**Crazyflie**

Esta carpeta contiene todos los archivos correspondientes a las funciones desarrolladas para el control del dron Crazyflie. Las funciones están desarrolladas en Python, pero son utilizadas desde MATLAB a través de funciones específicas que llaman al código Python. Esta integración permite controlar el dron Crazyflie desde MATLAB mediante la biblioteca cflib de Crazyflie.

Cada archivo está debidamente comentado y posee una breve explicación de su funcionamiento. Es importante mencionar que, para hacer uso de las funciones de Crazyflie, es necesario tener instaladas las dependencias de software descritas a continuación y en el manual de usuario presente en la documentación del proyecto.

**Contenido de la carpeta**

**Archivos base**

* **crazyflie\_python\_commands.py**: Archivo en Python con funciones para interactuar con el Crazyflie mediante la biblioteca cflib.
* **\_\_init\_\_.py**: Archivo de inicialización para la carpeta de scripts en Python.

**Funciones de conexión y desconexión**

* **crazyflie\_connect.m**: Establece una conexión entre MATLAB y un Crazyflie específico.
* **crazyflie\_disconnect.m**: Finaliza la conexión activa con el dron Crazyflie indicado.

**Funciones de lectura y configruación**

* **crazyflie\_detect\_flow\_deck.m**: Verifica si la placa Flow Deck está conectada al Crazyflie.
* **crazyflie\_get\_pid\_values.m**\_ Obtiene los valores de ganancias PID para los ejes X, Y y Z.
* **crazyflie\_get\_pid\_x.m**: Obtiene los valores de ganancias PID para el eje X.
* **crazyflie\_get\_pid\_y.m**: Obtiene los valores de ganancias PID para el eje Y.
* **crazyflie\_get\_pid\_z.m**: Obtiene los valores de ganancias PID para el eje Z.
* **crazyflie\_get\_pose.m**: Obtiene la pose actual del Crazyflie (posición y orientación).
* **crazyflie\_set\_pid\_values.m**: Configura valores de ganancias PID para los ejes X, Y y Z.
* **crazyflie\_set\_pid\_x.m**: Configura los valores de ganancias PID para el eje X.
* **crazyflie\_set\_pid\_y.m**: Configura los valores de ganancias PID para el eje Y.
* **crazyflie\_set\_pid\_z.m**: Configura los valores de ganancias PID para el eje Z.
* **crazyflie\_set\_pose.m**: Establece una posición absoluta en el espacio.
* **crazyflie\_set\_position.m**: Establece una pose absoluta en el espacio (posición y orientación).

**Funciones de control de movimiento**

* **crazyflie\_takeoff.m**: Inicia el despegue del Crazyflie.
* **crazyflie\_land.m**: Inicia el aterrizaje del Crazyflie.
* **crazyflie\_move\_to\_position.m**: Mueve al Crazyflie a una posición relativa específica.
* **crazyflie\_goto\_robotat.m**: Función para mover al Crazyflie a una posición absoluta dentro del ecosistema Robotat.

**Archivos adicionales**

* **ejemplo\_de\_uso.m**: Archivo de MATLAB con un ejemplo simple de uso de las funciones.

**Instalación de dependencias**

Para el correcto funcionamiento del código es necesario instalar algunas dependencias de Python y controladores específicos. A continuación, se detallan las principales dependencias y su instalación:

* cflib: Biblioteca principal para interactuar con el Crazyflie.
  + Instalación en cmd: pip install cflib
* Bibliotecas logging, time, sys, threading.
  + Estas bibliotecas son estándar en Python y no requieren instalación adicional.
* Controlador de dispositivo Crazyradio.
  + Descargue el programa Zadig desde [este enlace](https://zadig.akeo.ie/).
  + Conecte el dispositivo Crazyradio a un puerto USB de su ordenador.
  + Abra Zadig y verá que automáticamente detecta el controlador del dispositivo Crazyradio.
  + Presione el botón Instalar controlador y espere a que finalice el proceso.

Asegúrate de revisar el manual de usuario en la carpeta de documentación para instrucciones adicionales y posibles problemas de compatibilidad.

**Descripción de archivos y funciones**

A continuación, se presenta la descripción de cada uno de los archivos anteriormente listados, una explicación de su funcionamiento y consideraciones adicionales:

**crazyflie\_python\_commands.py**

Este archivo proporciona una serie de funciones para interactuar con el dron Crazyflie utilizando la biblioteca **cflib** (Crazyflie Library) en Python. Las funciones permiten conectar y desconectar el dron, obtener y configurar su posición y orientación, así como realizar acciones básicas como despegue, aterrizaje y movimiento a una posición específica. Estas funciones no son utilizadas directamente con el Crazyflie, sino que son llamadas por medio de funciones en MATLAB que se presentarán más adelante.

**Dependencias**

* cflib: Biblioteca principal para interactuar con el Crazyflie.
* logging: Configuración de los niveles de log.
* time: Manejo de pausas y tiempos de espera.
* sys: Interfaz con el sistema para mostrar mensajes de error.
* threading.Event: Para sincronizar eventos y datos en tiempo real.

**Funciones principales**

1. **connect(uri)**

* **Descripción**: Establece una conexión con el Crazyflie mediante el URI especificado.
* **Parámetros**:
  + uri (str): URI del Crazyflie a conectarse.
* **Retorno**: Objeto SyncCrazyflie para interactuar con el dron.
* **Errores**: Muestra errores si ocurre algún problema en la conexión con el dron Crazyflie.

1. **disconnect(scf)**

* **Descripción**: Cierra la conexión activa con el Crazyflie.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
* **Retorno**: No retorna ningún valor.

1. **detect\_flow\_deck(scf)**

* **Descripción**: Verifica si el accesorio Flow Deck está conectado al Crazyflie.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
* **Retorno**: 1 si el Flow Deck está detectado, 0 si no.

1. **get\_pose(scf)**

* **Descripción**: Obtiene la posición (x, y, z) y orientación (roll, pitch, yaw) del Crazyflie.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
* **Retorno**: Lista con valores [x, y, z, roll, pitch, yaw].

1. **set\_position(scf, x, y, z)**

* **Descripción**: Establece una posición absoluta para el Crazyflie
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + x, y, z (float): Coordenadas absolutas en metros.
* **Retorno**: No retorna ningún valor.

1. **set\_pose(scf, x, y, z, qx, qy, qz, qw)**

* **Descripción**: Establece una pose absoluta (posición y orientación en cuaterniones) en el espacio.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + x, y, z (float): Coordenadas de posición.
  + qx, qy, qz, qw (float): Componentes del cuaternión de orientación.
* **Retorno**: No retorna ningún valor.

1. **get\_pid\_values(scf)**

* **Descripción**: Obtiene los valores PID para los ejes X, Y y Z del Crazyflie.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
* **Retorno**: Diccionario con valores PID para cada eje {X, Y, Z}.

1. **set\_pid\_values(scf** **, p\_gains, i\_gains, d\_gains)**

* **Descripción**: Establece los valores PID de los ejes X, Y y Z.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + p\_gains, i\_gains, d\_gains (dict): Diccionarios con valores P, I y D para cada eje.
* **Retorno**: No retorna ningún valor.

1. **get\_pid\_x(scf), get\_pid\_y(scf) y get\_pid\_z(scf)**

* **Descripción**: Obtiene los valores PID solo para el eje correspondiente (X, Y o Z).
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
* **Retorno**: Diccionario con los valores {P, I, D} para el eje correspondiente.

1. **set\_pid\_x(scf), set\_pid\_y(scf) y set\_pid\_z(scf)**

* **Descripción**: Establece los valores PID solo para el eje correspondiente (X, Y o Z).
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + P, I, D (float): Valores de las ganancias P, I y D.
* **Retorno**: No retorna ningún valor.

1. **takeoff(scf, height=0.3, duration=1.0)**

* **Descripción**: Inicia el despegue del Crazyflie hasta una altura especificada.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + height (float): Altura objetivo del despegue en metros (0.3 metros por defecto).
  + duration (float): Duración del despegue en segundos (1.0 segundo por defecto).
* **Retorno**: No retorna ningún valor.

1. **land(scf, height=0.0, duration=2.0)**

* **Descripción**: Inicia el aterrizaje del Crazyflie hasta una altura específica.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + height (float): Altura objetivo de aterrizaje en metros (0.0 metros por defecto).
  + duration (float): Duración del aterrizaje en segundos (2.0 segundos por defecto).
* **Retorno**: No retorna ningún valor.

1. **move\_to\_position(scf, x, y, z, velocity=1.0)**

* **Descripción**: Mueve al Crazyflie a una posición específica en el espacio con una velocidad dada.
* **Parámetros**:
  + scf (SyncCrazyflie): Objeto de conexión del Crazyflie.
  + x, y, z (float): Coordenadas objetivo de posición.
  + velocity (float): Velocidad de movimiento en metros por segundo (1.0 m/s por defecto).
* **Retorno**: No retorna ningún valor.

El conjunto de funciones anteriormente descritas conforma a las funciones para interactuar con el dron Crazyflie. Estas serán llamadas por medio de funciones en MATLAB para interactuar con el dron desde la interfaz de trabajo de MATLAB.

Es importante mencionar que el archivo **crazyflie\_python\_commands.py** siempre debe estar dentro de la misma carpeta que las funciones de MATLAB para que estas funcionen correctamente. En caso desee moverse a una ubicación distinta que la de las funciones de MATLAB, deberá modificar la función **crazyflie\_connect.m** en la que se declara la ubicación del archivo en Python con las funciones (deberá usar una ruta relativa).

**crazyflie\_connect.m**

Establece una conexión entre MATLAB y un dron Crazyflie específico, identificando el dron mediante un número de identificación.

* **Parámetros**:
  + drone\_number: Número entero entre 1 y 12 que identifica al dron. Este número es obligatorio y se utiliza para seleccionar la URI correcta del dron.
* **Retorno**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
* **Errores**:
  + Verifica que drone\_number sea un número entero entre 1 y 12. Si no es válido, lanza un error.
  + Si ocurre un error durante la conexión, se muestra un mensaje detallado, indicando el error exacto capturado.
* **Función en Python**:
  + Llama a la función *connect* del módulo **crazyflie\_python\_commands.**

**crazyflie\_disconnect.m**

Finaliza la conexión activa con el dron Crazyflie y limpia el objeto de conexión de la memoria en MATLAB.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión activa con el dron Crazyflie. Este parámetro es obligatorio.
* **Retorno**:
  + No devuelve ningún valor.
* **Errores**:
  + Si ocurre un error durante la desconexión, se muestra un mensaje con detalles específicos del problema.
* **Función en Python**:
  + Llama a la función *disconnect* del módulo **crazyflie\_python\_commands**.

**crazyflie\_detect\_flow\_deck.m**

Detecta si el accesorio Flow Deck está instalado en el Crazyflie.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Muestra errores de conexión o de detección del Flow Deck en caso de fallos.
* **Función en Python**:
  + Llama a la función *detect\_flow\_deck* del módulo **crazyflie\_python\_commands.**

**crazyflie\_get\_pid\_values.m**

Obtiene los valores de las ganancias PID (proporcional, integral y derivativa) para los tres ejes (X, Y, Z) del dron Crazyflie.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión activa con el dron Crazyflie. Este parámetro es obligatorio.
* **Retorno**:
  + pid\_values: Estructura con tres campos (X, Y, Z), cada uno de los cuales contiene un vector con los valores PID [Kp, Ki, Kd].
* **Errores**:
  + Si ocurre un error al obtener los valores PID, se lanza un mensaje con información detallada del problema.
* **Función en Python**:
  + Llama a la función *get\_pid\_values* del módulo **crazyflie\_python\_commands**.

**crazyflie\_get\_pid\_x.m, crazyflie\_get\_pid\_y.m, crazyflie\_get\_pid\_z.m**

Obtiene los valores PÏD específicos para el eje correspondiente (X, Y o Z) del dron Crazyflie.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
* **Retorno**:
  + pid\_x/y/z: Estructura con los valores PID para el eje correspondiente, que contiene los campos P, I, y D (proporcional, integral y derivativo).
* **Errores**:
  + Si ocurre un error al obtener los valores PID para el eje correspondiente, se lanza un mensaje detallado.
* **Función en Python**:
  + Llama a la función *get\_pid\_x/y/z* del módulo **crazyflie\_python\_commands**.

**crazyflie\_get\_pose.m**

Obtiene la pose actual del dron Crazyflie, incluyendo su posición y orientación.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
* **Retorno**:
  + pose: Vector con la pose actual del dron, que incluye las coordenadas de posición y orientación.
* **Errores**:
  + Si ocurre un error al obtener la pose, se lanza un mensaje de error con información específica del problema.
* **Función en Python**:
  + Llama a la función *get\_pose* del módulo **crazyflie\_python\_commands**.

**crazyflie\_set\_pid\_values.m**

Configura las ganancias PID (P, I, D) en los tres ejes del dron.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + p\_gains, i\_gains, d\_gains: Diccionarios o estructuras con las ganancias.
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Muestra un mensaje detallado si ocurre un error durante la configuración.
* **Función en Python**:
  + Llama a la función *set\_pid\_values* del módulo **crazyflie\_python\_commands**.

**crazyflie\_set\_pid\_x.m, crazyflie\_set\_pid\_y.m, crazyflie\_set\_pid\_z.m**

Configura las ganancias PID para un eje específico (X, Y o Z) del dron Crazyflie.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + P, I, D: Valores de las ganancias.
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Si ocurre un error durante la configuración, se muestra un mensaje detallado.
* **Función en Python**:
  + Llama a la función *set\_pid\_x/y/z* del módulo **crazyflie\_python\_commands**.

**crazyflie\_set\_pose.m**

Establece la pose absoluta del dron, especificando su posición y orientación.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + x, y, z: Coordenadas de posición.
  + qx, qy, qz, qw: Componentes del cuaternión de orientación.
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Cualquier error durante el establecimiento de la pose muestra un mensaje detallado.
* **Función en Python**:
  + Llama a la función *set\_pose* del módulo **crazyflie\_python\_commands**.

**crazyflie\_set\_position.m**

Configura la posición absoluta del dron en el espacio.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + x, y, z: Coordenadas objetivo.
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Muestra un mensaje detallado en caso de error.
* **Función en Python**:
  + Llama a la función *set\_position* del módulo **crazyflie\_python\_commands**.

**crazyflie\_takeoff.m**

Despega el dron Crazyflie hasta una altura en un tiempo dado.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + *height* (opcional): Altura deseada para el despegue (por defecto es 0.5).
  + *duration* (opcional): Duración del despegue (por defecto es 1.0).
* **Retorno**:
  + Ninguno.
* **Errores**:
  + *height* y *duration* se ajustan si no cumplen los valores mínimos.
  + Cualquier error durante el despegue genera un mensaje.
* **Función en Python**:
  + Llama a la función *takeoff* del módulo **crazyflie\_python\_commands**.

**crazyflie\_land.m**

Lleva al dron Crazyflie a una altura deseada para realizar el aterrizaje en un tiempo específico.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + *height* (opcional): Altura final del aterrizaje (por defecto es 0.0).
  + *duration* (opcional): Duración del aterrizaje en segundos (por defecto es 2.0).
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Si *height* o *duration* son valores no válidos, se ajustan a los valores predeterminados.
  + Cualquier error durante el aterrizaje genera un mensaje detallado.
* **Función en Python**:
  + Llama a la función *land* del módulo **crazyflie\_python\_commands**.

**crazyflie\_move\_to\_position.m**

Mueve el dron Crazyflie a una posición absoluta específica en el espacio con una velocidad dada.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + x, y, z: Coordenadas objetivo. Todos deben ser valores numéricos.
  + v: Velocidad de desplazamiento
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Si x, y, o z no son numéricos, lanza un error.
  + Cualquier error durante el movimiento muestra un mensaje detallado.
* **Función en Python**:
  + Llama a la función *move\_to\_position* del módulo **crazyflie\_python\_commands**.

**crazyflie\_goto\_robotat.m**

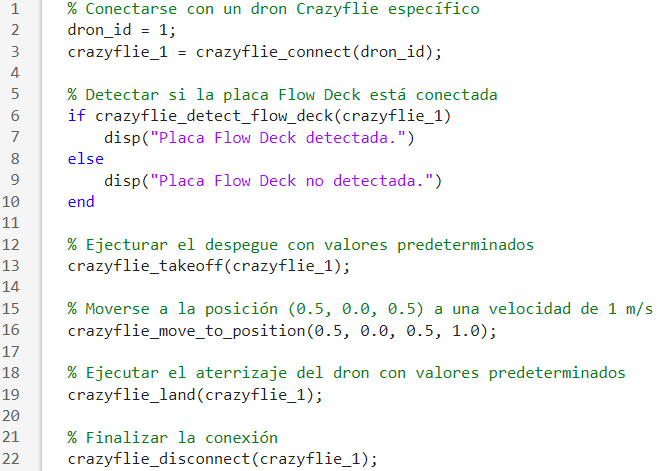
Mueve al dron Crazyflie a una posición absoluta dentro de la plataforma de pruebas del ecosistema Robotat con una velocidad especificada. Es importante que al utilizar esta función esté conectado a la red TCP/IP Robotat del laboratorio y que el dron Crazyflie tenga instalado el marcador reflectivo correspondiente (debe asegurarse que dicho marcador esté activo en el sistema de captura Robotat).

Además, debe tener en cuenta que el dron posee una posición absoluta cero por defecto (0, 0, 0), por lo que previo a volarlo y utilizar esta función debe actualizar la posición absoluta física del dron dentro de la plataforma Robotat (esta es distinta de cero). Es decir, debe obtener la posición actual del dron por medio de su marcador reflectivo y establecer esta como su posición absoluta mediante la función crazyflie\_set\_position.m.

* **Parámetros**:
  + scf: Objeto SyncCrazyflie que representa la conexión con el Crazyflie.
  + x, y, z: Coordenadas objetivo en el espacio 3D.
  + velocity: Velocidad de desplazamiento.
  + tcp\_obj: Objeto TCP utilizado para obtener la pose actual de la red Robotat.
  + agent\_id: Identificador del dron en el sistema de captura de movimiento Robotat.
* **Retorno**:
  + Ninguno.
* **Errores**:
  + Si ocurre un error al mover el dron, obtener la pose o actualizar la posición, se muestra un mensaje de advertencia.
* **Función en Python**:
  + Llama a las funciones *move\_to\_position* y *set\_position* del módulo **crazyflie\_python\_commands** para mover al dron y actualizar su posición absoluta durante el movimiento.

**Ejemplo de flujo de trabajo**

A continuación, un ejemplo básico para el uso de estas funciones:

****

**Figura 1.** Ejemplo simple de flujo de trabajo con funciones Crazyflie.

**Estado actual**

Esta carpeta está completa, y toda la funcionalidad ha sido testeada con la última versión de cflib. Se recomienda mantener actualizadas las dependencias y revisar las versiones de MATLAB y Python para garantizar la compatibilidad.