

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Adaptación del sistema de drones Crazyswarm al ecosistema
Robotat**

Protocolo de trabajo de graduación presentado por Julio Andrés Avila,
estudiante de Ingeniería Mecatrónica

Guatemala,

2023

Resumen

La Universidad del Valle de Guatemala cuenta con un laboratorio de Robótica llamado Robotat, el cual cuenta con un sistema de captura de movimiento y se trabajan líneas de investigación relacionadas con sistemas de control y robótica, además de ser el espacio donde se realizan las prácticas de laboratorio de los cursos de robótica. Está diseñado para realizar pruebas con agentes autónomos, entre ellos robots humanoides, robots móviles con ruedas, brazos robóticos y drones.

Entre los drones a utilizar se encuentran los drones Crazyflie 2.1, los cuales son de tamaño pequeño y pueden emplearse en conjunto para controlar enjambres de drones. Se planea realizar este control mediante el sistema Crazyswarm, el cual funciona a través de ROS2 en Linux. Con el objetivo de que este sistema de drones pueda utilizarse en prácticas de laboratorio y otras líneas de investigación en la Universidad, se adaptará la infraestructura de este sistema al ecosistema Robotat a través de los paquetes y funciones de ROS2. Esto podrá realizarse mediante la obtención de paquetes de información generados por los algoritmos de control y el sistema de captura de movimiento del Robotat, logrando una integración entre los drones y el laboratorio.

Antecedentes

Los drones Crazyflie se han utilizado en líneas de investigación relacionadas con sistemas de control y robótica de enjambre debido a su extensa documentación y versatilidad para desarrollo de algoritmos. Para poder realizar un control eficiente de múltiples agentes, es necesario implementar un sistema que sea capaz de obtener y procesar información acerca del comportamiento de los agentes en tiempo real.

Crazyflie 2.1

Los drones a trabajar durante esta línea de investigación son los Crazyflies 2.1, los cuales son cuadricópteros de tamaño pequeño, que pueden ser controlados mediante un sistema de control de enjambre llamado Crazyswarm, el cual es utilizado mediante ROS en Ubuntu.

Anteriormente, se trabajó con estos drones en una línea de investigación pero no en forma de enjambre, sino con un drone únicamente. Se desarrollaron herramientas para el uso individual de estos drones. En primer lugar, se comprobó la compatibilidad de comunicación entre un drone y Python para el análisis de datos, estos se guardan en bruto y se recomendó graficarlos posteriormente mediante Matlab. Se desarrolló una interfaz gráfica para prácticas de laboratorio, en la cual, se pudo observar el comportamiento del drone ante las propiedades de control determinadas, en este caso, un controlador PID y sus respectivas constantes. Cabe destacar que para poder hacer esta herramienta fue necesario plantear un modelo matemático a través del identificador de sistemas de Matlab. Como resultado de estas herramientas, se llegaron a planificar 2 prácticas de laboratorio para los cursos de Sistemas de Control. [1]

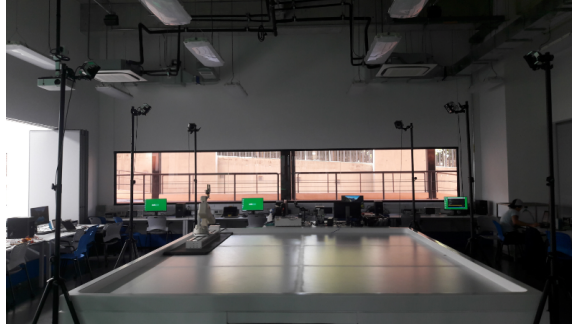


Figura 1: Ecosistema Robotat.

[2]

Ecosistema Robotat

El objetivo principal es utilizar este enjambre de drones en el ecosistema Robotat, el cual es un entorno tecnológico ubicado en el laboratorio de Robótica CIT-116 de la Universidad del Valle de Guatemala. [2] Se implementó un sistema de captura de movimiento mediante un set de 6 cámaras de la marca OptiTrack, las cuales están conectadas a un switch que se comunica con una computadora mediante UDP, también fue implementado un protocolo MQTT para establecer una comunicación Wi-Fi. La información recopilada es procesada mediante un algoritmo de Python y luego es enviada a un router que genera una red Wi-Fi local a la cual es posible conectar diferentes dispositivos y obtener datos como posiciones lineales y angulares. Las cámaras están colocadas de tal forma que rodean una tarima hecha de concreto y acero, como se muestra en la Figura 1.

Crazyswarm y ROS2

ROS es un sistema operativo para sistemas robóticos, el cual provee funciones especiales que facilitan el análisis y control en proyectos de robótica. Este suele trabajarse en sistemas basados en Linux como Ubuntu. A través de ROS2 puede trabajarse Crazyswarm, el cual es un sistema para controlar enjambres de drones Crazyflie 2.1, además se cuenta con un cliente llamado Bitcraze, el cual está basado en Python y permite la comunicación entre el servidor y los drones mediante telecomunicación por radiofrecuencia, este es un sistema independiente de ROS. El entorno Crazyswarm está siendo reemplazado por la versión 2, a la cual se le han estado trabajando mejoras.

El posicionamiento de los drones se ha realizado de múltiples maneras en distintas líneas de investigación, entre la documentación oficial de Crazyswarm puede encontrarse información sobre pruebas realizadas en sistemas de captura de movimiento, tales como OptiTrack. Para poder realizar experimentos con OptiTrack es necesario implementar un módulo con marcadores reflectivos y este debe estar diseñado a la medida para que el drone pueda ser detectado por las cámaras. Otros métodos utilizados para el posicionamiento de drones es mediante un dispositivo Kinect de la marca Microsoft, utilizado en consolas Xbox. [3]

Justificación

Los drones son dispositivos que han ganado popularidad en los últimos años por una característica especial, ya que al modelarlos como un sistema dinámico se presenta una dinámica no lineal, pero pueden ser controlados de forma satisfactoria con un controlador lineal, siendo este un control relativamente sencillo. Esto convierte a los drones en un excelente método de aprendizaje para diseñar y analizar modelos y algoritmos de control. Debido a esto, trabajar físicamente con un drone o inclusive, con un conjunto de estos, es una alternativa para poner en práctica los aprendizajes adquiridos a lo largo de una carrera universitaria relacionada con electrónica y para realizar investigación relacionada con sistemas de control clásico, moderno y robótica de enjambre. Para poder realizar esto, es necesario contar con un medio que le provea a los algoritmos de control la información necesaria para trabajar, incluyendo, pero no limitándose, a las posiciones espaciales y orientaciones de los respectivos drones. Debido a que se cuenta con un entorno que es capaz de capturar esta información física, el camino a tomar es la creación de un intermediario que permita la comunicación entre el sistema de captura y los algoritmos de control para drones.

El ecosistema Robotat fue creado como un entorno tecnológico que sirviera como un lugar de aprendizaje de sistemas robóticos y para desarrollar líneas de investigación relacionadas con sistemas de control y robótica. Para culminar el desarrollo de este entorno, es necesario integrar a todos los agentes autónomos que formarán parte de él, siendo los drones Crazyflie una parte importante del ecosistema.

Objetivos

Objetivo general

Adaptar el sistema de control de drones CrazySwarm al ecosistema Robotat mediante ROS2 y sus respectivos paquetes y funciones.

Objetivos específicos

- Emplear la información recibida por las cámaras de captura OptiTrack en el ecosistema Robotat para definir algoritmos de control del sistema de drones.
- Crear algoritmos que conviertan información de origen TCP/UDP a radio frecuencia para establecer una comunicación directa con CrazySwarm.
- Levantar la infraestructura de CrazySwarm y desarrollar pruebas iniciales de vuelo.



Figura 2: Crazyflie 2.1.
[4]

Marco teórico

Crazyflie

Los drones Crazyflie son micro drones de dimensiones significativamente pequeñas (92x92x29 mm) y una masa de 27 g, cuentan con cuatro motores con un diseño simétrico.

El control directo de estos drones se realiza mediante radio frecuencia, específicamente a una frecuencia de 2.4 GHz, la cual pertenece a la banda de radios ISM. Cuenta con un amplificador de radio de 20 dB, con el cual se puede trabajar en un radio de hasta 1 km. Entre sus características eléctricas, se encuentra un microcontrolador STM32F405, el cual corre a 168 MHz. Este microcontrolador presenta una gran ventaja al trabajar a una velocidad alta ya que se puede garantizar un procesamiento de señales y una ejecución de algoritmos eficientes. Cuenta con un cargador LiPo con modos de 100 mA hasta 980 mA, el tiempo de vuelo estimado con la batería cargada es de 7 minutos, con un tiempo de carga de 40 minutos. Según Bitcraze, es recomendado que cualquier carga adicional que soporte el drone no supere los 15 g. 2

Control y estimación de estado

La IMU del Crazyflie 2.1 cuenta con un acelerómetro/giroscopio de 3 ejes BMI080 y un sensor de presión BMP388. El control del drone se basa en un modelo de sistema dinámico. El modelado de sistemas dinámicos es utilizado en sistemas de control y robótica para diseñar un controlador basado en la naturaleza del sistema, obteniendo un modelo matemático de este. Las variables de estado son valores determinados de un sistema dinámico, el cual puede ser un sistema físico, mecánico, eléctrico, etc. En el caso de un drone cuadricóptero,

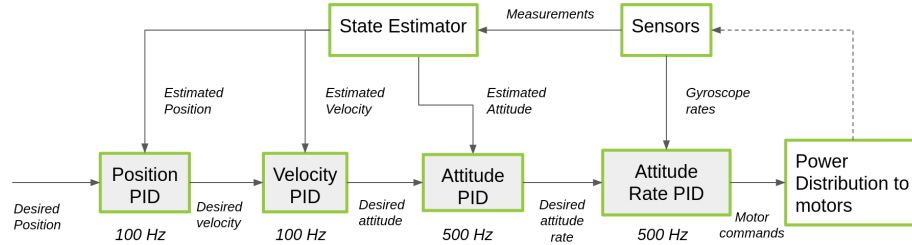


Figura 3: Arquitectura de Control.
[5]

las variables de estado son los ángulos de rotación alrededor de sus ejes (*roll*, *pitch*, *yaw*), su posición en el espacio y cómo estas variables cambian en el tiempo (sus derivadas). Para el caso de Crazyflie, las variables fundamentales a controlar son dichos ángulos y la altitud de este. Se utilizan estimaciones de estado para convertir las señales de los sensores en variables de estado y así poder realizar el respectivo control.

Debido a que se utilizan métodos externos para determinar el estado de los drones, siendo un sistema de captura de movimiento para este caso, es necesario combinar las mediciones internas del dron con las externas por lo que se utiliza el filtro de Kalman como observador de estado. Los observadores de estado son algoritmos que utilizan el modelo matemático del sistema y mediciones del mismo para dar una estimación de las variables de estado. El filtro de Kalman es un observador de estado que toma en cuenta el ruido generado por perturbaciones en los actuadores o sensores del sistema, modelándolos como ruido blanco no correlacionado. Los drones Crazyflie utilizan un filtro de Kalman extendido como un filtro recursivo que estima el estado actual del dron basado en las mediciones internas y externas y el modelo matemático del mismo. [5]

El control de los drones se realiza mediante una arquitectura de controladores PID en cascada, donde la entrada de control es la posición y la velocidad esperada, que luego determina ángulos de *pitch* y *roll* deseados, seguido de una velocidad angular y finalmente determina el *thrust* de los motores. 3

Sistemas de radio frecuencia

La comunicación por Radio Frecuencia (RF) es una tecnología para las telecomunicaciones que se da por un dispositivo que crea ondas electromagnéticas a altas frecuencias, donde las ondas se propagan en el espacio alcanzando a otro dispositivo receptor que procese la información a la misma frecuencia. Según la frecuencia de las ondas, las comunicación puede clasificarse en

- VLF: 3 kHz - 30 kHz
- LF: 30 kHz - 300 kHz
- HG: 3 MHz - 30 MHz
- VHF: 30 MHz - 300 MHz

- UHF: 300 MHz - 3 GHz
- SHF: 3 GHz - 30 GHz
- EHF: 30 GHz - 300 GHz

Las antenas convierten la información digital en señales eléctricas y luego en ondas electromagnéticas, o viceversa en el caso de la recepción de información. La ventaja que presenta este tipo de comunicación es que el dispositivo que procese tanto la información enviada como recibida puede ser un microcontrolador, siempre que este tenga la capacidad de conectarse a una antena y funcione a una frecuencia lo suficientemente alta como para procesar la señal de radio y digitalizar la información.

Protocolos de redes

Para crear una comunicación eficiente se debe trabajar en conjunto con protocolos de comunicación por redes, como TCP, el cual divide la información en paquetes y garantiza que estos lleguen correctamente a su destino. El protocolo TCP, que significa *Transmission Control Protocol*, es un protocolo de 3 vías donde primero se envía una solicitud inicial (*SYN*) desde el origen, luego el destino envía un paquete de confirmación (*SYN-ACK*) y finalmente el origen envía un último paquete (*ACK*) seguido de la información a transmitir. Este trabaja en conjunto con el protocolo IP, el cual se encarga únicamente asegurar una conexión entre el servidor de origen y destino mediante el sistema de direcciones de Internet.

El *User Datagram Protocol*, o UDP, es un protocolo de comunicación de Internet que, a diferencia del TCP, permite una comunicación rápida al no requerir que se establezca formalmente una conexión antes de iniciar con la transmisión de datos, sin embargo, esto también provoca que los paquetes puedan perderse en tránsito. [6]

Formato JSON

Para la estructura de los datos transferidos entre los distintos medios de comunicación, puede implementarse el formato *JavaScript Object Notation* (JSON), el cual es ligero computacionalmente y de fácil lectura y escritura. Adicionalmente, puede aplicarse a otros lenguajes de programación, como C, C++, Python, etc.

Sistemas de captura de movimiento

La captura de movimiento es una tecnología que ha ganado popularidad los últimos años al emplearse en aplicaciones como animación, videojuegos o investigación de sistemas mecánicos. Este proceso se da mediante cámaras de luz infrarroja, la cual se apunta hacia el sistema o conjunto de sistemas a capturar, los cuales deben contar con algún material que refleje la luz. En el caso particular de las cámaras de la marca Optitrack, las cuales son utilizadas en el Robotat en la Universidad del Valle de Guatemala (figura 4), cuentan con un set de emisores de luz infrarroja y utiliza pequeñas bolas cubiertas de material reflejante,



Figura 4: Cámara de captura OptiTrack.
[7]

conocidas como marcadores. Este tipo de captura de movimiento es conocido como óptica pasiva, ya que los marcadores reflejan la luz en lugar de generarla. [2]

Los marcadores pueden usarse de forma individual para rastrear únicamente posiciones, o bien pueden trabajarse en grupos de 3 con una forma en específico para analizar también la orientación de los cuerpos.

Ecosistema Robotat

Para transferir los datos obtenidos por el sistema de captura en el Robotat, las cámaras OptiTrack están conectadas a un Switch que se comunica con un servidor mediante un protocolo UDP. La información es procesada y enviada mediante un servidor de Python que posteriormente se comunica con un router Wi-Fi, el cual permite la comunicación entre dispositivos externos y agentes autónomos en el Robotat. Esta comunicación entre el router y los dispositivos se logra mediante un protocolo TCP, de esta forma puede accederse directamente a la información mediante la de composición de un documento de tipo JSON.

Cliente de Crazyflie y ROS2

ROS funciona con nodos, creando un servidor de Crazyflie que establece la comunicación directa entre los drones y la interfaz utilizada, que en este caso es ROS2 junto con la API de Python como se ve en la Figura 5.

El control de los drones se divide en cuatro capas independientes. La capa física transmite los paquetes de información desde y hacia el drone, donde puede implementarse una antena de radiofrecuencia para esto. El *link* implementa los canales para los paquetes, abstrayendo el medio físico e implementando un canal de transmisión y otro de recepción entre el Crazyflie. El CRTP, acrónimo de *Crazy Realtime Protocol*, implementa la información del puerto y el canal para crear la ruta del paquete a varios subsistemas. Finalmente, los subsistemas implementan las funcionalidades del Crazyflie que pueden ser controladas, habiendo sólo un puerto por subsistema.

El protocolo CRTP fue diseñado para permitir una priorización de paquetes y volver más

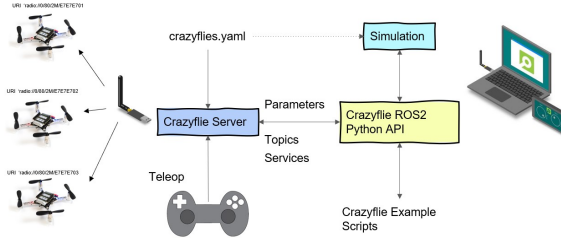


Figura 5: Estructura de control de Crazyswarm.
[8]

eficiente la comunicación con el drone, permitiendo enviar una trayectoria mientras el drone se controla en tiempo real, siempre que el puerto de la trayectoria sea de mayor prioridad. Cada paquete CRTP contiene un puerto (4 bits), un canal (2 bits), y una carga de hasta 31 bytes. Para crear la conexión del protocolo se debe habilitar el *link* USB usando un paquete de control USB, luego el *link* Radio mantiene dos contadores de paquete para garantizar que no habrá pérdida de paquetes. Finalmente, el subsistema *log* mantiene un estado de todos los bloques *log* y continua enviado información aún si se pierde el *link*[9]

Otro protocolo de comunicación para Crazyflie es el CPX (*Crazyflie Packet eXchange*), el cual fue diseñado para funcionar a bajo nivel y que se empleen otros protocolos por encima. Su función es solucionar el problema de distribuir paquetes a través de múltiples microcontroladores ya que este problema no existía cuando se diseñó el protocolo CRTP, por lo que es un protocolo complementario. Debido a que se utiliza entre microcontroladores, puede enviarse de múltiples formas, como Wi-Fi/TCP, SPI o UART dependiendo de los microcontroladores a comunicar y los módulos adicionales que se implementen en el drone. Este protocolo permite el manejo de paquetes grandes, donde paquetes por debajo de 30 bytes se entregan en un bloque y paquetes más grandes pueden separarse en bloques y un identificador especifica cual es el bloque final.

Para establecer la comunicación de Crazyswarm, el servidor de Crazyflie se conecta con múltiples drones mediante una o más antenas de radiofrecuencia. Se cuenta con dos *backends* a elegir, el "cpp" basado en la capa más baja y el "cflib", que funciona con Python a un nivel más alto. Este puede manejar aspectos de comunicación de bajo nivel, como recibir los parámetros del Crazyflie y convertirlos a parámetros de ROS2 para crear los parámetros de Crazyflie basado en la entrada.

Crazyflie cuenta con una máquina virtual, la cual funciona mediante Ubuntu y esta contiene todas las librerías para controlar los drones mediante una antena de radiofrecuencia USB. Estas librerías pueden visualizarse y editarse mediante Microsoft Visual Studio, que ya cuenta instalados los compiladores de C++ y Python. Desde Visual Studio, pueden iniciarse los códigos y, si la antena está conectada y un Crazyflie está encendido, pueden realizarse pruebas de movimiento y obtención de datos del drone. La máquina virtual también cuenta con el cliente de Crazyflie, desde el cual puede controlarse al Crazyflie mediante un control externo y es posible visualizar los ángulos de rotación del drone en tiempo real. En caso de que no se cuente con un control externo, puede utilizarse la aplicación para Android de Crazyflie y controlar al drone mediante *Bluetooth*.

Para poder realizar las pruebas del Crazyflie es necesario configurar el puerto de la antena

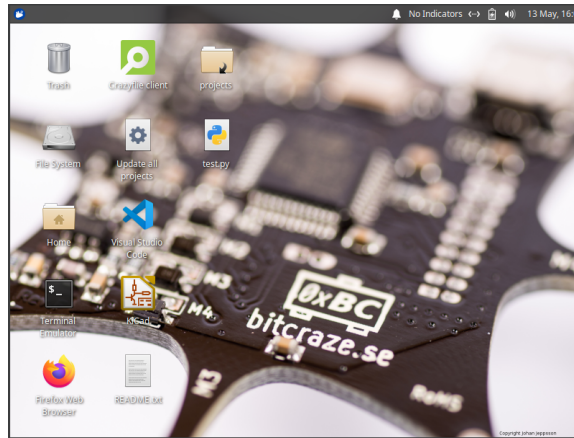


Figura 6: Máquina Virtual de Crazyflie.

durante la instalación de la máquina virtual, lo cual puede realizarse fácilmente mediante Oracle. También se deben configurar todos los dispositivos que se planeen utilizar en la máquina virtual, de lo contrario no serán reconocidos. Una vez instalada la máquina virtual se debe abrir una terminal e instalar ciertas librerías mediante el gestor de paquetes pip3. Las librerías son:

- Matplotlib.
- Pandas.
- Canvas.
- pyinstaller.

Finalmente, es necesario actualizar las librerías de los controladores. Esto se logra mediante la opción incluida en el escritorio de la máquina virtual, como se ve en la Figura 4 6. Sin embargo, se presenta un problema al actualizar las librerías ya que luego de hacerlo deja de funcionar el cliente para controlar el dron con un control externo, por lo que si se desea trabajar con las librerías de Python y con el cliente se recomienda instalar dos máquinas virtuales y que una de estas no se actualice.

Metodología

Con el objetivo de alcanzar la integración del sistema de drones y el sistema de captura de movimiento, se utilizará la siguiente metodología:

- Se realizarán pruebas en los drones con los códigos de Python proveídos por Bitcraze y analizar la estructura de control.
- Se levantará de la infraestructura de Crazyflie a través de ROS2 y realización de pruebas preliminares de control con múltiples drones.

- Se obtendrán los paquetes de información generados por OptiTrack para enviar retroalimentación en tiempo real sobre el estado de los drones.
- Se desarrollará un sistema que permita la interacción directa entre los paquetes de información de OptiTrack y los de CrazySwarm.
- Se realizarán pruebas de algoritmos de control y validación de la integración de sistemas.

Cronograma de actividades

		junio			julio				agosto					Septiembre		
		5 a 10	12 a 17	19 a 24	3 a 8	10 a 15	17 a 22	24 a 29	31 a 5	7 a 12	14 a 19	21 a 26	28 a 2	4 a 9	18 a 23	25 a 30
		1	2	3	1	2	3	4	1	2	3	4	5	1	3	4
1	Instalación de ROS2 y las librerías de CrazySwarm	x														
2	Levantamiento de la infraestructura de CrazySwarm		x	x												
3	Obtención y manipulación de datos de OptiTrack				x	x										
4	Desarrollo de algoritmos de integración entre Optitrack y CrazySwarm						x	x	x							
5	Realización de pruebas movimiento de múltiples drones dentro del ecosistema									x	x	x				
6	Validación de métodos de control y algoritmos de integración entre sistemas						x	x	x	x	x	x	x	x	x	x
7	Redacción de trabajo escrito	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Índice preliminar

1. Prefacio
2. Lista de figuras
3. Lista de cuadros
4. Resumen
5. Abstract
6. Introducción
7. Antecedentes
8. Justificación
9. Objetivos
10. Marco Teórico
 - a) Crazyflie 2.1
 - b) Control y estimación de estado
 - c) Comunicación por radiofrecuencia y protocolos de redes
 - d) Sistemas de captura de movimiento
 - e) Cliente de Crazyflie y su máquina virtual
11. Arquitectura de control
12. Filtro de Kalman

13. Protocolo CRTP
14. Algoritmos de control de Crazyswarm
15. Obtención de paquetes por TCP/IP y UDP
16. Recomendaciones
17. Bibliografía
18. Anexos

Referencias

- [1] F. Sanabria, “Diseño e implementación de una plataforma de pruebas para sistemas de control para el dron Crazyflie 2.0,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
- [2] C. Perafan, “Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
- [3] “Crazyswarm Documentation.” (), dirección: <https://crazyswarm.readthedocs.io/en/latest/>. (Accedido el: 21/4/2023).
- [4] “Crazyflie 2.1 | Bitcraze.” (), dirección: <https://www.bitcraze.io/products/crazyflie-2-1/>. (Accedido el: 13/5/2023).
- [5] “State estimation.” (), dirección: https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/sensor-to-control/state_estimators/. (Accedido el: 20/5/2023).
- [6] “¿Qué es el UDP?” (), dirección: <https://www.cloudflare.com/es-es/learning/ddos/glossary/user-datagram-protocol-udp/>. (Accedido el: 13/5/2023).
- [7] “Primex-40-Specs.” (), dirección: <https://optitrack.com/cameras/primex-41/specs.html>. (Accedido el: 21/4/2023).
- [8] “Overview-Crazyswarm2.” (), dirección: <https://imrclab.github.io/crazyswarm2/overview.html>. (Accedido el: 22/4/2023).
- [9] “CRTP-Communication with the Crazyflie.” (), dirección: <https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/crtp/>. (Accedido el: 6/5/2023).