



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Programa de
Sistemas Autónomos de Navegación Aérea y Submarina

“Navegación de un vehículo
aéreo no tripulado utilizando
odometría visual”

TESIS

Que presenta

ING. JONATHAN ZÁRATE CARTAS

Para obtener el grado de

MAESTRO EN CIENCIAS

En

Sistemas Autónomos de Navegación Aérea y Submarina

Directores de la Tesis:

Dr. Sergio Rosario Salazar Cruz

Dr. Filiberto Muñoz Palacios

Ciudad de México

Febrero, 2019

AGRADECIMIENTOS

Quiero agradecer inicialmente al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN) por financiarme los estudios de maestría. Al Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos (LANAVEX) 295536, a los profesores que compartieron sus conocimientos en el campo de la ciencia al que me he dedicado durante este tiempo, a mis compañeros con los que he convivido en estos años. A mis tutores, a los doctores Sergio Rosario Salazar Cruz y Filiberto Muñoz Palacios por su guía y correcciones para lograr concluir este trabajo.

Especialmente a Dios, a mi madre Adelfa Cartas Toledo y a mi hermana Jedy Zárate Cartas por cuidarme y ser mis guías hasta este momento, también por apoyarme y aconsejarme en las decisiones que he tomado a cada momento de mi vida.

Por último a los compañeros que me ayudaron y se involucraron compartiéndome sus conocimientos en el desarrollo de este trabajo.

RESUMEN

En este trabajo se desarrolló una estrategia de control para el seguimiento de trayectorias de un vehículo aéreo no tripulado utilizando el método de Odometría Visual para la obtención de la posición en tiempo real de un cuadricóptero. En primera instancia se definen las características generales de una aeronave no tripulada y el concepto de odometría visual. Posteriormente se desarrolla el estado del arte de los principales métodos de odometría visual existentes. Se obtiene el modelo matemático del cuadricóptero utilizando el método de Euler-Lagrange y se desarrolló la estrategia de control PD utilizada para el seguimiento de trayectorias. Después se describen todos los componentes que fueron utilizados (frame del cuadricóptero, unidad de procesamiento gráfico, cámara estéreo, autopiloto, etc.) para el desarrollo de este trabajo. Finalmente se muestran las pruebas experimentales, las comparaciones realizadas entre el GPS y la cámara estéreo y todos los resultados obtenidos.

Palabras clave: Cuadricóptero, PD, Odometría Visual, Modelado, Unidad de procesamiento gráfico, Seguimiento de trayectorias.

ABSTRACT

In this work one control strategie was developed for a quadcopter to track trajectories using the visual odometry method to location in real time of a quadcopter. Initially the general characteristics of unmanned aircraft and the concept of visual odometry were defined. The main methods of visual odometry and the state of art about this topic were developed. The dynamic modeling of quadcopter was obtained using the Euler-Lagrange method and PD control strategy to tracking trajectories was developed. Subsequently were described all components (quadcopter frame, graphics processing unit, stereo camera, autopilot,etc.) used and their joining to reach the goal to fly using visual odometry. Finally all proofs and results were shown comparating the performance of GPS and the stereo camera.

Keywords: Quadcopter, PD, Visual Odometry, Dynamic Modeling, Tracking Trajectories.

ÍNDICE GENERAL

| | |
|--|------------|
| AGRADECIMIENTOS | III |
| RESUMEN | IV |
| ABSTRACT | V |
| ÍNDICE GENERAL | VII |
| 1. Introducción | 1 |
| 1.1. Planteamiento del problema | 3 |
| 1.2. Hipótesis | 3 |
| 1.3. Objetivos | 3 |
| 1.3.1. Objetivo general | 3 |
| 1.3.2. Objetivos particulares | 3 |
| 1.4. Organización de la tesis | 4 |
| 2. Estado del arte | 5 |
| 2.1. Odometría Visual | 5 |
| 2.1.1. Tipos de odometría visual | 7 |
| 2.1.2. Ventajas de la odometría visual | 8 |
| 2.1.3. Principio de funcionamiento | 8 |
| 2.2. PTAM | 13 |
| 2.3. DSO | 14 |

| | |
|--|-----------|
| 2.4. Odometría Visual Semi-directa (SVO) | 15 |
| 3. Modelo dinámico y estrategia de control | 17 |
| 3.1. Cinemática del cuadricóptero | 17 |
| 3.2. Dinámica del cuadricóptero | 21 |
| 3.3. Control del vehículo | 29 |
| 3.3.1. Controlador PD | 29 |
| 3.3.2. Diseño del controlador | 35 |
| 4. Desarrollo de la plataforma experimental | 39 |
| 4.1. Componentes mecánicos y electrónicos de la plataforma | 41 |
| 4.1.1. Cámara estéreo | 41 |
| 4.1.2. Unidad de Procesamiento Gráfico (GPU) | 42 |
| 4.1.3. Autopiloto | 43 |
| 4.1.4. Motores (brushless) | 44 |
| 4.1.5. Propelas | 45 |
| 4.1.6. Variador electrónico de velocidad (ESC) | 45 |
| 4.1.7. Frame o marco de montaje | 46 |
| 4.1.8. Batería LiPo | 47 |
| 4.1.9. Módulo convertidor de voltaje | 48 |
| 4.2. NVIDIA®Jetson™TX2 + HKPilot32 | 48 |
| 4.3. Instalación del software | 51 |
| 4.3.1. Descarga e instalación del SDK ZED™Stereo Camera | 51 |
| 4.3.2. SDK ZED™Stereo Camera + ROS | 52 |
| 5. Resultados | 55 |
| 5.1. Resultados en simulación | 55 |
| 5.1.1. Control en posición | 56 |
| 5.2. Comparación entre GPS vs odometría visual | 59 |
| 5.3. Resultados experimentales | 62 |

| | |
|---|-----------|
| <i>ÍNDICE GENERAL</i> | IX |
| 5.3.1. Experimento 1: <i>Control en altura y posición</i> | 62 |
| 5.3.2. Experimento 2: <i>Seguimiento de una línea</i> | 64 |
| 5.3.3. Experimento 3: <i>Seguimiento de una trayectoria</i> | 66 |
| 6. Conclusiones y trabajo futuro | 71 |
| 6.1. Conclusiones | 71 |
| 6.2. Trabajo futuro | 71 |
| A. Códigos | 73 |
| A.1. Lectura de la pose | 73 |
| A.2. Envío de datos al autopiloto | 75 |
| BIBLIOGRAFÍA | 77 |

CAPÍTULO 1

Introducción

Una pregunta básica que alguna vez todos nos hemos hecho es saber en qué punto del planeta nos encontramos (ubicación) o también cómo llegar de un punto dado a una referencia (dirección). Los seres humanos responden a esta pregunta de forma natural, ya que cuentan con sus sentidos o con la utilización de tecnologías como el GPS. Este proceso que es natural para los seres humanos no lo es para el caso de los robots o máquinas, la necesidad de que estas máquinas reconozcan el entorno y las condiciones en las que operan, con el objetivo de tener robots completamente autónomos, se ha vuelto todo un reto para los investigadores que se han involucrado en el tema. Con las tecnologías actuales es posible conocer la ubicación de cualquier máquina o persona en el mundo, la más conocida y usada es el GPS, pero esta tecnología presenta algunas desventajas:

- En cuanto a la localización de un objeto o persona presentan un margen de error que puede variar desde los centímetros hasta los 2 metros para los dispositivos comerciales.
- Bajo ciertas condiciones climáticas (lluvia, tormenta eléctrica) o condiciones adversas del entorno (lugares subterráneos, entornos con edificios altos, zonas boscosas) este sistema no funciona o funciona de forma incorrecta.
- El GPS no funciona bajo el agua, por lo que los vehículos submarinos quedarían descartados en cuanto a su capacidad de localización y/o navegación autónoma.

Por lo cual es necesario resolver el problema de la localización utilizando otras tecnologías diferentes a la del GPS, entre las cuales destaca la **visión por computadora**. La visión por computadora es una disciplina que tiene como objetivo permitir la toma de decisiones basada en la percepción de objetos físicos y escenas capturadas en imágenes digitales. Esta representa el entorno visible a través de un conjunto de imágenes al que aplica algoritmos para estimar propiedades tales como la forma, la iluminación y la distribución de colores en una escena [17].

Uno de los aspectos que permanecen sin resolver es la complejidad de la visión por computadora, puesto que calcular la posición y orientación de un robot es un problema al que se han enfrentado muchos investigadores, sobre todo si se hace la comparación con el desempeño, rapidez y resolución de la visión humana.

Sin embargo, una de las aplicaciones más importantes de la visión por computadora es la navegación de robots móviles, la cual le permite al robot o vehículo poder conocer su posición y orientación sin necesidad de contar con un GPS, para llevar a cabo esto hay que tomar en cuenta lo siguiente:

- Tener sensores (cámaras) con la mayor resolución posible, para obtener información del entorno lo más precisa posible.

- Tener una buena cantidad de imágenes (frames) por segundo.

- Contar con un sistema de cómputo lo suficientemente robusto para que el procesamiento de las imágenes sea en el menor tiempo posible.

En este capítulo se presenta la sección 1.1 en la cual se realiza el planteamiento del problema a resolver, en la sección 2.2 se define la hipótesis a probar, en la sección 1.3 los objetivos generales y particulares a cumplir y en la sección 1.4 se muestran los capítulos en los que está organizado este trabajo.

1.1. Planteamiento del problema

El problema se puede plantear de la siguiente manera: cualquier vehículo no tripulado multirrotor necesita conocer su posición y orientación para poder navegar de manera autónoma, para lograr esto se suele utilizar el GPS, pero en ciertos ambientes la señal del GPS tiene difícil o nulo acceso, lo cual hace necesario el uso de otras técnicas de localización.

1.2. Hipótesis

Un vehículo es capaz de navegar de manera autónoma en ambientes interiores y exteriores utilizando la información de una o más cámaras sujetas a él y examinando los cambios producidos sobre las imágenes debido al movimiento.

1.3. Objetivos

1.3.1. Objetivo general

Implementar un sistema de odometría visual en un vehículo aéreo no tripulado multirrotor con el objetivo de lograr su navegación autónoma en ambientes donde la señal del GPS tenga difícil o nulo acceso mediante el uso de algoritmos de visión por computadora y la utilización de una cámara estéreo.

1.3.2. Objetivos particulares

1. Diseño de una estrategia de control PD y ajuste de las ganancias mediante su simulación en MATLAB®.
2. Implementar un control PD en el vehículo aéreo no tripulado multirrotor para el seguimiento de trayectorias.
3. Implementar en el vehículo aéreo no tripulado multirrotor el algoritmo de odometría visual utilizando la cámara estéreo ZED™ cam.

4. Llevar a cabo pruebas experimentales para comparar el desempeño del GPS y el método de visión por computadora en diferentes entornos.
5. Realizar pruebas de navegación autónoma del vehículo aéreo no tripulado multirotor utilizando la visión por computadora.

1.4. Organización de la tesis

Este trabajo está organizado en 6 capítulos los cuales comprenden:

- **Capítulo 1.** Introducción de manera general del método de visión por computadora y los objetivos a conseguir con este trabajo.
- **Capítulo 2.** Estado del arte de los trabajos de visión por computadora desarrollados hasta el día de hoy y las aplicaciones realizadas con los mismos.
- **Capítulo 3.** Modelado dinámico del multirotor y desarrollo del control elegido para la navegación autónoma del vehículo.
- **Capítulo 4.** Desarrollo de la plataforma experimental, descripción de cada componente electrónico y mecánico utilizado e instalación de los programas utilizados.
- **Capítulo 5.** Resultados de las pruebas experimentales y la comparación hecha entre el desempeño del GPS y la visión por computadora en diferentes entornos.
- **Capítulo 6.** Conclusiones y trabajo a futuro.

CAPÍTULO 2

Estado del arte

La odometría es el estudio de la estimación de la posición de vehículos (terrestres, aéreos o submarinos) durante la navegación. Este término también se usa a veces para referirse a la distancia que ha recorrido uno de estos vehículos.

En este capítulo se presenta la sección 2.1 en la que se define el concepto de odometría visual, sus tipos, ventajas y principio de funcionamiento; en la sección 2.2 se presenta el algoritmo PTAM, que fue uno de los primeros en ser desarrollados para la odometría visual, en la sección 2.3 se describen las características del algoritmo de *Odometría Directa Dispersa* y en la sección 2.4 se presenta el algoritmo de *Odometría Visual Semi-Directa*, su descripción, antecedentes y ventajas.

2.1. Odometría Visual

La odometría visual se refiere al proceso de estimar un movimiento o desplazamiento a partir de una secuencia de imágenes capturadas por una o varias cámaras montadas en un vehículo cualquiera. Los orígenes de la odometría visual se remontan a los años 80 con la tesis de Moravec [20], sin embargo ha sido hasta la última década en la que se han desarrollado algoritmos que funcionan en tiempo real de manera adecuada.

El problema particular de estimar el movimiento de un vehículo (*ego-motion*) a partir

de imágenes únicamente fue descrito en 1987 [18] por Moravec, Matthies y Shafer, basados en el trabajo de Moravec usaron un sistema binocular logrando un error relativo del 2% en un recorrido de 6 metros para un *rover* planetario. En el 2004 [22] Edwin Olson extendió dichos trabajos agregando un sensor de orientación. Este último demostró que el error en la localización crece con la distancia recorrida y agregando dicho sensor logró reducir el error de la posición relativa al 1.2% en un recorrido de 20 metros. En el 2006 [19] Cheng, Milella y Siegwart lograron un desempeño superior mejorando los algoritmos de detección de bordes, estos últimos usaron un escáner láser para refinar la posición relativa.

Los últimos desarrollos sobre odometría visual se han obtenido gracias al desarrollo y construcción de robots de exploración planetaria. Estos robots, específicamente del programa de exploración de Marte desarrollado por la NASA, han contribuido a este desarrollo con la búsqueda de un mecanismo de estimación del movimiento en 6 grados de libertad, en condiciones de terrenos inestables y de deslizamiento de ruedas, situación en la cual la odometría tradicional falla. A estos avances ha contribuido también una carrera de vehículos autónomos conocida como el *DARPA Grand Challenge*¹. En la Figura 2.1 se muestra un vehículo con todos los sistemas y sensores para su navegación autónoma usado en el *DARPA Urban Challenge*.

¹En esta carrera los vehículos deben llegar desde un punto de los Estados Unidos hasta otro sin intervención humana y disponiendo únicamente de un listado de puntos intermedios entre el principio del circuito y el final. *DARPA* es una organización de investigación del *Departamento de Defensa de los Estados Unidos*.



Figura 2.1: Vehículo terrestre con todos los sistemas y sensores utilizados para su navegación autónoma [31].

2.1.1. Tipos de odometría visual

Según el número de cámaras existen principalmente:

- **Odometría monocular:** Es la visión que utiliza una sola cámara, su principal característica es que se incrementa el *campo visual* y la *percepción de profundidad* se restringe, es decir, utilizar una sola cámara aumenta la complejidad del problema al no poder inferir la escala real de las imágenes en la escena observada. Por lo cual para poder determinar la distancia a la que se encuentran los objetos se tienen que obtener dos o más observaciones de una misma cámara en distintos instantes observando el mismo punto del entorno. Para obtener una escala real de la escena observada será necesario utilizar otros sensores de referencia.
- **Odometría estereoscópica:** Esta visión es la que integra una imagen tomada de forma simultánea desde dos posiciones equidistantes entre sí. Al tener dos perspectivas distintas de una misma imagen se puede obtener la distancia (profundidad), mediante

geometría epipolar, a la que se encuentra el objeto visto en 3D.

2.1.2. Ventajas de la odometría visual

- La odometría visual no es afectada por el deslizamiento de la rueda en un terreno desigual, el diámetro de estas, su mala alineación u otras condiciones adversas.
- Las trayectorias estimadas son más precisas con respecto a la odometría tradicional.
- Puede usarse complementariamente con otras tecnologías como GPS, Unidades de Medición Inercial (IMU en inglés), odometría por láser, entre otras.
- Funciona en ambientes donde la señal del GPS es de difícil o nulo acceso.

2.1.3. Principio de funcionamiento

En cuanto a su funcionamiento la odometría visual se divide en 5 etapas:

1. **Secuencia de imágenes (Image sequence):** En esta etapa las cámaras son calibradas (obtención de sus parámetros intrínsecos y extrínsecos) y se capturan las imágenes con las cuales se realizará la estimación del movimiento, esta secuencia de imágenes se obtiene a una frecuencia dada (para este trabajo de tesis las imágenes fueron obtenidas con una frecuencia de 10 imágenes por segundo).
2. **Detección y extracción de características (Feature detection):** Esto consiste en detectar puntos concretos en una imagen que puedan localizarse en imágenes posteriores, los tipos de características son:
 - Esquinas.
 - Regiones, manchas o gotas.
 - Bordes.

En la Figura 2.2 se muestra una imagen utilizando el detector de características SIFT.



Figura 2.2: Detección de características con el algoritmo SIFT.

Los algoritmos para la detección de este tipo de características deben ser eficientes computacionalmente, robustos e invariantes ante rotaciones o cambios de perspectivas dentro de la imagen. En la Figura 2.3 se muestra una tabla comparativa entre distintos detectores, sus propiedades y desempeño computacional.

| | Detector de esquinas | Detector de manchas | Invariante a rotaciones | Invariante a escalamientos | Invariante afin | Repetibilidad | Exactitud en localización | Robustez | Eficiencia |
|------------|----------------------|---------------------|-------------------------|----------------------------|-----------------|---------------|---------------------------|----------|------------|
| Harris | x | | x | | | +++ | +++ | ++ | ++ |
| Shi-Tomasi | x | | x | | | +++ | +++ | ++ | ++ |
| FAST | x | | x | x | | ++ | ++ | ++ | ++++ |
| SIFT | | x | x | x | x | +++ | ++ | +++ | + |
| SURF | | x | x | x | x | +++ | ++ | ++ | ++ |
| CENSURE | | x | x | x | x | +++ | ++ | +++ | +++ |

Figura 2.3: Comparación entre algunos detectores de características: *propiedades y desempeño* [4].

3. **Emparejamiento de características (Feature Matching):** Para obtener tal emparejamiento es necesario que tanto la imagen I_{k-1} y la imagen I_k tengan similitudes entre ambas que puedan ser medidas, es decir, que las imágenes I_{k-1} e I_k tengan características similares. En la Figura 2.4 se muestra un ejemplo de emparejamiento entre dos imágenes con características similares. Entre las medidas de similitud se encuentran SSD (*Sum of Square Differences*), NCC (*Normal Cross Correlation*), NNDR (*Nearest Neighbor Distance Radio*). Entre las diferentes metodologías para el emparejamiento de características se encuentran:

- Emparejamiento NO restringido (*Unconstrained Matching*).
- Emparejamiento restringido (*Constrained Matching*).



Figura 2.4: Emparejamiento de características entre imágenes capturadas con 100 *ms* de diferencia.

4. **Estimación del movimiento:** En esta etapa se estima el movimiento tanto traslacional como rotacional a partir de las características comunes f_{k-1} y f_k que han sido extraídas y emparejadas entre las dos imágenes I_{k-1} e I_k respectivamente. Existen 3 metodologías distintas para estimar la estimación del movimiento:

- **3D-3D:** En este caso tanto f_{k-1} y f_k están en 3D. Para llevar a cabo la estimación del movimiento es necesario triangular los puntos en 3D en cada instante de tiempo, utilizando, por ejemplo, una cámara estéreo.
- **2D-2D:** En este caso tanto f_{k-1} y f_k están en coordenadas 2D de la imagen.
- **3D-2D:** En este caso f_{k-1} está en 3D y f_k es su correspondiente reproyección en 2D en la imagen I_k .

Esta técnica requiere de ir tomando imágenes del entorno con una cámara en instantes discretos k . Las poses relativas denotadas como T_k y T_{k-1} de cámaras en posiciones adyacentes son calculadas a partir de características visuales y concatenadas para obtener las poses absolutas C_k con respecto al marco de referencia inicial en $k = 0$.

El conjunto de imágenes tomadas a una frecuencia dada se denota como:

$$I_{0:n} = \{I_{l,0}, \dots, I_{l,n}\}.$$

Considerando dos posiciones (*poses*) consecutivas en instantes de tiempo $k - 1$ y k respectivamente, estas se relacionan por las transformaciones de cuerpo rígido dada por la matriz:

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix},$$

donde $R_{k,k-1}$ representa la *matriz de rotación* y $t_{k,k-1}$ el *vector de traslación*. Finalmente el conjunto de las *poses* de la cámara $C_{0:n} = \{C_0, \dots, C_n\}$ contiene las transformaciones de la cámara con respecto a la coordenada en el cuadro inicial $k = 0$ (como se muestra en la Figura 2.5), por lo tanto, la *pose* más reciente (C_n) se calcula concatenando todas las transformaciones T_k , $k = \{1 \dots n\}$:

$$C_n = C_{n-1}T_n.$$

La odometría visual consiste en calcular las transformaciones relativas T_k a partir de las imágenes I_k e I_{k-1} e ir concatenando tales transformaciones para obtener, a partir de ello, la trayectoria total $C_{0:n}$ de la cámara.

Bosquejo del algoritmo

- a) Inicio del ciclo.
 - b) Captura de las imágenes: I_{k-1} , I_k .
 - c) Eliminación de la distorsión de las imágenes anteriores.
 - d) Uso de algún algoritmo para detectar las características en la imagen I_{k-1} y rastrear esas características en la imagen I_k . Una nueva detección se dispara si el número de características caen por debajo de un cierto umbral.
 - e) Cálculo de la matriz esencial.
 - f) Estimación de la matriz de rotación (R) y del vector de traslación (t) a partir de la matriz esencial calculada en el paso anterior.
 - g) Uso de una escala de información de alguna fuente externa (como un velocímetro) y concatenarla con los vectores de traslación y las matrices de rotación.
 - h) Concatenación de la estimación de la trayectoria de la cámara.
 - i) Fin del loop.
5. **Optimización local:** El principal problema de la odometría visual es el de la *deriva*. Cada transformación $T_{k,k-1}$ está concatenada con la *pose* del vehículo C_k , cada una de estas transformaciones $T_{k,k-1}$ tiene una incertidumbre y la incertidumbre de la pose de la cámara C_k depende de la incertidumbre de las transformaciones, a la acumulación de estos errores se le conoce como *error de propagación*, se puede deducir que dado este error la trayectoria estimada se desvía completamente de la trayectoria real conforme el tiempo crece. Para que este error converja a un valor lo más cercano al cero es necesario encontrar una solución factible que sea un óptimo local si es que no hubiera

otras soluciones con una mejor función de valor alrededor. Este óptimo puede ser un mínimo o máximo local [8]. Cada *pose* de la cámara C_k y cada transformación $T_{k,k-1}$ puede ser representado por un vector de seis elementos que contiene la posición (x, y, z) y la orientación (en ángulos de Euler ϕ, θ, ψ). Este vector de seis elementos se denota como $\vec{C}_k = (x, y, z, \phi, \theta, \psi)^T$ [5].

En la Figura 2.5 se ilustran los errores introducidos en la estimación del desplazamiento de un vehículo por medio de imágenes capturadas con una cámara sujeta al mismo.

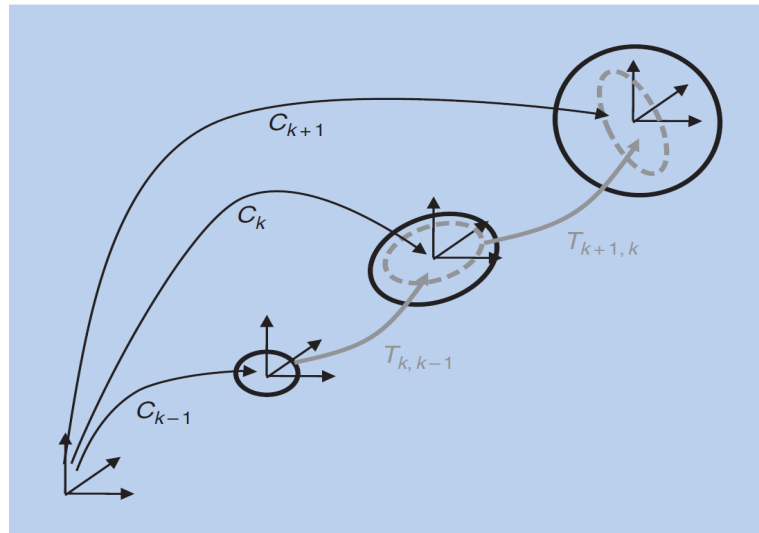


Figura 2.5: La incertidumbre de la *pose* de la cámara C_k es una combinación de la incertidumbre en C_{k-1} y la incertidumbre de la transformación $T_{k,k-1}$ [6].

2.2. PTAM

PTAM significa *Parallel Tracking and Mapping* y es una forma de implementación del algoritmo *Monocular SLAM*, el PTAM fue desarrollado en el 2007 [16] por Klein y Murray con el objetivo de resolver el mismo problema que MonoSLAM desde una estrategia completamente diferente.

Este algoritmo estima la posición de una cámara en un entorno 3D y mapea las posiciones de los puntos en objetos en el espacio analizando y procesando la entrada de la cámara en

tiempo real. El algoritmo se divide en dos partes principales, el *Seguimiento* de la cámara y el *Mapeo* de los puntos. Además que estos procesos corren en hilos paralelos (de forma asíncrona); para ello sólo es necesario que el *Seguimiento* funcione en tiempo real en tanto que no es necesario que el *Mapeo* se lleve a cabo en cada iteración.

2.3. DSO

DSO [2] se refiere a *Direct Sparse Odometry* y la formulación propuesta conjunta los métodos *Disperso + Directo*, esto con el objetivo de optimizar el error fotométrico definido directamente en las imágenes sin incorporar un antecedente geométrico. Entre las consideraciones que motivaron el desarrollo de este algoritmo se encuentran las siguientes:

- **Directo:** Uno de los principales beneficios de una *formulación directa* es que esta no requiere un punto para ser reconocible por sí mismo, permitiendo así una representación geométrica granulada más fina (profundidad inversa de pixel). Más aún, se puede verificar que todos los datos disponibles -incluyendo bordes y variaciones débiles de intensidad- generan un modelo completo y brindan más solidez en entornos pobres en texturas.
- **Disperso:** El principal inconveniente de agregar una geometría previa es la introducción de correlaciones entre los parámetros geométricos que ofrecen una optimización conjunta estadísticamente consistente no factible en tiempo real.

Basados en las consideraciones anteriores, la odometría directa dispersa se basa en una optimización continua del error fotométrico sobre una ventana de los cuadros más recientes, tomando en cuenta un modelo fotométricamente calibrado. A diferencia de los métodos directos existentes, todos los parámetros involucrados (intrínsecos, extrínsecos y valores de profundidad inversa) son optimizados en conjunto.

2.4. Odometría Visual Semi-directa (SVO)

Este algoritmo combina las ventajas de los métodos directos y basados en características [1]. Este utiliza un algoritmo de *alineación de imagen dispersa*, la cual es una aproximación directa eficiente para estimar el movimiento de una imagen a otra minimizando los errores fotométricos de las características que están en la intensidad de las esquinas y bordes. Los puntos 3D correspondientes a las características son obtenidos por medio de una estimación profunda robusta recursiva Bayesiana. Una vez que se establece una correspondencia es utilizado el *ajuste de puntos* para el refinamiento de la estructura y las poses de la cámara para obtener una alta exactitud.

La implementación propuesta requiere de sólo 2.5 ms para estimar la pose de una figura en una laptop estándar. La eficiencia del algoritmo mejora debido a 3 razones:

1. SVO extrae características sólo para las imágenes clave seleccionadas en un hilo paralelo, por lo tanto se desacopla de las fuertes restricciones del tiempo real.
2. El algoritmo de seguimiento directo propuesto elimina la necesidad de una asociación robusta de datos.
3. Contrario a los métodos directos previos, SVO requiere sólo una escasa reconstrucción del entorno.

Los métodos que de forma simultánea recuperan la pose de la cámara y la estructura de la escena puede dividirse en dos clases:

- **Basado en características:** La forma estándar de resolver es extraer un conjunto de características destacadas (esquinas, manchas, etc.) en cada imagen; emparejándolas en imágenes sucesivas usando descriptores de características invariantes; recuperadas de forma robusta tanto para el movimiento de la cámara y la estructura usando geometría epipolar; y finalmente, refinando la pose y la estructura a través de la reproyección de la minimización del error. La mayoría de los algoritmos de VO y V-SLAM utilizan una variante de estos procedimientos.

La desventaja de las aproximaciones *basadas en características* es su baja velocidad debido a la extracción de características y emparejamiento para cada imagen.

Para el caso de SVO la extracción de las características se realiza sólo en las imágenes clave seleccionadas, lo que reduce el costo computacional de forma significativa. una vez extraídas las características, se utiliza un método *directo* para el seguimiento de las mismas entre imágenes consecutivas con una precisión de sub-píxeles.

- **Métodos directos:** Los *métodos directos* estiman estructura y movimiento minimizando la medición del error basándose en los niveles de intensidad de los píxeles. La correspondencia entre píxeles es dada directamente por la geometría del problema, eliminando la necesidad de técnicas de asociaciones robustas de datos.

Usando una aproximación directa, los 6 DoF de la cámara puede recuperarse por medio de una *alineación cámara-modelo*, que es el proceso de alinear la imagen observada a una vista sintetizada del mapa 3D estimado. Los primeros métodos de OV seguían y mapeaban unas pocas manchas. Con el DTAM² se introdujo un método directo que calcula un *mapa de profundidad* denso de una sola cámara moviéndose en tiempo real. La pose de la cámara es hallada a través de la alineación directa de toda la imagen usando el mapa de profundidad. Sin embargo, inferir un mapa de profundidad denso a partir de imágenes monoculares es computacionalmente intensivo y típicamente se lleva a cabo con el paralelismo de una GPU³.

²*Dense Tracking and Mapping in Real-Time*

³*Unidad de Procesamiento Gráfico*

CAPÍTULO 3

Modelo dinámico y estrategia de control

Para el desarrollo del modelado dinámico del cuadricóptero es importante tomar en cuenta las siguientes consideraciones:

- Los modelos dinámicos y cinemáticos estarán basados en el método de Euler-Lagrange.
- Se considera una estructura rígida y simétrica.
- El centro de gravedad del cuadricóptero está en el centro del cuerpo rígido.
- Las propelas son rígidas.

En este capítulo se presenta la sección 3.1 donde se modela el cuadricóptero considerando su cinemática, en la sección 3.2 se modela considerando su dinámica y en la sección 3.3 se presenta la estrategia de control utilizada para la navegación autónoma del vehículo.

3.1. Cinemática del cuadricóptero

Para el modelado del cuadricóptero lo primero a considerar son todos sus movimientos posibles o grados de libertad, que para este vehículo en particular son 6. Los primeros 3 grados de libertad son los desplazamientos lineales, aquí se consideran todos los movimientos traslacionales; los otros 3 grados corresponden a los movimientos rotacionales del vehículo.

Para el desarrollo del modelo dinámico del vehículo, se definen dos marcos de referencia:

- **Inercial:** Con referencia fija a la Tierra, $\mathbf{I} = \{X_I, Y_I, Z_I\}$.
- **No inercial:** Con referencia fija al centro de masa del cuerpo, $\mathbf{B} = \{X_B, Y_B, Z_B\}$.

En la Figura 3.1 se ilustran los marcos de referencia, grados de libertad, momentos y fuerzas actuantes en el cuadricóptero.

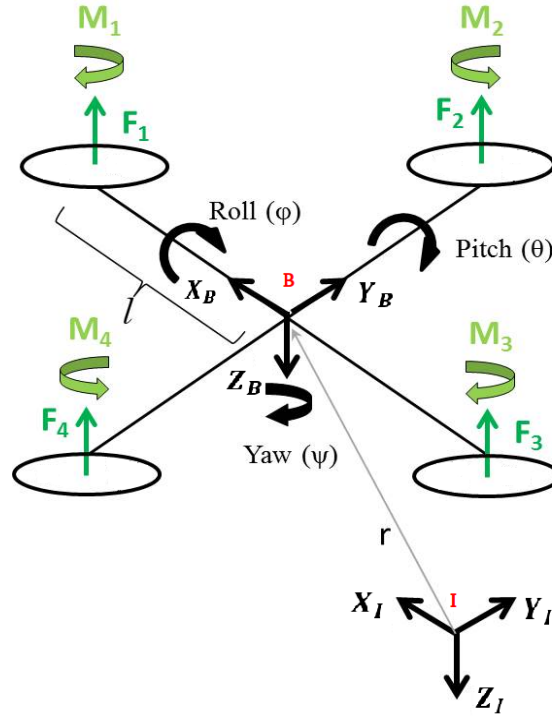


Figura 3.1: Fuerzas, Momentos y Ejes de rotación de un cuadricóptero [29].

Las posiciones traslacionales y rotacionales del cuadricóptero con respecto al marco inercial y no inercial respectivamente están definidas como:

$$\xi = [x, y, z]^T \in \mathbb{R}^3, \quad \eta = [\phi, \theta, \psi]^T \in \mathbb{R}^3, \quad (3.1)$$

La posición del vehículo denotada por (x, y, z) representa la posición del vehículo con respecto a los ejes X_I, Y_I, Z_I y los ángulos de roll o alabeo (ϕ), pitch o cabeceo (θ) y yaw o guiñada (ψ) representan las rotaciones del vehículo alrededor de los ejes X_B, Y_B, Z_B .

De igual forma, las velocidades traslacionales y rotacionales que experimenta el cuadricóptero están dadas por:

$$V = [u, v, w]^T \in \mathbb{R}^3, \quad \omega = [p, q, r]^T \in \mathbb{R}^3. \quad (3.2)$$

En primera instancia hay que describir la orientación total del cuadricóptero, es decir, obtener las coordenadas de un sistema móvil \mathbf{B} con respecto a un sistema de coordenadas fijo \mathbf{I} , y para ello se utilizan los ángulos de **Tait-Bryan**, estos tres ángulos describen la rotación general de un sistema móvil en el espacio euclidiano 3D a través de tres rotaciones sucesivas sobre cada uno de los ejes de referencia \mathbf{B} .

Las tres rotaciones sucesivas están dadas de la siguiente manera:

1. *Rotación alrededor del eje x*: El primer giro es el correspondiente al ángulo ϕ o de *roll* y se denota como:

$$\mathbf{R}(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\text{sen } \phi \\ 0 & \text{sen } \phi & \cos \phi \end{bmatrix}.$$

2. *Rotación alrededor del eje y*: El segundo giro es el correspondiente al ángulo θ o de *pitch*, pero este se realiza a partir del nuevo eje y_B para dejar al eje z_B en su posición final, y se denota como:

$$\mathbf{R}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \text{sen } \theta \\ 0 & 1 & 0 \\ -\text{sen } \theta & 0 & \cos \theta \end{bmatrix}.$$

3. *Rotación alrededor del eje z*: El tercer y último giro es el correspondiente al ángulo ψ o de *yaw*, similar al giro anterior, este se realiza a partir del nuevo eje z_B para llevar el móvil (cuadricóptero) a su posición final:

$$\mathbf{R}(z, \psi) = \begin{bmatrix} \cos \psi & -\text{sen } \psi & 0 \\ \text{sen } \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Con las rotaciones anteriores se define la matriz de rotación \mathbf{R}_I , la cual nos permite transformar las coordenadas expresadas en el marco no inercial al marco inercial, obteniéndose:

$$\mathbf{R}_I = \mathbf{R}(z, \psi) \cdot \mathbf{R}(y, \theta) \cdot \mathbf{R}(x, \phi) \quad (3.3)$$

$$\mathbf{R}_I = \begin{bmatrix} \cos \psi & -\text{sen } \psi & 0 \\ \text{sen } \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\text{sen } \theta \\ 0 & 1 & 0 \\ -\text{sen } \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\text{sen } \phi \\ 0 & \text{sen } \phi & \cos \phi \end{bmatrix}$$

$$\mathbf{R}_I = \begin{bmatrix} C\theta C\psi & C\psi S\theta S\phi - C\phi S\psi & S\phi S\psi + C\phi C\psi S\theta \\ C\theta S\psi & C\phi C\psi + S\theta S\phi S\psi & C\phi S\theta S\psi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix},$$

donde $S = \text{sen}$ y $C = \text{cos}$.

Con respecto al marco \mathbf{B} la matriz de rotación es la transpuesta de la matriz \mathbf{R}_I , esto es debido a su propiedad de ortonormalidad.

$$\mathbf{R}_B = \begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ C\psi S\theta S\phi - C\phi S\psi & C\phi C\psi + S\theta S\phi S\psi & C\theta S\phi \\ S\phi S\psi + C\phi C\psi S\theta & C\phi S\theta S\psi - C\psi S\phi & C\theta C\phi \end{bmatrix}.$$

Utilizando la matriz de rotación (3.3), la velocidad del vehículo expresada en el marco no inercial \mathbf{B} puede ser relacionada con la velocidad lineal expresada en el marco inercial \mathbf{I} mediante (3.4).

$$\dot{\xi} = J_1(\eta)V, \quad (3.4)$$

donde $J_1(\eta) = R_I$, la cual puede ser expresada de forma matricial como:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} C\theta C\psi & C\psi S\theta S\phi - C\phi S\psi & S\phi S\psi + C\phi C\psi S\theta \\ C\theta S\psi & C\phi C\psi + S\theta S\phi S\psi & C\phi S\theta S\psi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}.$$

La velocidad angular del vehículo en el marco \mathbf{B} y la velocidad angular en el marco \mathbf{I} se relacionan a partir de la matriz de rotación (3.3) y su propiedad de ortonormalidad [10]. Relacionando la derivada de la matriz ortonormal con una matriz anti-simétrica, se pueden obtener las ecuaciones cinemáticas de rotación del vehículo que establecen las relaciones entre las velocidades angulares definidas en el marco no inercial y el marco inercial, quedando de la siguiente forma:

$$\dot{\eta} = J_2(\eta)\omega, \quad (3.5)$$

$$\text{donde } J_2(\eta) = \begin{bmatrix} 1 & \sin \psi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix}.$$

De forma general la ecuación cinemática del vehículo está dada por:

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} J_1(\eta) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta) \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix}. \quad (3.6)$$

3.2. Dinámica del cuadricóptero

En el presente trabajo de tesis, el modelo dinámico del cuadricóptero se obtuvo utilizando la metodología de **Euler-Lagrange**. El vector de coordenadas generalizadas del vehículo, denotado por q , está dado por $q = [x, y, z, \theta, \phi, \psi]^T \in \mathbb{R}^6$, el cual contiene las posiciones lineales y angulares del vehículo. De manera similar el vector $\nu = [u, v, \omega, p, q, r]^T \in \mathbb{R}^6$ contiene las velocidades lineales y angulares del vehículo.

La ecuación de Euler-Lagrange está basada en el concepto de *energía cinética* y *energía potencial*, y se encuentra definida como:

$$\tau = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i}, \quad (3.7)$$

donde:

\mathcal{L} = Es el lagrangiano y se define como: $\mathcal{L} = T - V$

T = Energía cinética total.

V = Energía potencial total.

q_i = Vector de coordenadas generalizadas del cuadricóptero.

\dot{q}_i = La primera derivada con respecto al tiempo de q_i .

Debido a que la dinámica estudia el movimiento y las causas que lo producen, en el cuadricóptero tenemos entonces dos tipos de dinámicas a analizar: **Dinámica traslacional** y **Dinámica rotacional**.

Dinámica Traslacional

El lagrangiano traslacional se define como:

$$\mathcal{L}_{tras} = T_{tras} - V_{tras}, \quad (3.8)$$

donde la energía cinética traslacional del cuadricóptero está dada por:

$$T_{tras} = \frac{m}{2} \dot{\xi}^T \dot{\xi}, \quad (3.9)$$

y energía potencial del cuadricóptero está dada por:

$$V_{tras} = mgz, \quad (3.10)$$

donde g es la aceleración de la gravedad, z es la altura del alcanzada por el cuadricóptero y m es su masa.

Finalmente, el lagrangiano traslacional del cuadricóptero se obtiene al sustituir (3.9) y (3.10) en (3.8):

$$\mathcal{L}_{tras} = \frac{m}{2} \dot{\xi}^T \dot{\xi} - mgz. \quad (3.11)$$

La ecuación de Euler-Lagrange para la dinámica traslacional del vehículo se encuentra dada por:

$$\tau_\xi = \frac{d}{dt} \left(\frac{\partial L_{tras}}{\partial \dot{\xi}} \right) - \frac{\partial L_{tras}}{\partial \xi}, \quad (3.12)$$

donde τ_ξ representa el vector de fuerzas aplicadas al cuerpo y está definido en el marco inercial y cada término de (3.12) está dado por:

$$\frac{\partial L_{tras}}{\partial \xi} = -mgz,$$

$$\frac{\partial L_{tras}}{\partial \dot{\xi}} = m\dot{\xi},$$

$$\frac{d}{dt} \left(\frac{\partial L_{tras}}{\partial \dot{\xi}} \right) = m\ddot{\xi},$$

obteniéndose finalmente:

$$m\ddot{\xi} - mgz = \tau_\xi, \quad (3.13)$$

donde τ_ξ está relacionado con el vector de fuerzas F expresado en el cuerpo:

$$\tau_\xi = R_I F \in \mathbb{R}^3, \quad (3.14)$$

con F definido como:

$$F = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix},$$

donde:

$$U_1 = F_1 + F_2 + F_3 + F_4, \quad (3.15)$$

U_1 representa la fuerza de empuje resultante de los cuatro rotores que es la responsable de la altitud del cuadricóptero y la velocidad con que este asciende (z, \dot{z}) . Se considera negativa porque a pesar de ser paralela al eje z el empuje es contrario al sentido positivo mostrado en la Figura 3.1.

Sustituyendo la matriz de rotación R_I en (3.14) se obtiene:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} - mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} C\theta C\psi & C\psi S\theta S\phi - C\phi S\psi & S\phi S\psi + C\phi C\psi S\theta \\ C\theta S\psi & C\phi C\psi + S\theta S\phi S\psi & C\phi S\theta S\psi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix}. \quad (3.16)$$

Finalmente la **dinámica traslacional** queda expresada como:

$$\begin{aligned} \ddot{x} &= -\frac{U_1}{m}(\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \\ \ddot{y} &= -\frac{U_1}{m}(\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi) \\ \ddot{z} &= -\frac{U_1}{m}(\cos \phi \cos \theta) + g \end{aligned} \quad (3.17)$$

Dinámica Rotacional

El lagrangiano rotacional se define como:

$$\mathcal{L}_{rot} = \frac{1}{2} \dot{\eta}^T (J(\eta)) \dot{\eta}, \quad (3.18)$$

donde $\eta = [\theta, \phi, \psi]^T$ y $J(\eta)$ es el Jacobiano definido en coordenadas del marco de referencia inercial y se define como:

$$J(\eta) = W_\eta^T I W_\eta, \quad (3.19)$$

y W_η es el Jacobiano del marco de referencia no inercial que relaciona la velocidad angular ω con $\dot{\eta}$ y está definido como:

$$W_\eta = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix}, \quad (3.20)$$

dado que la estructura del cuadricóptero utilizado en este caso es simétrica la matriz de inercias queda definida como:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \quad (3.21)$$

donde I_{xx} , I_{yy} y I_{zz} son los momentos de inercia y representan una medida de la inercia rotacional del cuadricóptero sobre cada uno de sus ejes principales.

Sustituyendo (3.20) y (3.21) en (3.19) se obtiene:

$$J(\eta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix}, \quad (3.22)$$

$$J(\eta) = \begin{bmatrix} I_{xx} & 0 & -I_{xx}S\theta \\ 0 & I_{yy}C^2\phi + I_{zz}S^2\phi & (I_{yy} - I_{zz})C\phi S\phi C\theta \\ -I_{xx}S\theta & (I_{yy} - I_{zz})C\phi S\phi C\theta & I_{xx}S^2\theta + I_{yy}S^2\phi C^2\theta + I_{zz}C^2\phi C^2\theta \end{bmatrix}. \quad (3.23)$$

Para definir el vector de entrada de la dinámica rotacional definimos como l la distancia del centroide a las propelas y el torque producido por cada motor lo definimos como τ_{mi} ($i = 1,2,3,4$), donde l y τ_{mi} permiten a su vez a definir los torques de roll (τ_θ), pitch (τ_ϕ) y yaw (τ_ψ):

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix}. \quad (3.24)$$

Como se muestra en (3.24) los torques τ_ϕ y τ_θ dependen de las fuerzas lineales de empuje, en tanto τ_ψ depende del momento producido por cada motor. U_2 es la diferencia de empuje entre los rotores 2 y 4 que es responsable de la rotación en *roll* y su velocidad de cambio

$(\phi, \dot{\phi})$. U_3 representa la diferencia de empuje entre los rotores 1 y 3 que es responsable de la rotación en *pitch* y su velocidad de cambio $(\theta, \dot{\theta})$. Finalmente U_4 representa la diferencia de los torques en sentido horario y en sentido anti-horario que generan la rotación en *yaw* y su velocidad de cambio $(\psi, \dot{\psi})$.

La ecuación de Euler-Lagrange para la dinámica de orientación del vehículo se encuentra dada por:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}_{rot}}{\partial \dot{\eta}} \right) - \frac{\partial \mathcal{L}_{rot}}{\partial \eta} = \tau_{\eta}, \quad (3.25)$$

donde:

$$\frac{\partial \mathcal{L}_{rot}}{\partial \dot{\eta}} = \frac{1}{2} [J(\eta) + J(\eta)^T] \dot{\eta} = J(\eta) \dot{\eta}, \quad (3.26)$$

y al sustituir (3.26) en (3.25) se obtiene:

$$J(\eta) \ddot{\eta} + \dot{J}(\eta) \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} \dot{\eta}^T J(\eta) \dot{\eta} = \tau_{\eta}, \quad (3.27)$$

Definiendo la matriz de Coriolis como:

$$C = \dot{J}(\eta) - \frac{1}{2} \dot{\eta}^T J(\eta), \quad (3.28)$$

la cual define los efectos giroscópicos y centrífugos del sistema, la dinámica de orientación del vehículo se puede reescribir de forma general como:

$$J(\eta) \ddot{\eta} + C(\eta, \dot{\eta}) \dot{\eta} = \tau_{\eta}, \quad (3.29)$$

la matriz de Coriolis $C(\eta, \dot{\eta})$ tiene la estructura

$$C(\eta, \dot{\eta}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}, \quad (3.30)$$

donde:

$$\begin{aligned}
C_{11} &= 0 \\
C_{12} &= (I_{yy} - I_{zz})(\dot{\theta}C\phi S\phi + \dot{\psi}S^2\phi C\theta) + (I_{zz} - I_{yy})\dot{\psi}C^2\phi C\theta - I_{xx}\dot{\psi}C\theta \\
C_{13} &= (I_{zz} - I_{yy})\dot{\psi}C\phi S\phi C^2\theta \\
C_{21} &= (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi + \dot{\psi}S\phi C\theta) + (I_{yy} - I_{zz})\dot{\psi}C^2\phi C\theta + I_{xx}\dot{\psi}C\theta \\
C_{22} &= (I_{zz} - I_{yy})\dot{\phi}C\phi S\phi \\
C_{23} &= -I_{xx}\dot{\psi}S\theta C\theta + I_{yy}\dot{\psi}S^2\phi C\theta S\theta + I_{zz}\dot{\psi}C^2\phi S\theta C\theta \\
C_{31} &= (I_{yy} - I_{zz})\dot{\psi}C\phi S\phi C^2\theta - I_{xx}\dot{\theta}C\theta \\
C_{32} &= (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi S\theta + \dot{\phi}S^2\phi C\phi) + (I_{yy} - I_{zz})\dot{\phi}C^2\phi C\theta + I_{xx}\dot{\psi}C\theta S\theta \\
&\quad - I_{yy}\dot{\psi}S^2\phi C\theta S\theta - I_{zz}\dot{\psi}C^2\phi C\theta S\theta \\
C_{33} &= (I_{yy} - I_{zz})\dot{\phi}C\phi S\phi C^2\theta - I_{yy}\dot{\theta}S^2\phi C\theta S\theta - I_{zz}\dot{\theta}C^2\phi C\theta S\theta + I_{xx}\dot{\theta}C\theta S\theta
\end{aligned}$$

Las ecuaciones del movimiento rotacional se pueden representar con la siguiente expresión matricial:

$$\ddot{\eta} = J(\eta)^{-1}(\tau_{\eta} - C(\eta, \dot{\eta})\dot{\eta}). \quad (3.31)$$

En este trabajo de tesis se realiza la consideración de que el vehículo no va a realizar maniobras agresivas, por lo cual se tiene lo siguiente:

$$\begin{aligned}
\text{sen } \phi &\approx \text{sen } \theta \approx \text{sen } \psi \approx 0 \\
\text{cos } \phi &\approx \text{cos } \theta \approx \text{cos } \psi \approx 1
\end{aligned} \quad (3.32)$$

obteniéndose para $C(\eta, \dot{\eta})$ y $J(\eta)$ lo siguiente:

$$C(\eta, \dot{\eta}) = \begin{bmatrix} 0 & (I_{zz} - I_{yy})\dot{\psi} - I_{xx}\dot{\psi} & 0 \\ (I_{yy} - I_{zz})\dot{\psi} + I_{xx}\dot{\psi} & 0 & 0 \\ -I_{xx}\dot{\theta} & (I_{yy} - I_{zz})\dot{\phi} & 0 \end{bmatrix}, \quad (3.33)$$

$$J(\eta) = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \quad (3.34)$$

Sustituyendo (3.33) y (3.34) en (3.29) se obtiene:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \ddot{\eta} + \begin{bmatrix} 0 & (I_{zz} - I_{yy})\dot{\psi} - I_{xx}\dot{\psi} & 0 \\ (I_{yy} - I_{zz})\dot{\psi} + I_{xx}\dot{\psi} & 0 & 0 \\ -I_{xx}\dot{\theta} & (I_{yy} - I_{zz})\dot{\phi} & 0 \end{bmatrix} \dot{\eta} = \tau_{\eta}, \quad (3.35)$$

$$\begin{bmatrix} I_{xx}\ddot{\phi} - (I_{zz} - I_{yy})\dot{\theta}\dot{\psi} - I_{xx}\dot{\theta}\dot{\psi} \\ I_{yy}\ddot{\theta} + (I_{yy} - I_{zz})\dot{\phi}\dot{\psi} + I_{xx}\dot{\phi}\dot{\psi} \\ I_{zz}\ddot{\psi} + (I_{yy} - I_{zz})\dot{\theta}\dot{\phi} - I_{xx}\dot{\theta}\dot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_{\phi} \\ \tau_{\theta} \\ \tau_{\psi} \end{bmatrix}, \quad (3.36)$$

despejando las aceleraciones angulares y reduciendo términos se obtiene:

$$\begin{aligned} \ddot{\phi} &= \frac{(I_{yy} - I_{zz})\dot{\theta}\dot{\psi}}{I_{xx}} + \dot{\theta}\dot{\psi} + \frac{\tau_{\phi}}{I_{xx}} \\ \ddot{\theta} &= \frac{(I_{zz} - I_{xx})\dot{\phi}\dot{\psi}}{I_{yy}} + \dot{\phi}\dot{\psi} + \frac{\tau_{\theta}}{I_{yy}} \\ \ddot{\psi} &= \frac{(I_{xx} - I_{yy})\dot{\theta}\dot{\phi}}{I_{zz}} + \dot{\theta}\dot{\phi} + \frac{\tau_{\psi}}{I_{zz}} \end{aligned} \quad (3.37)$$

Finalmente, el modelo dinámico del vehículo cuadricóptero que toma en cuenta la dinámica traslacional y rotacional se encuentra dado por:

$$\ddot{x} = -\frac{U_1}{m}(\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \quad (3.38)$$

$$\ddot{y} = -\frac{U_1}{m}(\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi) \quad (3.39)$$

$$\ddot{z} = -\frac{U_1}{m}(\cos \phi \cos \theta) + g \quad (3.40)$$

$$\ddot{\phi} = \frac{l}{I_{xx}}U_2 + \dot{\theta}\dot{\psi}\left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right) \quad (3.41)$$

$$\ddot{\theta} = \frac{l}{I_{yy}}U_3 + \dot{\phi}\dot{\psi}\left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right) \quad (3.42)$$

$$\ddot{\psi} = \frac{1}{I_{yy}}U_4 + \dot{\theta}\dot{\phi}\left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right) \quad (3.43)$$

3.3. Control del vehículo

El concepto de **control automático** surge con la revolución industrial y los procesos que derivaron de esta transformación en la sociedad. Su objetivo principal es mantener las salidas de un proceso dado dentro de ciertos parámetros deseados para su óptimo funcionamiento. El principio de funcionamiento se basa en la realimentación o *feedback*, es decir, medir las salidas del proceso con el objeto de conocer la información del estado actual de la variable a controlar.

3.3.1. Controlador PD

Un controlador PD (Proporcional-Derivativo) es un mecanismo de control realimentado en el cual la parte Proporcional depende del error actual y la parte Derivativa trata predecir los errores futuros. Por lo cual la parte derivativa reduce el sobre-impulso logrando así mejorar el tiempo de estabilización del sistema. La relación de un controlador Proporcional-Derivativo entre la señal de control $u(t)$ y la señal de error $e(t)$ de un sistema está dada por la ecuación:

$$u_{PD}(t) = -\left(K_p e(t) + K_d \frac{de(t)}{dt}\right) \quad K_p > 0, K_d > 0 \quad (3.44)$$

donde:

K_p : Constante proporcional.

K_d : Constante derivativa.

En este trabajo la estrategia proporcional-derivativa será empleada para controlar las dinámicas de traslación y orientación del cuadricóptero. En particular, será aplicado de forma individual a sub-sistemas de segundo orden. Para garantizar la estabilidad de cada uno de tales sub-sistemas se presenta el siguiente teorema:

Teorema 1 *Considere el sistema lineal de segundo orden*

$$\ddot{\chi} + c_1\dot{\chi}(t) + c_2\chi(t) = u(t), \quad (3.45)$$

y la señal de control $u(t)$ definida como $u(t) = -K_d\dot{e}_\chi(t) - K_p e_\chi(t) + c_1\dot{\chi}(t) + c_2\chi(t) + \ddot{\chi}_d$, con $e_\chi(t) = \chi - \chi_d$ y $\dot{e}_\chi(t) = \dot{\chi} - \dot{\chi}_d$. Si $K_p, K_d > 0$ entonces el error de seguimiento $e(t)$ convergerá asintóticamente a cero, con lo cual se garantiza que $\lim_{t \rightarrow \infty} \chi(t) = \chi_d(t)$.

De tal manera que sustituyendo $u(t)$ en (3.45) obtendremos la ecuación para la dinámica del error

$$\ddot{e}_\chi + K_d\dot{e}_\chi + K_p e_\chi = 0, \quad (3.46)$$

formulando la representación de espacio de estados de la ecuación (3.46) se obtiene:

$$\begin{aligned} e_{\chi_1} &= e_\chi \\ e_{\chi_2} &= \dot{e}_{\chi_1}. \end{aligned} \quad (3.47)$$

Demostración:

A partir del sistema definido en (3.47) y reordenando los términos de la ecuación (3.46) se obtiene:

$$\begin{aligned} \dot{e}_{\chi_1} &= e_{\chi_2} \\ \dot{e}_{\chi_2} &= -K_d e_{\chi_2} - K_p e_{\chi_1}, \end{aligned} \quad (3.48)$$

reescribiendo (3.48) de forma matricial

$$\begin{bmatrix} \dot{e}_{\chi_1} \\ \dot{e}_{\chi_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_p & -K_d \end{bmatrix} \begin{bmatrix} e_{\chi_1} \\ e_{\chi_2} \end{bmatrix}, \quad (3.49)$$

a partir de la ecuación (3.48), la dinámica del error de seguimiento será asintóticamente estable si los valores propios de la matriz A definida como:

$$A = \begin{pmatrix} 0 & 1 \\ -K_p & -K_d \end{pmatrix}, \quad (3.50)$$

tienen parte real negativa. Lo cual se verifica si las raíces del polinomio característico

$$P(s) = s^2 + K_d s + K_p, \quad (3.51)$$

tienen parte real negativa. Lo anterior se satisface si las ganancias K_d y K_p se eligen como:

$$K_d > 0, K_p > 0,$$

con lo cual se prueba el Teorema 1 [25].

Análisis de respuesta transitoria y sintonización de las ganancias

Los sistemas de segundo orden, tal como su nombre indica, se pueden describir mediante una ecuación diferencial normalizada de segundo orden del tipo:

$$\frac{d^2 y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = k\omega_n^2 u(t), \quad (3.52)$$

y cuya función estándar de transferencia viene dada por:

$$G(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.53)$$

en la que el polinomio del denominador se le conoce como *polinomio característico*, cuyas raíces son los polos de la función de transferencia. Los parámetros que definen la función de transferencia son:

- ζ : Coeficiente o factor de amortiguamiento que determinará la la forma de la respuesta transitoria.

- ω_n : Es la frecuencia natural no amortiguada, corresponde a la frecuencia con la que oscilaría el sistema si no existiese amortiguamiento.

Con frecuencia, las características de desempeño de un sistema de control se especifican en términos de la respuesta transitoria para una entrada escalón unitario. La respuesta transitoria de un sistema para una entrada escalón unitario depende de las condiciones iniciales. Al especificar las características de la respuesta transitoria de un sistema de control para una entrada escalón unitario, es común especificar lo siguiente:

- t_s (secs) (*Tiempo de subida*).- Es el tiempo requerido para que la salida del sistema alcance por primera vez el estado estacionario. Se define como:

$$t_s = \frac{\pi - \phi}{\omega_n \sqrt{1 - \zeta^2}} \quad (3.54)$$

- t_p (secs) (*Tiempo de pico*).- Es el tiempo requerido para que la respuesta alcance el primer pico de sobrepaso y se define como:

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (3.55)$$

- t_e (secs) (*Tiempo de establecimiento*.-) Es el tiempo que transcurre desde que la salida del sistema comienza a evolucionar hasta que la respuesta se sitúa establemente en torno al 2% del valor de régimen estacionario. Viene dado por:

$$t_e = \frac{4}{\zeta \omega_n} \quad (3.56)$$

- SO (%) (*Sobreoscilación*).- Representa el sobrepico o sobreelongación del sistema en %, que es la diferencia entre el máximo valor pico de la respuesta y el valor de régimen estacionario. Este valor está dado por:

$$SO(\%) = e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \times 100 \quad (3.57)$$

En la Figura 3.2 se muestran los parámetros que acaban de ser descritos.

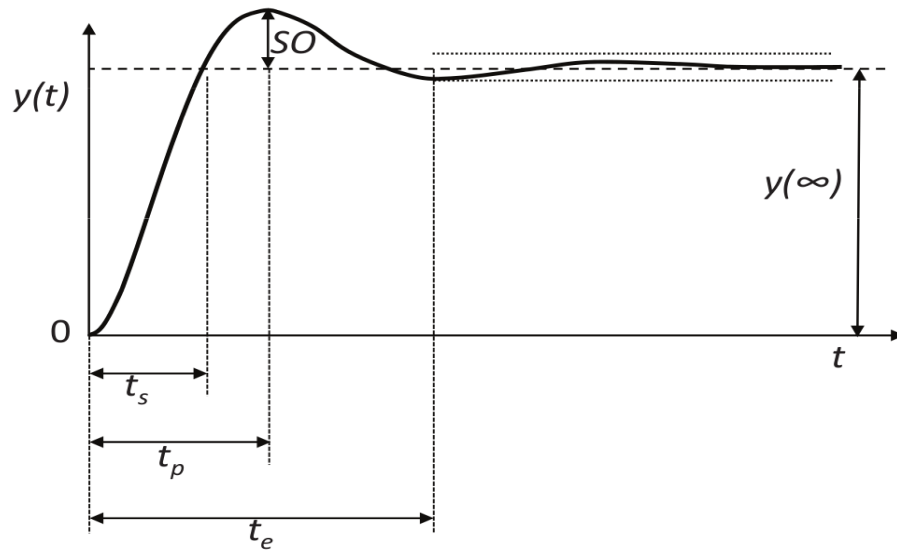


Figura 3.2: Parámetros que caracterizan la respuesta temporal de un sistema de segundo orden subamortiguado [9].

En general el coeficiente de amortiguamiento de un sistema debería diseñarse con valores entre 0.5 a 0.8, un buen criterio de diseño es utilizar un coeficiente de amortiguamiento ζ de aproximadamente 0.707 debido a que en estas condiciones el sistema es más robusto a las variaciones en los parámetros del sistema. Entre los principales métodos integrales para determinar la respuesta transitoria está el de la *Integral del valor absoluto del error ponderado en el tiempo (ITAE por sus siglas en inglés)*, que consiste en ponderar los errores que existen después de un largo tiempo más que los que existen al inicio de la respuesta. Este método se define como:

$$ITAE = \int_0^{\infty} t|e(t)|dt \quad (3.58)$$

que puede usarse para especificar una respuesta dinámica con un sobre-impulso y una oscilación relativamente pequeña. Los coeficientes óptimos generados por el criterio del ITAE según el orden del sistema se muestran en la Tabla 3.1:

Tabla 3.1: Coeficientes óptimos ITAE [26].

| Orden | Polinomio característico |
|-------|---|
| 1 | $s + \omega_n$ |
| 2 | $s^2 + 1.4\omega_n s + \omega_n^2$ |
| 3 | $s^3 + 1.75\omega_n s^2 + 2.15\omega_n^2 s + \omega_n^3$ |
| 4 | $s^4 + 2.1\omega_n s^3 + 3.4\omega_n^2 s^2 + 2.7\omega_n^3 s + \omega_n^4$ |
| 5 | $s^5 + 2.8\omega_n s^4 + 5\omega_n^2 s^3 + 5.5\omega_n^3 s^2 + 3.4\omega_n^4 s + \omega_n^5$ |
| 6 | $s^6 + 3.25\omega_n s^5 + 6.6\omega_n^2 s^4 + 8.6\omega_n^3 s^3 + 7.45\omega_n^4 s^2 + 3.95\omega_n^5 s + \omega_n^6$ |

De la Tabla 3.1 elegimos la ecuación característica de orden 2 para igualarla con la ecuación característica (3.51) del control PD utilizado con lo cual se obtiene:

$$s^2 + K_d s + K_p = s^2 + 1.4\omega_n s + \omega_n^2,$$

de manera que las ganancias buscadas quedan definidas de la siguiente forma:

$$K_p = \omega_n^2 \tag{3.59}$$

$$K_d = 1.4\omega_n \tag{3.60}$$

3.3.2. Diseño del controlador

La estructura de control propuesta para el sistema se presenta en el diagrama de bloques mostrado en la Figura 3.3.

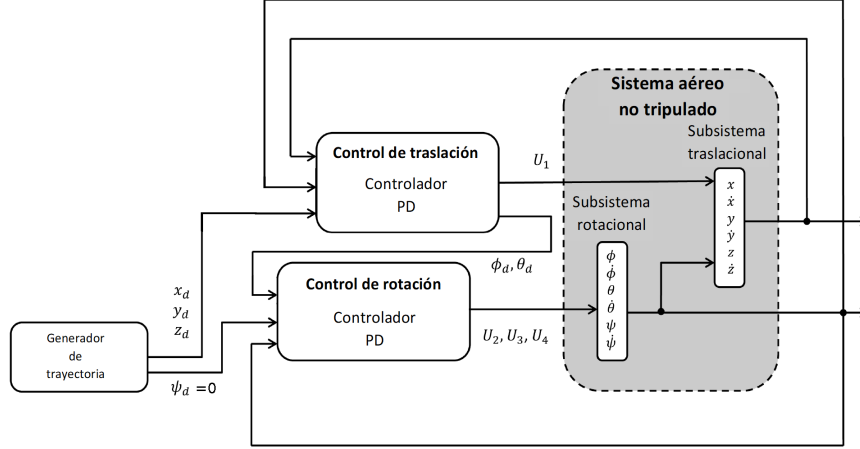


Figura 3.3: Diagrama a bloques del sistema de control [30].

En este diagrama se pueden observar los dos sistemas de control, tanto rotacional como traslacional y a continuación se desarrollan las ecuaciones para el control de estos subsistemas.

Control de orientación

El primer objetivo es estabilizar la dinámica rotacional del vehículo, estos son los ángulos de *roll*, *pitch* y *yaw*. Considerando los sub-sistemas (3.41), (3.42) y (3.43), se definen las siguientes leyes control para las dinámicas rotacionales.

$$U_2 = -\frac{I_{xx}}{l} [K_{d_\phi}(\dot{\phi} - \dot{\phi}_d) + K_{p_\phi}(\phi - \phi_d)] - \dot{\theta}\dot{\psi} \frac{I_{yy} - I_{zz}}{I_{xx}} + \ddot{\phi}_d, \quad (3.61)$$

$$U_3 = -\frac{I_{yy}}{l} [K_{d_\theta}(\dot{\theta} - \dot{\theta}_d) + K_{p_\theta}(\theta - \theta_d)] - \dot{\phi}\dot{\psi} \frac{I_{zz} - I_{xx}}{I_{yy}} + \ddot{\theta}_d, \quad (3.62)$$

$$U_4 = -I_{yy} [K_{d_\psi}(\dot{\psi} - \dot{\psi}_d) + K_{p_\psi}(\psi - \psi_d)] - \dot{\theta}\dot{\phi} \frac{I_{xx} - I_{yy}}{I_{zz}} + \ddot{\psi}_d. \quad (3.63)$$

Para obtener la dinámica del error, sustituimos (3.61) en (3.41):

$$\ddot{\phi} = -K_{d\phi}(\dot{\phi} - \dot{\phi}_d) - K_{p\phi}(\phi - \phi_d) + \ddot{\phi}_d, \quad (3.64)$$

reordenando los términos de (3.64) se obtiene el siguiente polinomio del error:

$$\ddot{e}_\phi(t) + K_{d\phi}\dot{e}_\phi(t) + K_{p\phi}e_\phi(t) = 0, \quad (3.65)$$

a partir del Teorema 1 las ganancias $K_{d\phi}$ y $K_{p\phi}$ se eligen mayores que cero para que el error de seguimiento $e_\phi(t)$ converja asintóticamente a cero.

Siguiendo el mismo procedimiento para *pitch* y *yaw* se obtienen los polinomios de error:

$$\ddot{e}_\theta(t) + K_{d\theta}\dot{e}_\theta(t) + K_{p\theta}e_\theta(t) = 0 \quad (3.66)$$

y

$$\ddot{e}_\psi(t) + K_{d\psi}\dot{e}_\psi(t) + K_{p\psi}e_\psi(t) = 0. \quad (3.67)$$

se eligen las ganancias $K_{d\theta}, K_{p\theta}, K_{d\psi}$ y $K_{p\psi}$ mayores que cero para que los errores de seguimiento $e_\theta(t)$ y $e_\psi(t)$ converjan asintóticamente a cero.

Control de traslación

A partir de las señales de referencia (x_d, y_d, z_d) de entrada, el control traslacional genera la señal de control U_1 y los ángulos de pitch y roll deseados ϕ_d, θ_d respectivamente, donde U_1 es la entrada de la dinámica del sub-sistema traslacional.

Control de altura

Considerando la dinámica en z dada por (3.40), la señal de control U_1 se propone como:

$$U_1 = \frac{m(g + u_{zv})}{(\cos \phi \cos \theta)}, \quad (3.68)$$

donde u_{zv} está dada por:

$$u_{zv} = -K_{pz}(z - z_d) - K_{dz}(\dot{z} - \dot{z}_d) + \ddot{z}_d, \quad (3.69)$$

sustituyendo (3.68) en (3.40) se obtiene:

$$\ddot{z} = u_{zv}, \quad (3.70)$$

de tal manera que la ecuación (3.70) queda de la siguiente forma:

$$\ddot{z} = -K_{pz}(z - z_d) - K_{dz}(\dot{z} - \dot{z}_d) + \ddot{z}_d, \quad (3.71)$$

reordenando los términos de (3.71) se obtiene el polinomio para la dinámica del error.

$$\ddot{e}_z(t) + K_{dz}\dot{e}_z(t) + K_{pz}e_z(t) = 0, \quad (3.72)$$

se eligen las ganancias K_{dz} y K_{pz} mayores que cero para que el error de seguimiento $e_z(t)$ converja asintóticamente a cero.

Control para el desplazamiento en X y Y

Considerando la dinámica en x y y dadas por (3.38) y (3.39) respectivamente.

$$\ddot{x} = -\frac{U_1}{m}(\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta), \quad (3.73)$$

$$\ddot{y} = -\frac{U_1}{m}(\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi), \quad (3.74)$$

se proponen dos controles virtuales μ_x y μ_y definidos como:

$$\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = -\frac{m}{U_1} \begin{bmatrix} \cos \psi & \sin \psi \\ \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \cos \phi \sin \theta \\ \sin \phi \end{bmatrix}, \quad (3.75)$$

asumiendo que las rotaciones en *pitch* y *roll* no serán bruscas entonces $\cos \phi \approx \cos \theta \approx 1$, $\sin \phi \approx \phi_d$ y $\sin \theta \approx \theta_d$ (3.75) queda como:

$$\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = -\frac{m}{U_1} \begin{bmatrix} \cos \psi & \sin \psi \\ \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix}, \quad (3.76)$$

invirtiendo la matriz dada en (3.76) obtendremos los ángulos deseados

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = -\frac{U_1}{m} \begin{bmatrix} \cos \psi & \text{sen } \psi \\ \text{sen } \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \quad (3.77)$$

sustituyendo μ_x y μ_y en (3.73) y (3.74) respectivamente obtenemos la siguiente dinámica:

$$\ddot{x} = \mu_x, \quad (3.78)$$

$$\ddot{y} = \mu_y, \quad (3.79)$$

las señales de los controles virtuales se proponen de la siguiente forma:

$$\mu_x = -K_{px}(x - x_d) - K_{dx}(\dot{x} - \dot{x}_d) + \ddot{x}_d, \quad (3.80)$$

y

$$\mu_y = -K_{py}(y - y_d) - K_{dy}(\dot{y} - \dot{y}_d) + \ddot{y}_d, \quad (3.81)$$

sustituyendo (3.80) en (3.78) y (3.81) en (3.79) se obtienen:

$$\ddot{x} = -K_{px}(x - x_d) - K_{dx}(\dot{x} - \dot{x}_d) + \ddot{x}_d, \quad (3.82)$$

y

$$\ddot{y} = -K_{py}(y - y_d) - K_{dy}(\dot{y} - \dot{y}_d) + \ddot{y}_d, \quad (3.83)$$

reordenando los términos (3.82) y (3.83) se obtienen los polinomios para las dinámicas del error:

$$\ddot{e}_x(t) + K_{dx}\dot{e}_x(t) + K_{px}e_x(t) = 0 \quad (3.84)$$

y

$$\ddot{e}_y(t) + K_{dy}\dot{e}_y(t) + K_{py}e_y(t) = 0 \quad (3.85)$$

se eligen las ganancias K_{dx}, K_{px}, K_{dy} y K_{py} mayores que cero para que los errores de seguimiento $e_x(t)$ y $e_y(t)$ converjan asintóticamente a cero.

CAPÍTULO 4

Desarrollo de la plataforma experimental

Para la construcción y desarrollo de la plataforma experimental se utilizaron los componentes mostrados en el diagrama a bloques de la Figura 4.1.

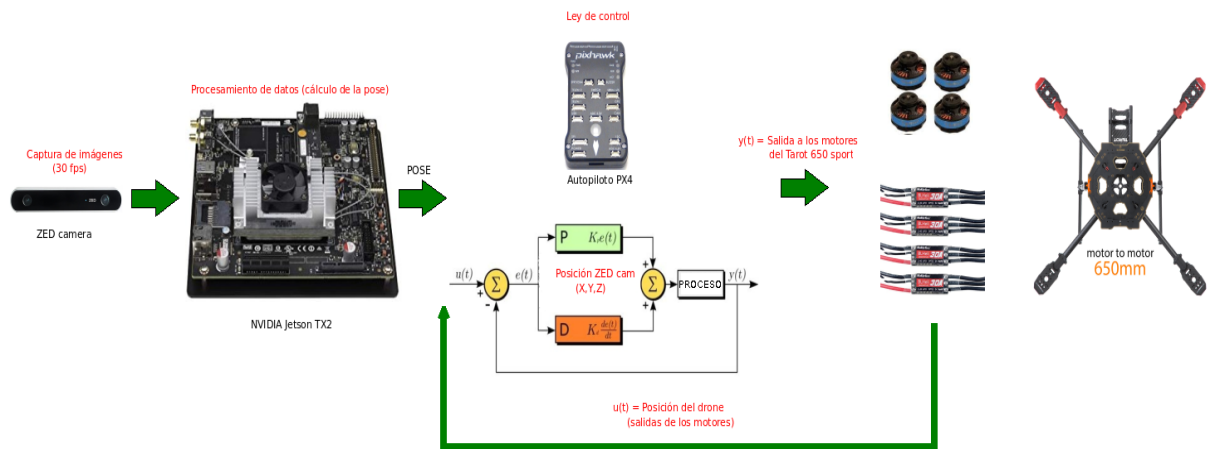


Figura 4.1: Diagrama a bloques de la plataforma experimental.

El diagrama a bloques anterior consta de los siguientes componentes:

- Cámara estéreo (ZED™ Stereo Camera).
- Computadora embarcada NVIDIA® Jetson™ TX2.
- Frame Tarot® 650 Sport (cuadricóptero).

- 4 motores brushless TL68P02 Tarot®4006/620KV.
- Batería LiPo Turnigy®4s/5000 mAh.
- Autopiloto Pixhawk PX4 2.4.8.
- Módulo convertidor de voltaje DC-DC XL6009.

En la Figura 4.2 se muestra la plataforma experimental junto con sus componentes mecánicos y electrónicos que fue construida para este proyecto.



Figura 4.2: Plataforma experimental utilizada.

En este capítulo se presenta la sección 4.1 en la que se describen los componentes electrónicos y mecánicos para el desarrollo de este trabajo, en la sección 4.2 se muestran las conexiones hechas para que el procesador gráfico y el autopiloto tuvieran comunicación entre sí y en la sección 4.3 se detallan los pasos seguidos para instalar el software necesario para la utilización de la cámara estéreo.

4.1. Componentes mecánicos y electrónicos de la plataforma

4.1.1. Cámara estéreo

Para la realización del presente proyecto se utilizó la cámara estéreo ZED™ de la compañía *Stereolabs*® mostrada en la Figura 4.3, la cual mediante la triangulación de las imágenes izquierda y derecha provee una escena en 3D de la imagen captada.



Figura 4.3: ZED stereo camera [28].

Entre las características con las que cuenta esta cámara se destacan [27]:

- Resolución máxima de 2K y hasta 100 *frames* por segundo.
- Percepción de hasta 20 mts en espacios cerrados.
- SDK que permite 6 grados de libertad.
- Mapeo espacial.

4.1.2. Unidad de Procesamiento Gráfico (GPU)

La computadora embarcada que se utilizó en este trabajo es la tarjeta Jetson™TX2 de NVIDIA®, la cual es la encargada de ejecutar los algoritmos de odometría visual. La Tabla 4.1 muestra las características principales de esta tarjeta.

Tabla 4.1: Especificaciones técnicas de la Jetson TX2 [3].

| Jetson TX2 | |
|-----------------------|---|
| GPU | NVIDIA®Pascal™, 256 núcleos CUDA |
| CPU | HMP Denver Dual 2/2 MB L2 + Quad ARM® A57/2 MB L2 |
| Video | Codificación a 4K x 2K 60 Hz (HEVC) Decodificación a 4K x 2K 60 Hz (soporte de 12 bits) |
| Memoria | 8 GB 128 bit LPDDR4 59.7 GB/s |
| Display | 2x DSI, 2x DP 1.2 / HDMI 2.0 / eDP 1.4) |
| CSI | Hasta 6 cámaras (2 vías) CSI2 D-PHY 1.2 (2.5 Gbps/vía) |
| PCIE | Gen 2 1x4 + 1x1 OR 2x1 + 1x2 |
| Almacenamiento | 32 GB eMMC, SDIO, SATA |
| Puertos | CAN, UART, SPI, I2C, I2S, GPIOs |
| USB | USB 3.0 + USB 2.0 |
| Conectividad | 1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth |
| Dimensiones | 50 mm x 87 mm (con 400 pines disponibles para conexiones tarjeta-a-tarjeta) |

4.1.3. Autopiloto

El autopiloto seleccionado para controlar al vehículo cuadricóptero es el modelo HKPilot32, el cual es un genérico de la PIXHAWK de *3DR robotics*® mostrado en la Figura 4.4.



Figura 4.4: Autopiloto HKPilot32 [24].

Este controlador de vuelo cuenta con un procesador micro ST 32bit ARM Cortex y está diseñado para soportar casi todo tipo de vehículos, desde terrestres, submarinos y aéreos. El autopiloto permite controlar el vuelo del cuadricóptero a través de un programa previamente cargado que contiene las leyes de control diseñadas en el capítulo 3 para que la velocidad de giro de los motores sea la adecuada para regular los ángulos de movimiento y el posterior desplazamiento del cuadricóptero. Entre las principales características de este autopiloto están:

- Procesador 32 bit ARM Cortex®M4 para Nuttx ROS.
- 14 salidas PWM/servo.
- Periféricos para conectividad (UART, I2C, CAN).
- Tarjeta de memoria MicroSD.

4.1.4. Motores (brushless)

Este tipo de motores al no tener escobillas permiten que no haya contacto físico entre el rotor y el estator, lo cual evita la fricción y el calentamiento del motor, el movimiento es posible gracias a los campos electromagnéticos generados en el estator que interactúan con imanes situados en el rotor con el objetivo de lograr una mayor eficiencia y potencia.

Los motores seleccionados fueron los TL68P02 4006/620KV de Tarot® , como el mostrado en la Figura 4.5, son de tamaño medio optimizados para las propelas Tarot®1355 y batería LiPo 4S, son ligeros de peso y generan un empuje de entre 650g y 800g cada uno con el regulador del radio-control al 50 %.



Figura 4.5: Motor Tarot®TL68P02 4006/620KV [11].

Entre las principales características de este motor encontramos:

- **Peso:** 82g

- **Corriente de reposo (A)@14.8V:** 0.8A

- **LiPo:** 4S

- **Corriente continua máxima:** 17.5A

4.1.5. Propelas

Las propelas son las encargadas de generar el empuje necesario para que el vehículo aéreo pueda elevarse al generar una diferencia de presión en el aire. Tienen una nomenclatura numérica 00x00 escrita que hace referencia a la longitud en pulgadas y al paso¹ de la propela también en pulgadas. En este caso se seleccionaron propelas de 12x4.5 pulgadas hechas de plástico como las mostradas en la Figura 4.6.



Figura 4.6: Propelas de 12x4.5" utilizadas.

4.1.6. Variador electrónico de velocidad (ESC)

Este dispositivo es el encargado de convertir las señales de control provenientes del autopiloto a señales que van a los motores y hacen que estos giren a una mayor o menor velocidad, es la etapa de potencia entre el autopiloto y los motores.

Entre sus especificaciones encontramos:

- **Consumo de corriente:** 30A continuo
- **Rango de voltaje:** 2-4s Lipo
- **Frecuencia de entrada:** 1 KHz
- **Tamaño:** 50 x 25 x 11 mm

¹Se refiere a una distancia fija que la propela avanzará hacia adelante por cada vuelta.



Figura 4.7: Variador electrónico de velocidad *AFRO*®30A [13].

4.1.7. Frame o marco de montaje

El frame utilizado para este proyecto fue el Tarot® que incorpora un tren de aterrizaje retráctil eléctrico, un centro de gravedad más bajo, un diseño más estable, además de que está fabricado con fibra de carbono con placas de circuito impreso (PCB) de soldadura para facilitar el montaje y cableado. En la Figura 4.8 se muestra el chasis armado.



Figura 4.8: Frame Tarot®650 Sport [12].

Entre sus principales características se tiene:

- **Material:** Placas de fibra carbono TORAY®.
- **Diámetro:** 600 mm

- **Peso total:** 1 Kg
- Diseñado especialmente para monitoreo, investigación aérea, mapeo y teledetección.

4.1.8. Batería LiPo

Las baterías recargables de Litio-Polímero tienen la características de ser ligeras, se pueden construir de diversas formas para adaptarlas a cualquier aplicación y tienen una alta capacidad para su tamaño relativamente pequeño.

Las nuevas generaciones de estas baterías permiten un número mayor de ciclos de vida y durabilidad gracias a la adición del grafeno en la estructura de la batería.



Figura 4.9: Batería LiPo Grafeno 4S 5000 mAh [14].

Entre las características de esta batería encontramos:

- **Capacidad:** 5000mAh
- **Peso:** 589g
- **Dimensiones:** 144x51x41mm
- **Balance de enchufe:** JST-XH
- **Enchufe de descarga:** XT-90

4.1.9. Módulo convertidor de voltaje

Los convertidores DC/DC son circuitos capaces de transformar un nivel de voltaje a otro de mayor o menor nivel. Existen dos tipos de convertidores o reguladores DC-DC: lineales y conmutados (switching). Los reguladores de tipo lineal como el clásico LM7805 o el LM317 son muy sencillos de utilizar pero no son eficientes energéticamente.

El convertidor de voltaje DC-DC Step-Up 2.5A XL6009 es un regulador de tipo conmutado y tiene como función entregar un voltaje de salida constante superior al voltaje de entrada frente a variaciones del voltaje de entrada o de carga. Soporta corrientes de salida de hasta 2.5A, voltaje de entrada entre 5V a 32V y voltaje de salida entre 7V a 32V. El voltaje de salida se selecciona mediante un potenciómetro multivuelta. Debido a su tamaño pequeño y poco peso fue seleccionada para alimentar la Jetson TX2, en la Figura 4.10 se muestra este convertidor DC-DC.

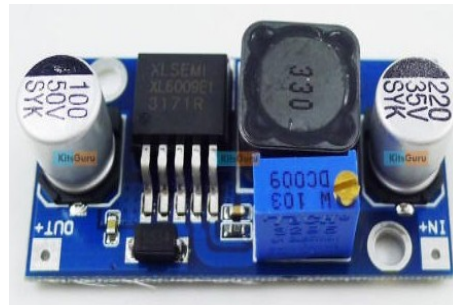


Figura 4.10: Convertidor DC-DC [15].

4.2. NVIDIA® Jetson™TX2 + HKPilot32

Para llevar a cabo la comunicación entre el GPU y el autopiloto se utilizaron el puerto serial UART1 (J17) de la Jetson™ y el puerto de telemetría (TELEM1) del HKPilot32. En la Figura 4.11 se muestran la configuración de los pines de la tarjeta Jetson™TX2 y en la tabla 4.2 se especifican la función de cada uno de estos pines.

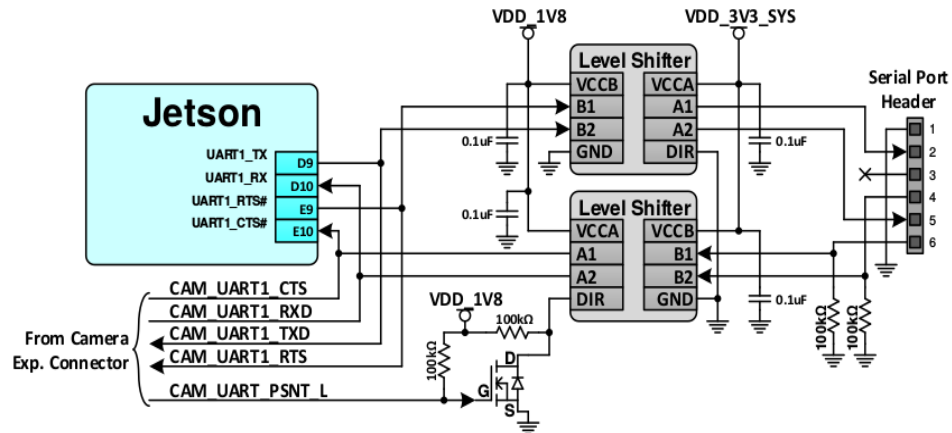


Figura 4.11: Configuración de los pines del puerto UART1 [21].

La conexión entre la tarjeta Jetson™TX2 y el autopiloto es necesaria ya que los algoritmos de visión encargados de obtener la posición del vehículo son ejecutados en esta tarjeta, estos datos de la posición del vehículo son enviados a su vez mediante comunicación serial al autopiloto y son utilizados por el HKPilot32 para llevar a cabo el control PD de posición previamente programado del vehículo.

Tabla 4.2: Distribución de los pines del puerto UART1 de la Jetson™TX2 [3].

| Pin# | Nombre de la señal | Uso/descripción | Tipo de pin |
|------|--------------------|---------------------|-------------|
| 1 | GND | Tierra | Tierra |
| 2 | UART1_RTS_3V3_L | UART petición envío | Salida |
| 3 | NC | Sin uso | Sin uso |
| 4 | UART1_RXD_3V3 | UART receptor | Entrada |
| 5 | UART1_TXD_3V3 | UART transmisor | Salida |
| 6 | UART1_CTS_3V3_L | UART limpiar envío | Entrada |

Los pines utilizados son los marcados con gris en la Tabla 4.2, estos fueron conectados con los pines correspondientes al TELEM1 del autopiloto HKPilot32 marcados en gris en la Tabla 4.3.

Tabla 4.3: Especificaciones técnicas del puerto serial UART1 de la Jetson TX2 [23].

| Pin | Señal | Voltaje |
|-----|---------------|---------|
| 1 | VCC | +5V |
| 2 | TX (salida) | +3.3V |
| 3 | RX (entrada) | +3.3V |
| 4 | CTS (entrada) | +3.3V |
| 5 | RTS (salida) | +3.3V |
| 6 | GND | GND |

La conexión se llevó a cabo de manera de la siguiente manera:

- Pin 1 UART1 a pin 6 TELEM1 PX4.
- Pin 4 UART1 a pin 2 TELEM1 PX4.
- Pin 5 UART1 a pin 3 TELEM1 PX4.

En la Figura 4.12 se muestra la ubicación del puerto UART1 en la tarjeta embarcada.

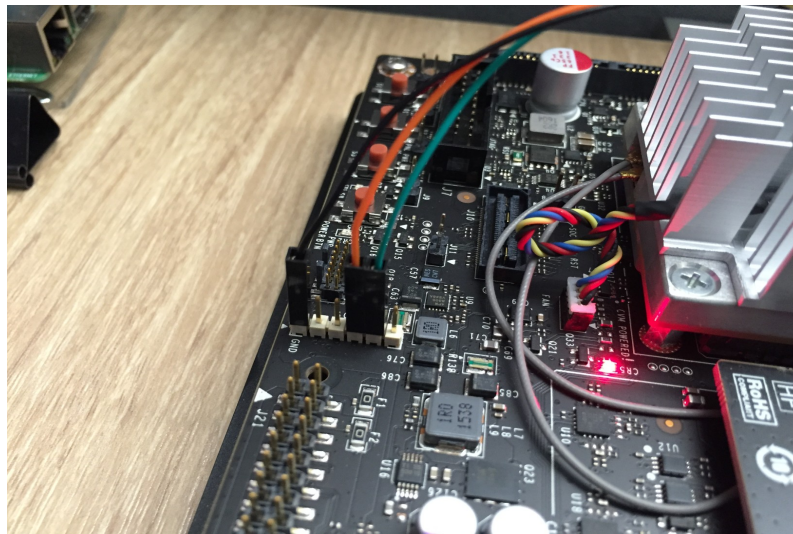


Figura 4.12: Conexión puerto UART1 [7].

4.3. Instalación del software

4.3.1. Descarga e instalación del SDK ZED™ Stereo Camera

El SDK² de la ZED™ camera es el conjunto de herramientas de software que nos permite utilizar las funciones de detección de profundidad, seguimiento, mapeo y grabación de la cámara.

Instalación

En la página [//www.stereolabs.com/developers/release/2.6/#sdkdownloads_anchor](http://www.stereolabs.com/developers/release/2.6/#sdkdownloads_anchor) encontramos el SDK de la ZED™ camera disponible para Windows, Linux y tarjetas embebidas Jetson™, en este caso seleccionamos la opción para la tarjeta Jetson™TX2 y su instalación se llevó a cabo de la siguiente manera:

1. Descargar el ejecutable `.run` en su versión más reciente que contiene los instaladores de la aplicación.
2. Desde la terminal acceder a la ubicación del ejecutable `.run` descargado y ejecutarlo con las instrucciones:
 - `chmod +x ZED_SDK_Linux_*.run`
 - `./ZED_SDK_Linux_*.run`
3. Descargar e instalar la última versión de la plataforma de computación en paralelo CUDA desarrollado por NVIDIA®), en el caso de la Jetson™TX2 esta plataforma se instala junto con la plataforma de desarrollo de la tarjeta.
4. Reiniciar la computadora.

²Software Development Kit

4.3.2. SDK ZED™ Stereo Camera + ROS

Para utilizar la cámara en integración con ROS³ y acceder a las funciones instaladas por su SDK descargamos la paquetería creada con ese objetivo ubicada en la página:

<https://github.com/stereolabs/zed-ros-wrapper>.

Instalación.

Pre-requisitos:

- Tener instalado el ROS Kinetic y haber creado un área de trabajo (workspace) respectivamente.
- Tener instalada la última versión del SDK de la cámara.
- Instalar la librería **Point Cloud Library**.

La instalación de esta paquetería se hizo de la siguiente forma:

- Abrir una terminal.
- `cd ~/catkin_ws/src`
- `git clone https://github.com/stereolabs/zed-ros-wrapper.git`
- `cd ..`
- `catkin build`
- `source ./devel/setup.bash`

Una vez descargado e instalado el paquete en ROS para el manejo de la cámara estéreo fue necesario configurar el archivo **zed.launch**⁴ modificando los siguientes parámetros:

- **“camera_model”**: Seleccionando el modelo ZED™(0).

³Robot Operating System

⁴Ubicado en la ruta `zed-ros-wrapper/zed_wrapper/launch`

- **“resolution”**: Seleccionando la resolución HD720 (2).
- **“frame_rate”**: Seleccionando 30 fps (30).

Posteriormente con la instrucción **roslaunch zed_wrapper zed.launch** publicamos todos los tópicos disponibles de la cámara:

- **Cámara izquierda**

- **/zed/rgb/image_rect_color**: Imagen a color rectificada (imagen izquierda en RGB por default).
- **/zed/rgb/image_raw_color**: Imagen a color sin rectificar (imagen izquierda en RGB por default).
- **/zed/rgb/camera_info**: Datos de calibración de la cámara izquierda.

- **Cámara derecha**

- **/zed/right/image_rect_color**: Imagen derecha a color rectificada.
- **/zed/right/image_raw_color**: Imagen derecha a color sin rectificar.
- **/zed/right/camera_info**: Datos de calibración de la cámara derecha.

- **Profundidad y nube de puntos**

- **/zed/depth/depth_registered**: Mapa de profundidad de la imagen izquierda (dado en flotante de 32 bits, en metros).
- **/zed/point_cloud/cloud_registered**: Nube de puntos a color.
- **/zed/confidence/confidence_image**: Imagen de confianza.
- **/zed/confidence/confidence_map**: Imagen de confianza. (valores en punto flotante)
- **/zed/disparity/disparity_image**: Imagen de disparidad.

- **Tracking**

- **/zed/odom:** Posición y orientación en 3D relativo al marco de la odometría (para odometría visual).
- **/zed/map:** Posición y orientación en 3D relativo al marco del mapa (algoritmo para fusión de sensores).

▪ **Inertial data**

- **/zed/imu/data:** Datos del acelerómetro, giróscopo y orientación con respecto al marco inercial de la Tierra.
- **/zed/imu/data_raw:** Datos del acelerómetro y giróscopo con respecto al marco inercial de la Tierra.

Para este caso únicamente utilizamos el tópico **/zed/odom** que publica la posición (X, Y, Z) de la cámara y su orientación (x, y, z, w) en cuaterniones.

CAPÍTULO 5

Resultados

En este capítulo se presenta la sección 5.1 que son los resultados en simulación de la estrategia de control utilizada, en la sección 5.2 se muestran los resultados de las comparaciones realizadas entre el GPS y la odometría visual y en la sección 5.3 se muestran los resultados experimentales en el seguimiento de tres trayectorias programadas utilizando odometría visual.

5.1. Resultados en simulación

Para presentar el funcionamiento del control PD desarrollado en la sección 3.3 de este trabajo, se muestran a continuación una serie de resultados obtenidos de la simulación hecha en MATLAB® para los controles en posición y orientación en el seguimiento de una trayectoria circular.

A partir de la ecuación característica de la forma estándar definida en (3.53) se obtiene el valor del coeficiente de amortiguamiento:

$$\zeta \approx 0.707$$

para realizar la simulación se sintonizaron las ganancias considerando las ecuaciones (3.59) y (3.60), fue elegido un tiempo de pico $t_p = 1s$ y a partir de la ecuación (3.55) se obtuvo un valor de $\omega_n \approx 4.4$, por lo cual las ganancias tienen los siguientes valores:

$$K_p \approx 20$$

$$K_d \approx 6$$

los parámetros del cuadricóptero mostrados en la Tabla 5.1 fueron los considerados en la prueba de simulación y en la Tabla 5.2 se muestran las ganancias obtenidas durante el proceso de sintonización y en las Figuras (5.1-5.4) se muestran las salidas de la simulación con estos valores de ganancias calculados.

Tabla 5.1: Parámetros para el modelo del cuadricóptero.

| Parámetro | Valor | Unidades |
|-----------|-------|------------|
| m | 2.5 | Kg |
| g | 9.81 | m/s^2 |
| I_x | 0.01 | Ns^2/rad |
| I_y | 0.01 | Ns^2/rad |
| I_z | 0.005 | Ns^2/rad |

Tabla 5.2: Parámetros para el controlador PD.

| Ganancias en orientación | Valor | Ganancias en posición | Valor |
|--------------------------|-------|-----------------------|-------|
| $K_{p\phi}$ | 20 | K_{px} | 20 |
| $K_{d\phi}$ | 50 | K_{dx} | 6 |
| $K_{p\theta}$ | 3 | K_{py} | 20 |
| $K_{d\theta}$ | 7 | K_{dy} | 6 |
| $K_{p\psi}$ | 1 | K_{pz} | 20 |
| $K_{d\psi}$ | 1 | K_{dz} | 6 |

5.1.1. Control en posición

Para generar una trayectoria circular se consideraron funciones sinusoidales para el eje X y Y como sus referencias. En las Figuras 5.1 y 5.2 se muestran los resultados de la simulación

para las posiciones en X y Y respectivamente considerando una trayectoria circular con un radio de 5 mts.

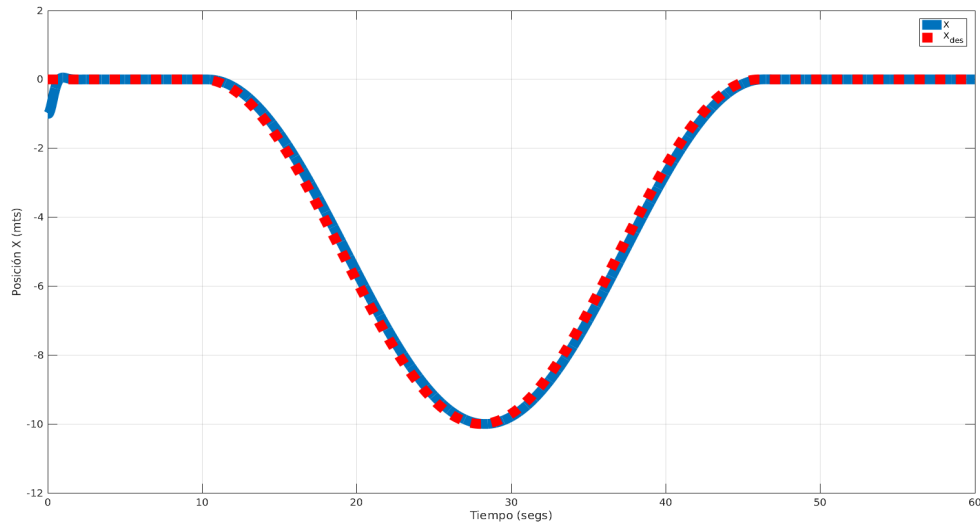


Figura 5.1: Simulación del seguimiento de la referencia en X.

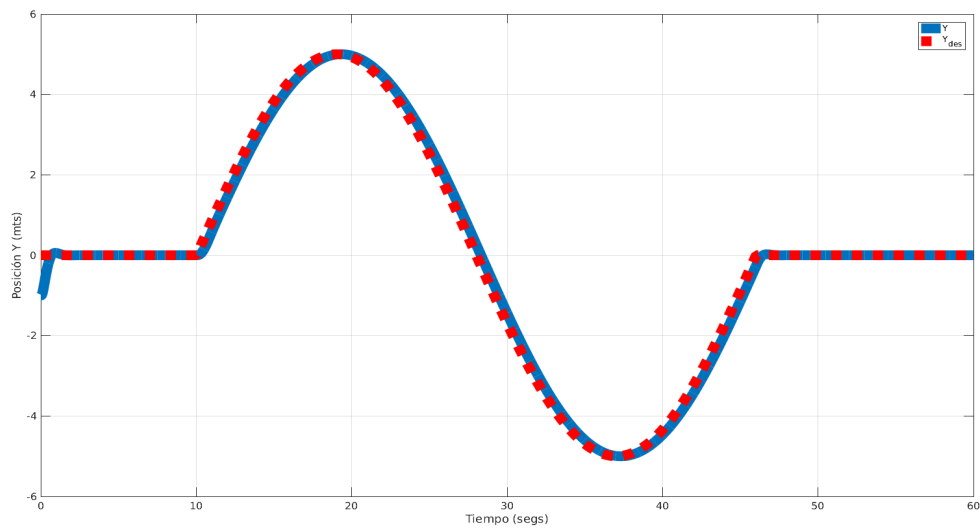


Figura 5.2: Simulación del seguimiento de la referencia en Y.

Durante el desplazamiento del vehículo aéreo no tripulado en los ejes X y Y se consideró una referencia en altura de 5 mts. El resultado de la simulación en altura se muestra en la Figura 5.3.

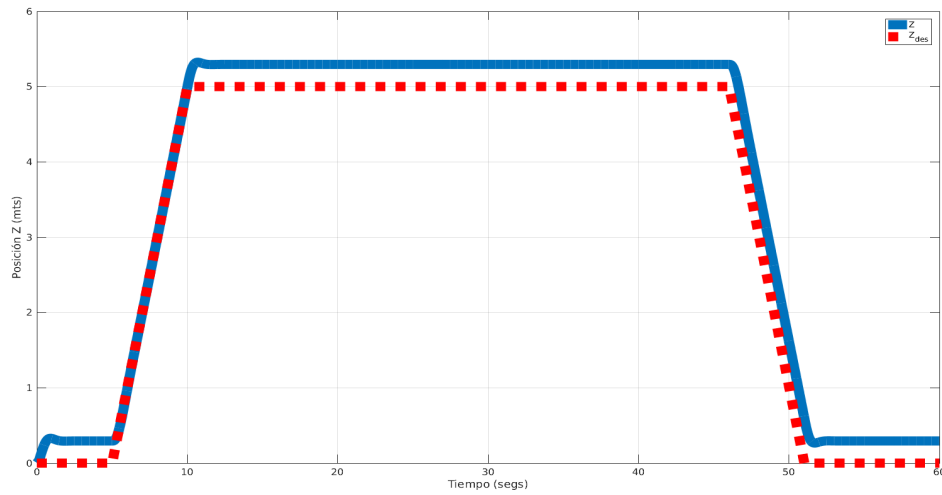


Figura 5.3: Simulación del seguimiento de la referencia en Z.

Finalmente en la Figura 5.4 se muestra el resultado de la simulación en el seguimiento de una trayectoria circular desde una perspectiva (X,Y,Z).

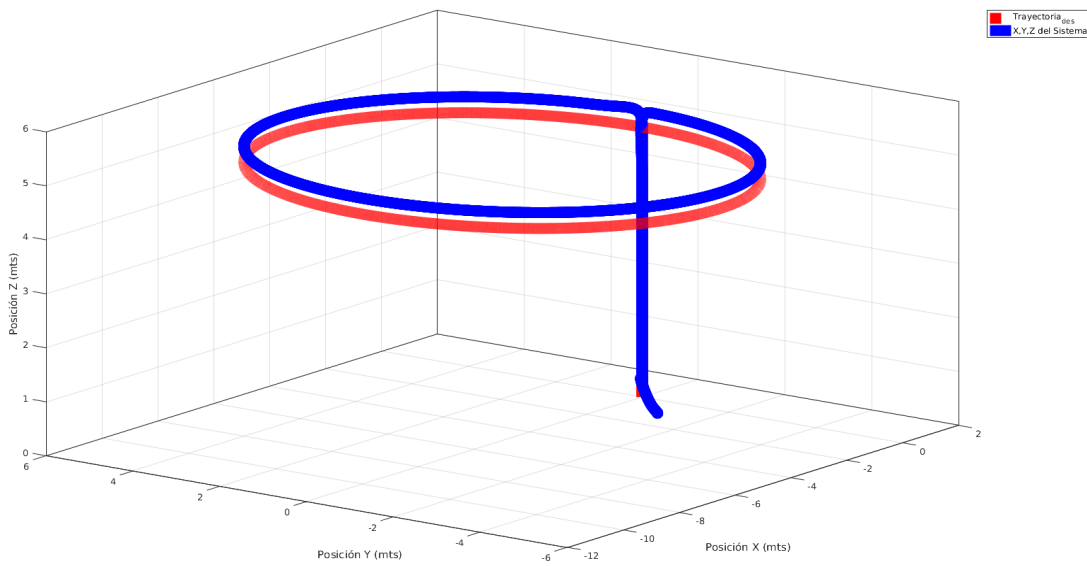


Figura 5.4: Simulación del seguimiento de la trayectoria en 3D.

5.2. Comparación entre GPS vs odometría visual

Antes de utilizar la cámara estéreo como sensor de posición se realizaron una serie de experimentos para comparar las mediciones obtenidas con el GPS y con los algoritmos de odometría visual. Los recorridos de prueba fueron hechos en espacios abiertos, la primera prueba, mostrada en las Figuras 5.5 y 5.6, fue realizada en un lugar donde la señal del GPS no tuviera interferencias, para este recorrido fue elegida la cancha de fútbol del CINVESTAV.

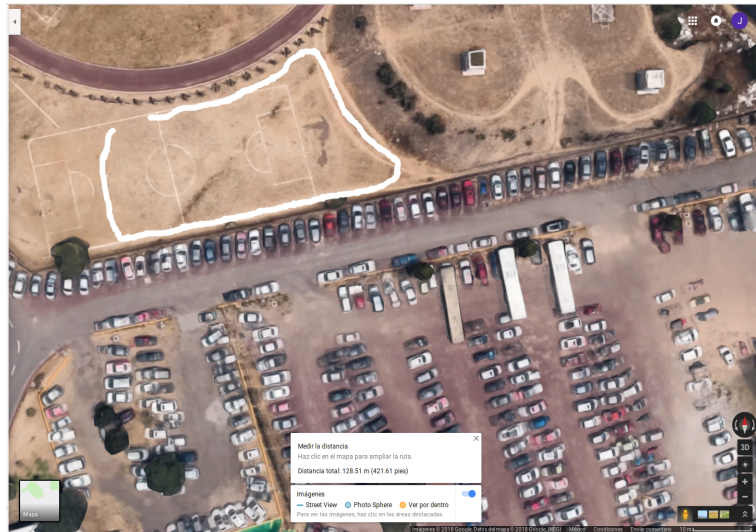


Figura 5.5: Recorrido parte trasera portería Este.

En la Figura 5.5 la línea blanca representa el recorrido realizado por el vehículo aéreo no tripulado llevando consigo el GPS y el sistema de odometría visual con el objetivo de obtener y comparar ambas mediciones de posición. En la Figura 5.6 se muestran ambas mediciones obtenidas, donde la línea punteada representa la posición con la odometría visual y la línea continua la posición con el GPS.

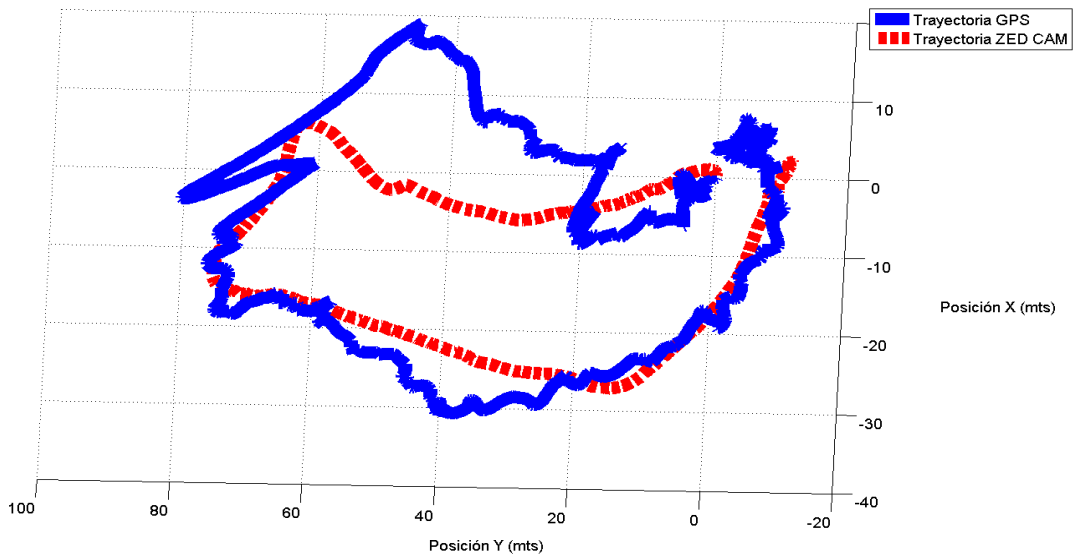


Figura 5.6: Mediciones GPS y cámara estéreo.

La segunda prueba, mostrada en las Figuras 5.7 y 5.8, fue hecha en un lugar donde la señal de GPS tuviera interferencias o un difícil acceso, para este recorrido fue escogida la rotonda conformada por los edificios de Servicios escolares, Espacios teóricos, Biblioteca, Banco de cerebros y el laboratorio UMI-LAFMIA del CINVESTAV.

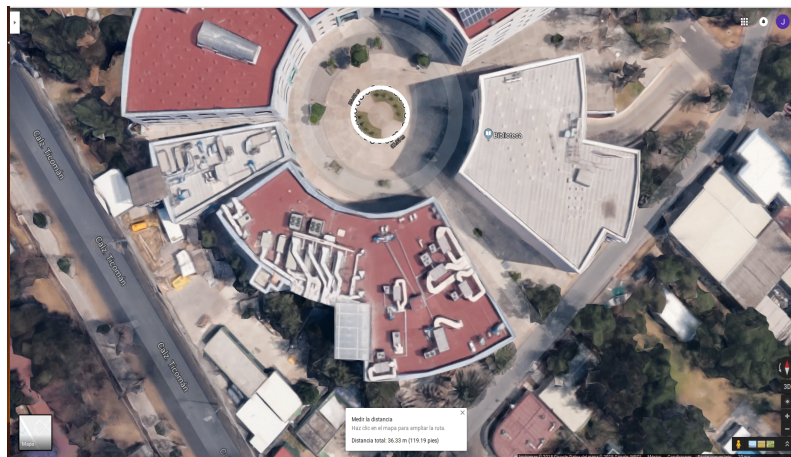


Figura 5.7: Recorrido rotonda externa UMI-LAFMIA.

De la misma forma, la línea blanca representa el recorrido realizado por el vehículo aéreo no tripulado llevando consigo el GPS y el sistema de odometría visual con el objetivo de

obtener y comparar ambas mediciones de posición.

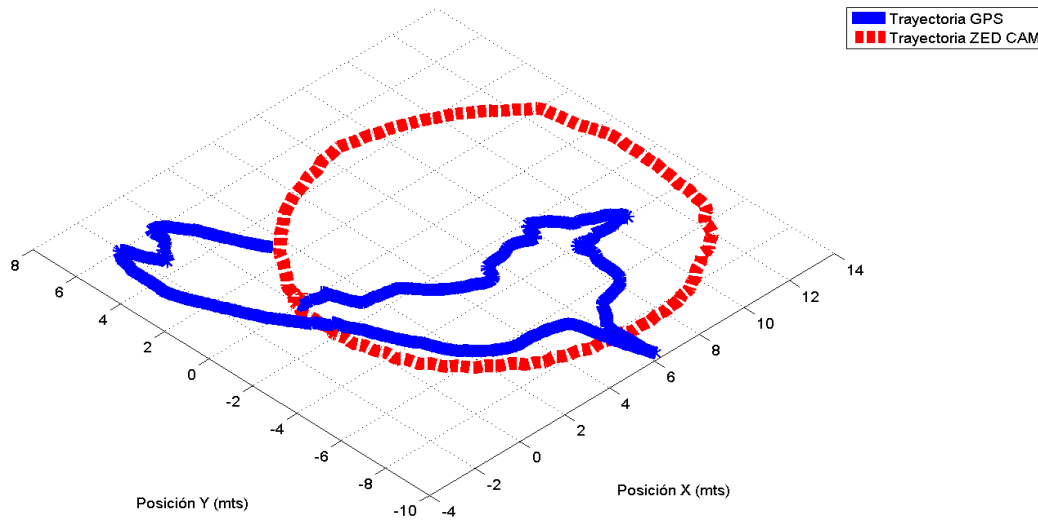


Figura 5.8: Mediciones GPS y cámara estéreo.

En la Figura 5.8 se muestran ambas mediciones obtenidas, y al igual que los resultados del experimento anterior la línea punteada representa la posición con la odometría visual y la línea continua la posición con el GPS.

En ambas pruebas las trayectorias trazadas por la cámara se asemejan más a las trayectorias reales, en cambio las trayectorias trazadas por el GPS tuvieron una diferencia más notable con las reales, en la segunda prueba esta diferencia se hizo más evidente dado que la señal del GPS tenía interferencia debido a los edificios que están alrededor de la rotonda, los resultados de estas pruebas justifican el uso de la odometría visual como sistema de navegación en vez del GPS. Cabe resaltar que la cámara requiere de escenas ricas en texturas para realizar de mejor manera la trayectoria recorrida por el vehículo, es decir, para recorridos en espacios abiertos sin puntos de referencia o texturas la cámara se “perderá”, en esta clase de espacios el GPS tiene una mucho mejor respuesta, como se muestra en la prueba 1 comparada con la prueba 2. La comparación entre ambas trayectorias es “cualitativa”, ya que no se contó con una forma de obtener la trayectoria “real” seguida por el vehículo.

5.3. Resultados experimentales

5.3.1. Experimento 1: *Control en altura y posición*

El primer experimento fue mantener una posición controlada en altura y posiciones en (x, y) , vuelo en estado estacionario, para este caso se seleccionó una altura de referencia de 1.5 mts, considerando el origen el punto desde donde despegó el cuadrirrotor, y este tendría que mantenerse en esa posición o alrededor de ese punto durante el tiempo de la prueba.

En la Figura 5.9 se muestran la referencia en x (línea punteada) y la trayectoria seguida por el cuadricóptero durante el tiempo de vuelo (línea continua), el vehículo estuvo aproximadamente dos minutos en vuelo durante los cuales siguió la trayectoria deseada con un error de ± 50 cms.

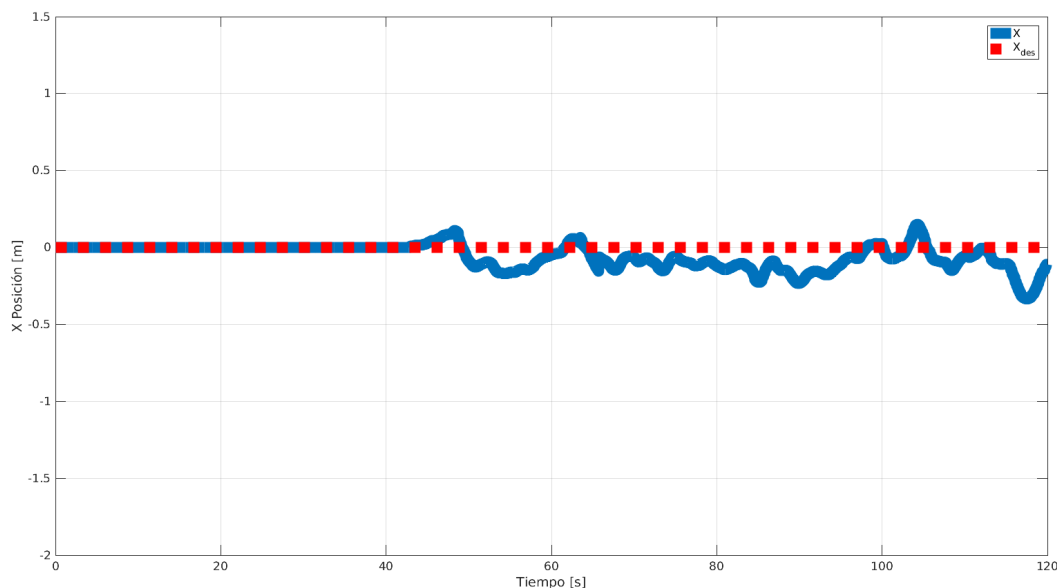


Figura 5.9: Seguimiento de la referencia en x con la cámara estereo.

Así mismo en la Figura 5.10 se muestran la referencia en y (línea punteada) y la trayectoria seguida por el cuadricóptero durante el tiempo de vuelo (línea continua), en este caso el vehículo siguió la trayectoria deseada con un error de casi ± 30 cms, fue un poco más exacta debido a que la carga estaba mejor balanceada en dicho eje.

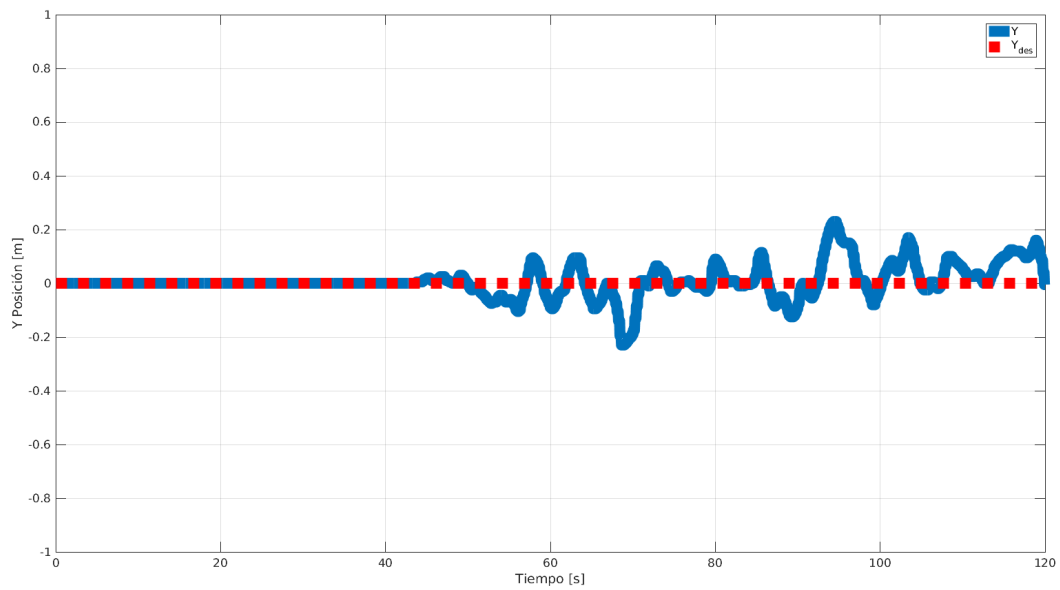


Figura 5.10: Seguimiento de la referencia en y con la cámara estereo.

Para el caso del seguimiento en el eje z mostrado en la Figura 5.11 el seguimiento de la referencia se realizó con el barómetro integrado en el autopiloto, este dispositivo muestra un mayor error que la cámara además de que el vehículo estaba volando en el límite de la carga que puede soportar, aquí el error fue de aproximadamente ± 70 *cms*.

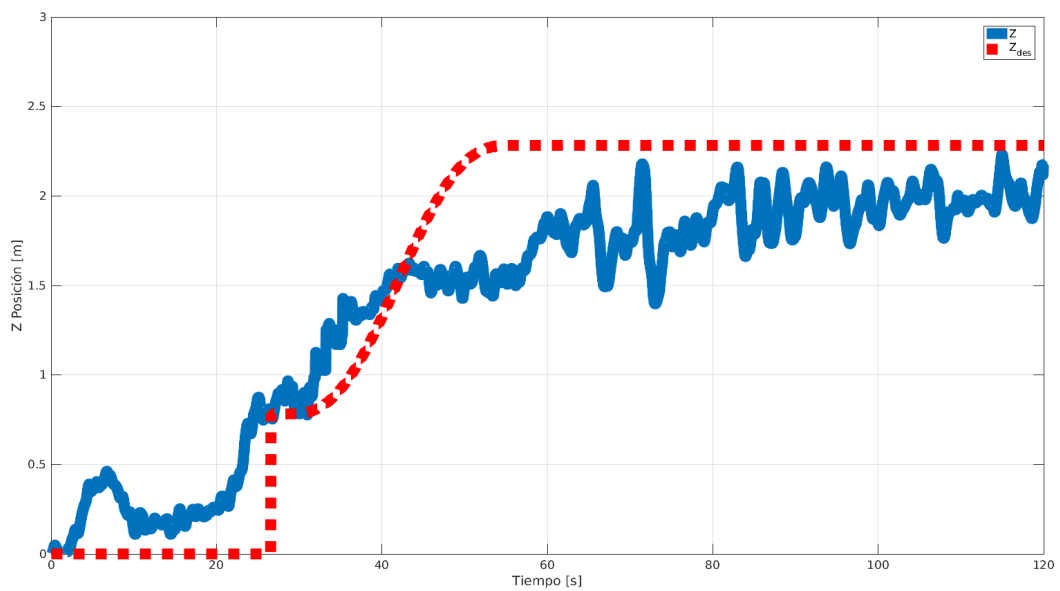


Figura 5.11: Seguimiento de la referencia en z con el barómetro del PX4.

Finalmente para este experimento en la Figura 5.12 se muestran las posiciones en (x, y) que el cuadricóptero mantuvo durante la prueba, los errores fueron aproximadamente en $x = \pm 30 \text{ cms}$ y en $y^+ = +20 \text{ cms}$, $y^- = -40 \text{ cms}$, el vehículo se mantuvo en vuelo estacionario dentro de un círculo de 50 cms de radio.

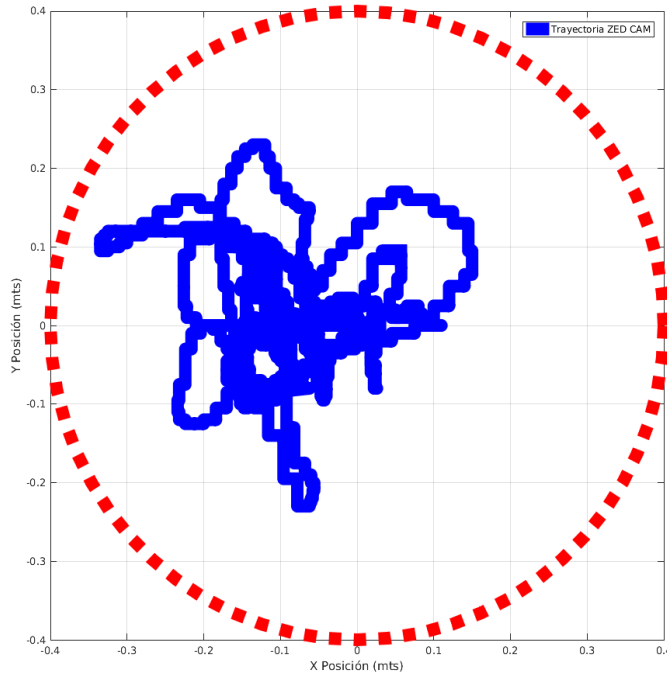


Figura 5.12: Seguimiento en Hover con la cámara estereo.

5.3.2. Experimento 2: *Seguimiento de una línea*

El segundo experimento consistió en que el cuadricóptero siguiera de manera autónoma una línea recta en el eje y a su vez que mantuviera una altura constante y una posición fija en el eje x . En las Figuras 5.13, 5.14 y 5.15 se muestran los resultados del seguimiento de la referencia en el eje x , y y z respectivamente.

Para el eje x se tuvo una referencia a 0 cms , para este caso el vehículo realizó el seguimiento de la referencia con un error de $\pm 30 \text{ cms}$.

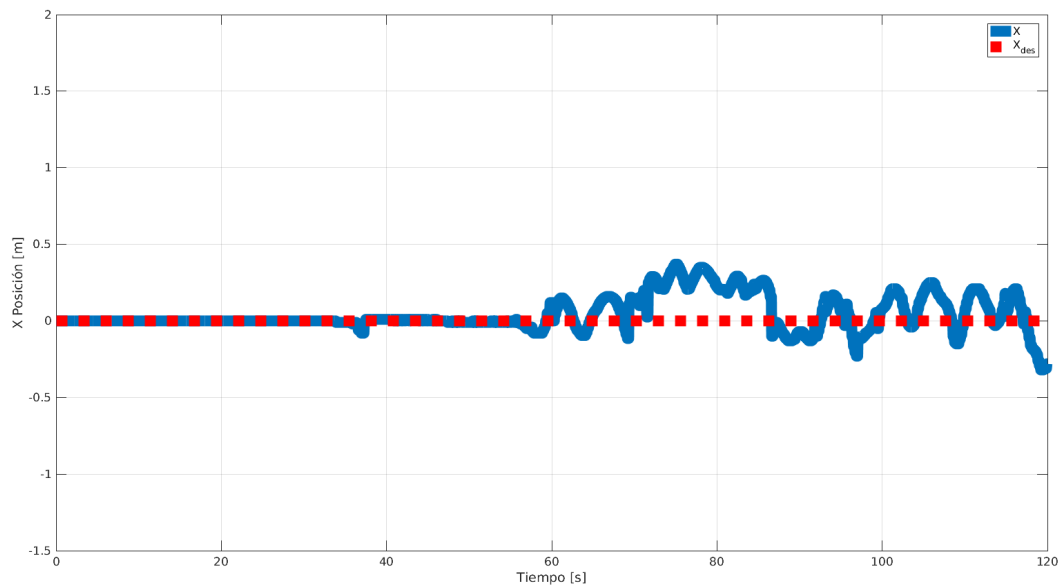


Figura 5.13: Seguimiento de la referencia en x con la cámara estereo.

Para el eje y el vehículo tenía que desplazarse una distancia de 3 *mts*, en este caso el seguimiento de la referencia se realizó con un error de aproximadamente ± 30 *cms*, logrando el vehículo una estabilización al finalizar su recorrido.

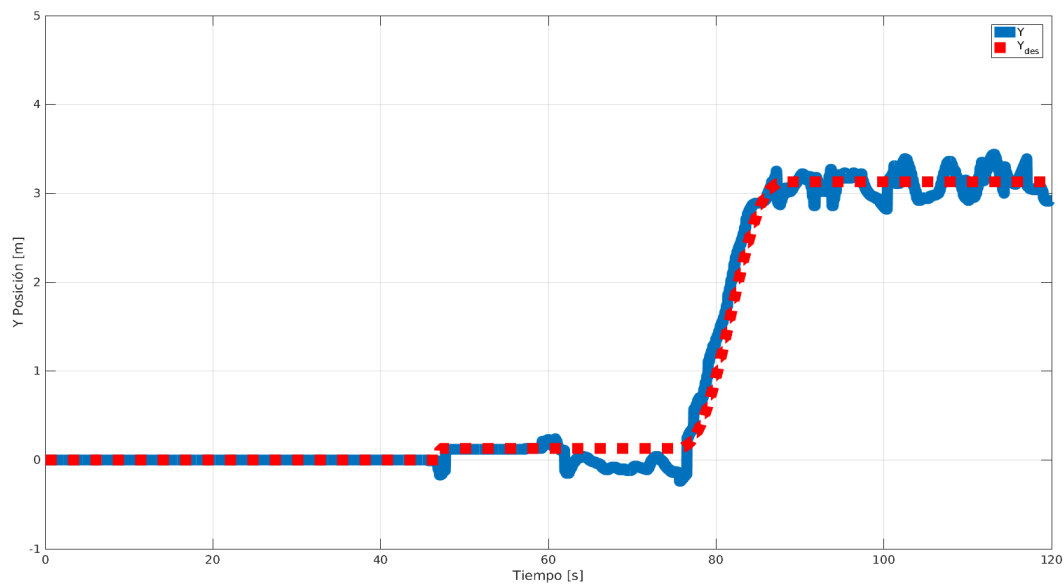


Figura 5.14: Seguimiento de la referencia en y con la cámara estereo.

Finalmente el cuadricóptero logró alcanzar y estabilizarse en la altura deseada, que para este caso era de 3.5 mts con un error mucho menor ($\pm 30 \text{ cms}$) con respecto al experimento anterior.

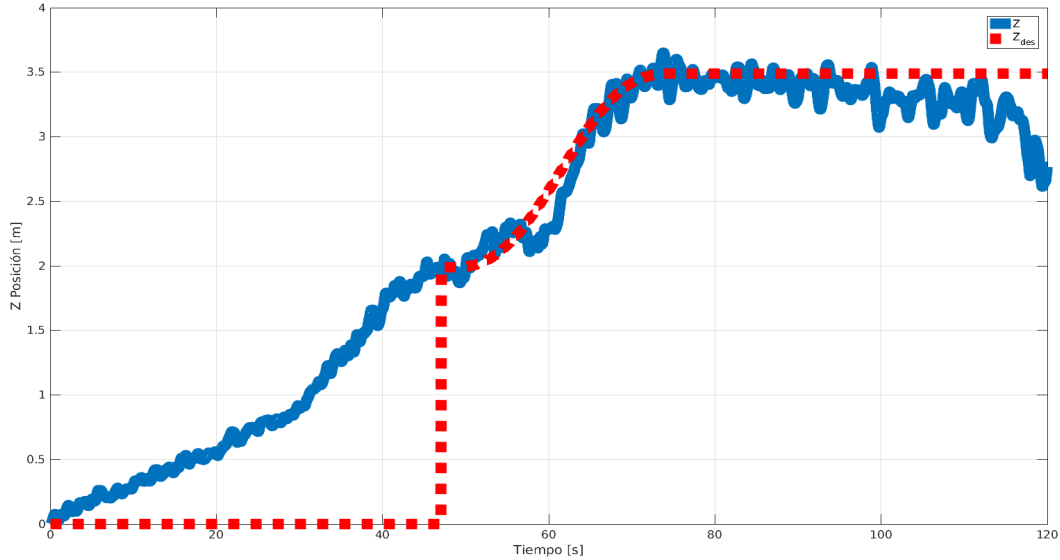


Figura 5.15: Seguimiento de la referencia en z con el barómetro del PX4.

5.3.3. Experimento 3: *Seguimiento de una trayectoria*

Como último experimento se probó una trayectoria deseada en los ejes x y y , en este caso se programó una trayectoria cerrada (cuadrado). Para llevar a cabo este recorrido el vehículo tuvo el siguiente orden:

1. Desplazamiento de 2 *mts* en el eje $+x$.
2. Desplazamiento de 3 *mts* en el eje $+y$.
3. Desplazamiento de 2 *mts* en el eje $-x$.
4. Desplazamiento de 3 *mts* en el eje $-y$.

En la primera y tercera parte de su recorrido el vehículo se tenía que desplazar 2 *mts* en el eje $+x$ (hacia el frente) y en el eje $(-x)$ (hacia atrás), la referencia fue seguida por el vehículo con un error de $\pm 50 \text{ cms}$, como se muestra en la Figura 5.16.

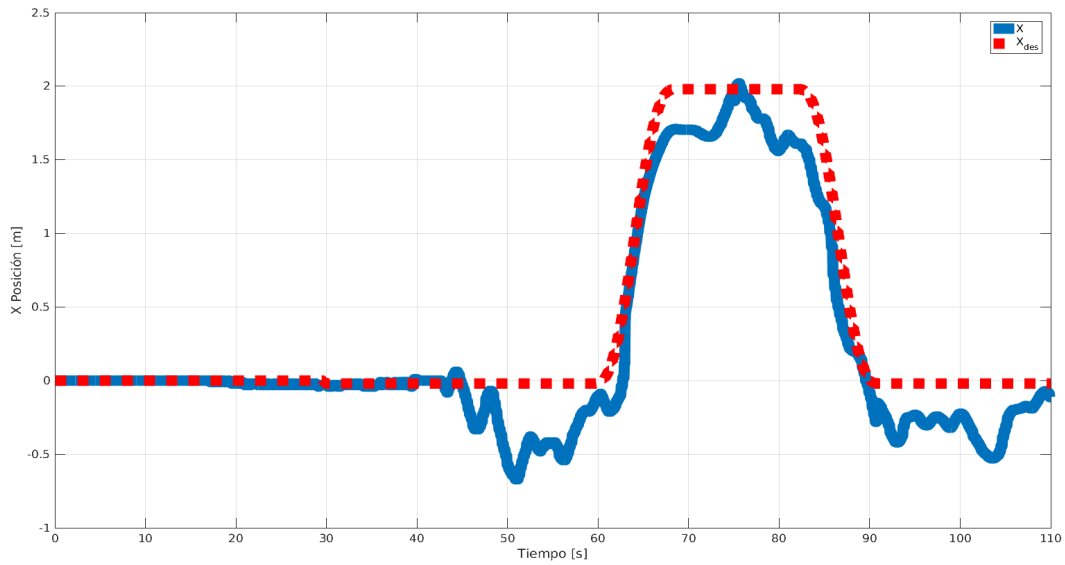


Figura 5.16: Seguimiento de la referencia en x con la cámara estereo.

La segunda y cuarta parte del recorrido consistió en un desplazamiento de 3 *mts* en el eje $+y$ (hacia la derecha) y eje $-y$ (hacia la izquierda), como en este eje la carga estaba mejor balanceada en el vehículo este siguió la referencia con un error de ± 20 *cms*, como se muestra en la Figura 5.17.

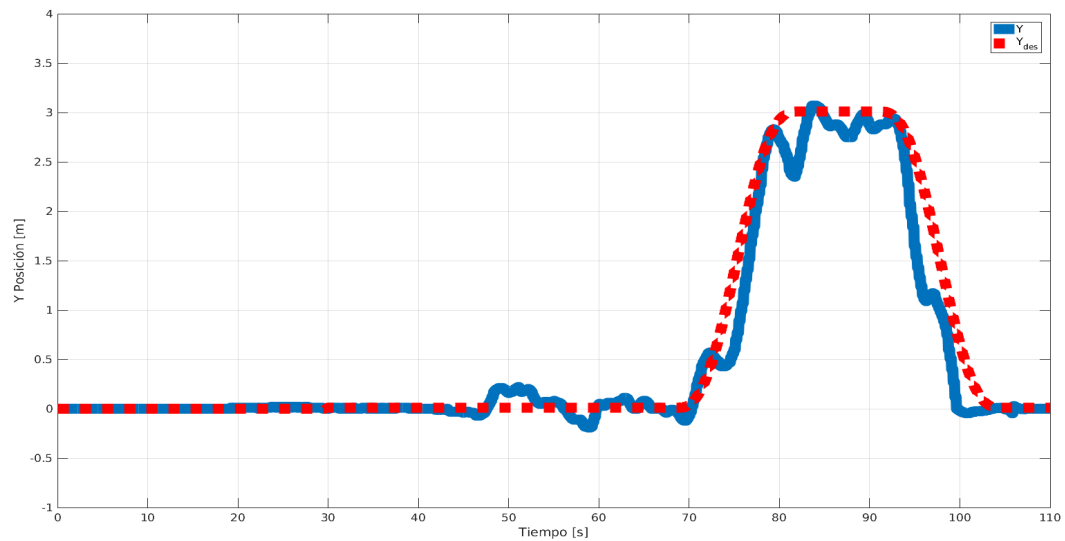


Figura 5.17: Seguimiento de la referencia en y con la cámara estereo.

Finalmente la referencia en el eje z (altura) el vehículo la siguió con un error de ± 60 *cms* debido a que la carga límite que puede soportar el vehículo, como se muestra en la Figura 5.18.

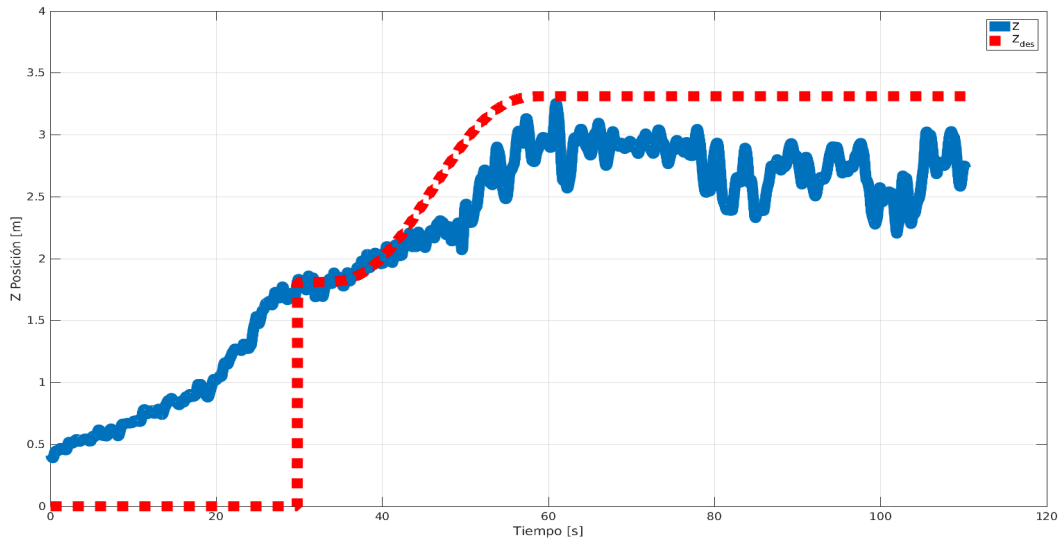


Figura 5.18: Seguimiento de la referencia en z con el barómetro del PX4.

En la Figura 5.19 se muestra el seguimiento de la trayectoria deseada desde una perspectiva $X - Y$, el vehículo trazó un cuadrado con un error no mayor a los 50 *cms*.

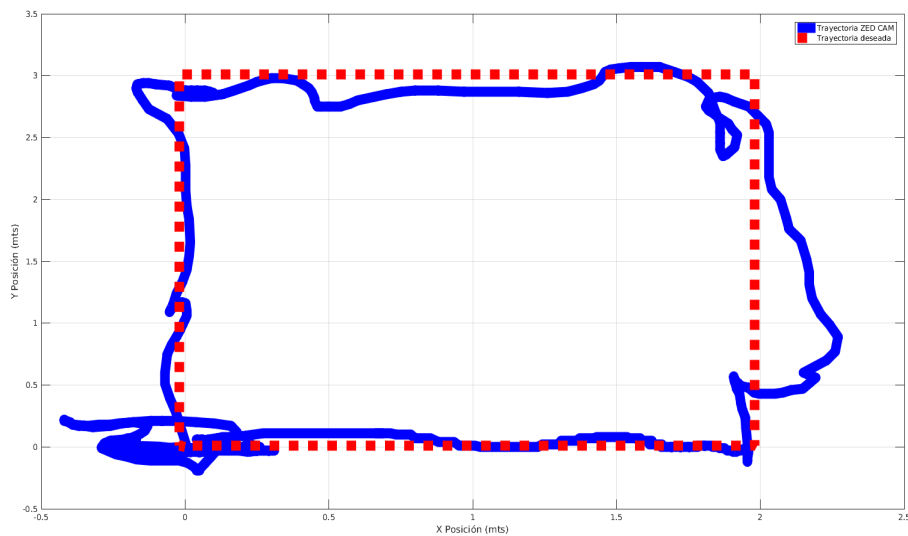


Figura 5.19: Seguimiento de la referencia en xy con la cámara estéreo.

En la Figura 5.20 se muestra el lugar en donde se llevaron a cabo las pruebas de odometría visual.



Figura 5.20: Área de pruebas con el cuadricóptero en el CINVESTAV-IPN.

CAPÍTULO 6

Conclusiones y trabajo futuro

6.1. Conclusiones

- Los resultados en simulación y experimentales del control PD nominal elegido fueron los esperados con el objetivo de lograr el seguimiento de trayectorias del vehículo aéreo no tripulado cuadrirotor.
- Se probó que el método de odometría visual tiene ventajas en cuanto a la obtención de la posición de un vehículo (terrestre, aéreo o submarino) en espacios donde la señal del GPS tiene difícil o nulo acceso.
- Entre las principales desventajas de la odometría visual con respecto a la tecnología GPS están que el costo computacional es mucho mayor con respecto a otros métodos de ubicación, el costo monetario para la adquisición del equipo necesario para el procesamiento de imágenes es superior al de un GPS comercial.

6.2. Trabajo futuro

- Para mejorar el seguimiento de las trayectorias deseadas y un mejor alcance de los parámetros deseados durante la navegación se pretende utilizar un controlador PID o un PD robusto para minimizar las perturbaciones.

- Después de las pruebas realizadas se recomienda cambiar el frame actualmente utilizado (Tarot™650 sport) por otro más robusto que pueda soportar una carga superior a los 1.5 Kg que todo el equipo utilizado en la odometría visual representa, el Tarot™680 Pro Hexacopter y el 960 Hexacopter pueden ser buenas opciones.
- Implementación de técnicas y algoritmos de nubes de puntos y SLAM para la detección y evasión de obstáculos durante la navegación.

ANEXOS A

Códigos

A.1. Lectura de la pose

En las Figuras A.1 y A.2 se muestra el código escrito para la lectura de la pose generada por la cámara y su escritura a un archivo **.txt*

```
#include <ros/ros.h>
#include <geometry_msgs/Twist.h>
#include <tf/transform_datatypes.h>
#include <nav_msgs/Odometry.h>
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
using namespace std;

double qt_x = 0, qt_y = 0, qt_z = 0, qt_w = 0;
double pos_x = 0, pos_y = 0, pos_z = 0;

void chatterCallback(const nav_msgs::Odometry::ConstPtr& msg)
{
    pos_x = msg->pose.pose.position.x;
    pos_y = msg->pose.pose.position.y;
    pos_z = msg->pose.pose.position.z;

    qt_x = msg->pose.pose.orientation.x;
    qt_y = msg->pose.pose.orientation.y;
    qt_z = msg->pose.pose.orientation.z;
    qt_w = msg->pose.pose.orientation.w;
}
```

Figura A.1: Función para guardar la posición lineal y orientación del UAV.

```

int main(int argc, char **argv)
{
    ros::init(argc, argv, "listener");

    ros::NodeHandle n;

    //Suscripción al tópico que publica la pose de la cámara
    ros::Subscriber sub = n.subscribe("/zed/odom", 1000, chatterCallback);

    ros::Publisher chatter_pub = n.advertise<geometry_msgs::Twist>("pose", 1000);

    ros::Rate loop_rate(10);

    ofstream myfile;

    //Ubicación del archivo ".txt" creado
    myfile.open("/media/nvidia/ZED/pose.txt");

    geometry_msgs::Twist pos;

    while (ros::ok())
    {
        //Conversion de cuaterniones a angulos de Euler
        tf::Quaternion q(qt_x, qt_y, qt_z, qt_w);
        tf::Matrix3x3 m(q);
        double roll, pitch, yaw;
        m.getRPY(roll, pitch, yaw);

        //Angulos de Euler en radianes
        pos.angular.x = pitch;
        pos.angular.y = roll;
        pos.angular.z = yaw;

        pos.linear.x = pos_x;
        pos.linear.y = pos_y;
        pos.linear.z = pos_z;

        //Escritura de la posición lineal al archivo ".txt"
        myfile <<fixed<<setprecision(3)<<pos.linear.x<<"\t"<<pos.linear.y<<"\t"<<pos.linear.z<<"\n";

        chatter_pub.publish(pos);

        ros::spinOnce();

        loop_rate.sleep();
    }

    myfile.close();

    return 0;
}

```

Figura A.2: Suscripción y escritura del tópico leído de la pose de la cámara.

A.2. Envío de datos al autopiloto

En las Figuras A.3 y A.4 se muestra el código escrito para el envío de la pose obtenida de la cámara al autopiloto por puerto serial.

```

#include <ros/ros.h>
#include <serial/serial.h>
#include <std_msgs/String.h>
#include <geometry_msgs/Twist.h>
#include <std_msgs/Empty.h>
#include <nav_msgs/Odometry.h>
#include <diagnostic_msgs/DiagnosticArray.h>
#include <sstream>

using namespace std;

serial::Serial ser;

int16_t m_xPos,m_yPos,m_zPos;
uint16_t seq,seq_flag;
uint8_t rtk_flag = 5, level;
uint8_t dato_enviar[8];
string bb;
const char *a;

//Función de envío de datos al autopiloto
void poseCallback(const nav_msgs::Odometry::ConstPtr& msg)
{
    seq = msg->header.seq;
    m_xPos = (int)(msg->pose.pose.position.x*100);
    //"-y" al autopiloto para que las coordenadas de la cámara y el UAV coincidan.
    m_yPos = -(int)(msg->pose.pose.position.y*100);
    m_zPos = (int)(msg->pose.pose.position.z*100);
}

void flagCallback(const diagnostic_msgs::DiagnosticArray& msg)
{
    seq_flag = msg.header.seq;
    bb = msg.status[0].values[4].value;
    a = bb.c_str();
    rtk_flag = (uint8_t)(*a);
}

```

Figura A.3: Función para la obtención de los datos a enviar.


```

int main(int argc, char **argv)
{
  ros::init(argc, argv, "talker_zed");
  ros::NodeHandle n;
  ros::Subscriber sub = n.subscribe("zed/odom", 10, poseCallback);
  ros::Rate loop_rate(10);

  try
  {
    ser.setPort("/dev/ttyTHS2");
    ser.setBaudrate(57600);
    serial::Timeout to = serial::Timeout::simpleTimeout(1000);
    ser.setTimeout(to);
    ser.open();
  }

  catch (serial::IOException& e)
  {
    ROS_ERROR_STREAM("Unable to open port ");
    return -1;
  }

  if(ser.isOpen())
    ROS_INFO_STREAM("Serial Port initialized");
  else
    return -1;

  dato_enviar[0] = 124;
  dato_enviar[1] = 236;

  while (ros::ok())
  {
    ros::spinOnce();

    // Posicion en X
    dato_enviar[2] = (unsigned int) m_xPos >> 8;
    dato_enviar[3] = (unsigned int) m_xPos & 0x00FF; //Dato de la pose
    // Posicion en Y
    dato_enviar[4] = (unsigned int) m_yPos >> 8;
    dato_enviar[5] = (unsigned int) m_yPos & 0x00FF;
    // Posicion en Z
    dato_enviar[6] = (unsigned int) m_zPos >> 8;
    dato_enviar[7] = (unsigned int) m_zPos & 0x00FF;

    ser.write(dato_enviar,8);

    loop_rate.sleep();
  }
  return 0;
}

```

Figura A.4: Inicialización del puerto serial y envío de la pose al autopiloto.

BIBLIOGRAFÍA

- [1] C. FORSTER, M. P., AND SCARAMUZZA, D. Svo: Fast semi-direct monocular visual odometry. *IEEE International Conference on Robotics and Automation (ICRA)* (June 2014), pp. 15–22.
- [2] COMPUTER VISION GROUP, FACULTY OF INFORMATICS, T. U. O. M. Dso: Direct sparse odometry. <https://vision.in.tum.de/research/vslam/dso?redirect=1>, 2016.
- [3] CORPORATION, N. Nvidia jetson systems. <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/>, 2018.
- [4] FRAUNDORFER, F., AND SCARAMUZZA, D. Visual odometry : Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics Automation Magazine* 19, 2 (June 2012), p. 79.
- [5] FRAUNDORFER, F., AND SCARAMUZZA, D. Visual odometry : Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics Automation Magazine* 19, 2 (June 2012), p. 85.
- [6] FRAUNDORFER, F., AND SCARAMUZZA, D. Visual odometry : Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics Automation Magazine* 19, 2 (June 2012), p. 85.
- [7] GITHUB, I. Dev board uart and pixhawk connection. <https://github.com/NVIDIA-AI-IOT/redtail/wiki/Dev-board-UART-and-Pixhawk-connection>, 2018.

- [8] GLOBAL, I. <https://www.igi-global.com/dictionary/local-optimization/40000>, 2004.
- [9] GOLATO, M. A. Análisis de respuestas transitorias en sistemas de segundo orden. Universidad Nacional de Tucumán, Abril 2016.
- [10] GUILHERME, V. R. Descripción y modelado del helicóptero quadrotor. In *Modelado y control de un helicóptero quadrotor*. Universidad de Sevilla, 2007, ch. 2, p. 14.
- [11] HELIPAL. Tarot 4006 high power brushless motor (620kv). <http://www.helipal.com/tl68p02-4006-620kv-brushless-motor.html>, 2018.
- [12] HELIPAL. Tarot-frame-tl65s01. <http://www.helipal.com/tarot-650-sport-quadcopter-frame-set.html>, 2018.
- [13] HOBBYKING. Controlador afro esc 30amp del multi-rotor de velocidad del motor (simonk firmware). https://hobbyking.com/es_es/afro-30a-multi-rotor-esc-simonk-firmware.html, 2018.
- [14] HOBBYKING. Turnigy graphene panther 5000mah 4s 75c battery pack w/xt90. https://hobbyking.com/es_es/turnigy-graphene-5000mah-4s-75c-lipo-pack-w-xt90.html, 2018.
- [15] INDIAMART. Xl 6009 dc to dc step up converter. <https://www.indiamart.com/proddetail/xl-6009-dc-to-dc-step-up-converter-15307513673.html>, 2018.
- [16] KLEIN, G., AND MURRAY, D. Parallel tracking and mapping for small ar workspaces. pp. 225–234.
- [17] LOZADA, R. P. *Sistema de odometría visual monocular para robots móviles*. Universidad Nacional Autónoma de México, 2013, p. 1.
- [18] MATTHIES, L., AND SHAFER, S. A. Error modeling in stereo navigation. *IEEE Journal of robotics and automation* 3 (June 1987), pp. 239 – 248.

- [19] MILELLA, A., AND SIEGWART, R. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*. (2006), p. 21.
- [20] MORAVEC, H. P. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Carnegie-Mellon University, 1980.
- [21] NVIDIA. Nvidia jetson tx1/tx2 developer kit carrier board. In *JETSON TX1/TX2 / DEVELOPER KIT CARRIER BOARD / SPECIFICATION* (2017), NVIDIA corporation, p. 28.
- [22] OLSON, E. A primer on odometry and motor control. Tech. rep., MIT, December 2004.
- [23] PX4. Pixhawk 1 flight controller. https://docs.px4.io/en/flight_controller/pixhawk.html, 2018.
- [24] PX4. Pixhawk1 series hkpilot32 autonomous vehicle 32bit control set with gps. http://m.wowdron.com/product/zoom.html?product_no=998, 2018.
- [25] RÍOS, A. C. *Estabilidad en los Sistemas de Ecuaciones Diferenciales*, vol. 5. Revista de Educación Matemática, Agosto 1993, pp. 63–66.
- [26] SHUAIB, A. O., AND AHMED, M. M. Robust pid control system design using itae performance index(dc motor model). *International Journal of Innovative Research in Science, Engineering and Technology* 3 (August 2014), pp. 15060–15067.
- [27] STEREO LABS. Meet zed. <https://www.stereolabs.com/zed/>, 2018.
- [28] STEREO LABS. Zed stereo camera. <https://www.wearvr.com/hardware/cameras/zed-stereo-camera>, 2018.
- [29] TALLA MOHAMED NABIL ELKHOLY, H. System modeling. In *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches*. The American University in Cairo, 2014, ch. 3, p. 33.

- [30] TALLA MOHAMED NABIL ELKHOLY., H. System modeling. In *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches*. The American University in Cairo., 2014, ch. 4, p. 45.

- [31] VOELCKER, J. Driverless cars. <https://sites.google.com/site/driverlesscars1234/darpa-urban-challenge>, 2007.