UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Desarrollo e implementación de algoritmos de control para un enjambre de drones Crazyflie 2.0 mediante un sistema de visión de cámaras OptiTrack.

Trabajo de graduación presentado por Kenneth Andree Aldana Corado
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Desarrollo e implementación de algoritmos de control para un enjambre de drones Crazyflie 2.0 mediante un sistema de visión de cámaras OptiTrack.

Trabajo de graduación presentado por Kenneth Andree Aldana Corado
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

Vo.Bo.:

(f) _____

Ing. Estuardo Mancio

Tribunal Examinador:

(f) _____

Ing. Estuardo Mancio

(f) _____

MSc. Carlos Esquit

(f) _____

Ing. Luis Pedro Montenegro

Fecha de aprobación: Guatemala, 5 de diciembre de 2018.

Prefacio

Índice

Lista de figuras

Lista de cuadros

Resumen

En el siguiente proyecto se va a desarrollar un conjunto de algoritmos para coordinación de enjambre de drones *Crazyflie 2.0* para la ejecución de formaciones en un ambiente analizado por un sistema de captura de movimiento, OptiTrack. Estos algoritmos se basan en la formación y control de los drones.

Los algoritmos serán desarrollados a través de aplicación de métodos de toma de decisiones en conjunto, la obtención de datos a través del sistema OptiTrack y la verificación de estos algoritmos por medio de simulaciones y pruebas físicas.

La ejecución de formaciones con enjambre de drones presenta diversos retos y por eso este trabajo busca solucionar los problemas como colisiones y evasión de obstáculos. Esto proporcionará autonomía a cada dron, esto porque serán capaz de ejecutar la trayectoria prevista pero con la capacidad de re-calcularla si se presenta algún obstáculo.

Se han desarrollado distintos trabajos que buscan la ejecución de formaciones con *Crazyflies*, pero se han encontrado con diversos problemas como el aumento del error de la posición conforme aumentan la cantidad de drones. En otros casos se han trabajado proyectos que permiten el calculo de las trayectorias de los drones para ejecutar las formaciones pero sin utilizar control, pues pueden calcular trayectorias pero no evadir obstáculos.

Este proyecto mezcla diversos temas, como sistemas de control, algoritmos de consenso que permitan al enjambre comunicarse para poder trasladarse o rotar, conocer las características del *Crazyflie 2.0* y del sistema OptiTrack para la implementación de la información que caracteriza a ambos sistemas y aplicar conceptos de robótica de enjambre para tomar en cuenta factores que reducen el óptimo funcionamiento del enjambre.

Abstract

In the following project we will develop a set of algorithms for coordinating swarm of *Crazyflie 2.0* drones for the execution of formations in an environment analyzed by a motion capture system, OptiTrack. These algorithms are based on drone formation and control.

The algorithms will be developed through application of ensemble decision making methods, data collection through the OptiTrack system and verification of these algorithms through simulations and physical tests.

The execution of drone swarm formations presents several challenges and that is why this work seeks to solve problems such as collisions and obstacle avoidance. This will provide autonomy to each drone, because they will be able to execute the planned trajectory but with the ability to re-calculate it if an obstacle occurs.

Different works have been developed that seek the execution of formations with *Crazyflies*, but they have encountered several problems such as the increase of the position error as the number of drones increases. In other cases, projects have been developed that allow the calculation of the trajectories of the drones to execute the formations but without using control, since they can calculate trajectories but not avoid obstacles.

This project mixes several topics, such as control systems, consensus algorithms that allow the swarm to communicate in order to move or rotate, to know the characteristics of the *Crazyflie 2.0* and the OptiTrack system for the implementation of the information that characterizes both systems and to apply swarm robotics concepts to take into account factors that reduce the optimal functioning of the swarm.

CAPÍTULO 1

Introducción

La tecnología se ha desarrollado a un ritmo más rápido los últimos años, es por ello que han aparecido nuevas herramientas para uso del mundo. Es por esto que en la Universidad del Valle de Guatemala proporciona cursos como sistemas de control y robótica los cuales son la base en la cuál se desarrollará el siguiente trabajo.

A continuación se presenta el proceso para el desarrollo de un algoritmo de coordinación de drones *Crazyflie 2.0*, éste algoritmo de desarrolla bajo la teoría de control moderno y grafos. Para ello se toma como inicio los algoritmos y resultados de trabajos previos, para su estudio y extracción de información que sea de utilidad. Con esta información se procede a realizar pruebas con los algoritmos para poder elegir el que sea útil para la coordinación de *Crazyflie 2.0*. Posteriormente se seleccionarán las formaciones de prueba y las métricas que evaluarán el resultado.

Este proyecto se divide, entonces, en las siguientes partes: estudio y desarrollo de algoritmo: en dónde se busca a través de un proceso iterativo encontrar el que mejor se adapte a la aplicación; formación de prueba: en dónde se va a define y estudia que tipo de grafo es exitoso para la aplicación; La métrica de evaluación para obtener datos estadísticos sobre la formación y velocidad de formación y la implementación del algoritmo en los *Crazyflies 2.0* obteniendo los datos por medio del sistema OptiTrack.

CAPÍTULO 2

Antecedentes

El *Crazyflie 2.0* es un cuadricóptero de código abierto de tamaño reducido [1]. Este dron ha sido utilizado por distintas instituciones educativas y de investigación. Gracias a la versatilidad del dispositivo, se han realizado distintas variaciones en sus aplicaciones de trabajo.

Figura 1: Cuadricóptero *Crazyflie 2.0* [1].

Varias instituciones e individuos que trabajar con estos drones, han implementado sistemas de control para su funcionamiento. Por ello se trabajan distintas simulaciones y pruebas para poder minimizar las pérdidas al momento de realizar las pruebas físicas. El uso de drones tiene un potencial para ejecución de tareas específicas en el mundo, por ello el poder darle instrucciones y que estos las ejecuten sin mayor percance es de suma importancia.

Se han trabajado distintos software que permite su control, tanto como dron individual como enjambre. Estos software han probado ser efectivos pero contienen deficiencias como la coordinación de distintos drones en ambientes con obstáculos.

2.1. Crazyswarm: software de código abierto para control de múltiples drones:

En 2017, Preiss, Wolfgang [2] y colaboradores trabajaron este software que permitía el control de múltiples drones. Este software utilizaba la información del entorno a partir de 14 cámaras VICON. La obtención de estos datos se realizaba a través de VICON Tracker. El desarrollo de esta plataforma se basa en la utilización de herramientas y librerías ROS, con lenguaje C++ en Ubuntu.

Durante la experimentación monitorearon el comportamiento del enjambre según la cantidad de drones. Como resultado se obtuvo que, entré más drones realizaban la tarea, el error de posición aumentaba. Esto se debe a que la aerodinámica del entorno cambiaba debido a los demás drones. En la Figura 2 se observa una formación de cuarenta drones y el sistema de cámaras VICON.

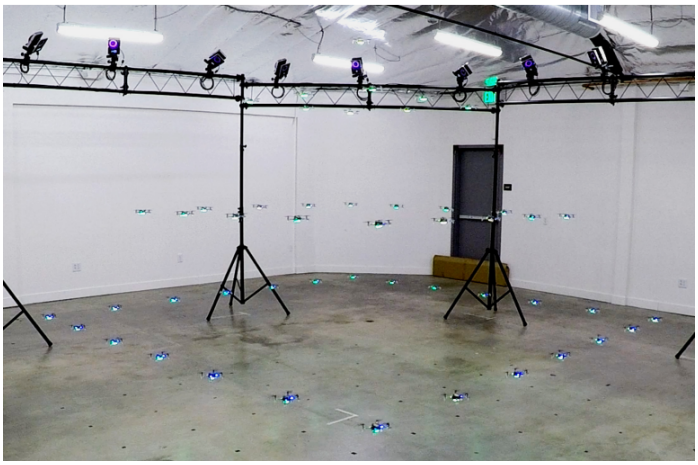

Figura 2: Formación de cuarenta Crazyflies en forma de pirámide [2].

2.2. Uso de distintos Crazyflies como UAV's:

Raúl Zahínos presentó resultados experimentales para el control de 9 Crazyflies 2.0 [3], comunicadas a partir de una computadora por medio de crazyradio PA con un alcance de hasta 1 km. Para poder obtener información del entorno se trabajó en un banco de pruebas de CATEC que les proporciona datos a partir de la convención ENU (East - North - Up). A pesar que el sistema controlado no es lineal, se utilizaron técnicas de control clásico, controlador PID, para poder trabajar con el sistema.

Inicialmente, la implementación del control de estos drones utilizaba el feedback del sistema VICON por medio de la librería Crazyflie-lib-python, software que proporciona bitcraze para el control de distintos drones por antena. El problema que surgió de esta librería es que no es viable utilizarla para trabajos a alta frecuencia. Por estos y demás problemas se migró a trabajar con el proyecto Crazyswarm. Con éste, aún presentaba un aumento en el error euclidiano al momento de trabajar a una mayor frecuencia y con varios drones pero era menor que con el software anterior. El resultado de este trabajo se puede

ver en la Figura 3 y Figura 4.

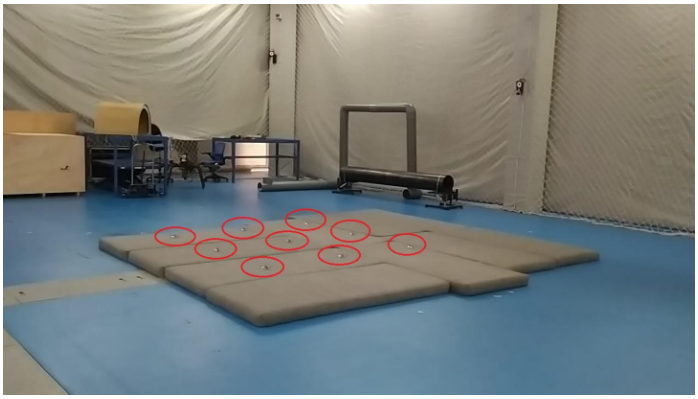

Figura 3: Nueve Crazyflies antes del despegue [3].

Figura 4: Nueve Crazyflies después del despegue [3].

2.3. Simulación de formación de drones:

Luka Jeroncic, en mayo del 2021, desarrolló un conjunto de simulaciones de enjambre de UAVs [4]. Para estas simulaciones, encontró que al trabajar con varios UAVs, se debía considerar que estos deben tener cooperación entre si. Estas pruebas buscaban evitar colisiones entre ellos y las colisiones con el entorno.

Las simulaciones fueron desarrolladas en Unreal Engine 4 y AirSim, herramientas que permitían el uso del sensor que proporciona la pose del dron (orientación y posición). Estos parámetros pueden ser modificados en AirSim. Los drones cuentan con sensores que permite la lectura de objetos en un rango de 4 metros a la redonda. Se realizaron 15 iteraciones, los datos obtenidos fueron la desviación estándar, el tiempo promedio en completar la formación y la cantidad de colisiones que sufrieron durante la formación.

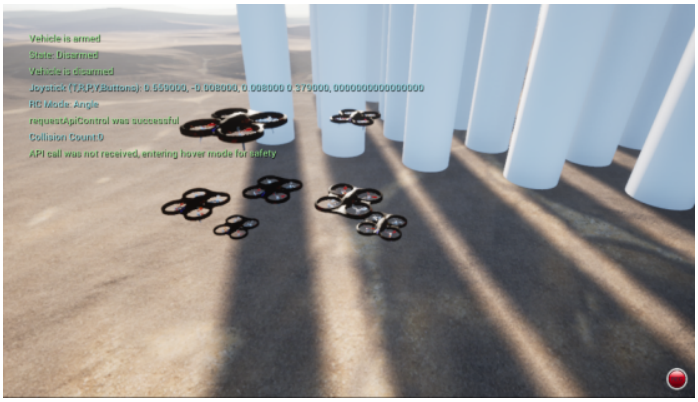

Figura 5: Pruebas de formación inicial con 6 drones [4].

2.4. Algoritmo para trayectorias de UAVs

En el 2020, Hao Zhuo y su equipo trabajaron un algoritmo de planificación de trayectorias para UAVs, este algoritmo se basa en restricciones exactas de posicionamiento para el sistema [5]. Este trabajo garantiza que el UAV pueda llegar a una posición indicada a través de la trayectoria mas corta desde el punto de salida hasta el punto final. Para esto se utilizan múltiples restricciones y la menor cantidad de correcciones de errores. La limitante de este trabajo es que no se tiene la retroalimentación del entorno por medio de un sistema de captura de movimiento, por lo que el algoritmo se basa en considerar el error de posicionamiento durante su vuelo debido a los factores presentes en el entorno. Esto lo trabajan a partir de puntos de corrección en el espacio de vuelo y así corregir el error, como se puede ver en la Figura 6. Con esto en mente, planifican la mejor trayectoria de forma que el UAV sea corregido por los puntos que se definieron como correctos para que el vuelo sea lo más corto y rápido posible.

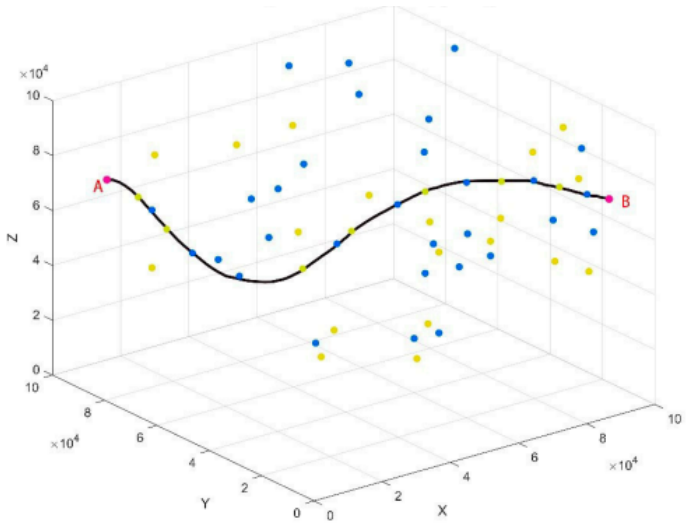

Figura 6: Algoritmo de trayectoria por medio de puntos [5].

2.5. Trabajos previos en UVG:

Los trabajos realizados en UVG, permiten tener una base y ver los comportamientos del *Crazyflie 2.0* en distintos escenarios. Los trabajos nos otorgan la ventaja de poder modificar y estudiar los controladores de los drones.

En el 2019, Gabriel Martinez realizó el diseño e implementación de una plataforma para el *Crazyflie 2.0* con el propósito de tener un sistema controlado para poder configurar y realizar pruebas de algoritmos de control de actitud para el dron [6]. Este trabajó fue la base para que en el 2021, Francis Sanabria trabajara en la implementación del cuadricótero a esta plataforma. El diseño de la plataforma con el dron se puede ver en la Figura 7.

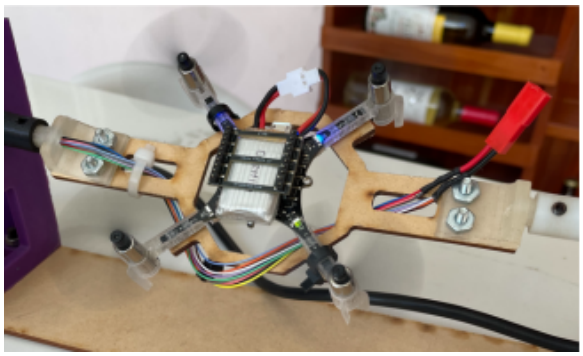

Figura 7: Plataforma con *Crazyflie 2.0* montado [7].

El trabajo de Sanabria permite configurar y estudiar el sistema de control del dron para entender cómo se orienta [7]. Esto lo logró trabajando con el control de actitud del *Crazyflie 2.0*. Para poder cumplir este objetivo, redactó una guía básica de cómo se utiliza el dron, en esta se indicaba qué pasos seguir para poder modificar el firmware del dron. Además, desarrolló una interfaz gráfica para poder controlar el dron de manera sencilla. La interfaz permite al usuario leer en tiempo real el ángulo de banqueo, modificar los parámetros del controlador PID y reiniciarlo a valores predeterminados del mismo y modificar el ángulo pitch al que se desea orientar el dron. En esta interfaz, también permite conectar, desconectar y tarar el dron. Como el objetivo de este proyecto era el estudio y análisis del comportamiento del dron, también cuenta con la función de guardar la data de la iteración. La interfaz se puede ver en la Figura 8. Las limitaciones de este trabajo fueron que no se tiene el comportamiento del dron con todos sus grados de libertad debido al diseño de la plataforma por lo que se tiene información del dron operando en un ambiente completamente controlado.

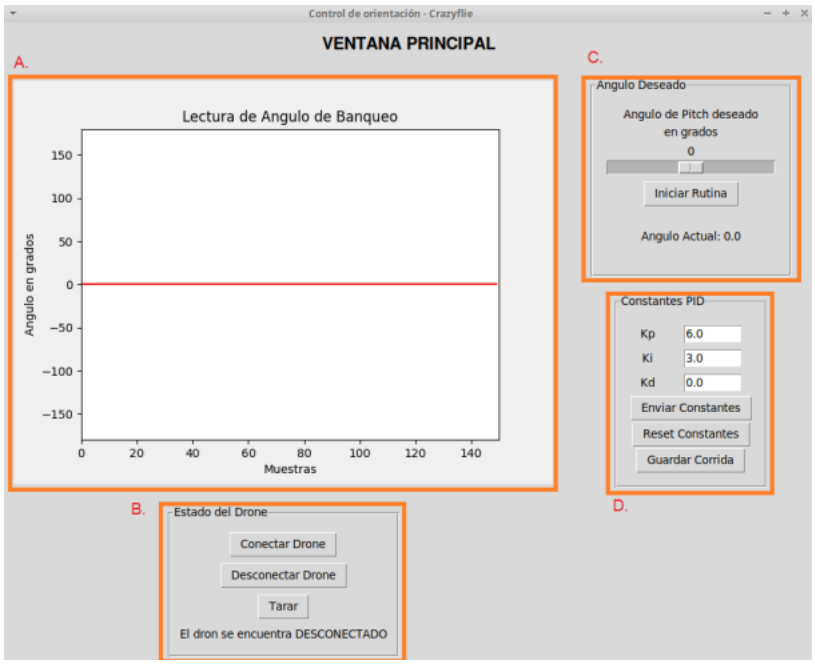

Figura 8: Interfaz gráfica [7].

Steven Castillo, trabajó una plataforma de pruebas en forma de *gimball* [8]. La ventaja de esta plataforma es que presenta 3 grados de libertad, que son los ángulos de vuelo del dron. Para el control de este dron, se utilizó el controlador de vuelo automático Pixhawk 1. Este controlador puede ser modificado a través de una herramienta en Matlab/Simulink. En esta herramienta modificaba los cambios de velocidad de los motores para posteriormente realizar análisis de esfuerzos de manera teórica y por simulaciones para las piezas del sistema.

Andrea Peña, en el 2019 desarrolló un algoritmo para el control de robots de dos ruedas, en su caso utilizó e-puck. Para su trabajo, hizo pruebas con distintos controladores y así comprobar cuál sería más efectivo. Como resultado de su experimentación, utilizó 3 tipos distintos de controlador para aumentar la eficiencia de la energía y tiempo de formación, en dónde destaca el uso de un coseno hiperbólico para optimizar el uso de energía. Para las pruebas del controlador, trabajó el código en matlab para poder realizar simulaciones que le permitieran ver el comportamiento de los agentes. Cómo parte del trabajo, también obtuvo gráficos de cómo se comportaba la velocidad de cada agente con cada controlador y en la Figura 9 se puede observar la trayectoria que siguieron los agentes de inicio a fin. Adicionalmente implementó un control de velocidad para que no excediera los límites físicos del robot. Por último, trabajó en la formación de los robots implementando evasión de obstáculos, lo cuál era útil por su orientación de drones exploradores. Esto no lo pudo implementar de forma física pero, si trabajó la simulación en Webots cómo se puede observar en la Figura 10

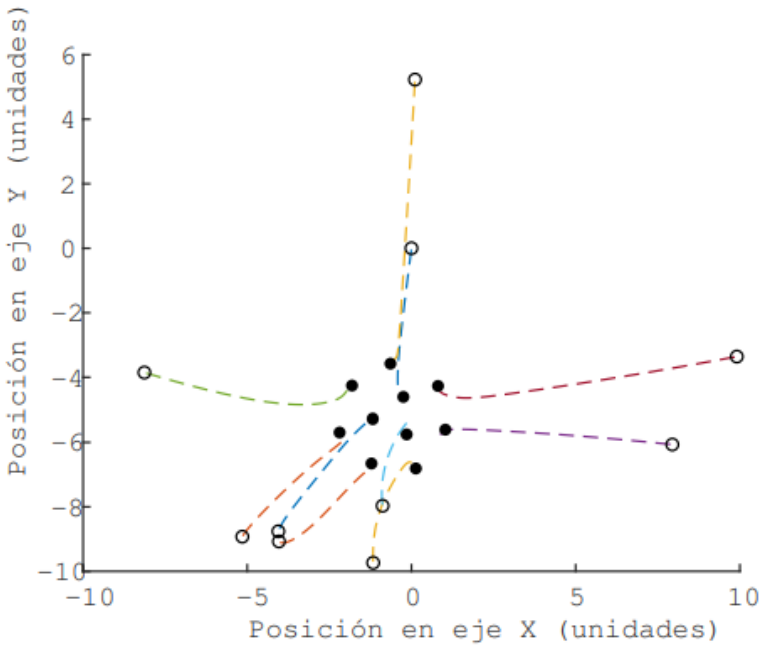

Figura 9: Trayectorias seguidas por los agentes para su formación [9]

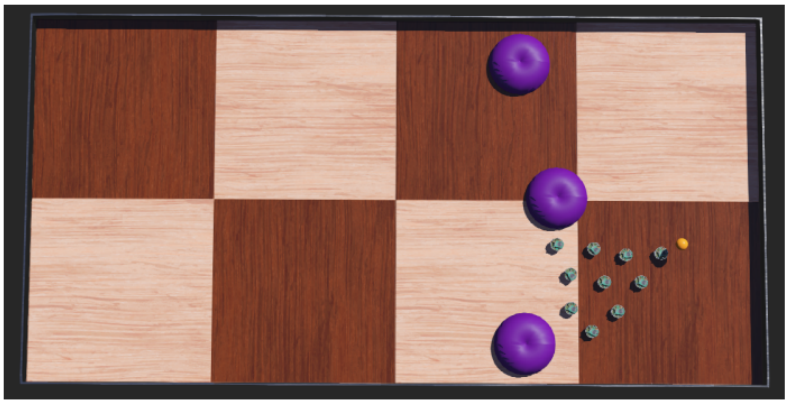

Figura 10: Simulacion en Webots de los algoritmos [9]

Es importante mencionar que los trabajos previos no permiten su implementación para la realización de enjambres de drones, por lo que se restringue la posiblidad de utilizar los drones para ejecutar formaciones o trayectorias específicas.

CAPÍTULO 3

Justificación

El propósito de este proyecto es desarrollar algoritmos de enjambre de drones *Crazyflie 2.0* que utilicen un sistema de captura de movimiento OptiTrack con el fin de realizar las formaciones que se le soliciten de forma autónoma. Esto permitirá crear enjambres inteligentes utilizando sistemas de control para la ejecución de trayectorias y evasión de obstáculos, utilizando la realimentación del ecosistema Robotat.

Estos algoritmos evitarán colisiones entre los drones del enjambre y con cualquier cuerpo que esté dentro del entorno de trabajo. Los obstáculos pueden estar en cualquier parte del ecosistema, inclusive dentro del área en dónde el enjambre ejecutará la formación. El algoritmo permitirá que si los drones ya están en la formación deseada, si un obstáculo se aproxima, se movilizarán para esquivarlo.

Por medio de las simulaciones, se obtendrán datos estadísticos que permitan ver cómo reaccionan los drones al algoritmo cuando sea ejecutado, con el fin de ver cuántos drones podrán ser controlados.

Los enjambres de drones pueden ser utilizados para aplicaciones específicas, por ejemplo, en el campo de la agricultura, para monitorear, vigilar o aplicar plaguicidas al cultivo. Con un enjambre, se tiene un área de trabajo más grande, por lo que el tiempo que tarda en realizarse una tarea se reduce.

CAPÍTULO 4

Objetivos

4.1. Objetivo General

Desarrollar algoritmos de coordinación de enjambre de drones *Crazyflie 2.0* para la ejecución de formaciones, utilizando técnicas de control y el sistema de captura de movimiento OptiTrack.

4.2. Objetivos Específicos

- Evaluar algoritmos de control y formación para drones *Crazyflie 2.0*.
- Validar los algoritmos desarrollados por medio de simulaciones computarizadas.
- Definir los parámetros que necesita transmitir el sistema OptiTrack al *Crazyflie 2.0* para que el dron pueda conocer su entorno.
- Realizar pruebas físicas con los algoritmos para verificar el correcto funcionamiento del *Crazyflie 2.0*.
- Obtener estadísticas de posición y trayectoria del *Crazyflie 2.0* durante la ejecución de la formación.

CAPÍTULO 5

Alcance

El resultado de este trabajo proporciona un algoritmo funcional que permite la coordinación de un enjambre de drones desde una posición aleatoria en el espacio. Para ello se utiliza un sistema de posicionamiento que permite que cada dron sepa en qué posición de la formación debe ubicarse, así como la posición de los demás para poder determinar quienes serán los drones adyacentes.

El conjunto de simulaciones fueron desarrolladas y verificadas a través de Matlab. Las simulaciones constan de dos formaciones distintas, Es importante resaltar que en este *software* no se toman en cuenta las limitaciones físicas que proporcionan los drones y el sistema de captura de movimiento a utilizar. Seguido, se transcribirá el código de Matlab para el lenguaje que utilice el firmware de los *Crazyflie 2.0*. Las pruebas realizadas en la simulación consta de dos formaciones de 8 agentes cada una.

Por último se implementará el algoritmo en los drones para las pruebas físicas. Con esto se cambiará un sistema omnisapiente de las posiciones de los drones por el sistema OptiTrack y los agentes por los drones para poder comprobar como reaccionan antes las limitantes de cada herramientas. Esto nos permitirá validar los algoritmos o realizar los cambios necesarios para su correcto funcionamiento, siendo verificado por medio de métricas de evaluación que permitan conocer la velocidad, energía utilizada y posición final de los drones.

CAPÍTULO 6

Marco teórico

El algoritmo desarrollado se basa, principalmente, en la utilización de la teoría de grafos y control moderno, es por ello que en la siguiente sección se describen conceptos importantes que se utilizarán en el resto del trabajo. Adicionalmente, se proporciona información útil de otros temas como el sistema OptiTrack, el cuál también tiene participación en este trabajo.

6.1. Sistema OptiTrack

El sistema de captura de movimiento, OptiTrack, consiste en múltiples cámaras sincronizadas alrededor de un volumen de captura objetivo. Éste captura imágenes 2D con cada cámara. Posteriormente se calculan las posiciones 2D y los datos de posición de cada cámara se comparan para calcular las posiciones en 3D por medio de triangulación. Estas posiciones en 2D se obtienen por medio la detección de la luz reflejada que es emitida por las cámaras, como se puede ver en la Figura 11. Por esto se recomienda minimizar la iluminación ambiental para evitar interferencias como luz solar [10].

Figura 11: Cámara Prime x 41 del sistema OptiTrack en UVG.

El conjunto de cámaras se recomienda instalarse alrededor de la periferia del volumen objetivo, las cámaras a distinta altura para obtener distintas perspectivas del volumen. Las cámaras Primex 41 cuentan con las siguientes especificaciones técnicas:

- **Resolución:** 2048 × 2048
- **Velocidad de fotogramas:** 180 Hz.
- **Precisión 3D:** +/- 0.10 mm.
- **Rango para marcadores pasivos:** 30 mm.
- **Rango para marcadores activos:** 45 m.

6.2. *Crazyflie 2.0*

El *Crazyflie 2.0* es un dron de 4 motores, cuadricótero, de bajo costo y ligero. Este permite ser controlado por medio de Bluetooth LE y por radiofrecuencia, utilizando un componente adicional, Crazyradio PA. El *Crazyflie 2.0* se carga en 40 minutos y permite un tiempo de vuelo de 7 minutos. Tiene integrado diversos sensores como un giroscopio, acelerómetro y magnetómetro de 3 ejes cada uno, así como un sensor de presión.

6.3. Teoría de grafos

Un grafo se forma por medio de vértices y aristas, las cuales se encargan de conectar los vértices. Cómo uso general, se presentan como una pareja de conjuntos (V, E), en dónde V es el conjunto que define a los vértices y E el de aristas, en dónde E se representa a través de pares, de forma que cumple $(a, b) \in V$ [11].

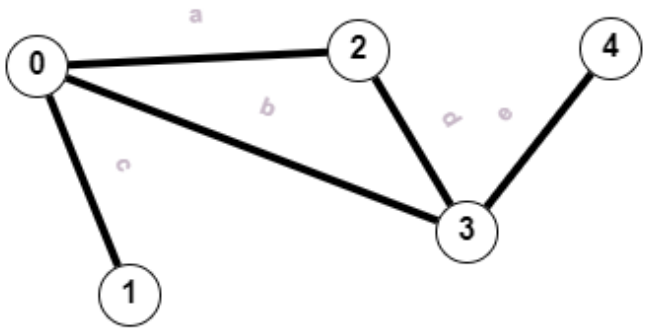

Figura 12: Representación de un grafo no dirigido

Si se observa la Figura 12, la forma formal de escribir este grafo sería: $A = (V, E)$ en dónde $V = 0, 1, 2, 3, 4$, siendo este el conjunto de los vértices y $E = (0, 1), (0, 2), (0, 3), (2, 3), (3, 4)$, siendo este el conjunto de aristas.

6.3.1. Definiciones básicas

Clasificación de los grafos:

- Simple: este grafo consiste en que, cualquier par de vértices tienen, cómo máximo, una arista que los une.
- Conexo: un grafo es conexo si para cualquier par de vértices (a, b) existe, por lo menos, un camino entre ayb.
- Completo: este tipo de grafo tiene para todo par de vértices (a, b) una arista que los une. Quiere decir que todos los vértices están unidos entre sí.
- Ponderado: este tipo de grafos tienen un número (magnitud) definido en cada arista, es decir, un peso.

6.4. Algoritmos de consenso

Cuando se tiene una red de agentes dinámicos, es común encontrarse con el problema de consenso. La solución para este problema consiste en que, cada agente debe llegar a un

consenso, esto se refiere a un estado en el que se consigue un acuerdo para poder alcanzar una meta o valor en común según el interés dependiente del estado de los agentes que conforman la red. Este estado se alcanza cuando se tiene constante comunicación entre la red por medio de enlaces de comunicación o sensores. Este intercambio de información se le llama algoritmo de consenso. Los algoritmos de consenso tienen el propósito de que cada agente actualice su información a partir de la información de los otros agentes [12].

Para estas redes multi-agentes, se trabaja la estructura con grafos para indicar que tipo de comunicación habrá entre la red. Se utilizan dos tipos grafos, los dirigidos y los no dirigidos. En la Figura 13 se observan ejemplos de ambos tipos. Para el primer caso, cada arista es representada por una flecha, en dónde la cola indica el agente que envía la información, mientras que la punta indica el agente receptor de la información. Para el segundo caso, las aristas son lineas continuas, estos se utilizan en redes en dónde la comunicación entre los agentes es bidireccional [13].

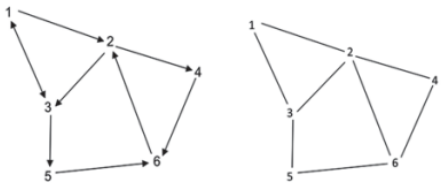

Figura 13: Ejemplos de grafo dirigido (izquierda) y no dirigido (derecha) [13].

En los algoritmos se requiere de una estructura de intercambio de información, la cuál puede ser una estructura sin líder o una con líder y seguidores. La primera estructura funciona de forma que, cada agente se comunica con sus vecinos, y estos no siguen a ningún agente en específico. Esta estructura permite que, aunque no se le de prioridad a una orden deseada a los agentes, eventualmente la salida de cada agente converge a un valor común. En la segunda estructura mencionada, el agente líder puede afectar a los seguidores siempre que se encuentre en su conjunto vecino pero, presenta el problema que no recibe retroalimentación de sus seguidores.

6.4.1. Modelo de votantes:

El modelo de votantes hace que un agente considere las situación actual de cada vecino, entonces elige al azar a un vecino y cambia su situación por la del vecino, si es distinta a la propia. Este modelo es bastante robusto, pues este modelo ayuda a converger en la decisión correcta, por lo que presenta un alto grado de precisión [14].

6.4.2. Regla de la mayoría:

Este modelo es de los más sencillos para la toma de decisiones colectivas. Para este modelo, el agente comprueba a su grupo de vecinos y cuenta la recurrencia de cada opción. Los agentes pueden cambiar su decisión a la opción más frecuente que encontró entre sus vecinos, esta regla es rápida pero menos precisa que el modelo del votante [14].

6.5. Robótica de enjambre

La robótica de enjambre es el estudio del diseño de un gran número de agentes relativamente sencillos para que ejecuten un comportamiento colectivo a partir de las interacciones entre los agentes o entre los agentes y el entorno. Las interacciones tienen un papel clave en los enjambres, pues es necesario que cada agente tenga las capacidades de detección local y comunicación [14].

El rendimiento medio de un enjambre, depende del tamaño del enjambre si el espacio de trabajo se mantiene constante. Esto sucede porque, cuando se agregan más robots al sistema, la densidad de agentes va aumentando, y llegada una determinada cantidad, se tendrá el rendimiento óptimo del enjambre. Si pasado este punto, sigue aumentando la cantidad de robots, el rendimiento disminuye porque hay una mayor interferencia entre cada uno y esto ralentiza las acciones, esto puede aumentar hasta el punto que el enjambre no pueda moverse debido a las interferencias [14].

CAPÍTULO 7

Desarrollo del algoritmo de control

El capítulo a continuación presenta el proceso del desarrollo del algoritmo de control. Las simulaciones del algoritmo se desarrollaron en Matlab ya que es una herramienta que permite realizar cálculos complejos y permite representar información de forma sencilla.

7.1. Grafo para formaciones

Para la definición del modelo matemático a utilizar, se realizaron distintas pruebas para poder encontrar el que permitiera el mejor comportamiento al formarse. Para esto se propuso un un grafo totalmente rígido, mostrado en la Figura 14, de 8 nodos.

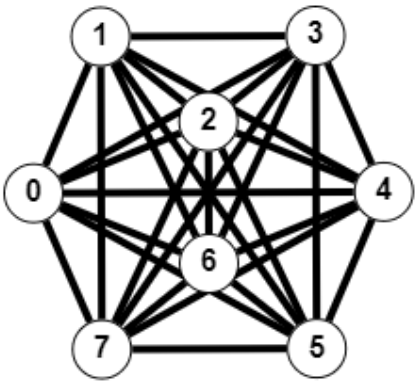

Figura 14: Grafo de prueba para formaciones

7.1.1. Matrices de adyacencia

Para el grafo se construyó dos matrices totalmente rígidas, es decir, una matriz de adyacencia para un grafo completo, para ver cómo se comportaba. Para ello se trabajaron dos matrices con distintos pesos para poder ver el comportamiento según la estructura de la formación.

Matriz de adyacencia rígida para la formación 1

$$\begin{bmatrix} 0 & 1 & 1 & \sqrt{2} & 1 & \sqrt{2} & \sqrt{2} & \sqrt{3} \\ 1 & 0 & \sqrt{2} & 1 & \sqrt{2} & \sqrt{3} & 1 & \sqrt{2} \\ 1 & \sqrt{2} & 0 & 1 & \sqrt{2} & 1 & \sqrt{3} & \sqrt{2} \\ \sqrt{2} & 1 & 1 & 0 & \sqrt{3} & \sqrt{2} & \sqrt{2} & 1 \\ 1 & \sqrt{2} & \sqrt{2} & \sqrt{3} & 0 & 1 & 1 & \sqrt{2} \\ \sqrt{2} & \sqrt{3} & 1 & \sqrt{2} & 1 & 0 & \sqrt{2} & 1 \\ \sqrt{2} & 1 & \sqrt{3} & \sqrt{2} & 1 & \sqrt{2} & 0 & 1 \\ \sqrt{3} & \sqrt{2} & \sqrt{2} & 1 & \sqrt{2} & 1 & 1 & 0 \end{bmatrix}$$

Matriz de adyacencia rígida para la formación 2

$$\begin{bmatrix} 0 & 2 & 2 & \sqrt{3} & 2 & \sqrt{8} & \sqrt{3} & \sqrt{7} \\ 2 & 0 & 2 & 1 & \sqrt{8} & 2 & \sqrt{8} & \sqrt{5} \\ 2 & 2 & 0 & 1 & \sqrt{8} & \sqrt{8} & 2 & \sqrt{5} \\ \sqrt{3} & 1 & 1 & 0 & \sqrt{7} & \sqrt{5} & \sqrt{5} & 2 \\ 2 & \sqrt{8} & \sqrt{8} & \sqrt{8} & 0 & 2 & 2 & \sqrt{3} \\ \sqrt{8} & \sqrt{8} & 2 & \sqrt{5} & 2 & 0 & 2 & 1 \\ \sqrt{8} & \sqrt{8} & 2 & \sqrt{5} & 2 & 2 & 0 & 1 \\ \sqrt{7} & \sqrt{5} & \sqrt{5} & 2 & \sqrt{3} & 1 & 1 & 0 \end{bmatrix}$$

En este caso, cada arista del grafo representa las distancias que deben mantener los agentes para realizar la formación, lo cual determina realmente la comunicación entre cada agente del enjambre. Es importante resaltar que al momento de utilizar los grafos completos, se está definiendo una red de comunicación compleja debido a que cada robot debe saber la posición de todos los agentes. Es por ello que adicional al controlador, en Matlab se realiza un sub algoritmo para que los agentes empiecen en posiciones aleatorias en un espacio de tres dimensiones.

7.2. Métricas de evaluación

El proceso del desarrollo del algoritmo conlleva una gran cantidad de iteraciones para poder comprobar qué tan efectivo está siendo el controlador según la posición en la que empiezan los robots del enjambre. Es por ello que se define un conjunto de métricas para poder comparar y verificar cómo se comporta el algoritmo.

7.2.1. Éxito en la formación

Los aspectos principales a tomar en cuenta en las métricas es la cantidad de energía utilizada por el sistema y si la formación cumple realmente con los pesos que se establecieron en la matriz de adyacencia. Para ello, se utiliza el error cuadrático medio:

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ij}-d_{ij})^2 \tag{1}$$

en dónde x_{ij} representa la distancia instantánea entre el agente i y el agente j , d_{ij} es la distancia que deben tener los agentes entre sus adyacentes, en este caso es la posición ij de la matriz de adyacencia. Para esto, como se trabajó con grafos totalmente rígidos, se tiene la rúbrica que la formación debe tener un error cuadrático medio menor a los límites indicados, en este caso es 0.001 para éxito y 0.07 para fallo.

7.3. Ecuaciones de consenso básicas

Como base del algoritmo desarrollado, es importante conocer los distintos tipos de ecuaciones existentes para poder conocer el comportamiento que tendrían. Por ello se utilizó la ecuación de consensus, la cual permite la convergencia de los robots en un punto. Posterior a su simulación, se trabajó con la ecuación de control de formaciones para la realizar la geometría indicada en la matriz de adyacencia. Debe mencionarse el problema canónico de Rendezvous, que consiste en el problema de cómo hacer converger en una posición o lugar común y específico a N agentes. Es por ello que se puede observar en ambas simulaciones que la formación y consenso se lleva a cabo en el centroide de la posición indicial de los robots.

En la Figura 15, se puede observar cómo funciona la ecuación de consensus, en dónde no presenta cambios bruscos respecto a la posición y todos los agentes llegan a converger al mismo punto, comprobado a través de la velocidad de los mismos puesto que todas las velocidades convergen en 0 pues están en la posición deseada. En la Figura 17, se ve el resultado de la ecuación de formación en dónde los agentes cumplen con su función de realizar la formación indicada. Al igual que la ecuación de consensus, sus velocidades también convergen a 0. Para ambos casos se mantiene que se forman en el centroide de la posición inicial de los robots.

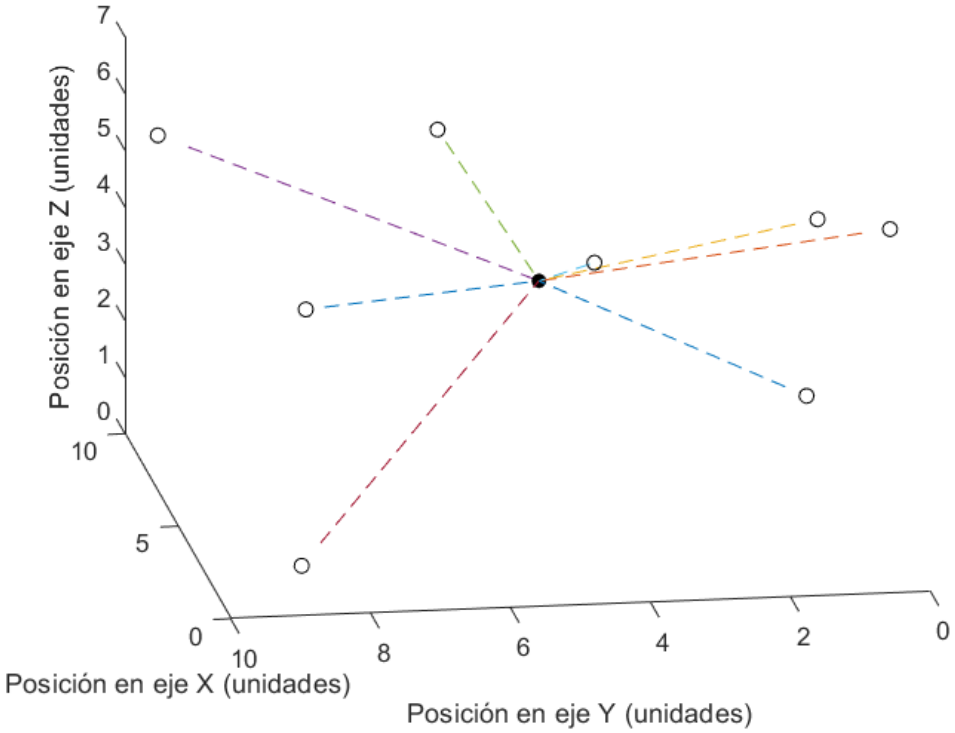

Figura 15: Trayectorias de los agentes con ecuación de consensus

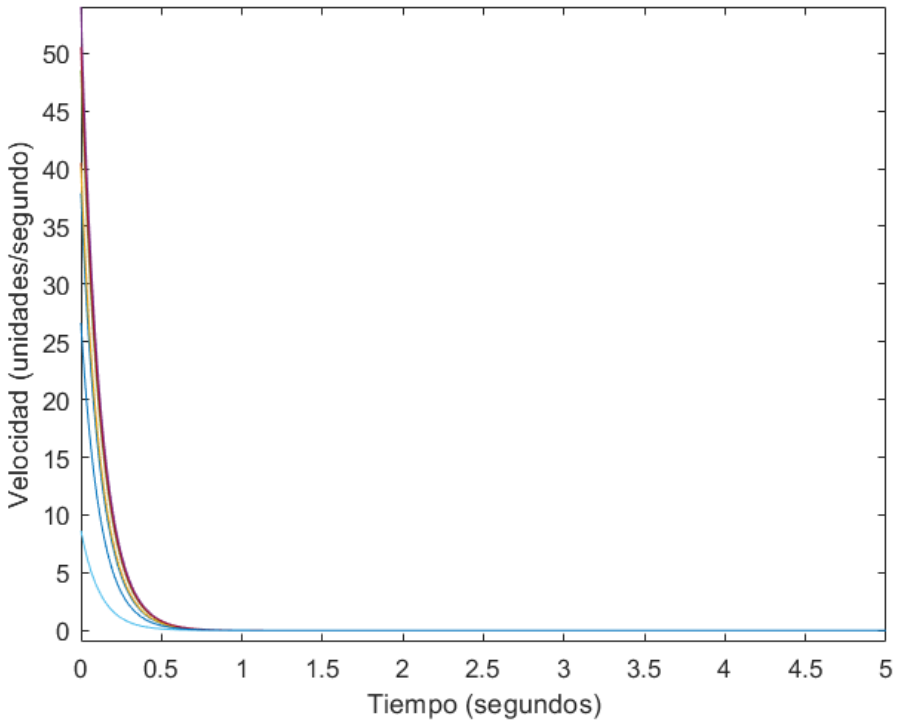

Figura 16: Magnitud de la velocidad de los agentes con ecuación de consensus

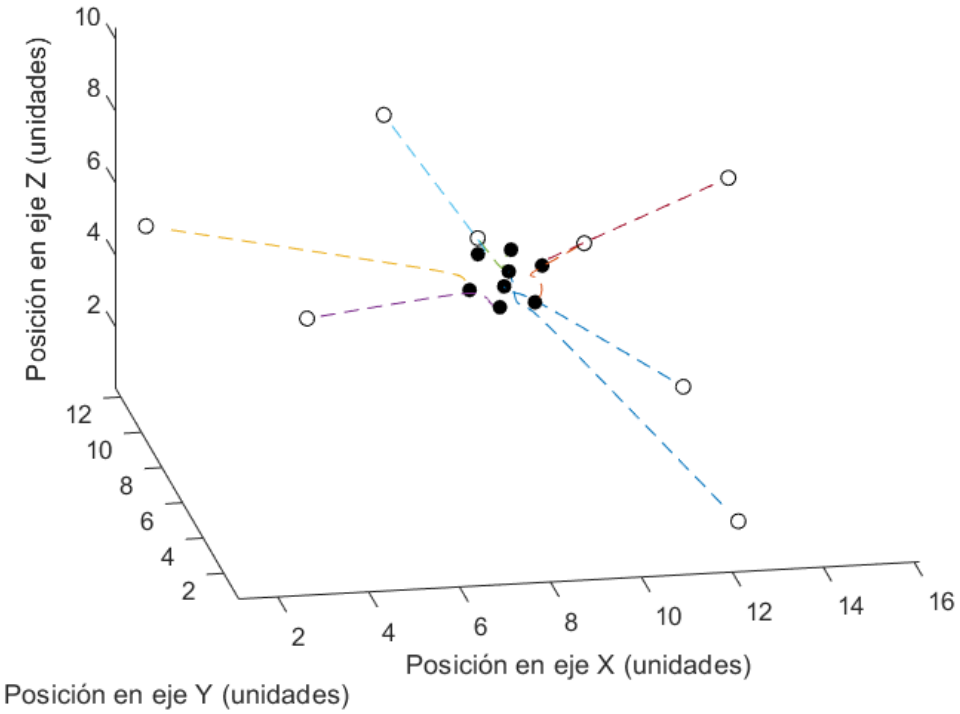

Figura 17: Trayectorias de los agentes con ecuación de formación

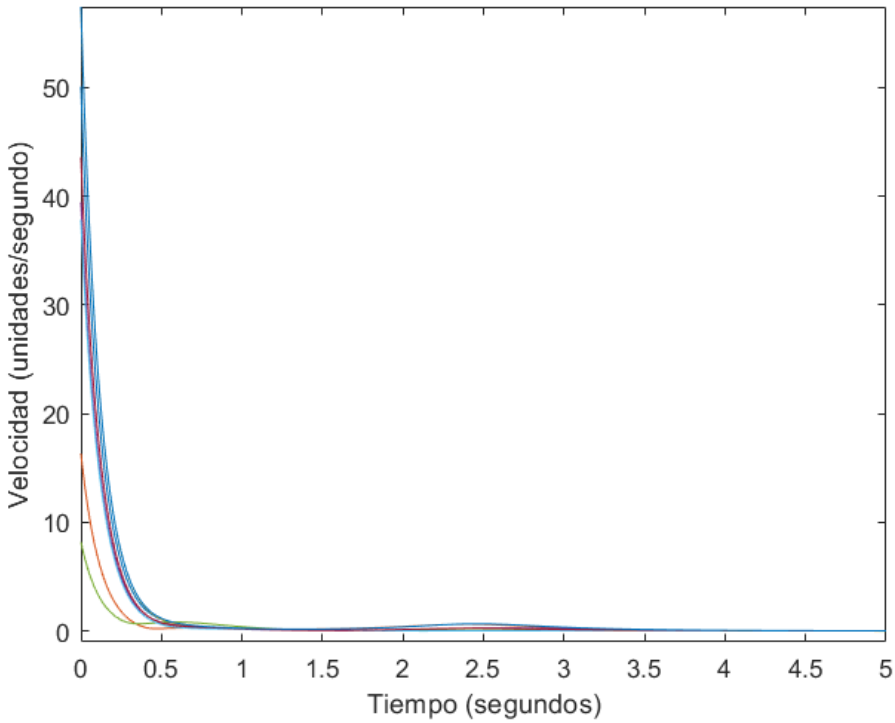

Figura 18: Magnitud de la velocidad de los agentes con ecuación de formación

7.3.1. Ecuación dinámica de consenso modificada

La ecuación de consenso con pesos tiene otras aplicaciones más relevantes, es por ello que se manipuló la tensión entre las aristas del grafo entre los agentes, esto con el propósito de aprovechar las ventajas que cada aplicación proporcionaba. Para este caso se trabajó con la ecuación de evasión de colisiones, puesto que permitía que se formaran con la restricción que, cada robot no entre en el radio físico del otro y con la ecuación de formación, que permite que los agentes se posicionen al lado de sus vecinos para que puedan formarse según la formación requerida.

Por ello, se trabajó el algoritmo con una ecuación dinámica, esto consiste en que, primero se acciona la ecuación para que los agentes se acerquen hasta un radio en dónde cada uno estuviera lo suficiente cerca del otro para poder empezar a formarse con el peso indicado del grafo, una vez en esta distancia, se cambia al controlador combinado mencionado anteriormente para que se ejecute la formación. El resultado de esta combinación se puede observar en la Figura 19 y 21, en dónde se cumple la formación definida en la matriz de adyacencia. Posteriormente, se puede visualizar en la Figura 20 y 22, cómo se comporta la velocidad de los agentes, en dónde se puede apreciar el cambio de velocidad debido al cambio de la ecuación.

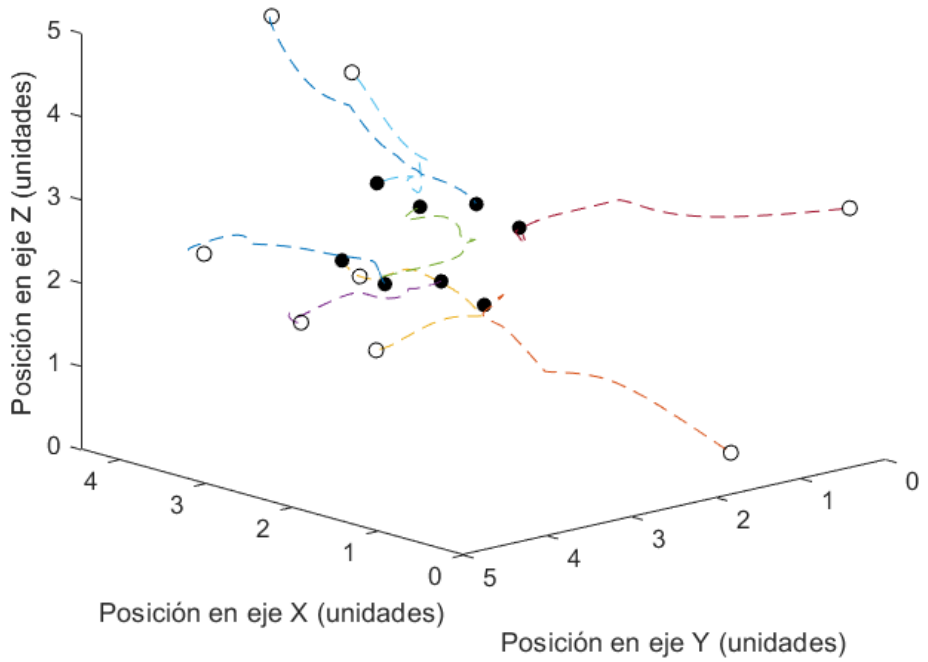

Figura 19: Trayectorias de los agentes con ecuación combinada para el grafo 1

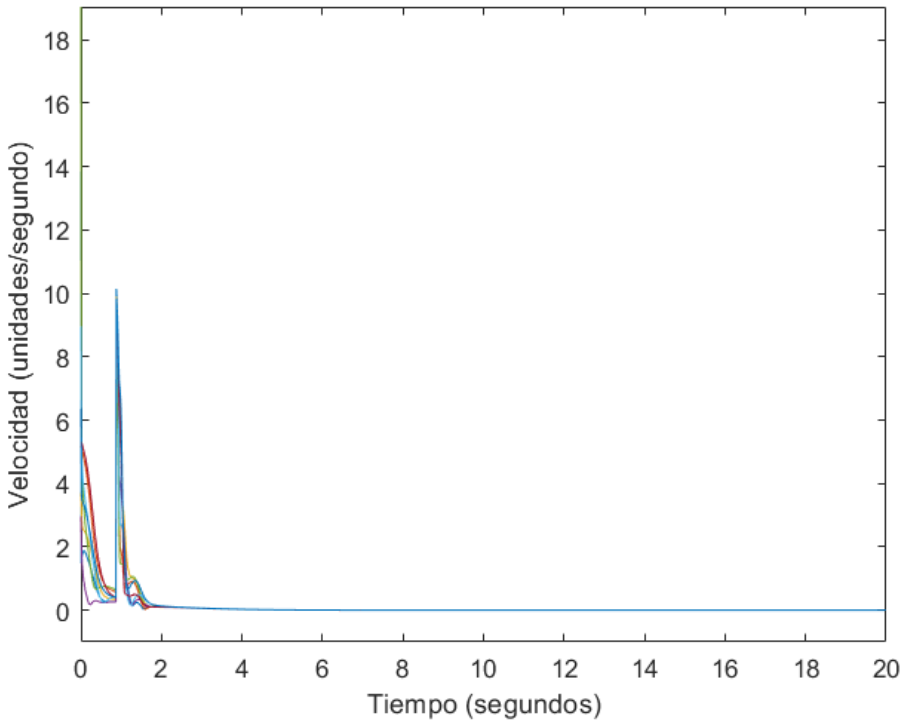

Figura 20: Magnitud de la velocidad de los agentes con ecuación combinada para el grafo 1

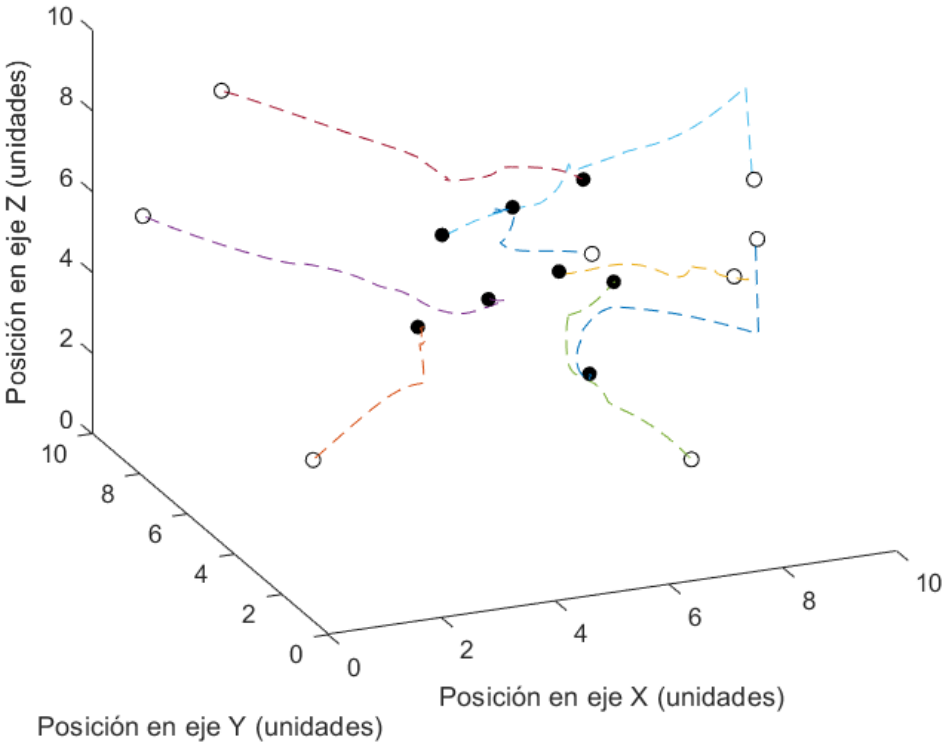

Figura 21: Trayectorias de los agentes con ecuación combinada para el grafo 2

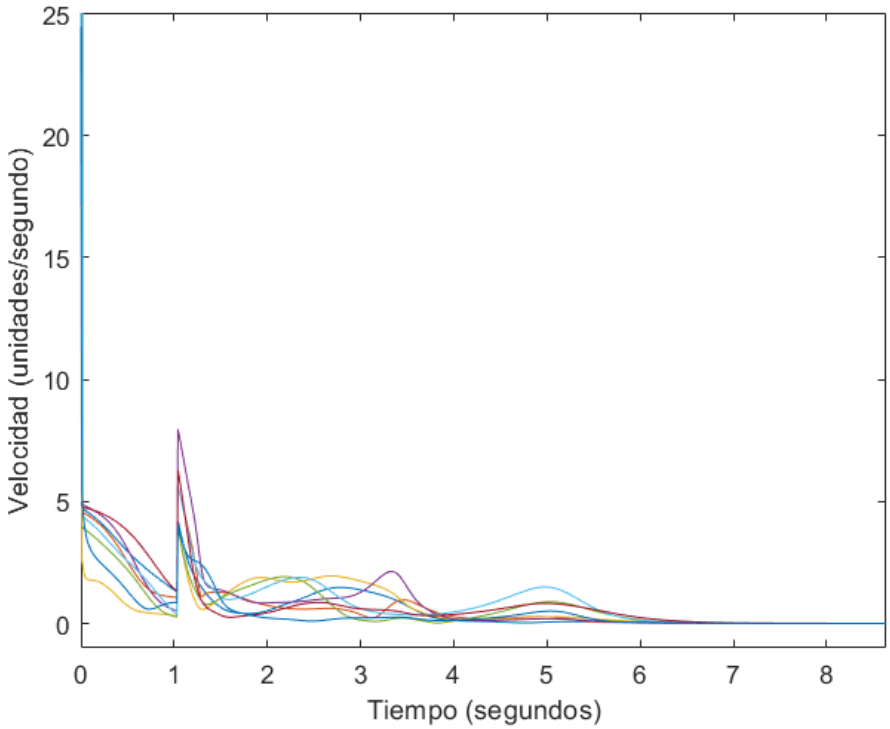

Figura 22: Magnitud de la velocidad de los agentes con ecuación combinada para el grafo 2

Modelo	Grafo	Exitosa	Fallida
Modificado	1	71 %	29 %
Modificado	2	63 %	37 %

Cuadro 1: Porcentaje de formación exitosa. El cuadro muestra que porcentaje de las simulaciones logró formarse correctamente.

CAPÍTULO 8

Conclusiones

El uso de grafos completamente rígidos presenta una red de comunicación compleja para la formación en 3 dimensiones pero, resulta ser una herramienta útil debido a que, por la cantidad de grados de libertad que presenta cada dron, resulta complejo la realización de grafos mínimamente rígidos.

El uso de un control variable permite que la formación se realice en un lapso menor pero tiene un costo alto de energía, por ello debe implementar una limitante de velocidad para reducir este costo y así evitar el deterioro de los actuadores físicos.

CAPÍTULO 9

Recomendaciones

Implementar un control para la aceleración de los agentes para evitar cambios drásticos en la velocidad.

Realizar pruebas con grafos mínimamente rígidos para comprobar si se obtiene alguna ventaja de utilizarlos en casos que únicamente formación.

CAPÍTULO 10

Bibliografía

[1] "Crazyflie 2.0." dirección: https://store.bitcraze.io/products/crazyflie2.

[2] J. A. Preiss, W. Honig, G. S. Sukhatme y N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," en *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, págs. 3299-3304.

[3] R. Zahínos, "Sistema de coordinación y control de múltiples vehículos aéreos no tripulados en testbed de interiores," Universidad de Sevilla, 2020.

[4] L. Jeroncic, "Drone Swarm Simulator," Tesis de mtría., The Arctic University of Norway, Tromsø, Norway, 2021.

[5] H. Zhou, H.-L. Xiong, Y. Liu, N.-D. Tan y L. Chen, "Trajectory planning algorithm of UAV Based on System Positioning Accuracy Constraints," *Electronics*, vol. 9, n.º 2, pág. 250, 2020. DOI: 10.3390/electronics9020250.

[6] G. Martínez, "Plataforma de pruebas y ajustes de sistemas de control para vehículos multirrotores," Tesis de licenciatura, Universidad Del Valle de Guatemala, 2019.

[7] F. J. 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, 2021.

[8] S. J. C. Lou, "Implementación y desarrollo de plataforma tipo gimbal para pruebas de control de drones de mediana potencia," Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.

[9] A. M. P. Echeverría, "Algoritmo de sincronización y control de sistemas de robots multi-agente para misiones de búsqueda," Tesis de licenciatura, Universidad Del Valle de Guatemala, 2019.

[10] "Optritrack for Robotics." dirección: www.optitrack.com.

[11] K. Ruohonen, *Graph theory*, ép. Lecture notes. 2013. dirección: http://gen.lib.rus.ec/book/index.php?md5=4c908924a3ae2a7170811eea4201729f.

[12] E. Hernández, "Sistema multi-cámara para la reconstrucción volumétrica de objetos 3D," Tesis de mtría., Universidad Zaragoza, 2019.

[13] J. Ávila, "Diseño e implementación de estrategias de consenso líder-seguidor para un sistema multi-agentes," Tesis de mtría., Universidad autónoma del estado de Hidalgo, 2019.

[14] H. Hamann, *Swarm Robotics: A Formal Approach*. Springer, 2018.