Buenas tardes, mi nombre es Julio Avila y hoy mostraré mis avances a la fecha de mi trabajo de graduación titulado “Adaptación del sistema de drones Crazyswarm al ecosistema Robotat”. La última vez había complido con dos de los objetivos, emplear las cámaras de captura de movimiento Optitrack para definir algoritmos de control y levantar la infraestructura de Crazyswarm para hacer pruebas de vuelo, los avances realizados a la fecha dan por concluido el último objetivo, crear algoritmos que conviertan la información de origen TCP/UDP a radiofrecuencia para establecer una comunicación directa con Crazyswarm.

Para cumplir con este objetivo, se creó un servidor en Python que funciona como intermediario entre ROS y otras computadoras. Luego de que la información de las cámaras es procesada en Motive, esta se transmite al Router de la red Robotat y el ordenador con ROS la recibe, esta comunicación ya se había establecido, el principal cambio realizado es que ahora ROS no corre una rutina para los Crazyflies, sino que ahora establece un servidor a través de un puerto en el router, este servidor recibe y lee los paquetes enviados por comunicación TCP. Gracias a este servidor, ahora cualquier computadora con Matlab u otro lenguaje de programación que permita la creación de clientes TCP, puede comunicarse con Crazyswarm.

El paquete de datos enviado al servidor muestra la siguiente estructura, un byte representa la fuente del paquete, otri byte corresponde al largo del paquete, este parámetro puede utilizarse de distintas formas dependiendo de la función a utilizar, puede tomarse como cantidad de Crazyflies a utilizar o el largo de la trayectoria. El siguiente byte corresponde al tipo de paquete, debido a que se espera que próximamente pueda hacerse una comunicación de amabas vías y pueda enviarse información en tiempo real, se apartaron los valores del 0 al 127 para enviar data y los valores del 128 al 255 para comandos. Finalmente, el payload corresponde a la concatenación de las trayectorias, las cuales se generan como 3 matrices de posiciones espaciales pero se convierten en vectores y se concatenan para una comunicación más eficiente. Esto último implica que el servidor debe recibir el vector y re ordenarlo y separarlo de nuevo en las matrices, para esto se aprovecha el valor del len.

Hasta ahora se han creado 6 funciones en Matlab para la comunicación con Crazyswarm, consisten en connect, que genera un objeto de tipo TCP, goto que envía a un crazyflie especificado a una posición indicada, traj que envía la cantidad de crazyflies deseados y las matrices de posicionamiento, test que confirma si la conexión del cliente aún está activa, land que aterriza todos los crazyflies indicados al mismo tiempo, y disconnect, que elimina el objeto tcp y corta la comunicación con el servidor.

Finalmente, creó un código en Matlab que genera 7 tipos de trayectorias distintas, este código se hizo lo más versátil posible, de modo que solo es necesario ingresar el tiempo que dure la trayectoria y el número de Crazyflies, y automáticamente genera la mejor trayectoria posible manteniendo simetría en los patrones para que sea estética la formación. El diseño de las trayectorias se basó en el movimiento armónico simple y en la parametrización vectorial de líneas espaciales como en las integrales de línea. En pantalla se ve un ejemplo de cómo se hizo esta parametrización, n corresponde a un vector de tiempo discretizado con un periodo de muestreo de 1 segundo, k es un número entero positivo que sirve para desfasar los movimientos de los drones y N mayúscula es la cantidad de crazyflies a utilizar. Mezclando este concepto con otras parametrizaciones, se alcanzaron formas como la que se muestra.