;
    %% Inercia del sistema completo
    Ixx=Ix+(m1*(l1^2)*(sin(q1))^3)/3+(m2*l2*(sin(q2))^3/3)+(m2*l2*(sin(q1)^2));
    Iyy=Iy+(m1*(l1^2)/3)+(m2*(l2^2)/3)+(m2*(l1^2));
    Izz=Iz+(m1*(l1^2)*(cos(q1))^3)/3+(m2*l2*(cos(q2))^3/3)+(m2*l2*(cos(q1)^2));
    %% Efecto del torque del manipulador en el UAV
    Tr=T1+(m2*g*lc1*cos(q1))+(m2*g*lc2*cos(q1-q2))+T2;
    %% Parametros del Quadrotor y del controlador.
    pos=1400;    acw=1600;    alpha1=pos*2;    alpha2=acw*2;    alpha3=pos;    alpha4=acw;
    alpha5=50;    alpha6=80;    b1=(1/Ixx);    b2=(1/Iyy);    b3=(1/Izz);
    a1=((Iyy-Izz)/Ixx);    a3=((Izz-Ixx)/Iyy);    a5=((Ixx-Iyy)/Izz);
    %% Declaracion de las variables de error.
    z1=(phir-phi);    z2=phip-phirp-(alpha1*z1);    z3=tetar-teta;
    z4=tetap-tetarp-(alpha3*z3);    z5=epsir-epsi;    z6=epsip-epsirp-(alpha5*z5);
    %% Señales de control
    U2=(1/b1)*((z1)-(a1*x4*x6)-(b1*Tr)-((alpha1)*(z2+(alpha1*z1)))-
    (alpha2*z2));%
    U3=(1/b2)*(z3 - (a3*x2*x6)-(((g*(m1+m2))*(yp*sin(teta)))/(Iyy)) -
    (alpha3*(z4 + (alpha3*z3))) - (alpha4*z4));
    U4=(1/b3)*(z5 - (a5*x2*x4) - (alpha5*(z6 + (alpha5*z5))) -
    (alpha6*z6));
    sys = [U2,U3,U4];
```

Figura A11. Código empleado para representar la dinámica Rotacional del vehículo con el manipulador acoplado.

A.3 CODIGOS IMPLEMENTADOS EN EL HARDWARE

En la Figura A12 se muestra el código escrito en Visual Basic empleado para la descriptación de la información de telemetría proporcionada por el DJI datalink 2.4 Ghz.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;

namespace DriverMatlab
{
    public class Datos
    {
        private static byte[] key11;
        private static byte[] bux;
        private static byte[] bux1;
        private static UInt16 commandID;
        private static UInt16 SN;
        private static ushort version;
        private static ushort flag;
        private static byte encrypType;
        private static uint GSVoltageWarningType;
        private static float GSMotorVoltage;
        private static uint GSGSP_Count;
        private static uint GSFwClock;
        private static ushort GSTLstatus;
        private static ushort GSOSd_flag;
        private static ushort GSfailsafe;
        private static short GSHeliAttiRoll;
        private static short GSHeliAttiPitch;
        private static short GSHeliAttiYaw;
        private static short GSHeliAttiVX;
        private static short GSHeliAttiVY;
        private static short GSHeliAttiVZ;
        private static double GSHeliAttiLati;
        private static double GSHeliAttiLon;
        private static float GSHeliAttiAlti;
        private static byte GSHeliAttiWpIdx;
        private static byte GSFwLedRed;
        private static byte GSFwLedWhite;
        private static byte GSFwLedGreen;
        private static ushort GSFwServovol;
        private static byte GSFwpitch_ATV;
        private static byte GSFwthrottle_ATV;
        private static ushort GSFwWpCnt;
        private static double GSFwWpTargetlat;
        private static double GSFwWpTargetlon;
        private static float GSFwWpTargetalt;
    }
}
```

```
public static void Bateria(byte[] b, out byte version, out UInt16 SN, out UInt16
commandID, out UInt16 flag, out byte encrypType, out UInt32 GSVoltageWarningType, out
Single GSMotorVoltage, out UInt32 GSGSP_Count, out UInt32 GSFwclock, out UInt16
GSTLstatus, out UInt16 GS0sd_flag, out UInt16 GSfailsafe)
{
    key11 = new byte[] { 15, 2, 0, 1, 0, 0x12, 0x30, 9, 9, 1, 6, 0x12, 13, 5,
7, 0x90 };
    //bux = new byte[]
{0,137,213,65,139,21,106,248,196,155,224,52,246,106,136,108,184,205,19,177,118,13,172,
143,135,52,25,37,1};
    bux = b;
    ushort num = 0;
    bux1=tea_decrypt(bux, key11);

    version = Convert.ToByte(bux1[num]);
    SN = BitConverter.ToUInt16(bux1, num = (ushort)(num + 1));
    commandID = BitConverter.ToUInt16(bux1, num = (ushort)(num + 2));
    flag = BitConverter.ToUInt16(bux1, num = (ushort)(num + 2));
    encrypType = Convert.ToByte(bux1[num = (ushort)(num + 2)]);
    num = (ushort)(num + 1);
    GSVoltageWarningType = BitConverter.ToUInt32(bux1, num);
    num += 4;
    GSMotorVoltage = BitConverter.ToSingle(bux1, num);
    num += 4;
    if (num <= (bux1.Length - 4))
    {
        GSGSP_Count = BitConverter.ToUInt32(bux1, num);
        num += 4;
    }
    else { GSGSP_Count = 0; }
    if (num <= (bux1.Length - 4))
    {
        GSFwclock = BitConverter.ToUInt32(bux1, num);
        num += 4;
    }
    else { GSFwclock = 0; }
    if (num <= (bux1.Length - 2))
    {
        GSTLstatus = BitConverter.ToUInt16(bux1, num);
        num += 2;
    }
    else { GSTLstatus = 0; }
    if (num <= (bux1.Length - 2))
    {
        GS0sd_flag = BitConverter.ToUInt16(bux1, num);
        num += 2;
    }
    else { GS0sd_flag = 0; }
    if (num <= (bux1.Length - 2))
    {
        GSfailsafe = BitConverter.ToUInt16(bux1, num);
        num += 2;
    }
    else { GSfailsafe = 0; }
}
}
```

```
public static void Posicion(byte[] b, out byte version, out UInt16 SN, out UInt16
commandID, out UInt16 flag, out byte encrypType, out Int16 GSHeliAttiRoll, out Int16
GSHeliAttiPitch, out Int16 GSHeliAttiYaw, out Int16 GSHeliAttiVX, out Int16
GSHeliAttiVY, out Int16 GSHeliAttiVZ, out Double GSHeliAttiLati, out Double
GSHeliAttiLon, out Single GSHeliAttiAalti, out UInt32 GSHeliAttiWpIdx, out UInt32
GSFWclock)
{
    key11 = new byte[] { 15, 2, 0, 1, 0, 0x12, 0x30, 9, 9, 1, 6, 0x12, 13, 5,
7, 0x90 };
    bux = b;
    ushort num = 0;
    bux1 = tea_decrypt(bux, key11);

    version = Convert.ToByte(bux1[num]);
    SN = BitConverter.ToUInt16(bux1, num = (ushort)(num + 1));
    commandID = BitConverter.ToUInt16(bux1, num = (ushort)(num + 2));
    flag = BitConverter.ToUInt16(bux1, num = (ushort)(num + 2));
    encrypType = Convert.ToByte(bux1[num = (ushort)(num + 2)]);
    num = (ushort)(num + 1);

    GSHeliAttiRoll = BitConverter.ToInt16(bux1, num);
    num += 2;
    GSHeliAttiPitch = BitConverter.ToInt16(bux1, num);
    num += 2;
    GSHeliAttiYaw = BitConverter.ToInt16(bux1, num);
    num += 2;
    GSHeliAttiVX = BitConverter.ToInt16(bux1, num);
    num += 2;
    GSHeliAttiVY = BitConverter.ToInt16(bux1, num);
    num += 2;
    GSHeliAttiVZ = BitConverter.ToInt16(bux1, num);
    num += 2;
    GSHeliAttiLati = BitConverter.ToDouble(bux1, num);
    num += 8;
    GSHeliAttiLon = BitConverter.ToDouble(bux1, num);
    num += 8;
    GSHeliAttiAalti = BitConverter.ToSingle(bux1, num);
    num += 4;
    GSHeliAttiWpIdx = bux1[num];
    num++;
    if (num <= (bux1.Length - 4))
    {
        GSFWclock = BitConverter.ToUInt32(bux1, num);
        num += 4;
    }
    else { GSFWclock = 0; }
}
```

```

    public static void GSFWS(byte[] b, out byte version, out UInt16 SN, out UInt16
commandID, out UInt16 flag, out byte encryptType, out uint GSFWLedRed, out uint
GSFWLedWhite, out uint GSFWLedGreen, out UInt16 GSFWServovol, out byte GSFWpitch_ATV, out
byte GSFWthrottle_ATV, out UInt16 GSFWWpCnt, out Double GSFWWpTargetlat, out Double
GSFWWpTargetlon, out Single GSFWWpTargetalt, out UInt32 GSFWclock)
    {
        key11 = new byte[] { 15, 2, 0, 1, 0, 0x12, 0x30, 9, 9, 1, 6, 0x12, 13, 5,
7, 0x90 };
        bux = b;
        ushort num = 0;
        bux1 = tea_decrypt(bux, key11);
        version = Convert.ToByte(bux1[num]);
        SN = BitConverter.ToUInt16(bux1, num = (ushort)(num + 1));
        commandID = BitConverter.ToUInt16(bux1, num = (ushort)(num + 2));
        flag = BitConverter.ToUInt16(bux1, num = (ushort)(num + 2));
        encryptType = Convert.ToByte(bux1[num = (ushort)(num + 2)]);
        num = (ushort)(num + 1);
        GSFWLedRed = bux1[num];
        num++;
        GSFWLedWhite = bux1[num];
        num++;
        GSFWLedGreen = bux1[num];
        num += 4;
        GSFWServovol = BitConverter.ToUInt16(bux1, num);
        num += 2;
        GSFWpitch_ATV = bux1[num];
        num += 4;
        GSFWthrottle_ATV = bux1[num];
        num += 4;
        GSFWWpCnt = BitConverter.ToUInt16(bux1, num);
        num += 2;
        GSFWWpTargetlat = BitConverter.ToDouble(bux1, num);
        num += 8;
        GSFWWpTargetlon = BitConverter.ToDouble(bux1, num);
        num += 8;
        GSFWWpTargetalt = BitConverter.ToSingle(bux1, num);
        num += 4;
        if (num <= (bux1.Length - 4))
        {
            GSFWclock = BitConverter.ToUInt32(bux1, num);
            num += 4;
        }
        else { GSFWclock = 0; }
    }

private static byte[] TobyteArray(uint[] Data)
{
    int num = Data.Length << 2;
    byte[] buffer = new byte[num];
    for (int i = 0; i < num; i++)
    {
        buffer[i] = (byte)(Data[i >> 2] >> ((i & 3) << 3));
    }
    return buffer;
}

```

