Trajectoires crazyswarm

Les trajectoires de Crazyswarm sont représentées par des polynômes définis par morceaux de degré 7, dans chacune des 3 directions ainsi que pour le "yaw" (lacet). La trajectoire complète est découpée en plusieurs étapes et on utilise exactement un polynôme par étape. Dans la suite, on ne se soucie pas du yaw (tous les coefficient sont mis à 0).

Constructeur de trajectoires

Il s'agit d'un programme python qui utilise les modules d'interpolation de **scipy**, et se lance avec la commande `python3 traj.py` :

- on choisit les points par lesquels on veut que la trajectoire passe un à un sur un graphe 2D; après chaque clic, on doit spécifier la hauteur du point. Une fois la trajectoire construite, on valide en quittant la fenêtre popup.
- un fichier `trajectoire.csv` est généré.
- un fichier `trajectoire.py` qui charge déjà la trajectoire avec **crazyswarm** est généré afin de la tester facilement.

Les premier et dernier points de la trajectoire sont toujours (0,0,0).

> **Note:** Chaque ligne du fichier `trajectoire.csv` représente un polynôme et donc une trajectoire. Toutes ces trajectoires durent un même temps T (3s par défaut).

Charger la trajectoire avec crazyswarm

Dans le script **crazyswarm**, il est possible d'allouer une trajectoire à un ou plusieurs drones en même temps. Pour cela, il est nécessaire de configurer les positions de départ de chaque drone, ce qui translate la trajectoire calculée, de sorte que tous les drones soient synchronisés à partir de leur point de départ.

Il faut donc modifier le fichier `ros_ws/src/crazyswarm/launch/crazyflies.yaml` de façon a initialiser les positions des différents drones.

> **Note:** La documentation plus précise (mais parfois lacunaire) de crazyswarm peut être trouvée ici: https://crazyswarm.readthedocs.io/en/latest/

Ensuite, la forme d'un script **crazyswarm** est la suivante:

```
swarm = Crazyswarm()
trajectory = uav_trajectory.Trajectory()
trajectory.loadcsv('trajectory.csv')
```

Pour assigner à un drone ou à un groupe de drones une trajectoire:

- ```
  single_cf = swarm.allcfs.crazyflies[0]
  single_cf.uploadTrajectory(id,0,trajectory)
  ```
- ```
  cf1 = swarm.allcfs.crazyflies[0]
  cf2 = swarm.allcfs.crazyflies[1]
  cf1.uploadTrajectory(id,0,trajectory)
  cf2.uploadTrajectory(id,0,trajectory)
  cf1.setGroupMask(1)
  cf2.setGroupMask(1)
  ```

Enfin:

- `cf1.startTrajectory(id)`
- `swarm.allcfs.startTrajectory(id, groupMask=1)`

> **Note:** il est nécéssaire de faire suivre chaque lancement de trajectoire par un `swarm.timeHelper.sleep(T_trajectory)` pour que le script attende la fin des trajectoires avant de continuer.

Lancer le vol (simulation & réel)

En supposant que le script **crazyswarm** ait été enregistré dans `script.py` :

- `python3 script.py --sim` pour la simulation
- `python3 script.py` pour un vol réel

> **Note:** La documentation **crazyswarm** assure que si un vol est fructueux en simulation, alors il le sera aussi en réel.