

Sistema de carga modular y aterrizaje de precisión para enjambre de drones

**José Alberto Castro Villasana, José Eduardo Castro Villasana,
Ana Bárbara Quintero García**

Una Tesis presentada a la Facultad de Ingeniería en Conformidad con los
Requisitos para el Grado de:

Ingeniero en Tecnologías Electrónicas y Robótica, Ingeniero en Mecatrónica



Universidad de Monterrey
Departamento de Ingeniería y Tecnologías
San Pedro Garza García, México

30 de noviembre 2024

Asesor: Fermín Castro Aragón

Contents

1	Introducción	4
1.1	Contexto General y Motivación	4
1.1.1	Contexto del Problema	4
1.1.2	Importancia del Problema	5
1.2	Planteamiento del Problema	5
1.2.1	Definición Clara del Problema	5
1.2.2	Preguntas de Investigación	5
1.3	Objetivos	6
1.3.1	Objetivo General	6
1.3.2	Objetivos Específicos	6
1.4	Justificación	6
1.4.1	Relevancia del Proyecto	6
1.4.2	Impacto Potencial	7
1.5	Justificación	7
1.5.1	Relevancia del Proyecto	7
1.5.2	Impacto Potencial	8
1.6	Alcance y Limitaciones	8
1.6.1	Alcance del Trabajo	8
1.6.2	Limitaciones del Estudio	8
1.7	Estructura de la Tesis	9
2	Estado del Arte	10
2.1	Estado del Arte	10
2.2	Hardware	10
2.2.1	Drones	10
2.2.2	Types of Drone Charging Stations	11
2.2.3	Automatic Drawer Mechanism	16
2.2.4	Motors and Propellers	17
2.3	Electronics	18
2.3.1	Movimiento a Motores y Módulos ESC	18
2.3.2	Controladores de Vuelo	18
2.3.3	Computadoras Complementarias y Sistemas Embebidos	19
2.3.4	Tipos de Protocolos de Comunicación	20
2.3.5	Sensores de Proximidad	20
2.4	Software	21
2.4.1	FC Firmware	21
2.4.2	Ground Control Systems	22

2.4.3	Companion Computer Operative Systems	23
2.4.4	ROS2 Distributions	24
2.4.5	Arquitectura de ROS 2	26
2.4.6	Computer Vision	27
2.4.7	What is an ArUco?	29
2.4.8	Aruco vs Embedded Aruco	31
2.4.9	Aruco Detection	33
2.4.10	Comunicación ROS2 - Pixhawk	35
2.4.11	Sensores para la Navegación en Ambientes Exteriores e Interiores	36
2.4.12	Fusión de Sensores usando el Filtro de Kalman en Pixhawk	38
3	Desarrollo	42
3.1	Diseño y Desarrollo de la Base de Carga	42
3.1.1	Especificaciones y Requerimientos	42
3.1.2	Selección e Implementación del Mecanismo de Movimiento del Cajón	42
3.1.3	Diseño CAD de la Base de Carga	43
3.1.4	Lista de Materiales	44
3.1.5	Proceso de Manufactura de la Estructura de la Base de Carga	45
3.1.6	Circuito Electrónico de la Estación de Carga	45
3.2	Instrumentación del Dron	46
3.2.1	Especificaciones y Requerimientos	46
3.2.2	Lista de Materiales para la Instrumentación del Dron	46
3.2.3	Diseños CAD del Dron	46
3.2.4	Círculo de Distribución de Energía	47
3.2.5	Integración de Componentes Electrónicos	47
3.3	Software Configuration	47
3.3.1	Configuración de la Computadora en Tierra	47
3.3.2	Configuración de la Computadora Auxiliar del Dron	48
3.3.3	Configuración de la Estación de Control en Tierra	48
3.4	Sistema de Visión	48
3.4.1	Especificaciones y Requerimientos	48
3.4.2	Calibración de la Cámara	48
3.4.3	Generación de Marcadores Aruco	49
3.4.4	Detección de Marcadores e-Aruco en Tiempo Real	49
3.5	Sistema de Comunicación	49
3.5.1	Especificaciones y Requerimientos	49
3.5.2	Protocolo de Comunicación entre el Dron y la Base de Carga	50
4	Resultados	51
4.1	Resultados del Diseño y Manufactura de la Base de Carga	51
4.1.1	Diseño CAD Final del 1er prototipo de la Estación de Carga	51
4.1.2	Diseño e Instrumentación del Drone	51
4.1.3	Movimiento del Cajón	51
4.1.4	Vuelo del Drone	51
4.1.5	Carga y Descarga de Batería	51
4.1.6	Detección del ArUco	51
4.1.7	Sistema de Comunicación	51
4.2	Ánalisis de Desempeño General	51

4.2.1	Resultados de Integración Global	51
5	Conclusiones	52

Chapter 1

Introducción

1.1 Contexto General y Motivación

1.1.1 Contexto del Problema

En los últimos años, los drones han evolucionado para convertirse en herramientas esenciales en una amplia gama de industrias. Estos vehículos aéreos no tripulados varían en tamaño y capacidad, pero comparten la característica de acceder a áreas difíciles o peligrosas, lo que ha impulsado su uso en sectores como la inspección de infraestructuras, la agricultura de precisión y las operaciones de vigilancia y seguridad. A nivel comercial, los drones están diseñados para realizar tareas repetitivas de manera autónoma, lo que ha revolucionado la eficiencia en dichas áreas.

En el ámbito de la inspección de infraestructuras, los drones son utilizados para monitorear y evaluar el estado de torres eléctricas, aerogeneradores, líneas de transmisión y otras construcciones críticas, permitiendo detectar fallos estructurales sin la necesidad de exponer a los trabajadores a riesgos innecesarios [?]. En agricultura, los drones facilitan la gestión de cultivos, monitoreo de riego y la topografía, lo que permite a los agricultores optimizar la productividad y reducir costos operativos [?]. En vigilancia y seguridad, se emplean para monitorear incendios forestales, realizar labores de seguridad fronteriza y evaluar daños tras desastres naturales [?].

A pesar de sus múltiples ventajas, uno de los mayores desafíos en la operación continua de drones es la duración limitada de la batería. En aplicaciones profesionales, los drones suelen tener una autonomía de vuelo entre 20 y 30 minutos, lo cual resulta insuficiente para operaciones prolongadas como la inspección de grandes áreas o la vigilancia a largo plazo [?]. Además, las condiciones climáticas adversas pueden afectar significativamente el rendimiento y estabilidad de los drones en vuelo, lo que limita aún más su capacidad operativa [?]. Estos factores hacen necesario interrumpir las operaciones para recargar las baterías, lo cual afecta tanto la eficiencia como los costos operativos.

Para abordar esta limitación, se han diseñado estaciones de carga automáticas que permiten recargar las baterías sin intervención humana. Sin embargo, las estaciones actuales presentan varios inconvenientes: la mayoría no están diseñadas para gestionar múltiples drones de manera

simultánea, lo que limita su eficiencia en operaciones a gran escala, como las que utilizan enjambres de drones [?]. Además, el aterrizaje preciso sobre estas estaciones sigue siendo un reto importante, lo que puede afectar el éxito del proceso de carga [?].

1.1.2 Importancia del Problema

Con el creciente uso de drones en sectores como la agricultura, la logística y la vigilancia, la capacidad de mantener operaciones prolongadas y autónomas se ha vuelto crucial. Estos sectores demandan drones capaces de completar misiones largas sin interrupciones frecuentes para la recarga de baterías [?].

Aquí es donde entra en juego la necesidad de un sistema de carga modular. En este contexto, la modularidad se refiere a la posibilidad de montar varios sistemas de carga en un mismo espacio físico, lo que permite a múltiples drones recargarse simultáneamente. Esto optimiza el uso de los recursos y maximiza la eficiencia operativa de los enjambres de drones, al reducir los tiempos de espera entre recargas [?]. Esta solución es especialmente valiosa en aplicaciones que requieren la coordinación de varios drones en tiempo real, donde cualquier retraso en la recarga podría impactar negativamente en la misión.

Además, un sistema de aterrizaje preciso es clave para garantizar que los drones se posen correctamente sobre las estaciones de carga, lo que permite automatizar completamente el proceso y reducir la necesidad de intervención humana. Este tipo de solución tiene el potencial de mejorar la eficiencia y autonomía de las operaciones con drones, abriendo nuevas posibilidades para su adopción en áreas donde las limitaciones actuales de autonomía y recarga son un obstáculo [?].

1.2 Planteamiento del Problema

1.2.1 Definición Clara del Problema

Uno de los mayores desafíos para la operación continua y autónoma de enjambres de drones es la falta de estaciones de carga modulares y autónomas que permitan la recarga simultánea de múltiples drones. Actualmente, las soluciones de carga disponibles en el mercado presentan limitaciones significativas, como la imposibilidad de gestionar de manera eficiente la carga de varios drones al mismo tiempo, y que aseguren un aterrizaje preciso en un mismo espacio sin intervención humana. Estas limitaciones no solo disminuyen la autonomía de los drones, sino que también afectan la eficiencia operativa, ya que requieren intervenciones frecuentes para maniobrar los drones de manera manual y requiere mayor espacio en casi de utilizar las soluciones que se encuentran actualmente en el mercado.

1.2.2 Preguntas de Investigación

- ¿Cómo se puede desarrollar una estación de carga modular que permita la carga simultánea de múltiples drones?
- ¿Qué tecnologías de localización y visión pueden ser implementadas para

asegurar un aterrizaje preciso? - ¿Cuales serían los recursos necesarios para implementar una estación de carga y aterrizaje de precisión en un enjambre de drones? - ¿Qué tipo de carga es la más adecuada para la carga de un dron en cuestión de eficiencia y rendimiento?

1.3 Objetivos

1.3.1 Objetivo General

Diseñar y manufacturar una base de carga para cuadricópteros de arquitectura abierta, desarrollar un sistema de comunicación entre el drone y la estación de carga, e instrumentar el drone con de sensores de localización y un sistema de visión, para asegurar una integración eficaz y segura entre el cuadricóptero y su sistema de carga.

1.3.2 Objetivos Específicos

- Realizar un diseño CAD para la base de carga, asegurando que sea compatible con cuadricópteros de arquitectura abierta.
- Manufacturar una base de carga para el enjambre de drones.
- Adaptar el diseño actual del dron de arquitectura abierta a la base de carga.
- Instrumentar un cuadricóptero con onboard computer, flight controller, cámara de visión and sensores de localización.
- Programar un sistema de comunicación entre la base de carga y el cuadricóptero para la trasnferencia de datos relevantes para su interacción.
- Programar un sistema que permita controlar el drone desde una onboard computer.
- Programar un sistema de visión por computadora que detecte en tiempo real la pose de un Aruco Marker embebido en la base de carga.

1.4 Justificación

1.4.1 Relevancia del Proyecto

La operación continua y autónoma de enjambres de drones enfrenta varios obstáculos, entre ellos, la falta de soluciones eficientes de carga que permitan la recarga simultánea de múltiples drones de manera autónoma y sin intervención humana. Aunque este proyecto no busca solucionar completamente este desafío, sí establece las bases fundamentales para desarrollar una solución viable. Al diseñar una estación de carga modular y autónoma con un sistema de aterrizaje preciso, este trabajo proporciona un punto de partida sobre el cual se pueden realizar futuras investigaciones y mejoras.

El proyecto es relevante para sectores como la logística, la vigilancia y la agricultura de precisión, donde la operación autónoma de drones puede generar grandes beneficios. Al presentar una solución modular que permite recargar varios drones en un mismo espacio físico, se sientan las bases para una futura implementación a mayor escala, lo que potencialmente podría reducir los costos operativos y mejorar la eficiencia de las operaciones [?]. Aunque no se pretende resolver el problema de manera completa en este trabajo, los avances presentados contribuirán al desarrollo de soluciones reales en el futuro.

1.4.2 Impacto Potencial

A pesar de que el sistema desarrollado en este proyecto no es una solución final, su implementación tiene el potencial de influir en futuras investigaciones y desarrollos en el campo de la autonomía de drones. Las tecnologías de aterrizaje preciso y carga simultánea que se exploran aquí podrían ser fundamentales para el diseño de soluciones más avanzadas y escalables, capaces de abordar las necesidades operativas de sectores que dependen cada vez más de los drones.

El impacto potencial de este proyecto se reflejará en la capacidad de inspirar investigaciones que mejoren la eficiencia de los enjambres de drones en aplicaciones como la entrega de mercancías, la vigilancia de grandes áreas y la optimización de recursos en la agricultura. Si bien las soluciones actuales enfrentan limitaciones, este trabajo ofrece una plataforma sólida sobre la cual se podrían implementar mejoras significativas en futuras etapas [?].

Memoria actualizada Gracias por la aclaración. Aquí tienes una versión ajustada de las secciones, teniendo en cuenta que el proyecto establece las bases para una solución real:

1.5 Justificación

1.5.1 Relevancia del Proyecto

La operación continua y autónoma de enjambres de drones enfrenta varios obstáculos, entre ellos, la falta de soluciones eficientes de carga que permitan la recarga simultánea de múltiples drones de manera autónoma y sin intervención humana. Aunque este proyecto no busca solucionar completamente este desafío, sí establece las bases fundamentales para desarrollar una solución viable. Al diseñar una estación de carga modular y autónoma con un sistema de aterrizaje preciso, este trabajo proporciona un punto de partida sobre el cual se pueden realizar futuras investigaciones y mejoras.

El proyecto es relevante para sectores como la logística, la vigilancia y la agricultura de precisión, donde la operación autónoma de drones puede generar grandes beneficios. Al presentar una solución modular que permite recargar varios drones en un mismo espacio físico, se sientan las bases para una futura implementación a mayor escala, lo que potencialmente podría reducir los costos operativos y mejorar la eficiencia de las operaciones [?]. Aunque no se pretende resolver el problema de manera completa en este trabajo, los avances presentados contribuirán al desarrollo de soluciones reales en

el futuro.

1.5.2 Impacto Potencial

A pesar de que el sistema desarrollado en este proyecto no es una solución final, su implementación tiene el potencial de influir en futuras investigaciones y desarrollos en el campo de la autonomía de drones. Las tecnologías de aterrizaje preciso y carga simultánea que se exploran aquí podrían ser fundamentales para el diseño de soluciones más avanzadas y escalables, capaces de abordar las necesidades operativas de sectores que dependen cada vez más de los drones.

El impacto potencial de este proyecto se reflejará en la capacidad de inspirar investigaciones que mejoren la eficiencia de los enjambres de drones en aplicaciones como la entrega de mercancías, la vigilancia de grandes áreas y la optimización de recursos en la agricultura. Si bien las soluciones actuales enfrentan limitaciones, este trabajo ofrece una plataforma sólida sobre la cual se podrían implementar mejoras significativas en futuras etapas [?].

1.6 Alcance y Limitaciones

1.6.1 Alcance del Trabajo

Este proyecto establece las bases para una solución modular y autónoma de recarga de drones. Se enfoca en el diseño, manufactura e instrumentación de una estación de carga que permita gestionar múltiples drones simultáneamente, utilizando tecnologías de visión por computadora y sensores de localización. El trabajo se llevará a cabo dentro de un entorno controlado y se validará en condiciones simuladas para asegurar que la solución es viable desde un punto de vista técnico.

Aunque el proyecto no resuelve el problema de la carga autónoma a gran escala, proporciona un primer paso hacia una solución más avanzada. Se presentará el diseño de un sistema de carga modular que permita recargar varios drones en el mismo espacio, sentando así las bases para desarrollos posteriores, y físicamente se realizará un prototipo funcional de uno de los modulos de carga para un drone. Las pruebas se realizarán en un cuadricóptero de arquitectura abierta con algunas modificaciones en el diseño que ayudarán a la carga y a la localización de los sensores de visión y localización. [?].

1.6.2 Limitaciones del Estudio

Este estudio se limita a la validación técnica del sistema en un entorno controlado y con un tipo de drone específico (cuadricóptero de arquitectura abierta). No se abordará la escalabilidad a otros tipos de drones ni su implementación en condiciones climáticas adversas. Además, el proyecto no incluye pruebas prolongadas ni la implementación en aplicaciones comerciales.

Las tecnologías de visión y localización implementadas también estarán limitadas a las condiciones de prueba, y su precisión en escenarios más complejos no será evaluada en este trabajo. El

objetivo es demostrar la viabilidad técnica de los componentes clave, más que ofrecer una solución final completamente funcional [?].

1.7 Estructura de la Tesis

El capítulo 2 presentará una revisión detallada del estado del arte en sistemas de carga para drones abarcando temas de mecánica, electrónica y programación. El capítulo 3 describirá la metodología empleada para el desarrollo del prototipo. El capítulo 4 discutirá los resultados obtenidos durante las pruebas, y el capítulo 5 incluirá las conclusiones y recomendaciones para futuros trabajos.

Chapter 2

Estado del Arte

2.1 Estado del Arte

El propósito de este capítulo es brindar una visión clara y detallada de los trabajos e investigaciones previas en el campo relacionado con la carga de drones. Se presenta una revisión de la literatura, las tecnologías existentes y las limitaciones identificadas que justifican la necesidad de la investigación propuesta.

2.2 Hardware

2.2.1 Drones

There are different categories or names for what can be considered a drone, such as UAV (Unmanned Aerial Vehicle), UAS (Unmanned Aerial Systems), RPAS (Remotely Piloted Aircraft System), or Multicopter (Figure 2.1). A specific definition is given by the official documentation [?], “A multicopter is a mechanically simple aerial vehicle whose motion is controlled by speeding or slowing multiple downward thrusting motor/propeller units.” It is known that each category has distinct functionalities, and therefore, a multicopter is not exactly the same as a UAV. According to the official documentation [?], “A multicopter becomes a UAV or Drone when it is capable of autonomous flight.”

Additionally, according to Piotr et al., ”Drones or Unmanned Aerial Systems (UAV - Unmanned Aerial Vehicle or UAS - Unmanned Aerial Systems) are the aircrafts, which are able to fly without a pilot and passengers on board” [?]. Another definition comes from The Office of the Secretary of Defense of the United States of America: “A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or non-lethal payload.”

There are various configurations of drones, each with different capabilities and specific applications. Figure 2.2 shows the Quad Frame, one of the most common and basic configurations. Quadrotor drones, with four motors and propellers, are ideal for applications like photography,



Figure 2.1: Multicopter.

surveillance, and light tasks.

Figure 2.3 presents the Hexa Frame, with six motors and propellers, offering greater stability and payload capacity. These drones are suitable for tasks requiring the transport of heavier equipment, such as advanced sensors or high-resolution cameras.

In Figure 2.4, the Octo Frame configuration is shown, featuring eight motors and propellers. Drones with this configuration are ideal for industrial applications due to their higher payload capacity and redundancy, which improves safety in case of motor or propeller failures.

Finally, Figure 2.5 shows the Octo Quad Frame, combining the advantages of eight motors with the efficiency of a compact configuration. These types of drones are used when high payload capacity and precise stability control are required.

2.2.2 Types of Drone Charging Stations

Currently, there is a growing trend towards the concept of charging or docking stations for drones. According to Grlj et al., "A docking station for UAVs is a multipurpose system that enables them to land safely, take off, recharge and/or replace batteries, and transfer data and payload" [?].

In this context, various types of charging stations can be observed, classified according to the following specifications [?]:

- Mobility (fixed and mobile)
- Charging method (two electrodes, multiple electrodes, wireless, etc.)
- Automatic battery exchange (recharging spare batteries)
- Positioning (active and passive)
- Drone storage (yes or no)

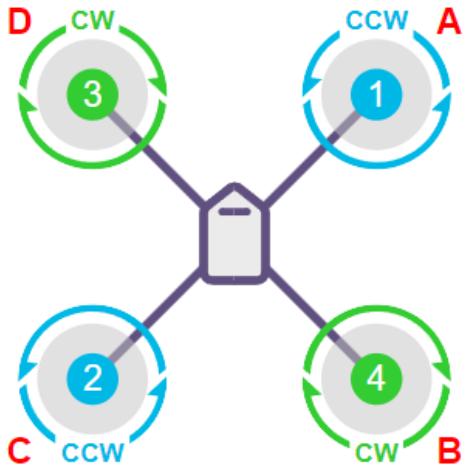


Figure 2.2: Quad X Frame.

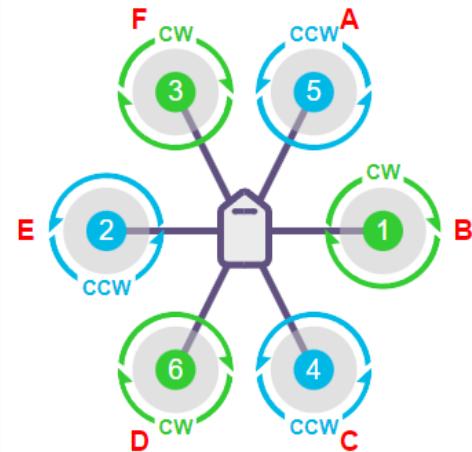


Figure 2.3: Hexa X Frame.

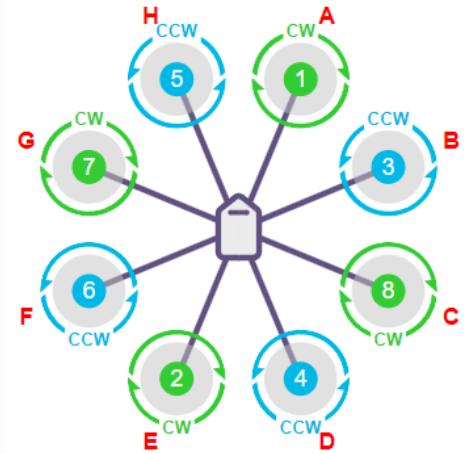


Figure 2.4: Octo X Frame.

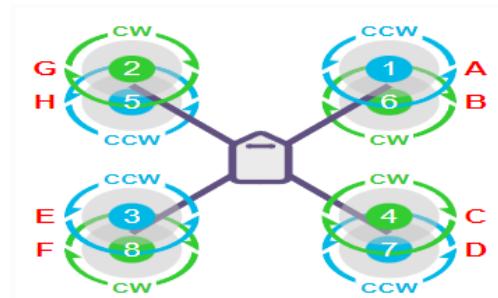


Figure 2.5: Octo Quad X Frame.

Figure 2.6: Drone configurations.

- Package delivery (with storage, without storage)
- Type of landing (precision, vision-based, etc.)
- Type of landing platform (self-leveling)

Fixed Charging Stations

The main objective of fixed charging stations is to provide a stable platform that allows drones to recharge without needing to move the infrastructure during the operation process. These stations are often simpler in design and are placed in strategic locations, such as cities or industrial areas. According to Grlj et al., many of these fixed stations use an active or passive positioning system

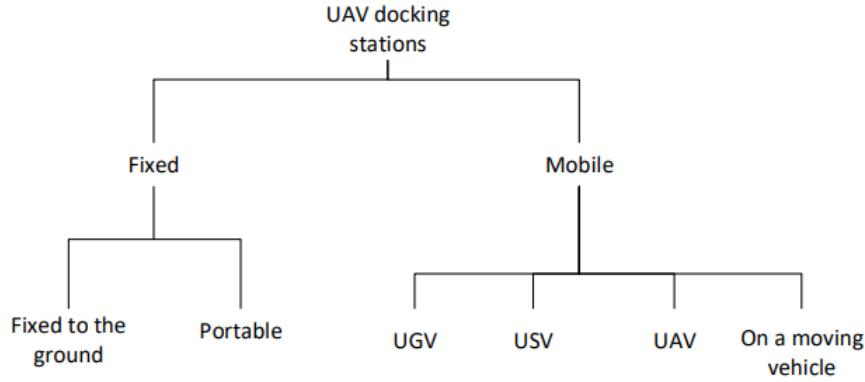


Figure 2.7: Drone Charging Stations Classification [?].

to ensure the drone lands accurately, and they may feature charging options through electrical contacts or even wireless charging [?].

Advantages of Fixed Charging Stations

Fixed stations are particularly useful in situations where the drone needs a precise landing in a constant location. These stations can use visual markers, such as ArUco [2] markers, to guide the drone during the final landing phase, ensuring high accuracy in the maneuver. Additionally, they may include self-leveling platform systems to guarantee the stability of the drone upon landing, especially on uneven terrain.

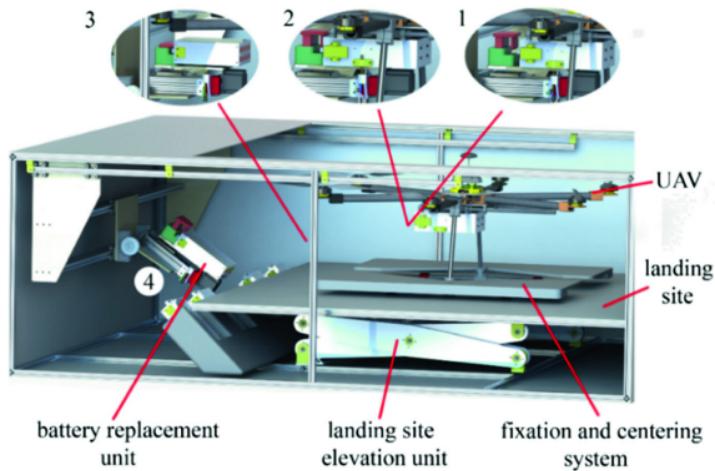


Figure 2.8: Fixed docking station with its subsystems. The labels (1–4) show the battery replacement mechanism in operation [?].



Figure 2.9: Battery replacement docking station [?].

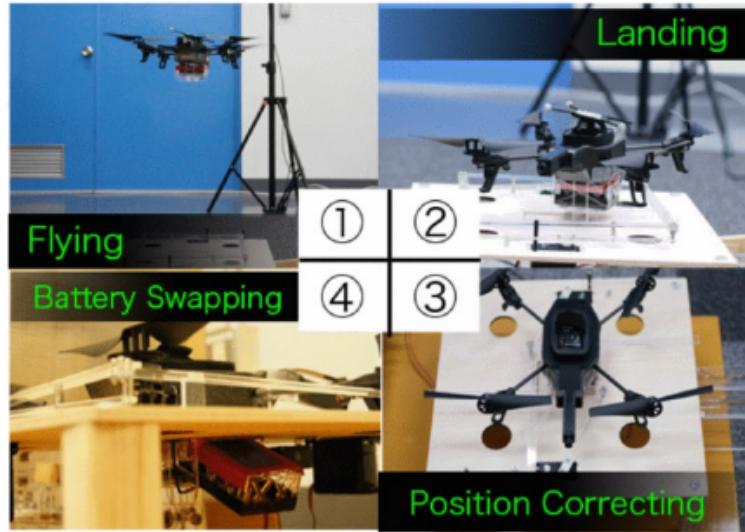


Figure 2.10: Endless Flyer [?], docking station with battery replacement mechanism. The labels in the center show the different landing stages and the order in which they are realized.

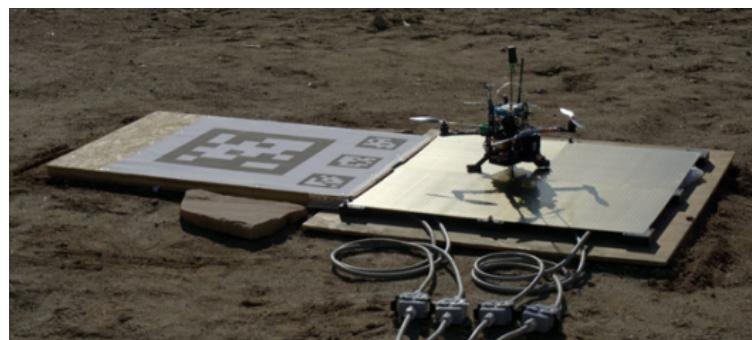


Figure 2.11: Docking station with recharging pad with the custom made UAV from [?]. In the figure, the QR codes used in the approach and landing phase can be seen .

Mobile Charging Stations

Mobile charging stations, on the other hand, provide greater operational flexibility, as they are mounted on surface vehicles, such as Unmanned Ground Vehicles (UGVs) or even Unmanned

Surface Vehicles (USVs). These stations extend the range of drones by moving to locations that require real-time logistical support, such as in search and rescue operations [?].

Advantages of Mobile Charging Stations

Mobile stations provide a dynamic alternative for supporting drones in missions that require continuous movement. For example, some stations are mounted on autonomous ground vehicles that can move to locations where the drone requires assistance, significantly increasing the operational range of the UAV. These types of stations can also incorporate complex systems to exchange batteries, such as robotic arms that ensure the safe removal and insertion of the drone's battery without needing to pause the mission for long recharge periods.

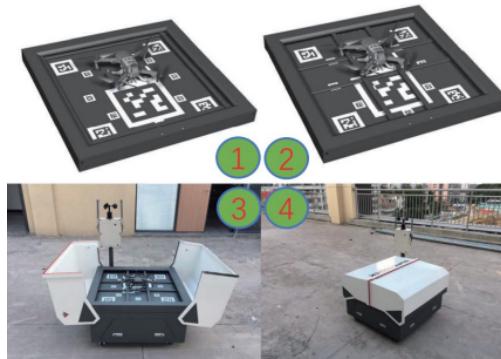


Figure 2.12: Moving docking station with storage system and battery replacement system. Labels in the figure (1–4) show the order in which the active positioning mechanism and storing mechanism take place [?].

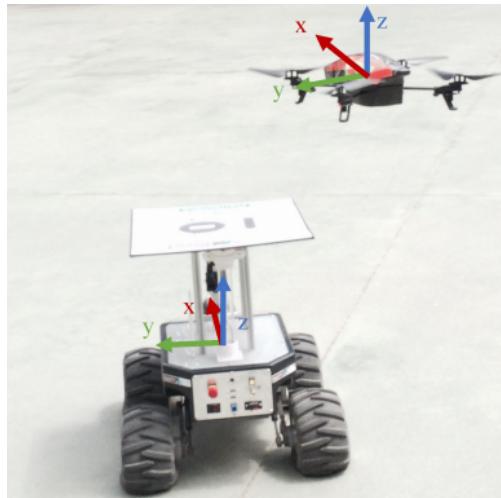


Figure 2.13: UGV–UAV proposed moving docking station solution.



Figure 2.14: USV–UAV landing system.

2.2.3 Automatic Drawer Mechanism

Types of Mechanisms

The automatic drawer mechanism in the charging station for drone storage is similar to the mechanisms found in other automatic storage solutions. Here, we'll discuss the main mechanisms that could be used in this context, including the advantages and disadvantages of each. The mechanism chosen for this project was the belt and V-pulley due to its easy accessibility and low cost.

Pistons Pistons are a common choice for automated movement in many types of machinery. They use compressed air or hydraulic fluid to extend and retract, making them well-suited for precise and strong movements. For an automatic drawer for drones, pistons could provide a robust and stable means to open and close the drawer. However, this approach may have higher costs and complexity compared to other mechanisms, as it requires pneumatic or hydraulic infrastructure and regular maintenance.

Belt and Toothed Pulley This mechanism uses a toothed belt and pulley to move the drawer. It offers greater precision compared to V-belts because the toothed design prevents slipping, making it a suitable choice for environments where precise positioning is critical. However, the increased complexity and cost of toothed belts make them less accessible, and they may require more maintenance due to the increased friction between the teeth.

Belt and V-Pulley This mechanism involves a belt shaped like a V and a pulley, which ensures good friction and prevents slipping while moving the drawer. The belt and V-pulley mechanism is simple, inexpensive, and easy to source, which made it the choice for this project. It offers a balance between precision and accessibility, being a practical solution for the needs of an automatic drone storage drawer. Its main disadvantage is that it may lack the precise positioning capabilities of toothed pulleys, but for the current application, this is acceptable.

2.2.4 Motors and Propellers

Motors

Motors are an essential component for the flight of a drone, and they are typically brushless motors due to their high efficiency and durability. According to the guide, the selection of the motor depends on factors such as the drone's size, total weight, and the desired type of flight. Key considerations include:

- **Kv Rating:** Indicates the revolutions per minute (RPM) per applied volt without load. A lower Kv provides more torque, making it ideal for larger drones with bigger propellers.
- **Compatibility:** The motor must be compatible with the electronic speed controller (ESC) and the propellers.
- **Mounting:** The drone's frame design should allow for a secure installation of the motors.

Propellers

Propellers generate the necessary lift for flight. According to the guide, they are a critical factor for the drone's performance and should be selected based on the motors and the size of the drone. Important characteristics include:

- **Size (diameter and pitch):** Larger propellers generate more lift but require more powerful motors. The "pitch" determines how much air the propeller displaces per revolution.
- **Material:** Propellers can be made of plastic, wood, or carbon fiber. Carbon fiber propellers are lighter and stronger, making them ideal for high-performance drones.
- **Balancing:** Poorly balanced propellers can cause vibrations, affecting stability and the quality of sensor data.

LiPo Batteries

LiPo (Lithium Polymer) batteries are the primary energy source for most drones. They are lightweight, high-capacity, and capable of delivering the current required by the motors and other components. Key aspects include:

- **Cells (S):** Each cell has a nominal voltage of 3.7V. For example, a 3S battery has 11.1V ($3 \times 3.7V$). The number of cells must be compatible with the motors and ESCs.
- **Capacity (mAh):** Determines the flight time. Higher capacity provides longer flights but also increases weight.
- **Discharge Rate (C):** Indicates how much current the battery can safely provide. It must meet the motors' requirements.

- **Charging and Safety:** LiPo batteries require specific chargers and must be handled carefully to prevent fires.

2.3 Electronics

2.3.1 Movimiento a Motores y Módulos ESC

En los drones, los motores sin escobillas (*brushless*) requieren de módulos *Electronic Speed Controller* (ESC) que permitan controlar su velocidad y dirección de rotación con precisión. Los ESC convierten la corriente directa (DC) de la batería en la corriente alterna (AC) que los motores necesitan, ajustando la frecuencia y voltaje de la señal para controlar su velocidad de giro.

El funcionamiento del ESC depende de la tecnología de modulación de frecuencia, donde la tasa de cambio determina la velocidad del motor. Como señala Rohde y Schwarz, “los ESC modernos no solo regulan la velocidad, sino que también monitorean la temperatura y el voltaje, protegiendo el motor contra sobrecargas” [?]. En el contexto de un sistema de carga modular para drones, estos ESC permiten un control ágil y seguro del sistema de propulsión, esencial en las maniobras de aterrizaje y acoplamiento en la estación de carga.

2.3.2 Controladores de Vuelo

Los controladores de vuelo son los componentes centrales que gestionan la estabilidad, navegación, y la ejecución de comandos de vuelo en los drones. Existen varios modelos en el mercado, cada uno con sus características específicas y aplicaciones recomendadas. Entre los controladores más relevantes destacan:

Principales Controladores de Vuelo

- **DJI Naza:** Este controlador es ampliamente utilizado en drones comerciales debido a su facilidad de configuración y su sistema de estabilización mejorado. Sin embargo, presenta limitaciones en personalización y flexibilidad para proyectos de investigación avanzada.
- **APM (ArduPilot Mega):** La plataforma APM, desarrollada por ArduPilot, es una opción altamente personalizable y adecuada para proyectos de código abierto. No obstante, se enfrenta a restricciones en términos de potencia de procesamiento, limitando su uso en aplicaciones complejas.
- **Holybro Kakute:** Este controlador ofrece buena integración de sensores y es ideal para drones pequeños y de carrera, pero carece de las capacidades avanzadas de procesamiento necesarias en aplicaciones autónomas y de carga pesada.

Pixhawk 6X:

Entre todos los controladores mencionados, el **Pixhawk 6X** destaca como la opción más avanzada y robusta. Según el equipo de desarrollo de Pixhawk, “el Pixhawk 6X fue diseñado para proporcionar un alto nivel de procesamiento con la mayor compatibilidad para sensores avanzados y módulos de comunicación” [?]. Las principales ventajas del Pixhawk 6X incluyen:

- **Procesamiento y Estabilidad:** Equipado con procesadores de alto rendimiento que soporan algoritmos avanzados de estabilización y control de vuelo.
- **Modularidad:** Permite una integración fluida con computadoras complementarias y sistemas externos, lo cual es fundamental para aplicaciones autónomas.
- **Sensores Integrados:** Incluye un acelerómetro y giroscopio de alta precisión, además de soporte para sensores externos como magnetómetros y GPS.
- **Compatibilidad:** Altamente compatible con sistemas de comunicación como UART, I2C, y CAN, lo cual facilita su integración en sistemas complejos como estaciones de carga autónomas.

2.3.3 Computadoras Complementarias y Sistemas Embebidos

En aplicaciones avanzadas de drones, los controladores de vuelo a menudo necesitan el apoyo de sistemas embebidos adicionales, conocidos como computadoras complementarias o *companion computers*, para gestionar procesamiento de datos intensivo, algoritmos de inteligencia artificial, o transmisión de video en tiempo real.

Principales Computadoras Complementarias

- **Raspberry Pi 4 y 5:** La **Raspberry Pi 4** y su sucesora, la **Raspberry Pi 5**, son opciones económicas y potentes. Gracias a su procesador de 64 bits y sus 4 GB o más de RAM, estas computadoras pueden ejecutar sistemas operativos complejos y manejar procesamiento intensivo, como procesamiento de imágenes y algoritmos de reconocimiento de patrones. Según Raspbian, la Raspberry Pi 5 tiene un 30% más de rendimiento comparado con la versión anterior, haciéndola ideal para drones que requieren alto rendimiento a bajo costo [?].
- **Odroid XU4:** Con un procesador Exynos 5422 octa-core, el Odroid XU4 ofrece un alto rendimiento en aplicaciones de procesamiento en tiempo real. Sin embargo, su consumo energético es mayor, lo cual limita su uso en aplicaciones de bajo consumo como drones ligeros.
- **Jetson Nano:** Esta computadora, desarrollada por NVIDIA, incluye una GPU integrada que facilita tareas de inteligencia artificial y visión computacional. Aunque su potencia en procesamiento de gráficos es sobresaliente, su costo y complejidad de integración superan los beneficios en aplicaciones donde el costo es un factor limitante.

- **Arduino:** En el contexto de una estación de carga, el Arduino es ideal para manejar tareas específicas de bajo procesamiento, como el control de motores de posicionamiento. Su simplicidad y bajo consumo energético lo hacen adecuado para tareas de control periférico, complementando bien a una Raspberry Pi que maneje la gestión central de la estación.

2.3.4 Tipos de Protocolos de Comunicación

Para conectar de manera efectiva los diferentes sistemas y módulos, es necesario implementar protocolos de comunicación estandarizados que aseguren una transmisión confiable de datos. En la estación de carga modular, estos protocolos permiten la comunicación entre la Raspberry Pi, el Arduino, y el controlador Pixhawk, creando un sistema integrado.

Principales Protocolos de Comunicación

- **UART (Universal Asynchronous Receiver-Transmitter):** Ideal para conexiones simples de bajo costo, el UART permite la transmisión punto a punto de datos asíncronos entre dispositivos, lo cual es útil para comunicación básica entre la Raspberry Pi y el Arduino.
- **SPI (Serial Peripheral Interface):** Este protocolo permite una comunicación sincrónica de alta velocidad y es adecuado para aplicaciones que requieren transferencia de datos rápida y simultánea entre la Raspberry Pi y el Pixhawk.
- **I2C (Inter-Integrated Circuit):** Con solo dos líneas (SDA y SCL), I2C permite la conexión de múltiples dispositivos, haciéndolo ideal para la integración de sensores y periféricos adicionales en el sistema de carga.

2.3.5 Sensores de Proximidad

Para medir con precisión la posición de componentes móviles como el cajón de la estación de carga, se utilizan sensores de proximidad que detectan cuando el cajón está completamente extendido o retraído. Estos sensores garantizan una operación segura y evitan daños en el sistema al proporcionar retroalimentación precisa al controlador.

Tipos de Sensores de Proximidad

- **Interruptores de Límite (Limit Switches):** Estos sensores mecánicos detectan la posición final del cajón mediante contacto físico y son ideales para aplicaciones de alta confiabilidad y bajo costo.
- **Sensores Inductivos:** Son capaces de detectar objetos metálicos sin necesidad de contacto, lo cual es útil en aplicaciones donde se desea evitar el desgaste mecánico o en entornos con mucho movimiento.

- **Sensores Capacitivos y Ópticos:** Adecuados para detectar objetos no metálicos, ofrecen una detección precisa y son comúnmente empleados en aplicaciones de precisión alta.

2.4 Software

En esta sección, hablaremos de todos los temas relacionados con software que se han investigado para el desarrollo de este proyecto. Se abordarán temas como los firmwares de los controladores de vuelo, las estaciones de control en tierra, los sistemas operativos de las computadoras auxiliares, documentación de ROS 2, visión por computadora, marcadores ArUco, entre otros.

2.4.1 FC Firmware

La definición de firmware según ... es [], en el caso de los controladores de vuelo, el firmware es el software que se ejecuta en el controlador de vuelo y se encarga de gestionar los sensores, actuadores y algoritmos de control necesarios para mantener la estabilidad y control del vehículo. Existen varios firmwares de controladores de vuelo disponibles en la actualidad, cada uno con sus propias características y ventajas. En este proyecto, se han investigado dos de los firmwares más populares y ampliamente utilizados en la industria de los drones y vehículos no tripulados: PX4 y ArduPilot.

Firmware PX4

El firmware PX4 ha sido una pieza fundamental en el desarrollo de sistemas de control de vuelo para drones y vehículos no tripulados, destacándose por su naturaleza de código abierto y su flexibilidad para muchas aplicaciones. Según la documentación oficial, “*PX4 is a powerful open source autopilot flight stack running on the NuttX RTOS*” [1]. Este sistema es ampliamente reconocido por su arquitectura modular, que permite la integración de nuevos sensores, actuadores y algoritmos de control, facilitando adaptaciones específicas para cumplir con los requisitos de diferentes proyectos [1].

Entre sus características más destacadas, se encuentra su capacidad para soportar una amplia variedad de tipos de vehículos. En este sentido, PX4 “*supports many different vehicle frames/types, including: multicopters, fixed-wing aircraft (planes), VTOLs (hybrid multicopter/fixed-wing), ground vehicles, and underwater vehicles*” [1]. Esta capacidad de soportar múltiples configuraciones es esencial, ya que permite aplicar el mismo marco de trabajo a diversas plataformas con modificaciones mínimas.

Además, PX4 es parte integral de un ecosistema más amplio que incluye estaciones de control en tierra como QGroundControl, Mission Planner y hardware específico como Pixhawk. La documentación menciona que “*PX4 is a core part of a broader drone platform that includes the QGroundControl ground station, Pixhawk hardware, and MAVSDK for integration with companion computers, cameras, and other hardware using the MAVLink protocol*” [1]. Esta integración asegura una comunicación fluida entre los diferentes componentes del sistema, lo cual es crucial para el monitoreo en tiempo real y la planificación de misiones.

Finalmente, PX4 ofrece “*flexible and powerful flight modes and safety features*” que son vitales para proyectos que requieren maniobras complejas, como el aterrizaje autónomo preciso en estaciones de carga [1]. Estas funcionalidades avanzadas de control y su capacidad de integrarse con tecnologías de visión por computadora y sensores externos consolidan a PX4 como una herramienta clave en el desarrollo de tecnologías de alta relevancia para vehículos autónomos.

Firmware ArduPilot

ArduPilot, al igual que PX4, es un software de código abierto que se ejecuta en una amplia variedad de hardware, permitiendo la creación y uso de sistemas autónomos de vehículos no tripulados para diferentes aplicaciones. Según la documentación oficial, “*ArduPilot provides a comprehensive suite of tools suitable for almost any vehicle and application. As an open source project, it is constantly evolving based on rapid feedback from a large community of users*” [2]. Esta flexibilidad y capacidad de adaptación han convertido a ArduPilot en una herramienta esencial para proyectos de automatización y robótica que buscan una solución versátil y robusta.

Uno de los aspectos más destacados de ArduPilot es que, a diferencia de PX4, éste no fabrica hardware, por lo cual “*ArduPilot firmware works on a wide variety of different hardware to control unmanned vehicles of all types*” [2]. Esto significa que puede integrarse con distintos tipos de controladores, sensores y dispositivos, transformando prácticamente cualquier máquina móvil en un vehículo autónomo con la simple adición de un paquete de hardware adecuado.

Como se mencionó anteriormente, el firmware es el código que se ejecuta en el controlador, y la elección del tipo de firmware depende del vehículo y la misión. Como se menciona en la documentación, “*You choose the firmware to match your vehicle and mission: Copter, Plane, Rover, Sub, or Antenna Tracker*” [2]. Esta versatilidad permite a los desarrolladores seleccionar la configuración más apropiada para sus necesidades específicas, optimizando así el proceso de desarrollo y operación.

ArduPilot también se complementa con estaciones de control en tierra (GCS), que funcionan como la interfaz entre el usuario y el controlador del vehículo. Una de las herramientas más completas en este ámbito es Mission Planner, descrita como “*a full-featured GCS supported by ArduPilot*”, que ofrece interacción fácil y rápida con el firmware al usuario [2].

Por último, mencionar que tanto PX4 como Ardupilot, tienen funcionalidades similares como la calibración de sensores, planificación de misiones, configuración de parámetros, entre otros. Sin embargo, la elección de uno u otro dependerá de las necesidades específicas del proyecto y de la compatibilidad con el **hardware** disponible.

2.4.2 Ground Control Systems

Las estaciones de control en tierra (GCS, por sus siglas en inglés) son herramientas auxiliares ampliamente utilizadas para la configuración, calibración y monitoreo de drones. Aunque no son el foco principal de este proyecto, desempeñan un papel importante en la preparación del sistema,

facilitando la configuración de parámetros, la calibración de sensores y la supervisión durante vuelos de prueba.

Entre las herramientas utilizadas se encuentran QGroundControl y Mission Planner, que destacan por su compatibilidad con los firmwares PX4 y ArduPilot, respectivamente. Estas plataformas proporcionan interfaces intuitivas para ajustar parámetros del vehículo, planificar misiones, y visualizar datos de telemetría en tiempo real. En el contexto de este proyecto, han sido clave para garantizar que el dron esté correctamente configurado antes de integrarse con la estación de carga modular.

A pesar de su utilidad durante las etapas de preparación y pruebas, el sistema propuesto no depende directamente de estas herramientas para su funcionamiento final. La operación autónoma del dron y la estación de carga se basa en la interacción directa entre el firmware, los sistemas de visión, y los controladores de vuelo, sin necesidad de intervención continua desde una estación de control en tierra.



Figure 2.15: Interfaz de QGroundControl.



Figure 2.16: Interfaz de Mission Planner.

Figure 2.17: Estaciones de control en tierra utilizadas durante las etapas de configuración y pruebas.

2.4.3 Companion Computer Operative Systems

Para poder correr una companion computer se necesita elegir dentro de las multiples opciones de sistemas operativos disponibles, algunos de los más populares son Ubuntu y Raspberry Pi OS. A continuación, se describen las características y ventajas de cada uno de estos sistemas operativos.

Ubuntu

Ubuntu es un sistema operativo de código abierto basado en Linux que ha ganado popularidad por su estabilidad y amplia comunidad de soporte, es importante saber que dentro de Ubuntu y sus versiones, las versiones pares son las que tienen mayor estabilidad. Según la documentación oficial de Ubuntu, este sistema es “*designed for security, reliability, and ease of use*” [4]. En el contexto de los companion computers para drones y otros vehículos autónomos, Ubuntu se utiliza frecuentemente debido a su compatibilidad con herramientas de robótica como ROS (Robot Operating System), lo

que facilita la integración y desarrollo de software avanzado de control y automatización.

Ubuntu ofrece soporte para arquitecturas ARM, lo que permite su instalación y ejecución en dispositivos como la Raspberry Pi 4 y 5. Esta capacidad es fundamental para proyectos que requieren procesamiento local eficiente, manejo de datos de sensores y comunicación en tiempo real. Adicionalmente, la flexibilidad de Ubuntu permite la personalización de su entorno para adaptarse a las necesidades específicas del proyecto, ya sea para ejecutar nodos de control de vuelo o procesamiento de imágenes en tiempo real [4].

Raspberry Pi OS

Raspberry Pi OS, es el sistema operativo oficial desarrollado y optimizado para dispositivos Raspberry Pi. La documentación de Raspberry Pi OS lo describe como “*a Debian-based operating system specifically tuned for the Raspberry Pi hardware*” [5]. Su principal ventaja es su optimización para el hardware de Raspberry Pi, lo que garantiza un rendimiento óptimo y un uso eficiente de los recursos disponibles.

Raspbian incluye una serie de herramientas preinstaladas que facilitan el desarrollo y prototipado, haciendo que sea una opción preferida para proyectos educativos y de investigación. La compatibilidad con Python y otras bibliotecas de programación facilita la implementación de scripts y software necesarios para muchas aplicaciones de robótica.

En comparación con otros sistemas operativos, Raspbian es ligero y permite un **arranque rápido**, lo cual es beneficioso en escenarios donde se requiere un inicio rápido del sistema.

2.4.4 ROS2 Distributions

Las distribuciones de ROS 2 (Robot Operating System 2) proporcionan entornos estandarizados para el desarrollo de aplicaciones robóticas, cada una adaptada a diferentes necesidades y capacidades de hardware. Una de las ventajas de ROS 2 es su capacidad para **facilitar la comunicación entre distribuciones** mediante el uso de topics, lo que permite que nodos en distintas versiones de ROS 2 se comuniquen y colaboren de manera efectiva en un mismo proyecto [6]. En la siguiente sección se abordarán más detalles sobre las distribuciones de ROS 2 más recientes y estables.

Distribución Humble

La distribución **ROS 2 Humble Hawksbill** es conocida por ser una versión con **soporte a largo plazo (LTS)**, lo que garantiza actualizaciones constantes y confiables. Esta distribución está diseñada para proyectos que requieren estabilidad y una alta compatibilidad con diferentes sistemas y paquetes. Es ideal para hardware como la **Raspberry Pi 4** y es compatible con **Ubuntu 22.04** y versiones anteriores, como Ubuntu 20.04, lo que la hace accesible para configuraciones de hardware más estándar.

De acuerdo con la documentación oficial de ROS 2, Humble es una de las versiones más recomendadas para proyectos que buscan consistencia a lo largo del tiempo, debido a su enfoque en

evitar cambios disruptivos [?].

Características principales:

- **LTS (Long-Term Support):** Asegura soporte prolongado para actualizaciones y corrección de errores.
- **Estabilidad:** Mejoras comprobadas en la comunicación entre nodos, garantizando confiabilidad.
- **Compatibilidad amplia:** Funciona con la mayoría de las bibliotecas y paquetes existentes en el ecosistema de ROS 2.
- **Optimización para ARM:** Ideal para plataformas como la Raspberry Pi 4, aprovechando recursos limitados de manera eficiente.

En general, Humble es la opción preferida para quienes priorizan estabilidad en sus proyectos de investigación o aplicaciones a largo plazo.

Distribución Jazzy

La distribución **ROS 2 Jazzy Jalisco**, más reciente que Humble, está diseñada para sacar ventaja del hardware más moderno, como la **Raspberry Pi 5**, y es compatible con sistemas operativos avanzados como **Ubuntu 24.04**. Esta distribución incluye mejoras significativas en rendimiento y nuevas funcionalidades, pero se considera más experimental en comparación con versiones LTS.

Según la documentación de Jazzy, su principal enfoque es facilitar el desarrollo de aplicaciones robóticas que aprovechen las últimas herramientas y simuladores, además de mejorar la velocidad en la comunicación entre nodos [?].

Características principales:

- **Nuevas funcionalidades:** Introducción de características experimentales y ajustes de rendimiento.
- **Menor latencia:** Comunicación más rápida entre nodos, clave para aplicaciones en tiempo real.
- **Integración avanzada:** Mejoras en simulación y depuración, aprovechando herramientas como Gazebo y Rviz.
- **Orientación al hardware moderno:** Diseñada para dispositivos como la Raspberry Pi 5, que ofrecen mayor capacidad de procesamiento.



Figure 2.18: ROS 2 Humble



Figure 2.19: ROS 2 Jazzy Jalisco

2.4.5 Arquitectura de ROS 2

ROS 2 (Robot Operating System 2) es un conjunto de bibliotecas y herramientas diseñadas para desarrollar aplicaciones de robótica. Es la evolución de ROS 1, creada para resolver problemas relacionados con la escalabilidad, seguridad y soporte a sistemas en tiempo real. ROS 2 utiliza una arquitectura basada en el middleware **DDS (Data Distribution Service)**, que facilita la comunicación entre nodos en sistemas distribuidos, asegurando interoperabilidad y un bajo tiempo de respuesta [6].

Conceptos básicos de ROS 2

ROS 2 está organizado en torno a varios elementos fundamentales que permiten la interacción entre las diferentes partes del sistema robótico. Estos elementos, que definen su arquitectura modular, son esenciales para comprender cómo se estructura y funciona un sistema basado en ROS 2 [6]:

- **Nodos:** Son las unidades básicas de ejecución en ROS 2. Cada nodo cumple una función específica, como recopilar datos de un sensor o controlar un actuador. Los nodos están diseñados para ser independientes y comunicarse entre sí mediante topics, servicios o acciones.
- **Messages (Mensajes):** Son estructuras de datos que se envían entre nodos para compartir información. Los mensajes tienen un formato definido que asegura que todos los nodos comprendan el contenido.
- **Topics:** Representan canales de comunicación para el intercambio de mensajes entre nodos. La comunicación mediante topics es asincrónica, lo que permite que los nodos publiquen y reciban información sin necesidad de esperar respuestas inmediatas.
- **Servicios (Services):** Permiten la comunicación síncrona entre nodos. A diferencia de los topics, los servicios funcionan en un esquema de solicitud y respuesta, lo cual es útil para operaciones específicas.

- **Actions:** Son similares a los servicios, pero están diseñados para operaciones de larga duración. Permiten a los nodos recibir actualizaciones del progreso de una tarea y cancelarla si es necesario.
- **Launch Files:** Facilitan la configuración y el inicio simultáneo de varios nodos. Esto simplifica la gestión de aplicaciones complejas, especialmente en entornos distribuidos.
- **Parámetros:** Son valores configurables que los nodos utilizan para ajustar su comportamiento sin necesidad de modificar el código.

La combinación de estos elementos permite que los sistemas robóticos diseñados en ROS 2 sean altamente escalables y flexibles, adaptándose tanto a proyectos pequeños como a sistemas complejos.

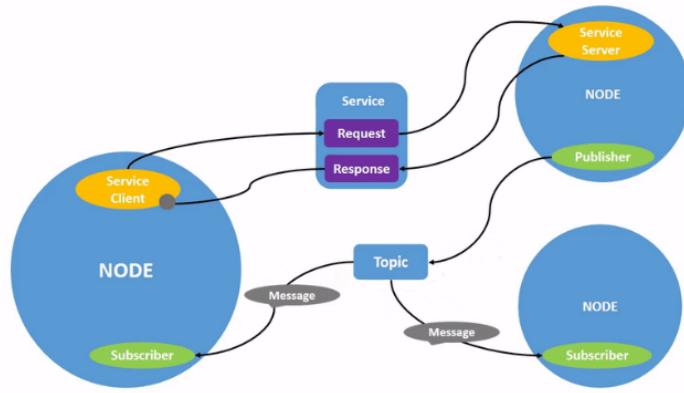


Figure 2.20: Arquitectura de ROS 2 [6].

Beneficios clave de ROS 2

La arquitectura de ROS 2 trae consigo varias ventajas que la hacen ideal para sistemas robóticos distribuidos y modernos. Entre los beneficios clave se encuentran los siguientes [6]:

2.4.6 Computer Vision

La visión por computadora es una rama de la programación que permite a los sistemas interpretar y procesar información visual del entorno. Este campo es clave en aplicaciones que requieren análisis de imágenes y videos en tiempo real, como la navegación autónoma de robots, detección de objetos y sistemas de inteligencia artificial. Para facilitar el desarrollo de estas aplicaciones, existen herramientas y bibliotecas especializadas como OpenCV, que juega un papel fundamental en la implementación práctica de visión por computadora [7].

Table 2.1: Beneficios clave de ROS 2

Beneficio	Descripción
Multithreading	Permite la ejecución de nodos en paralelo, optimizando el rendimiento.
Uso de topics	Comunicación asincrónica entre nodos para el intercambio de datos.
Nodos y servicios	Los nodos interactúan mediante servicios y topics para tareas específicas.
Archivos de lanzamiento (Launch Files)	Configuran y ejecutan múltiples nodos al mismo tiempo.
Comunicación rápida	Basada en DDS, proporciona comunicación de baja latencia entre nodos.
Mensajes personalizados	Permite definir y utilizar estructuras específicas para cada aplicación.
Soporte a tiempo real	Habilita aplicaciones críticas donde el tiempo de respuesta es esencial.

OpenCV Library

OpenCV (Open Source Computer Vision Library) es una de las bibliotecas más utilizadas en la industria y en la academia para el desarrollo de soluciones de visión por computadora. Según su documentación oficial, OpenCV es “*an open-source computer vision and machine learning software library containing more than 2500 optimized algorithms*” [7]. Estas herramientas permiten realizar tareas como procesamiento de imágenes, reconocimiento de objetos y seguimiento de movimiento, siendo compatibles con lenguajes de programación como Python y C++.

Gracias a su flexibilidad y facilidad de uso, OpenCV se adapta tanto a principiantes como a expertos. Además, sus algoritmos optimizados permiten ejecutar procesos en tiempo real, lo que la convierte en una herramienta ideal para proyectos de robótica y aplicaciones autónomas [7].

Camera Calibration

Antes de implementar cualquier sistema de visión por computadora en aplicaciones que demanden precisión espacial, es fundamental realizar una calibración adecuada de las cámaras utilizadas. Este proceso corrige las distorsiones inherentes a las lentes y permite obtener mediciones precisas del entorno. Según la documentación de OpenCV, “*Camera calibration is the process of estimating the parameters of the lens and the image sensor of an imaging device*” [?].

La calibración implica determinar parámetros intrínsecos (como la distancia focal y el punto principal) y extrínsecos (relativos a la posición y orientación de la cámara) que permiten mapear las coordenadas 2D de las imágenes capturadas a coordenadas 3D del mundo real.

OpenCV proporciona funciones que permiten realizar este proceso de manera eficiente mediante la detección de patrones en imágenes, como tableros de ajedrez o arreglos de círculos. El flujo típico de calibración incluye:

- Capturar imágenes de un patrón conocido desde diferentes ángulos.

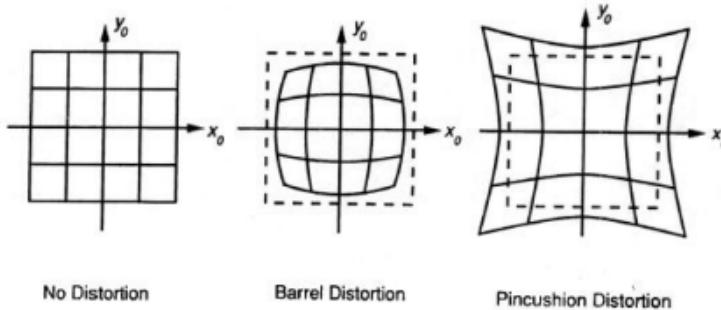


Figure 2.21: Tipos de distorsiones en las cámaras.

- Identificar puntos de interés en las imágenes capturadas (esquinas del patrón).
- Usar algoritmos de optimización para calcular parámetros intrínsecos y coeficientes de distorsión.

El resultado de la calibración permite corregir las distorsiones en imágenes y videos, mejorando la precisión de las aplicaciones de visión por computadora. Esto es especialmente útil en proyectos que requieren análisis espacial preciso, como la navegación autónoma y el posicionamiento de objetos en tiempo real.

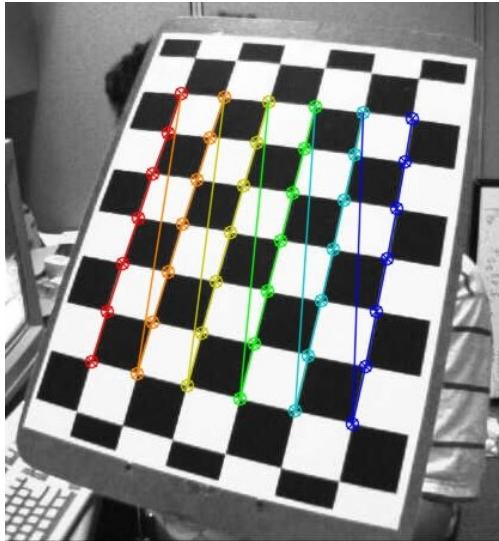


Figure 2.22: Patrón de tablero de ajedrez utilizado en la calibración de cámaras con OpenCV.

2.4.7 What is an ArUco?

ArUco, cuyo nombre proviene de la combinación de "Artificial" y "Uco" (por la Universidad de Córdoba, donde se desarrolló), es una biblioteca de código abierto ampliamente reconocida en el ámbito de la visión por computadora para la detección de marcadores fiduciales en imágenes. Esta

tecnología es fundamental para la estimación de la pose de la cámara con respecto a los marcadores cuando la cámara ha sido previamente calibrada. De acuerdo con la documentación, “*ArUco is an OpenSource library for detecting squared fiducial markers in images*” [8]. La detección de estos marcadores es crucial en aplicaciones que requieren una estimación precisa de la posición y orientación de objetos en el espacio tridimensional.

Historia de los Marcadores Aruco

Los marcadores ArUco se originaron como una solución para superar las limitaciones de otras tecnologías de detección de patrones, colores o figuras. El objetivo era desarrollar una técnica que proporcionara alta fiabilidad incluso bajo occlusiones parciales y condiciones de iluminación variables. Los primeros estudios se centraron en la generación automática de marcadores con un diseño que asegurara su unicidad y facilidad de detección. Estos marcadores están compuestos por un patrón binario rodeado de un borde negro, lo que mejora su visibilidad y robustez en diferentes condiciones de iluminación [8].

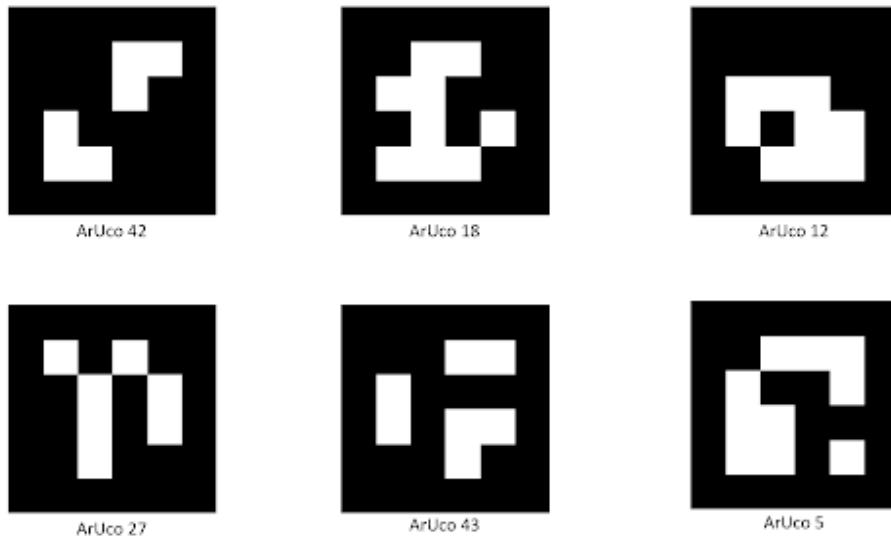


Figure 2.23: ArUco’s and ID’s examples

Aplicaciones Comunes

Los marcadores ArUco se utilizan en una variedad de aplicaciones, que incluyen la calibración de cámaras, la realidad aumentada, y el control y navegación de robots y drones. Una de las ventajas de usar ArUco es su capacidad para actuar como puntos de referencia en entornos 3D, permitiendo a los sistemas de visión por computadora calcular la pose de la cámara. Según la documentación, “*Markers can be used as 3D landmarks for camera pose estimation*” [?]. Esta característica hace que los marcadores sean esenciales en sistemas de seguimiento y posicionamiento donde la precisión es crítica.

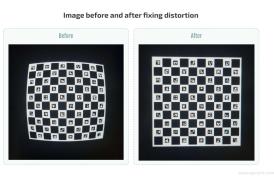


Figure 2.24: Camera Calibration with ArUco

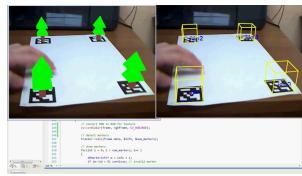


Figure 2.25: Augmented Reality with ArUco

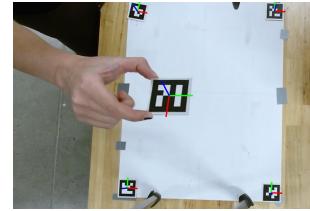


Figure 2.26: Real Time Tracking with ArUco

Figure 2.27: Common applications of ArUco markers.

Formatos de Marcadores

Los marcadores ArUco están compuestos por un borde negro externo y una región interna que codifica un patrón binario único. Dependiendo del diccionario que se esté utilizando, el número de bits en el marcador varía, lo que afecta la probabilidad de confusión con otros marcadores y la distancia de detección. Una mayor resolución de los marcadores permite que estos se detecten desde distancias más lejanas, pero puede requerir mayor procesamiento [8].

La biblioteca ArUco también soporta la creación de diccionarios personalizados, lo que permite a los desarrolladores adaptar los marcadores a las necesidades específicas de sus proyectos. “*The design of a dictionary is important since the idea is that their markers should be as different as possible to avoid confusions*” [?]. Esta flexibilidad es especialmente útil en proyectos donde es crucial mantener la unicidad y fiabilidad de la detección de marcadores en entornos complejos.

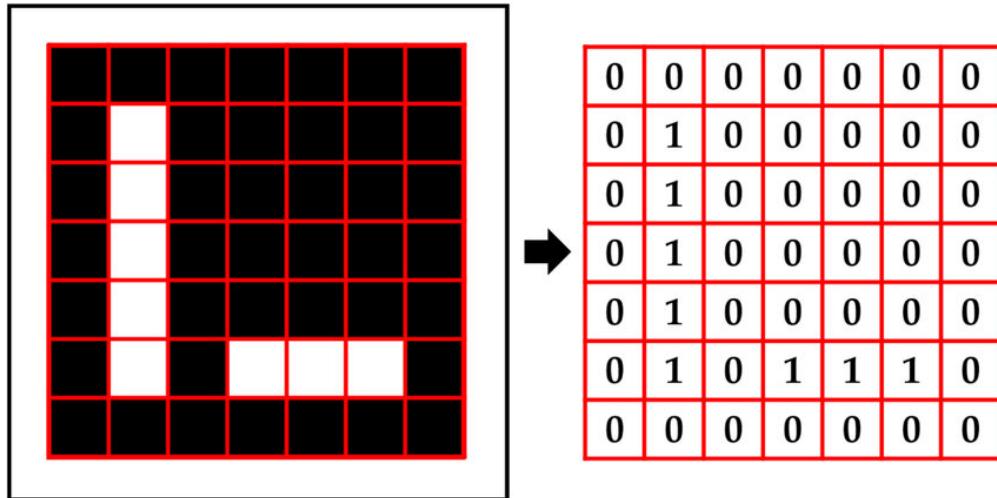


Figure 2.28: Bits of a 7x7 ArUco marker

2.4.8 Aruco vs Embedded Aruco

Los marcadores ArUco y los marcadores Embedded ArUco (e-ArUco) son tecnologías utilizadas en visión por computadora para tareas de detección y estimación de pose. Aunque ambos comparten

una base común en cuanto a su diseño y algoritmos de detección, tienen diferencias importantes que los hacen adecuados para distintas aplicaciones, especialmente en el contexto de operaciones de alta precisión.

Diferencias Principales

La principal diferencia entre los marcadores ArUco tradicionales y los marcadores Embedded ArUco radica en la optimización de estos últimos para detecciones de alta precisión en un rango amplio de distancias. Según Khazetdinov et al. (2021), “*a new type of fiducial marker called embedded ArUco (e-ArUco) was developed specially for a task of robust marker detection for a wide range of distances*” [9]. Los e-ArUco están diseñados para mantener su detectabilidad y precisión en escenarios donde los marcadores ArUco estándar podrían no ser tan efectivos, como cuando se requiere una precisión milimétrica en aplicaciones de aterrizaje de UAVs.

Otra diferencia significativa es que los marcadores e-ArUco están diseñados para mejorar la robustez de la detección, minimizando los errores que podrían surgir por condiciones de iluminación cambiantes y occlusiones parciales. Estos marcadores se basan en los algoritmos de detección de ArUco, lo que permite su implementación sin cambios sustanciales en los sistemas existentes basados en ArUco [9].



Figure 2.29: Generación de ArUco Grande / Chico

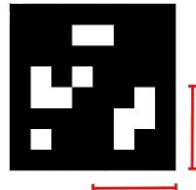


Figure 2.30: Cálculo del centro del ArUco Grande

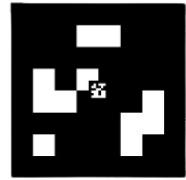


Figure 2.31: Sobreposición del ArUco Chico en el centro del Grande

Figure 2.32: Proceso de generación de marcadores e-ArUco.

Casos de Uso

Los marcadores ArUco tradicionales como se mencionó en la sección anterior, se utilizan comúnmente en aplicaciones de realidad aumentada, estimación de poses y navegación de robots y drones. Estos marcadores son versátiles y se pueden adaptar a diversas aplicaciones que no requieren una precisión extrema, siendo ideales para proyectos de localización, realidad aumentada, etc.

Por otro lado, los marcadores Embedded ArUco (e-ArUco) están específicamente diseñados para situaciones donde la precisión es más crítica. Un ejemplo destacado es el uso en el aterrizaje de UAVs (vehículos aéreos no tripulados), donde se necesita una detección precisa y confiable a diferentes distancias. En un estudio realizado por Khazetdinov et al., tuvo como resultado “*an average landing accuracy was 2.03 cm with a standard deviation of 1.53 cm*” al usar marcadores e-ArUco y un algoritmo de aterrizaje implementado en ROS y probado en el simulador Gazebo [9].

Esta capacidad hace que los e-ArUco sean ideales para entornos donde se requiere una precisión milimétrica, como en operaciones de aterrizaje autónomo de alta precisión.

2.4.9 Aruco Detection

La detección de marcadores ArUco es un proceso esencial en la visión por computadora que permite la identificación y estimación de la pose de los marcadores en imágenes. La siguiente sección detalla los algoritmos de detección, la implementación en OpenCV y los parámetros que afectan la precisión de la detección.

Algoritmos de Detección

La detección de los marcadores ArUco se basa en algoritmos de visión por computadora que identifican contornos y patrones específicos en las imágenes. Según la documentación de OpenCV, el proceso de detección comienza con la identificación de cuadrados en la imagen y la verificación de si estos contienen un patrón binario válido correspondiente a un marcador ArUco [?]. El algoritmo implementado en OpenCV utiliza la segmentación de contornos y la detección de bordes para encontrar regiones cuadradas que luego se comprueban para determinar si coinciden con los patrones en el diccionario de ArUco.

Una vez que se identifica un marcador, el algoritmo calcula la pose de la cámara en relación con el marcador utilizando la proyección inversa. Este proceso es particularmente importante en aplicaciones donde se necesita calcular la posición y orientación de la cámara para la navegación y el control de robots y drones.

Implementación en OpenCV

OpenCV ofrece una implementación robusta para la detección de marcadores ArUco a través del módulo `cv::aruco`. La función principal para la detección es `cv::aruco::detectMarkers`, que se encarga de identificar los marcadores en una imagen y devolver sus esquinas y los IDs correspondientes. La documentación de OpenCV resalta que “*the function detects the markers and returns their IDs and corner positions in the image*” [?].

El siguiente ejemplo muestra cómo usar OpenCV para detectar marcadores ArUco en Python:

```
import cv2 import cv2.aruco as aruco

Cargar la imagen
image = cv2.imread('image_path.jpg')

Definir el diccionario de ArUco
aruco_dict = aruco.Dictionary_get(aruco.DICT_6X6_250)

Detectar los marcadores
```

```

corners, ids, _ = aruco.detectMarkers(image, aruco_dict)

Dibujar los marcadores detectados
if ids is not None: aruco.drawDetectedMarkers(image, corners, ids)

```

Después de obtener las esquinas y los IDs de los marcadores, es posible utilizar esta información para calcular la pose de la cámara en relación con el/los marcadores detectados. Para calcular el centro de un marcador ArUco, se pueden utilizar las esquinas detectadas y el tamaño del marcador para estimar su posición en la imagen.

Parámetros de Precisión

La precisión de la detección de marcadores ArUco depende de varios factores, incluidos la calidad de la imagen, el tamaño del marcador, y los parámetros de calibración de la cámara. Según la documentación de OpenCV, la calibración precisa de la cámara es fundamental para minimizar los errores en la estimación de la pose [?]. Entre los parámetros que afectan la detección se encuentran:

- **Distorsión de la lente:** La corrección de la distorsión de la lente mejora la precisión de la detección.
- **Resolución del marcador:** Marcadores con mayor resolución permiten una detección más precisa a distancias más largas, pero requieren más recursos de procesamiento.
- **Iluminación y contraste:** La detección puede verse afectada por condiciones de iluminación variables, por lo que es importante que la imagen tenga un buen contraste entre el marcador y el fondo.

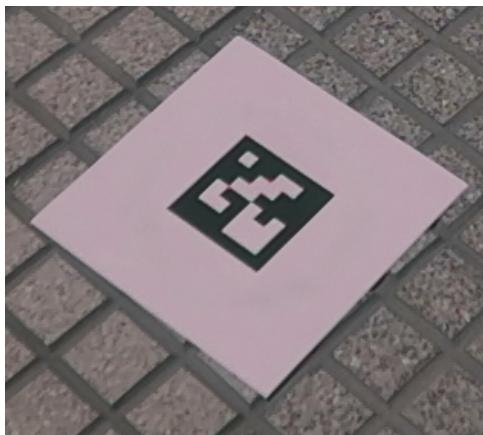


Figure 2.33: ArUco no pose estimation

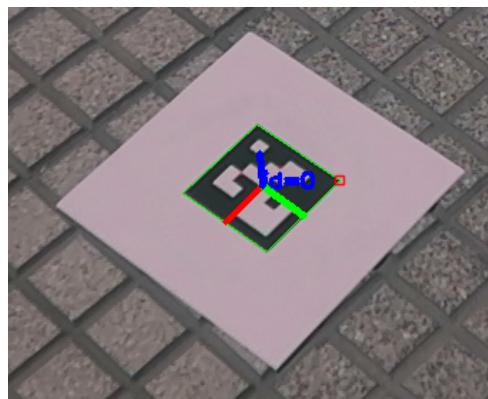


Figure 2.34: ArUco pose estimation

2.4.10 Comunicación ROS2 - Pixhawk

La comunicación entre ROS 2 y Pixhawk se basa en el uso de MAVLink, un protocolo ligero que permite la transferencia bidireccional de datos entre el controlador de vuelo y una computadora auxiliar (companion computer). Esta integración es crucial para enviar comandos de vuelo en tiempo real y monitorizar el estado del dron.

Protocolo MAVLink

MAVLink (Micro Air Vehicle Link) es un protocolo de comunicación de alto rendimiento diseñado para sistemas de vehículos no tripulados. Según la documentación oficial, “*MAVLink is a very lightweight, header-only message marshalling library for micro air vehicles*” [?]. Este protocolo utiliza un sistema basado en paquetes de mensajes que facilita la transmisión de datos entre la estación de control y el vehículo no tripulado.

Los mensajes de MAVLink están estructurados en comandos específicos que permiten controlar varios aspectos del vuelo y la telemetría, incluyendo parámetros como la posición, velocidad, estado de la batería, y comandos de control de vuelo. Cada mensaje está identificado por un ID único, lo que facilita su procesamiento y permite una comunicación eficiente en tiempo real.

Opciones de Integración de MAVLink en ROS 2

Para facilitar la integración de MAVLink con ROS 2, existen varias opciones de comunicación que permiten enviar y recibir datos entre una computadora auxiliar y el controlador de vuelo Pixhawk. Las dos soluciones más populares son MAVROS y Micro XRCE-DDS.

MAVROS es un conjunto de nodos de ROS que actúa como una interfaz entre ROS y MAVLink, permitiendo que ROS se comunique con el controlador de vuelo mediante topics y servicios. MAVROS permite la publicación y suscripción de mensajes de MAVLink a través de topics en ROS, facilitando la ejecución de comandos de vuelo, obtención de datos de telemetría y control de misiones.

MAVROS implementa una serie de topics predefinidos en ROS, tales como:

- `/mavros/setpoint_position/local`: Para establecer puntos de referencia de posición.
- `/mavros/state`: Proporciona el estado actual del dron, incluyendo el modo de vuelo y si el vehículo está armado.
- `/mavros imu/data`: Datos de IMU para el monitoreo de la orientación y aceleración.
- `/mavros/battery`: Muestra datos de la batería como porcentaje de carga y voltaje.

Con MAVROS, los desarrolladores pueden enviar comandos de control, como armar el dron, cambiar el modo de vuelo, o establecer puntos de referencia para la navegación autónoma. Esto se realiza publicando mensajes en los topics adecuados y ajustando parámetros en tiempo real mediante la interfaz de MAVLink [?].

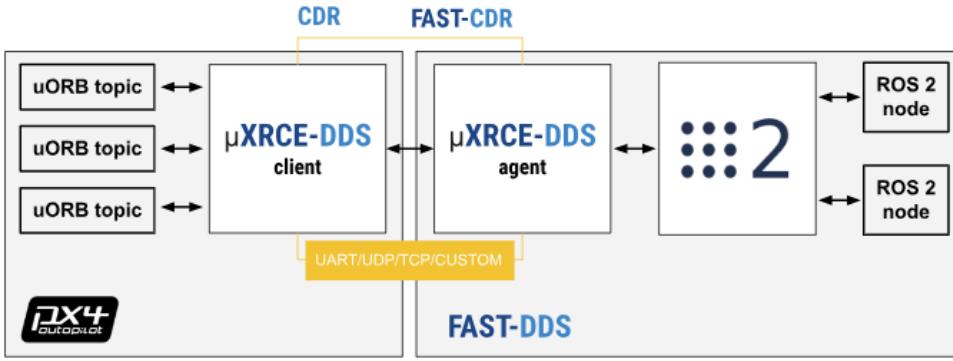


Figure 2.35: Comunicación ROS 2 - Pixhawk con Micro XRCE-DDS.

Micro XRCE-DDS es una implementación ligera del DDS (Data Distribution Service) que permite la comunicación entre ROS 2 y el controlador de vuelo de forma eficiente en sistemas con limitaciones de recursos. Esta opción es útil en contextos donde el tamaño y rendimiento del sistema son cruciales, como en drones pequeños o con hardware de procesamiento limitado.

Con Micro XRCE-DDS, ROS 2 puede comunicarse directamente con PX4 utilizando el middleware DDS. Esto permite la creación de un sistema distribuido y escalable, en el cual los nodos de ROS 2 pueden enviar y recibir mensajes MAVLink a través de topics DDS, de manera similar a MAVROS, pero con una carga de procesamiento optimizada [?].

2.4.11 Sensores para la Navegación en Ambientes Exteriores e Interiores

Para lograr el control preciso de la posición del dron mediante los topics de ROS 2 como `/mavros/setpoint_position`, es necesario contar con sensores que permitan obtener datos de posicionamiento confiables en función del entorno en el que opere el dron. A continuación, se describen los sensores comúnmente utilizados en ambientes exteriores e interiores para proporcionar información de posición.

Sensores para Ambientes Exteriores

En entornos exteriores, los drones suelen utilizar sensores de posicionamiento global (GPS) para determinar su ubicación en tiempo real. A continuación se presentan los sensores más comunes en exteriores:

- **GPS (Sistema de Posicionamiento Global):** El GPS es el sensor principal para la navegación en exteriores. Proporciona coordenadas de latitud, longitud y altitud que permiten al dron determinar su posición global. En ROS 2, estos datos se suelen publicar en topics como `/mavros/global_position/global` y se integran en los sistemas de control de posición.
- **RTK-GPS (Real-Time Kinematic GPS):** Para aplicaciones de alta precisión, como el aterrizaje de precisión o la navegación en áreas restringidas, se utiliza RTK-GPS. Este sistema mejora la precisión del GPS mediante la corrección de datos en tiempo real, alcanzando

precisiones de centímetros. RTK-GPS es compatible con MAVLink y permite un control de posición altamente preciso en ROS 2.

- **IMU (Unidad de Medición Inercial):** Aunque no proporciona datos de ubicación directa, la IMU complementa al GPS detectando cambios en orientación y aceleración. Estos datos son cruciales para mantener la estabilidad y proporcionar información sobre la orientación del dron durante el vuelo.

Sensores para Ambientes Interiores

En ambientes interiores, donde las señales de GPS suelen ser débiles o inexistentes, se emplean otros sensores para determinar la posición y navegación del dron. Los sensores utilizados en interiores incluyen:

- **Cámaras de Visión (Monocular o Estéreo):** Las cámaras de visión permiten la localización visual mediante algoritmos de SLAM (Simultaneous Localization and Mapping) o detección de marcadores, como ArUco, para calcular la posición relativa del dron en interiores. Con ROS 2, estas cámaras pueden integrarse con librerías como OpenCV y utilizar los topics de imagen para el procesamiento en tiempo real.
- **LIDAR (Light Detection and Ranging):** Los sensores LIDAR emiten pulsos de luz para medir distancias y obtener mapas detallados del entorno. Estos sensores son útiles para la navegación y el mapeo en tiempo real en interiores y se integran en ROS 2 mediante topics de nube de puntos. Son particularmente efectivos para evitar obstáculos y realizar posicionamiento preciso en interiores.
- **Sensores UWB (Ultra-Wideband):** Los sistemas UWB utilizan señales de radio para calcular distancias precisas entre el dron y puntos de referencia (beacons) fijos. Estos sensores son útiles en áreas cerradas, proporcionando precisión en el rango de centímetros, y permiten al dron calcular su posición relativa sin depender de la visión.
- **IMU (Unidad de Medición Inercial):** En interiores, la IMU sigue siendo un componente esencial para detectar cambios de orientación y movimiento. La fusión de datos de la IMU con otros sensores, como LIDAR o cámaras, ayuda a mantener la estabilidad y proporciona estimaciones de posición precisas mediante filtros de fusión, como el filtro de Kalman.

Fusión de Sensores para Precisión en Navegación

Para mejorar la precisión y estabilidad del control de posición, se recomienda la fusión de datos de múltiples sensores, un proceso que combina la información de cada sensor para obtener una estimación más confiable de la posición y orientación del dron. La fusión de sensores se implementa comúnmente en ROS 2 utilizando filtros de Kalman o técnicas avanzadas de estimación de estados, permitiendo el uso eficiente de los topics de setpoint de posición en diferentes entornos.

Estos sensores permiten al dron obtener datos de posición en tiempo real y de alta precisión, ya sea en exteriores o interiores, facilitando así el uso de los topics de posicionamiento en ROS 2 para control de navegación autónoma.

2.4.12 Fusión de Sensores usando el Filtro de Kalman en Pixhawk

La fusión de sensores es un proceso crucial en la navegación de drones, permitiendo integrar datos de múltiples sensores para obtener una estimación de posición y orientación más precisa. En el Pixhawk, este proceso se realiza mediante el filtro de Kalman extendido (EKF), el cual combina datos de sensores como GPS e IMU en exteriores, y cámaras, LIDAR o UWB e IMU en interiores. A continuación, se detallan los requisitos y configuraciones necesarias para que el sistema de fusión de sensores funcione adecuadamente en cada tipo de entorno.

Fusión de GPS e IMU en Exteriores

En entornos exteriores, el GPS es el sensor principal para la localización global, mientras que la IMU proporciona datos de orientación y aceleración. La fusión de estos sensores en Pixhawk se logra mediante el uso del EKF, que mejora la precisión y confiabilidad de la navegación al compensar posibles errores en las lecturas del GPS y la IMU.

- **Configuración de Pixhawk para GPS e IMU:** Para activar la fusión de GPS e IMU en Pixhawk, es necesario habilitar el EKF y configurar parámetros específicos, tales como:
 - **EKF2_GPS_CHECK:** Habilita la verificación de la calidad de la señal de GPS, garantizando que solo se utilicen datos de GPS cuando la señal sea suficiente.
 - **EKF2_HGT_MODE:** Establece la fuente de altitud para el filtro de Kalman. En exteriores, se recomienda configurarlo para usar datos de GPS o de barómetro.
 - **EKF2_AID_MASK:** Activa el soporte para sensores adicionales. Para la fusión de GPS e IMU, este parámetro debe incluir *GPS* y *IMU*.
- **Requisitos de Precisión de GPS e IMU:**
 - **GPS**:** Es ideal contar con un GPS de alta precisión o un sistema RTK-GPS en aplicaciones donde se requiera una precisión de centímetros.
 - **IMU**:** Es importante que la IMU esté correctamente calibrada y que el Pixhawk esté montado en una estructura estable para reducir el ruido de las lecturas.
- **Proceso de Fusión con EKF:** El EKF en Pixhawk realiza un cálculo continuo de la posición y velocidad del dron al combinar los datos de GPS e IMU. El GPS proporciona la ubicación global (latitud, longitud y altitud), mientras que la IMU contribuye con datos de orientación y aceleración. El EKF utiliza estos datos para reducir el error acumulativo y generar una estimación de posición precisa y estable.

Desafíos en la Fusión de Sensores en Ambientes Dinámicos

La fusión de sensores en Pixhawk enfrenta desafíos en entornos dinámicos o con interferencias, como en interiores con superficies reflectantes para LIDAR o en exteriores con pérdida de señal GPS. Para minimizar estos problemas, es importante realizar calibraciones periódicas de todos los sensores y ajustar los parámetros del EKF según el entorno y las condiciones de vuelo.

Mediante la fusión de sensores con el EKF en Pixhawk, los drones pueden operar de forma autónoma y precisa tanto en exteriores como en interiores, adaptándose a las condiciones y requerimientos específicos de cada entorno.

Fusión de Sensores en Interiores sin GPS

Para lograr la navegación autónoma en interiores, donde el GPS no está disponible, se emplean sensores alternativos que permiten al dron estimar su posición relativa y mantener un vuelo estable. La fusión de estos sensores se realiza mediante el filtro de Kalman extendido (EKF) en el Pixhawk, que combina datos de diferentes sensores para obtener una estimación precisa de posición y orientación. A continuación se describen los sistemas de posicionamiento más destacados y los requisitos para lograr una fusión de sensores efectiva en interiores.

Sensores Alternativos para Navegación en Interiores

En entornos cerrados, el uso de tecnologías como cámaras, beacons, y sistemas de seguimiento óptico permite al dron obtener información de posición sin depender del GPS. Estos sistemas requieren que se defina manualmente el origen de la posición inicial, usando una estación de control en tierra (GCS) como Mission Planner o mediante scripts en Lua. Algunos de los sensores más destacados incluyen:

- **Intel RealSense T265 y Luxonis OAK-D:** Ambas son cámaras de visión estéreo que pueden proporcionar estimaciones de posición mediante técnicas de SLAM (Simultaneous Localization and Mapping). Estas cámaras permiten calcular la posición y la orientación del dron en tiempo real, y pueden integrarse en el Pixhawk usando el EKF para fusionar estos datos con la IMU. La configuración requiere calibrar correctamente las cámaras y establecer el origen de posición.
- **MarvelMind Beacons y Pozyx Beacons:** Estos sistemas basados en balizas (beacons) usan señales de radio para determinar la posición relativa del dron. Los beacons se instalan en ubicaciones fijas dentro del entorno, y el dron calcula su posición en función de la distancia a cada beacon. Este tipo de sistema es muy preciso y puede proporcionar una estimación de posición estable, aunque requiere de una infraestructura inicial de beacons correctamente ubicados en el área.
- **ModalAI VOXL y Nokov Indoor Optical Tracking:** Estos sistemas combinan visión por computadora y algoritmos de seguimiento óptico para proporcionar datos de posicionamiento

precisos en interiores. ModalAI VOXL, por ejemplo, es un módulo que integra una computadora de alto rendimiento con cámaras de visión para permitir una navegación autónoma sin GPS. El Pixhawk puede fusionar estos datos de posición y orientación en tiempo real mediante el EKF, mejorando la estabilidad del vuelo en interiores.

- **Vicon Positioning System y OptiTrack Motion Capture System:** Estos sistemas de captura de movimiento utilizan cámaras de alta precisión colocadas en el entorno para rastrear marcadores ubicados en el dron. Estos sistemas son muy efectivos para aplicaciones de navegación de alta precisión, como en laboratorios o áreas de prueba, y permiten obtener una posición con precisión milimétrica. Sin embargo, requieren un entorno controlado y la instalación de múltiples cámaras en el área.
- **Optical Flow:** Este sensor detecta el movimiento relativo entre el dron y la superficie debajo de él. Aunque no proporciona una posición absoluta, el flujo óptico es útil para mantener la estabilidad del dron en interiores y para pequeñas correcciones de posición. La combinación de datos de flujo óptico con la IMU permite una navegación precisa a nivel local.

Requisitos para la Fusión de Sensores sin GPS

La fusión de sensores sin GPS en interiores requiere de un entorno controlado donde estos sistemas de posicionamiento puedan operar de manera efectiva. Los siguientes son algunos de los requisitos necesarios para configurar y utilizar estos sensores en Pixhawk:

- **Definir el Origen de la Posición:** Al utilizar sensores de visión, beacons o sistemas de captura de movimiento, el origen de la posición debe definirse manualmente a través del GCS o con un script en Lua para que el dron pueda interpretar correctamente su posición en relación con el entorno.
- **Calibración de Sensores de Visión y Beacons:** La precisión de los datos de posición depende de una calibración adecuada de los sistemas de visión y beacons. Esto incluye la alineación de cámaras y beacons, así como la configuración de parámetros en el EKF para optimizar la precisión de los datos.
- **Calidad de IMU:** La IMU es fundamental para la fusión de sensores en interiores, ya que proporciona datos de orientación y aceleración que complementan los datos de posicionamiento relativo. Las IMUs de bajo costo pueden experimentar un alto nivel de deriva, por lo que es recomendable utilizar IMUs de mayor calidad o realizar una calibración frecuente.
- **Uso del Filtro de Kalman (EKF) en Pixhawk:** El EKF en Pixhawk debe configurarse para aceptar las entradas de los sensores seleccionados. Esto incluye activar los parámetros necesarios en el Pixhawk, tales como `EKF2_AID_MASK` para habilitar `VISION_POSITION`, `RANGE_FINDER`, o `FLOW` dependiendo del sensor utilizado, y ajustar los valores de ruido según las especificaciones del sistema de posicionamiento.

Estos sensores y configuraciones permiten que el dron mantenga un vuelo autónomo y preciso en entornos interiores sin GPS, proporcionando alternativas para aplicaciones de navegación autónoma en espacios cerrados.

Chapter 3

Desarrollo

Este capítulo describe el proceso de diseño, implementación y configuración de la estación de carga modular y la instrumentación del dron. Se detallan las metodologías y herramientas utilizadas para llevar a cabo el desarrollo, con un enfoque en la creación de una solución eficaz para el sistema de carga.

3.1 Diseño y Desarrollo de la Base de Carga

3.1.1 Especificaciones y Requerimientos

A continuación se presentan las especificaciones técnicas y los requisitos para la base de carga, garantizando compatibilidad y eficiencia para un sistema de carga en enjambres de drones:

1. Diseño modular y apilable para facilitar el almacenamiento de múltiples drones en un mismo metro cuadrado.
2. Las dimensiones mínimas requeridas para la estación de carga debe ser de mínimo 10 porciento mayores en ancho, largo y alto que las del dron.
3. La estación de carga deberá contar con un mecanismo de carga para la recarga de la batería del dron.
4. Debe contar con por lo menos un método de posicionamiento visible para el dron.

3.1.2 Selección e Implementación del Mecanismo de Movimiento del Cajón

En la sección 2.2.3 se explican en detalle los diferentes mecanismos de movimiento que se consideraron para la estación de carga, evaluando sus ventajas y desventajas en términos de precisión, durabilidad, costo y facilidad de integración en el diseño modular. Se analizaron opciones como pistones lineales, bandas dentadas y otros sistemas de transmisión, con el objetivo de seleccionar una solución que cumpliera con los requisitos de funcionalidad sin elevar demasiado los costos. Tras esta evaluación, se optó por un mecanismo de Polea y Banda tipo V, el cual, además de ser una

opción económica, utiliza componentes ya disponibles, como el motor. Este sistema permite un movimiento horizontal suficiente para posicionar el dron con la precisión necesaria para el proceso de carga, a la vez que facilita el apilamiento modular de la estación. La orientación del movimiento no obstruye la estructura general de la estación, lo cual asegura que el diseño cumpla con el requisito de apilabilidad y compatibilidad en espacios compartidos, manteniéndose compacto y adaptable sin comprometer la estructura modular de la estación.

Para este mecanismo, se seleccionaron diferentes componentes, incluyendo una banda de alta resistencia para soportar las cargas previstas, poleas (con baja fricción) para optimizar el movimiento y reducir el desgaste del sistema en operaciones continuas, y un motor ... que tiene un torque de ... y cumple con el torque necesario para mover los 3 kg de peso del drone.

Diseños CAD y Componentes

1. Motor de 12V
2. Poleas
3. Banda
4. Baleros
5. Chumaceras
6. Soporte del Motor

Mecanismo Polea y Banda tipo V

(Imagen del Mecanismo)

3.1.3 Diseño CAD de la Base de Carga

Para cumplir con los requerimientos de diseño modular / apilable y compatibilidad con el drone en cuestión a las dimensiones de este. Se llevó a cabo el siguiente diseño en CAD:

- El diseño de la estructura principal de la estación de carga se realizó en Fusion 360, considerando las dimensiones mínimas requeridas para el dron y el mecanismo de carga. Imagen
- Se tomó en cuenta la dimensión necesaria para que el dron tenga un espacio de por lo menos 10 cm de separación con la pared del cajón, esto para evitar alguna colisión del dron con la estructura de la base de carga al momento de intentar aterrizar, además se disminuyó la altura del cajón para que el dron pueda aterrizar sin problemas. Por otro lado se usarán rieles que se obtuvieron previamente reutilizados, por lo cual la estación de carga física se encuentra salida de la base de carga por unos 5 cm al momento de estar cerrada, pero en el diseño CAD el diseño si se encuentra cerrado por completo. Imagen
- Se añadió un espacio de 25 cm para colocar la parte electrónica del cajón. Imagen

3.1.4 Lista de Materiales

- Perfiles de Aluminio
 - 2 x Perfiles de 75 cm
 - 2 x Perfiles de 50 cm
 - 2 x Perfiles de 25 cm
- 1 x Fuente de Poder de 12V
- 1 x Arduino Mega
- 1 x Raspberry Pi 4
- 1 x Driver JBT
- 2 x Sensor de Proximidad Inductivo
- 1 x Joystick
- 1 x Cargador de Lipo battery
- Tronillería para perfil de aluminio
- 2 x Rieles de 75 cmd
- 1 x Motor de 12V
- 1 x Soporte de Motor impreso en 3D
- 1 x Banda de 1 m
- 2 x Poleas de 5 cm de diámetro
- 2 x Baleros de 5 cm de diámetro
- 2 x Chumacera de 5 cm de diámetro impresas en 3D
- 2 x MDF de 9 mm de 50 x 50 cm
- 8 x Separadores de MDF impresos en 3D
- 1 x Router WiFi
- 16 x Codos para perfil de aluminio impresos en 3D

3.1.5 Proceso de Manufactura de la Estructura de la Base de Carga

1. **Corte del Material:** Para iniciar el proceso de manufactura, se realizaron cortes precisos de los perfiles de aluminio y las placas de MDF según las dimensiones especificadas en la lista de materiales. Este paso es fundamental para asegurar que todas las piezas se ensamblen correctamente en el diseño modular.

Imagen del proceso de corte de perfiles de aluminio y MDF

2. **Ensamblaje de la Estructura:** Una vez cortadas las piezas, se procedió a ensamblar la estructura principal de la estación de carga utilizando tornillos y tuercas para fijar los perfiles de aluminio y las placas de MDF usando los separadores impresos. Se verificó que todas las piezas estuvieran alineadas y niveladas para garantizar la estabilidad y resistencia de la base de carga.

Imagen del proceso de ensamblaje de la estructura de la base de carga

3. **Instalación del Mecanismo de Movimiento:** Despues de ensamblar la estructura principal, se instaló el mecanismo de polea y banda tipo V para permitir el movimiento horizontal del cajón. Se colocaron las poleas, la banda y el motor en las ubicaciones previamente definidas, asegurando que el sistema de movimiento funcionara correctamente y sin obstrucciones.

Imagen del proceso de instalación del mecanismo de movimiento

3.1.6 Circuito Electrónico de la Estación de Carga

Diagrama de Conexiones

Añadir diagrama de conexiones

Construcción del Circuito

Añadir imágenes de la construcción del circuito

Programación del Circuito

Explicar el código de programación del circuito y su funcionamiento, mencionar que en los anexos se encuentra el código fuente del arduino y raspberry pi 4.

3.2 Instrumentación del Dron

3.2.1 Especificaciones y Requerimientos

Para garantizar la compatibilidad y funcionalidad en el sistema de carga, el dron debe cumplir con las siguientes especificaciones:

- Las dimensiones de las piezas deberán adaptarse al frame del drone cuadricoptero de arquitectura abierta que fue comprado.
- El dron deberá tener un circuito de carga compatible con la estacion de carga.
- El drone deberá contar con una cámara y un sistema de visión para la detección de marcadores Aruco.
- Se deberán integrar sensores de localización para la navegación en exterior.
- Se deberá integrar algún microprocesador como computadora auxiliar para el procesamiento de datos y la comunicación con la estación de carga.

3.2.2 Lista de Materiales para la Instrumentación del Dron

- 1 x Frame de cuadricóptero abierto (especificar modelo)
- 1 x Controlador de vuelo (especificar modelo)
- 1 x Cámara (compatible con detección Aruco)
- 1 x Raspberry Pi 4 (Companion Computer)
- 1 x Sensor de proximidad (modelo de preferencia)
- 1 x Módulo GPS (compatible con el controlador de vuelo)
- Cableado y conectores
- Material de montaje impreso en 3D (soportes específicos para cada componente)

3.2.3 Diseños CAD del Dron

Se presentan a continuación los diseños CAD de las piezas que se han desarrollado para el dron, adaptadas a su frame original para integrar los componentes electrónicos necesarios y cumplir con los requisitos de carga y posicionamiento.

- Se diseñaron nuevos soportes para el microprocesador y la cámara, asegurando una integración estable y precisa en el frame.
- Se implementó un espacio de montaje para los sensores de localización y el circuito de carga.

Imagen de los diseños CAD del dron con las piezas modificadas

3.2.4 Circuito de Distribución de Energía

El circuito de distribución de energía proporciona alimentación segura y estable a los componentes del dron. La selección de baterías y reguladores de voltaje fue optimizada para asegurar la duración de vuelo y protección de cada componente.

- Se seleccionaron baterías de litio-polímero (LiPo) de alta capacidad para maximizar la autonomía.
- Reguladores de voltaje se añadieron para adaptar el suministro a la Raspberry Pi, la cámara y el controlador de vuelo.

Imagen del circuito de distribución de energía en el dron

3.2.5 Integración de Componentes Electrónicos

Cada componente fue ensamblado y cableado de acuerdo con el diseño modular del dron. A continuación se describen los pasos del proceso de integración:

1. **Montaje de la Cámara:** La cámara se fijó en la parte de abajo del dron, permitiendo una visibilidad óptima para la detección de marcadores Aruco.

Imagen del montaje de la cámara en el dron

2. **Instalación del Controlador de Vuelo y Módulo GPS:** El controlador de vuelo se instaló en el centro del frame para optimizar el balance y calibración, y el módulo GPS se colocó en una de las patas.

Imagen del montaje del controlador de vuelo y módulo GPS

3. **Instalación del Micróprocesador:** La Raspberry Pi 4 se integró en un soporte impreso en 3D en la parte superior, y el microprocesador se colocó sobre este soporte. Un cable de comunicación serial se conectó entre la Raspberry Pi y el controlador de vuelo que se encuentra debajo de el microprocesador.

Imagen del montaje del micróprocesador en el dron

3.3 Software Configuration

3.3.1 Configuración de la Computadora en Tierra

Se documenta la configuración de la computadora en tierra, incluyendo las herramientas y librerías necesarias para la comunicación y el procesamiento de datos:

- Instalación de ROS 2 y configuración de nodos para la comunicación con el dron.
- Configuración de los parámetros de red para la comunicación WiFi.

Imagen del entorno de configuración en la computadora de tierra

3.3.2 Configuración de la Computadora Auxiliar del Dron

La Raspberry Pi se configuró como una computadora auxiliar, encargada del procesamiento de datos y la comunicación en tiempo real con la estación de carga.

- Instalación de ROS 2 y conexión de nodos de comunicación.
- Configuración de módulos de procesamiento de imágenes para la detección de Aruco.

Imagen de la configuración del entorno en la Raspberry Pi 4

3.3.3 Configuración de la Estación de Control en Tierra

Se detalla la configuración de la estación de control utilizando Mission Planner:

- Instalación de Mission Planner para la visualización y gestión de parámetros de vuelo.
- Configuración de la calibración de sensores, incluyendo el acelerómetro, el giroscopio y el GPS.

Imagen de la configuración de sensores en Mission Planner

3.4 Sistema de Visión

3.4.1 Especificaciones y Requerimientos

El sistema de visión debe cumplir con los siguientes requisitos para asegurar la precisión en la detección de marcadores Aruco:

- Obtención y aplicación de las matrices de calibración de la cámara.
- Generar marcadores Aruco de diferentes tamaños (Embedded ArUcos) para su detección en tiempo real.

3.4.2 Calibración de la Cámara

El proceso de calibración incluyó la captura de múltiples imágenes de una cuadrícula de calibración y la generación de los parámetros intrínsecos y extrínsecos.

Figure

Imagen del proceso de calibración y los resultados obtenidos

Figure 3.1: Imagen del proceso de calibración y los resultados obtenidos

3.4.3 Generación de Marcadores Aruco

Se generaron marcadores Aruco adaptados al sistema, incluyendo tamaños y patrones específicos para maximizar la precisión en la detección desde diferentes alturas.

Imagen de los diferentes tamaños y diseños de los marcadores Aruco

3.4.4 Detección de Marcadores e-Aruco en Tiempo Real

El proceso que se llevó a cabo para la detección de marcadores Aruco en tiempo real se describe a continuación:

1. **Captura de Imágenes:** Se capturaron imágenes de la cámara en tiempo real, utilizando la webcam conectada a la Raspberry Pi 5 que se encuentra en el dron.
2. **Procesamiento de Pose de los ArUcos:** Se aplicó un algoritmo para la detección de la pose de los marcadores ArUco el cual se describe a continuación:
 - Se utilizó la librería OpenCV para el procesamiento de imágenes y la detección de marcadores Aruco.
 - Se implementó un algoritmo de detección de bordes y esquinas para identificar los marcadores en las imágenes.
 - Se mide el centroide de los marcadores y se calcula la matriz de posición y orientación para determinar la ubicación del dron.
3. **Visualización en Tiempo Real:** Se mostraron los marcadores detectados en tiempo real, permitiendo la interacción con el sistema de control en tierra.

Imagen de la detección de marcadores Aruco en tiempo real

3.5 Sistema de Comunicación

3.5.1 Especificaciones y Requerimientos

Para asegurar una comunicación efectiva entre el dron y la estación de carga, el sistema de comunicación debe cumplir con los siguientes requisitos:

- Utilización de un protocolo basado en **WiFi** para la transferencia de datos entre el dron y la estación de carga.
- Implementación del middleware DDS (Data Distribution Service) de ROS 2 para publicación y suscripción de tópicos en tiempo real.

3.5.2 Protocolo de Comunicación entre el Dron y la Base de Carga

El protocolo de comunicación establece la interacción entre el dron y la estación de carga, utilizando ROS 2 y DDS para la transmisión de mensajes en tiempo real.

Diagrama del flujo de comunicación entre el dron y la base de carga

Diagrama de Comunicación WiFi

Este diagrama muestra la estructura de comunicación dentro de la red WiFi, destacando el flujo de datos desde el dron hasta la estación de carga.

Diagrama de Comunicación de Nodos ROS 2

El esquema de nodos en ROS 2 especifica los tópicos y servicios clave utilizados para la comunicación entre el dron y la estación de carga.

Diagrama de comunicación de nodos en ROS 2

Chapter 4

Resultados

Este capítulo presenta los resultados obtenidos en el desarrollo y prueba de la base de carga modular y el dron instrumentado para la interacción con la estación de carga.

4.1 Resultados del Diseño y Manufactura de la Base de Carga

4.1.1 Diseño CAD Final del 1er prototipo de la Estación de Carga

4.1.2 Diseño e Instrumentación del Drone

4.1.3 Movimiento del Cajón

4.1.4 Vuelo del Drone

4.1.5 Carga y Descarga de Batería

4.1.6 Detección del ArUco

4.1.7 Sistema de Comunicación

Documentación de los tiempos de respuesta, confiabilidad y estabilidad de la comunicación. Las métricas de latencia y tasa de éxito en la transferencia de datos se presentan en gráficos para facilitar el análisis.

4.2 Análisis de Desempeño General

4.2.1 Resultados de Integración Global

Comparación entre el desempeño esperado y el desempeño real en la interacción entre el dron y la estación de carga.

Chapter 5

Conclusiones

Texto de introduccion: Estabamos buscando hacer esto y llegamos a esto

Evaluacion de resultados

Posibles mejoras

Bibliography

- [1] PX4 Documentation, “PX4 is a powerful open source autopilot flight stack running on the NuttX RTOS.” Disponible en: . [*Último acceso: mes, año*].
- [2] ArduPilot Documentation, “ArduPilot firmware works on a wide variety of different hardware to control unmanned vehicles of all types.” Disponible en: <https://ardupilot.org/ardupilot/index.html>. [*Último acceso: mes, año*].
- [3] QGroundControl Documentation, “QGroundControl provides full flight control and vehicle setup for PX4 or ArduPilot powered vehicles.” Disponible en: <https://docs.qgroundcontrol.com/en/>. [*Último acceso: mes, año*].
- [4] Ubuntu Documentation, “Designed for security, reliability, and ease of use.” Disponible en: <https://ubuntu.com>. [*Último acceso: mes, año*].
- [5] Raspberry Pi OS Documentation, “A Debian-based operating system specifically tuned for the Raspberry Pi hardware.” Disponible en: <https://www.raspberrypi.org/documentation/raspbian/>. [*Último acceso: mes, año*].
- [6] ROS 2 Documentation, “Benefits and architecture of ROS 2 for distributed systems in robotics.” Disponible en: <https://docs.ros.org/en/>. [*Último acceso: mes, año*].
- [7] OpenCV Documentation, “An open-source computer vision and machine learning software library containing more than 2500 optimized algorithms.” Disponible en: <https://docs.opencv.org/>. [*Último acceso: mes, año*].
- [8] ArUco Documentation, “ArUco is an OpenSource library for detecting squared fiducial markers in images.” Disponible en: . [*Último acceso: mes, año*].
- [9] Khazetdinov, M., et al., “Embedded ArUco Markers for UAV Landing: High Accuracy Detection in Wide Distance Range,” 2021. DOI: [DOI number or URL if available].