

# Compte rendu : Voice-mission

## I- Définition de la mission :

La mission voice-mission est un script Python qui permet à l'utilisateur de contrôler un drone via des commandes vocales simples. Grâce à la bibliothèque `speech_recognition`, l'utilisateur peut donner des instructions telles que "avancer de 5 mètres", "reculer", ou "atterrir". Le script traduit la commande vocale en actions et envoie les commandes correspondantes au drone en utilisant l'API `olymp`. Ce système facilite l'interaction avec le drone en rendant le contrôle intuitif et mains libres. cette mission peut être utiliser dans les applications suivantes :

- 1- Mission de recherche et sauvetage
- 2- Photographie aérienne et prise de vues
- 3- Surveillance en temps réel

## II- Les entrées et la sortie de la mission :

**1- Les entrées :** Pour que cette mission fonctionne, vous avez uniquement besoin de l'ID du drone en tant qu'entrée.

**2- Les sorties :** aucun sortie

## III- Explication du code de la mission :

Le code présenté permet de commander un drone à l'aide de commandes vocales simples, grâce à la bibliothèque `speech_recognition` pour capter et convertir la parole en texte. Lorsqu'une commande est donnée, elle est analysée à l'aide d'expressions régulières pour en extraire l'action (comme avancer, reculer, tourner, etc.) et les paramètres associés, comme la distance ou l'angle. Une fois la commande traitée, le drone exécute l'action correspondante via la bibliothèque `olymp`, en utilisant des commandes comme `moveBy` ou `Landing`. En parallèle, l'interface graphique du programme change de couleur pour indiquer l'action en cours, ce qui aide l'utilisateur à suivre l'état de la mission. Un flux vidéo en temps réel est également ouvert pour visualiser ce que voit le drone pendant l'exécution des commandes. Ce système, qui utilise des threads pour gérer les commandes en parallèle, offre ainsi une manière interactive et fluide de piloter un drone avec la voix tout en affichant des informations visuelles pertinentes.

## IV- Explication des fonctions du script :

- **`check_gps_fix(self)`** : Cette fonction vérifie si le GPS du drone est correctement fixé. Si le GPS est valide (c'est-à-dire qu'il ne renvoie pas des valeurs par défaut comme 500.0), elle enregistre les coordonnées géographiques de la position de départ du drone (latitude et longitude) en tant que "position home".

- **return\_to\_home(self)** : Cette fonction permet au drone de revenir à sa position d'origine (home) en utilisant la commande return\_to\_home(). Elle vérifie en continu que le drone atteint les coordonnées de la position d'origine avant de procéder à l'atterrissage.
- **start\_video\_stream()** : Ouvre un flux vidéo en temps réel à partir du drone et l'affiche sur une fenêtre à l'écran à l'aide de OpenCV. La vidéo est redimensionnée pour s'adapter à une taille plus petite tout en préservant la qualité de l'image. La fenêtre se ferme lorsque l'utilisateur appuie sur la touche 'q'.
- **extract\_command\_data(command)** : Cette fonction utilise des expressions régulières pour analyser la commande vocale et en extraire l'action et la valeur associée. Par exemple, si la commande est "avancer de 5 mètres", elle retournera l'action "avancer" et la valeur 5.
- **execute\_command(command)** : Après avoir extrait les informations de la commande vocale, cette fonction exécute l'action correspondante via le drone. Selon l'action détectée (avancer, reculer, tourner, etc.), elle appelle les commandes appropriées comme moveBy, rotation, ou Landing.
- **listen\_for\_commands()** : C'est la fonction principale qui écoute en continu les commandes vocales via le microphone. Lorsqu'une commande est détectée, elle est convertie en texte et envoyée à la fonction execute\_command() pour être traitée.