

Projet double :
Asservissement "rigide" de drone

Joanne STEINER, Hugo LEVY-FALK

2019

Résumé

Table des matières

1	Introduction	2
2	Déroulement du projet	3
3	Modélisation du problème	4

Chapitre 1

Introduction

Le campus de Metz de CentraleSupélec dispose de drones appelés les quadricoptères Bebop 2 (Parrot). Ces derniers sont capables de réaliser des mouvements brusques ce qui rend leur pilotage complexe. Il faut donc les manipuler avec précautions, ce qui revient à sous-exploiter leurs capacités.

Jusqu'à présent, le drone était asservi à l'aide d'une cible présente dans son champ visuel. Le drone suivait la cible (bleu) et se positionnait en face de cette dernière. Toutefois, le drone se déplaçait lentement et se montrait prudent. De plus, une fois face à la cible, le drone n'était pas stable. Il oscillait verticalement face à la cible. Ce résultat avait été obtenu suite à un projet d'élèves.

L'objectif de ce projet est donc de rendre l'asservissement du drone plus "rigide" et donc d'améliorer et de mieux exploiter l'utilisation de ce dernier.

Chapitre 2

Déroulement du projet

Le projet comporte les étapes suivantes :

- Etape 1 : Prise en main de ROS et du drone présent à la smartroom ;
- Etape 2 : Réduction du bruit causé par le calcul de la position du drone par rapport à l'image
- Etape 3 : Choix du correcteur simple boucle
- Etape 4 : Choix du correcteur double boucle
- Etape 5 : Ecriture d'un guide de fonctionnement

Chapitre 3

Modélisation du problème

Le drone diffuse un flux optique, qui permet de calculer sa position vis-à-vis d'une cible. On peut donc dans un premier temps modéliser l'asservissement de cette manière :

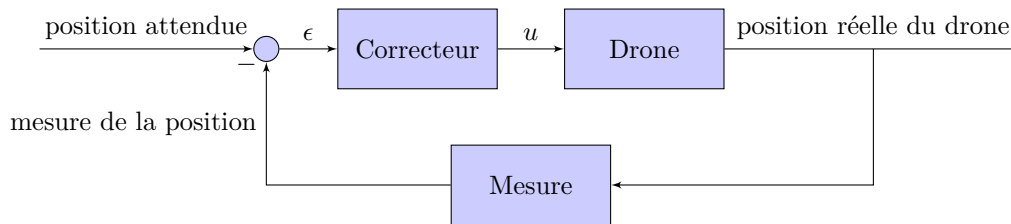


FIGURE 3.1 – Modélisation simple de l'asservissement

La mesure peut se décomposer en trois éléments :

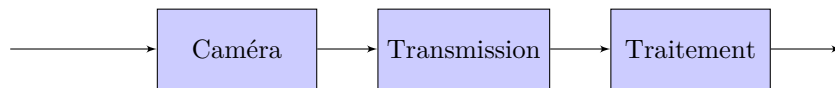


FIGURE 3.2 – Décomposition de la mesure

Comme le traitement ne se fait pas directement sur le drone, il existe des retards aléatoires dus au temps de transit sur le réseau, ainsi qu'au temps de calcul sur le poste de contrôle. Ces retards limitent la fréquence des mesures à environ 5Hz.

Chapitre 4

Réalisation du projet

4.1 Prise en main de ROS et du drone présent à la smartroom

Pour la prise en main de ROS, nous avons suivi les étapes du tutoriel de M. Frezza-Buet. Cela nous a permis de comprendre comment fonctionne ROS et de pouvoir l'utiliser de manière basique et simple.

4.2 Réduction du bruit causé par le calcul de la position du drone par rapport à l'image

Nous avons constaté qu'une forte incertitude existait lors du calcul de la position du drone par rapport au panneau. En effet, celle-ci, même si le drone était immobile, bougeait beaucoup et faisait des "sauts". Il s'agissait d'un bruit de type de Poisson, difficile, voire impossible, à traiter en automatique pour obtenir un asservissement satisfaisant. En effet, le drone, avec ce bruit, aurait été asservi à ce dernier, rendant impossible l'obtention d'une position fixe. Notre drone aurait fait de grands écarts par rapport à la position attendue.

Afin de régler ce problème, l'idée proposée a été de modifier l'algorithme en place. Ce dernier calcul la position des trois carrés bleus dans l'image renvoyée par le drone par les position H, R et L. Cette détermination se faisait par gngt, qui faisait beaucoup de bruit. Nous avons donc utilisé un nouveau script Python qui profite de la labellisation afin de déterminer la position des panneaux.

Dans cette partie, nous ne nous intéresserons qu'au script donné dans le fichier `src\detect_targets\script\find_targets.py`. Il permet, à partir d'un seuil RGB défini, de trouver les parties "bleues" de l'image. Une fois ces positions trouvées, on ne garde que les trois plus grandes parties et on en calcule le centre de masse. Les résultats obtenus sont plutôt satisfaisants.

4.3 Premier asservissement sommaire

Le premier asservissement qui a été réalisé n'utilisait qu'une seule boucle et asservissait la position du drone. Pour ce premier asservissement sommaire, nous avons estimé la courbe de la réponse fréquentielle de notre drone afin d'en déduire le type de correcteur à utiliser.

Nous savons que selon l'axe z, le drone répond seulement en vitesse. On peut approximer sa réponse fréquence par une pente nulle en basse fréquence, puis une pente [-1] en haute fréquence. Afin d'asservir le drone, nous aurons donc besoin d'un correcteur de type proportionnel-intégrale

$$C(p) = K_p + \frac{K_i}{p}$$

Concernant les axes x et y, nous savons que le drone ne peut être correctement asservi en vitesse. En effet, le réglage de la vitesse et de l'accélération du drone se fait selon un angle d'inclinaison : le drone accélère en permanence s'il conserve le même angle ce qui ne permet pas le maintien d'une vitesse constante. La réponse fréquentielle selon ses axes est donc représentée par une pente de [-1] dans les basses fréquence et une pente de [-2] dans les hautes fréquences. Afin de corriger ces axes, nous allons donc utiliser une correction de type Proportionnelle-Dérivée :

$$C(p) = K_p + K_d p$$

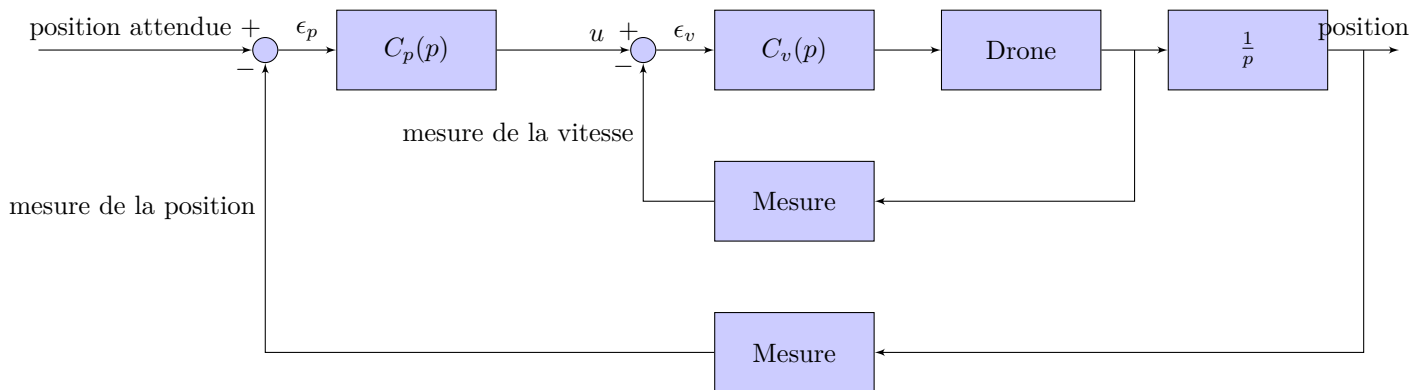


FIGURE 4.1 – Modélisation simple de l'asservissement

4.4 Deuxième asservissement utilisant une double-boucle d'asservissement

4.5 Installation du module et utilisation de l'interface graphique

Modules pour régler les seuils de couleur (on pourrait songer à utiliser des cibles rouges, à condition de modifier les seuils) et le correcteur : la valeur de la saturation en vitesse et en accélération peuvent se régler ainsi que les différentes valeurs du PID (il est toutefois déconseillé de modifier les valeurs, celles-ci ayant été adaptés au problème).