



PROJET D'INGÉNIEUR EN ÉQUIPE

SPECTACLE DRONAUTIQUE

Document Technique

Équipe 1 :
OKABAYASHI Aimi
POTEL Pierre
RAMAMBASON Jeanne
LAMPE Quentin
PRISER Victor
LOMBARD Agathe
BENNATAN Adrien
FREON Dimitri

Mai 2021

Table des matières

I	Introduction	1
II	Choix des technologies	1
II.1	Le Bitcraze Crazyflie	1
II.2	Optitrack	1
II.3	Les différents softwares	2
III	Installation des softwares	2
III.1	Installation de ROS	2
III.2	Installation de Crazyswarm	3
IV	Configuration des softwares	4
IV.1	Configuration des drones	5
IV.2	Configuration des radios	5
IV.3	Configuration des coordonnées des drones	6
IV.4	Configuration du système de motion tracking	6
V	Inventaire	7

I Introduction

L'objectif de notre PIE était de faire un spectacle de dronautique à l'aide de plusieurs drones qui volaient de manière synchronisée. Ce document résume les différents choix techniques que nous avons fait et détaille ce que nous avons réalisé durant notre PIE en 2020/2021.

II Choix des technologies

Pour ce projet nous avons plusieurs choix pour ce qui est des drones et des technologies de capture de positions. Concernant les drones, nous avons hésité entre les DJI Tello EDU et les Bitcraze Crazyflie. Nous avons finalement opté pour les Bitcraze Crazyflie car l'U2IS en possédait déjà deux au moment du début de notre projet. Pour le système de capture des positions nous avons opté pour le système Optitrack, déjà présent à l'U2IS également.

II.1 Le Bitcraze Crazyflie



Le Bitcraze Crazyflie est un quadcoptère créé par l'entreprise Bitcraze. Nous utilisons dans notre PIE la version 2.1 de ces drones. Ils sont parfaits pour notre utilisation car ils sont petits et légers, 10 cm par 10 cm et 27 g. Les drones possèdent une plateforme de développement de vol open-source et peuvent être reliés à un ordinateur *via* Bluetooth ou WIFI. Ils possèdent également une charge utile de 15g, ce qui nous sera utile pour disposer sur les drones les capteurs nécessaires à la détection de leurs positions *via* Optitrack. La durée d'une charge n'est pas très longue, environ 7 minutes, mais sera suffisante pour effectuer un petit spectacle. Ce drone peut également être commandé *via* un téléphone portable, grâce à l'application Crazyflie 2.0, disponible sur Android et sur IOS. Pour notre PIE nous avions à notre disposition 2 drones et 4

autres drones devraient arriver après la fin du projet.

II.2 Optitrack

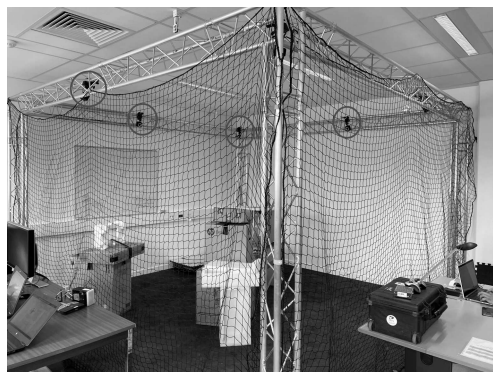


FIGURE 1 – La cage muni d'Optitrack située à l'U2IS

Le système Optitrack est un système de capture de positions de drones grâce à des caméras. Ce système se décompose en plusieurs caméras haute fréquence toutes disposées autour d'une cage et qui permettent de faire du motion tracking. Le laboratoire de l'ENSTA l'U2IS possède déjà une cage d'environ 16 m² équipée de 6 caméras Optitrack toutes reliées à un ordinateur central comme le montre la photo 1. C'est ce dispositif que nous avons utilisé pour localiser les drones dans l'espace. Pour cela, nous avons attaché grâce à de la patafixe des petites boules sur nos drones, et ce sont ces petites boules qui seront détectées par le système et dont nous aurons les coordonnées. Le système est relié à un ordinateur de l'U2IS qui possède déjà le logiciel Motive, qui est le logiciel développé par Optitrack permettant de faire du motion tracking. C'est grâce à ce logiciel

là que nous pourrions localiser les drones dans l'espace. Il est nécessaire d'avoir ce système car

les drones ne sont pas capables de se localiser par eux-mêmes, il faut donc récupérer leurs positions respectives grâce à un système à part pour leur transmettre dans un second temps et pourvoir correctement les diriger.

II.3 Les différents softwares

Pour notre PIE nous avons eu besoin d'installer et de configurer deux softwares principaux : ROS et Crazyswarm.

II.3.1 ROS

ROS est un ensemble d'outils informatiques sous formes de logiciels libres open source, permettant de développer des logiciels pour la robotique. C'est cet ensemble d'outils qui va nous permettre de contrôler les drones.

II.3.2 Crazyswarm

Crazyswarm est un ensemble de logiciel développés par des chercheurs de l'Université de Californie du Sud et qui permet de contrôler un essaim de drones pour réaliser un spectacle par exemple. C'est donc exactement la plateforme qu'il nous faut. Crazyswarm nécessite un système de motion capture tel que Optitrack pour fonctionner. Le langage utilisé est principalement Python, avec un peu de C/C++. Le tout est basé sur les outils ROS, qui est donc nécessaire pour que le tout fonctionne. Crazyswarm permet de faire le lien entre le système de motion capture et les drones, de tracker plusieurs drones à la fois et de les commander indépendamment les uns des autres mais aussi de faire des simulations de trajectoires. On trouvera plus de détails sur ce logiciel dans le papier des auteurs.

III Installation des softwares

L'ensemble des logiciels à installer fonctionnent sur Ubuntu, nous vous conseillons donc de travailler exclusivement sur Ubuntu. En effet, même si vous disposez de Windows Subsystem for Linux, Crazyswarm ne marche pas dessus. Nous avons rencontré beaucoup de difficultés à travailler sur les machines virtuelles, (notamment à communiquer avec les ports USB *via* le terminal) le mieux est donc d'avoir un ordinateur avec Ubuntu d'installer nativement.

L'installation suit le tutoriel des concepteurs de Crazyswarm, avec quelques ajouts là où nous avons rencontrés des difficultés. Comme nous voulons utiliser des drones en vrai et pas juste faire des simulations, on se place dans le cadre "Physical Robots and Simulations" du tutoriel.

III.1 Installation de ROS

La première étape est d'installer ROS. Pour cela, on se rend sur le site de ROS pour suivre leur tutoriel. La première chose importante à faire est de télécharger et d'installer la version de ROS qui correspond à la version d'Ubuntu de votre machine. Nous vous conseillons d'avoir la version 20.04.x d'Ubuntu, c'est cette dernière que nous avons utilisée. La version de ROS correspondante est ROS Noetic. Le reste est indiqué sur le tutoriel correspondant à cette version, on commence par configurer l'ordinateur pour accepter le software de packages.ros.org :

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc)
main" > /etc/apt/sources.list.d/ros-latest.list'
```

On met en place les clés nécessaires à l'installation :

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key  
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

Il faut ensuite être sûr que les packages natifs à votre ordinateur sont bien un jour, pour cela il suffit d'exécuter la commande

```
sudo apt update
```

On installe ensuite la version Noetic complète de ROS :

```
sudo apt install ros-noetic-desktop-full
```

Normalement, à ce moment-là, ROS est installé sur votre machine. Vous pouvez vous en assurer en exécutant la commande suivante :

```
source /opt/ros/noetic/setup.bash
```

Cette commande est très importante et il sera nécessaire de l'exécuter avant d'exécuter un programme ROS sur votre terminal. Autrement dit, dès que vous allez utiliser ROS ou Crazyswarm sur un terminal, il faudra l'exécuter avant. Nous vous conseillons de l'exécuter dès que vous arrivez en PIE, comme ça vous êtes sûr que votre terminal est bien configuré pour marcher avec ROS. Vous pouvez aussi automatiser l'exécution de cette commande en la mettant dans le fichier `~/.bashrc` et en tapant ensuite dans un terminal :

```
source ~/.bashrc
```

ROS sera alors automatiquement disponible dès que vous lancez un nouveau terminal.

Les packages de ROS ne sont pas obligatoires, mais vous pouvez toujours les installer pour être sûr que tout fonctionne bien. Vous pouvez vérifier que vous avez bien la bonne version de ROS en exécutant

```
rosversion -d
```

Qui devrait vous renvoyer noetic. Vous avez maintenant installé ROS, il faut ensuite installer Crazyswarm.

III.2 Installation de Crazyswarm

La première étape est de configurer la variable d'environnement `$CSW_PYTHON` avec notre version de Python. Normalement, à l'ENSTA, on travaille avec Python 3 mais vérifiez que vous l'avez bien d'installer sur votre machine, par exemple en tapant la commande

```
python
```

Qui devrait vous lancer un environnement de travail Python en vous indiquant la version de Python installée sur votre machine. Vous pouvez le quitter avec `exit()`. Après avoir vérifié que python est bien installé, tapez la commande :

```
export CSW_PYTHON=[python2 ou python3]
```

En remplaçant les crochets par la version de Python que vous avez. Vous pouvez maintenant installer les dépendances grâce à la commande :

```
sudo apt install git swig lib${CSW_PYTHON}-dev ${CSW_PYTHON}-numpy  
${CSW_PYTHON}-yaml ${CSW_PYTHON}-matplotlib gcc-arm-embedded libpcl-dev  
libusb-1.0-0-dev sdcc ros-noetic-vrpn
```

Il est possible que cette commande ne marche pas, par exemple que vous ayez des erreurs avec `gcc-arm-embedded`. Dans ce cas, vous pouvez essayer de l'installer en cherchant sur internet ou alors installer à la place `arm-none-eabi-gcc` et le remplacer dans la commande. Ensuite, cette commande fait en sorte d'installer différents packages Python : `dev`, `numpy` et `matplotlib`. Donc, si ces différents packages sont déjà installés sur votre machine, nous pouvez les enlever de la commande. Sinon, vous pouvez les installer indépendamment grâce à `pip`. C'est une des étapes qui nous a causé le plus de soucis et à la fin nous avons simplement exécuté :

```
sudo apt install git swig arm-none-eabi-gcc libpcl-dev libusb-1.0-0-dev sdcc  
ros-noetic-vrpn
```

Un fois cette étape réussie, il faut cloner le répertoire git de CrazySwarm :

```
git clone https://github.com/USC-ACTLab/crazyswarm.git
```

Vous pouvez ensuite vous placez dans le répertoire qui s'appelle `crazyswarm` et qui vient d'être créé et build le script grâce aux commandes :

```
cd crazyswarm
```

puis

```
./build.sh
```

Normalement à cette étape-là vous avez installé CrazySwarm. Vous pouvez le vérifier en allant dans le dossier scripts :

```
cd ros_ws/src/crazyswarm/scripts
```

Et en lançant, non pas le script `pytest` comme dit dans le tutoriel car il n'existe pas mais par exemple le script `graphVisualization.py` :

```
python -m graphVisualization.py --launch
```

Si vous avez une simulation qui se lance, alors vous avez installé ROS et CrazySwarm correctement !

IV Configuration des logiciels

Maintenant que vous avez installé ROS et CrazySwarm, vous devez configurer les Crazyflies ainsi que les Crazyradios. C'est une étape assez longue et compliquée. Nous avons déjà effectué une partie de la configuration, vous n'avez peut-être pas à tout refaire. Il sera par contre indispensable de le faire si les 4 nouveaux drones sont arrivés quand vous reprenez le projet.

IV.1 Configuration des drones

Nous avons configuré les deux Crazyflies qui ont une étiquette avec les chiffres “1” et “2”. Les deux adresses de ces drones sont respectivement :

1. 0xE7E7E7E701
2. 0xE7E7E7E702

Pour les configurer il faut utiliser le client Python Crazyflies en reliant les drones grâce à un câble usb. Vous ouvrez ce client en tapant `cfclient` dans le terminal une fois que Crazyswarm est installé. Vous pouvez alors relier le drone *via* un câble usb, cliquer sur **Connect**. Une fois les drones connecter, allez dans le menu **Connect** puis **Configure 2.x** et vous pourrez changer les adresses des drones avec leur adresse respective.

Une fois les adresses configurées, il faut mettre à jour les firmwares des drones. Pour cela il faut d’une part brancher et connecter les radios au client Python Crazyflies et avoir les drones à portée de main. On suit ensuite la configuration pour upload le firmware précompilé :

1. Branchez la batterie au drone.
2. Allumez le Crazyflie grâce au bouton d’allumage.
3. Mettez le drone en mode bootloader en appuyant sur la bouton d’allumage pendant 3 secondes. Les led doivent se mettre à clignoter en bleu.
4. Exécuter la commande :

```
roslaunch crazyflie_tools flash --target nrf51 --filename prebuilt/cf2_nrf.bin
```

Si cette commande ne marche pas, vérifiez que vous avez bien exécuté la commande

```
source /opt/ros/noetic/setup.bash
```

 dans votre terminal comme indiqué à la partie III.1.

5. Éteignez et rallumez le drone en le mettant en mode bootloader.
6. Enfin, exécutez la commande

```
roslaunch crazyflie_tools flash --target stm32 --filename prebuilt/cf2.bin
```

Normalement, à ce moment là, les drones sont bien à jour.

IV.2 Configuration des radios

Pour ce qui est des radios, chacune des deux peut contrôler environ une quinzaine de drones. Les deux antennes que nous avons sont donc suffisante pour notre PIE. nous avons mis à jour les firmware des deux radios dont nous disposons comme détaillé dans le `README.md` du fichier `~/crazyswarm/crazyradio-firmware`. Nous avons rencontré un problème majeur lors d’une des étapes de la mise à jour : les codes python n’arrivaient pas à trouver le module `usb`. Nous avons tenté de les installer *via* un

```
pip install usb
```

Mais nous avions toujours l’erreur. Voilà comment nous l’avons comprise : le code python qui est fourni avec Crazyswarm est en Python 2.7 or quand nous installons le module `usb` *via* `pip`, nous obtenions la version du module pour Python 3. Nous avons juste copié le fichier du module `usb` du dossier `/usr/local/lib/python3.x/dist-packages` vers le dossier `/usr/local/lib/python2.7/dist-packages`. Le script pour configurer les antennes s’est alors mis à marcher.

Il faut maintenant configurer votre machine pour utiliser les radio USB sans être root. Pour cela vous pouvez utiliser le script :

```
./pc_permissions.sh
```

IV.3 Configuration des coordonnées des drones

Nous avons opté pour un arrangement avec plusieurs marqueurs pour repérer les drones dans l'espace. Vous trouverez une photo des positionnement des marqueurs sur la photo 2. Une fois les capteurs fixés sur les drones grâce à de la patafix, il faut lister tous les crazyflies dans le fichier `~/crazyswarm/ros_ws/src/crazyswarm/launch/crazyflies.yaml`. Ce fichier doit détailler les différents drones avec leur identifiant, leur channel et leur position initiale. Deux drones ne doivent pas avoir deux positions initiale identiques. L'identifiant correspond aux derniers chiffres de l'adresse des drones.

```
crazyflies:
- id: 1
  channel: 100
  initialPosition: [1.5, 1.5, 0.0]
  type: default
- id: 2
  channel: 110
  initialPosition: [1.5, 1.0, 0.0]
  type: medium
```



FIGURE 2 – Positionnement des marqueurs sur les drones.

IV.4 Configuration du système de motion tracking

L'objectif maintenant est de configurer le système de motion tracking Optitrack *via* le logiciel Motive. Le logiciel et Optitrack est déjà relié à l'ordinateur présent à l'U2IS, l'objectif principal est donc de relier notre machine personnelle à l'ordinateur de l'U2IS pour relier Optitrack et Crazyswarm. Nous n'avons pas réussi à relier ces deux composants. Nous avons commencé par configurer les différents fichiers de Crazyswarm afin de fonctionner avec Optitrack, car Crazyswarm peut fonctionner avec différents systèmes tels que Vicon, Qualisys, Optitrack. Pour cela, on met l'adresse IP d'Optitrack qu'on trouve sur la logiciel Motive dans l'onglet View puis Streaming Pannel dans le fichier `ros_ws/src/crazyswarm/launch/hover_swarm.launch`.


```
# tracking
  motion_capture_type: "optitrack" # one of none,vicon,optitrack,qualisys,vrpn
  object_tracking_type: "libobjecttracker" # one of motionCapture,libobjecttracker
  send_position_only: False # set to False to send position+orientation; set to True to send
  # vicon_host_name: "vicon"
  optitrack_local_ip: "127.0.0.1" # only needed if optitrack is selected
  optitrack_server_ip: "147.250.35.30" # only needed if optitrack is selected
```

Nous avons aussi sélectionné le bon système de capture à la ligne 10 de `ros_ws/src/crazyswarm/launch/mocap_helper`. Il faut aussi être connecté au réseau Ethernet de l'ENSTA pour que cela fonctionne. Normalement, après avoir fait ça, nous étions censé pouvoir nous connecter à Optitrack *via* la commande :

```
roslaunch mocap_helper.launch
```

Encore une fois, il faut bien avoir pensé à exécuter la commande `source ros_ws/devel/setup.bash` avant le `roslaunch`. Cependant, lorsque nous exécutons cette commande, nous avons l'erreur suivante :

```
ERROR: cannot launch node of type [crazyswarm/mocap_helper]: crazyswarm
ROS path [0]=/opt/ros/noetic/share/ros
ROS path [1]=/opt/ros/noetic/share
```

Nous avons essayé de chercher partout d'où pouvait venir cette erreur sans succès. Nous avons alors posé une question sur la github de CrazySwarm mais au moment de la fin du PIE, nous n'avons pas eu de réponse qui résolvait notre problème. Vous pouvez trouver cette question [ici](#).

V Inventaire

- 2 Drones Crazyflie 2.0 (2 équipés d'une led bleu et de 4 capteurs) + accessoires
 - Contenu d'une boîte :
 - 1 carte de contrôle Crazyflie 2.0
 - 1 batterie LiPo 240mAh
 - Moteur CC sans noyau 5 x 7 mm
 - 6 supports de moteur de 7 mm
 - 1 tampon de batterie en caoutchouc
 - 5 hélices CCW
 - 5 hélices CW
 - 1 plateau de support de batterie
 - 2 connecteurs mâles courts
 - 2 connecteurs mâles longs
- 2 Drones DJI Tello
- 4 Clés Wifi TP-Link