TP techdays

Alexis OFFERMANN

May 27, 2024

1 Plan du TP

1. Simulation
   - Explication du main
   - Lancement du fichier UAV_SIMULATOR_MAIN_DMR2_5.m
   - Explication du modèle Simulink
   - Test du mode classique position de px4
   - Test du mode attitude : Réglage du PID, voir Section 3
   - On prend les valeurs suivantes ...
   - Affichage des valeurs et tracés de courbes
   - On peut passer aux expérimentations
2. Expérimentations
   - Mise en charge des batteries
   - configuration du bridge esp32
   - Configuration de la motion capture
   - Affichage du mode de vol dans Qground control

2 Introduction

Avant de commencer on s'assure d'avoir téléchargé le fichiers à l'adresse suivante `Techdays_gipsa-lab-matlab-simulink-uav.zip` ici :
`https://filesender.renater.fr/?s=download&token=2ae06c1d-6b36-47a0-b2cc-a92a87627a7c`

On ouvre ensuite le fichier UAV_SIMULATOR_MAIN_DMR2_5.m. Tous les paramètres du drone y figurent ainsi que nombres de réglages pour la commande. Vous pouvez parcourir le fichier pour comprendre les différents paramètres.

On exécute le fichier une première fois pour rajouter les constantes dans le workspace.

Ensuite on ouvre le fichier : PX4_UAV_SIMULATOR.slx

Explication des différents blocs.

On peut lancer la simulation directement en appuyant sur le bouton play en haut de l'écran.
Le mode par défaut utilise la loi de commande en position de px4[1].

Pour afficher les différentes courbes de la simulation on ouvrira le fichier UAV_SIMULATOR_PLOTS.m
On l'exécute pour afficher les différentes courbes

[1] `https://docs.px4.io/main/en/flight_stack/controller_diagrams.html`

Dans ce TP on propose de changer cette loi de commande pour un simple PID. Le but étant de montrer la facilité avec laquelle on peut changer de loi de commande et donc tester différents algorithmes.

Pour contrôler le drone en position à l'aide de commande en attitude/thrust, nous allons utiliser un simple contrôleur linéaire présenté en section suivante.

3 Contrôle du drone par des commandes en attitude/thrust

On propose un algorithme de commande simple qui consiste à transformer la consigne de position du repère inertiel vers le repère du drone, avant d'appliquer un contrôle type PID. La sortie de commande en X et Y est appliquée directement sur les entrées de tangage et roulis respectivement du drone. La commande en Z est appliquée sur la poussée (thrust) du drone. Le lacet est directement réglé dans les boucles internes du drone.

- On note $X_{uav} = [x, y, z, v_x, v_y, v_z, \Psi]^T$ l'état du drone et $X_{ref} = [x_r, y_r, z_r, 0, 0, 0, \Psi_r]^T$ la position et l'orientation de référence.
- La commande linéaire en attitude + thrust est calculée de la manière suivante :

$$\begin{aligned}
\Phi_d &= \sin(\Psi) \cdot \left[k_p \cdot (x_{ref} - x) - k_d \cdot v_x + k_i \cdot \int (x_{ref} - x)\right] && (1)\\
&\quad - \cos(\Psi) \cdot \left[k_p \cdot (y_{ref} - y) - k_d \cdot v_y + k_i \cdot \int (y_{ref} - y)\right] && (2)\\
\Theta_d &= \cos(\Psi) \cdot \left[k_p \cdot (x_{ref} - x) - k_d \cdot v_x + k_i \cdot \int (x_{ref} - x)\right] && (3)\\
&\quad + \sin(\Psi) \cdot \left[k_p \cdot (y_{ref} - y) - k_d \cdot v_y + k_i \cdot \int (y_{ref} - y)\right] && (4)\\
\Psi_d &= \Psi_{ref} && (5)\\
T_d &= T_{hover} + (k_{tp} * (z_ref - z) + k_{td} * v_z + k_{ti} \cdot \int (z_{ref} - z)) && (6)
\end{aligned}$$

  Où k_p, k_d, k_{tp} et k_{td} sont des gains a paramétrer. T_{hover} correspond à la valeur de thrust qui compense uniquement la gravité, pour le drone simulé, sa valeur est de 0.52.

- Il faut ensuite saturer ces valeurs :

$$\begin{aligned}
\Phi'_d &= saturate(\Phi_d, -0.5, 0.5) && (7)\\
\Theta'_d &= saturate(\Theta_d, -0.5, 0.5) && (8)\\
\Psi'_d &= \Psi_d && (9)\\
T'_d &= saturate(T_d, 0, 1) && (10)
\end{aligned}$$

4 Et ensuite

Par la suite on pourra utiliser les gains suivants ...

On peut également ré-afficher les courbes de la simulation.

On propose finalement de tester une autre loi de commande par retour d'état intégrale réglé par LQR.
On pourra tester cette loi de commande au cours des expérimentations.
En parlant de ça, allons en salle de vol !

5 Annexe : formalisme contrôle linéaire

$$m\vec{\dot{v}} = -mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \implies \begin{cases} \ddot{x} &= \frac{T}{m}(c_\phi s_\theta c_\psi + s_\phi s_\psi) \\ \ddot{y} &= \frac{T}{m}(c_\phi s_\theta s_\psi - s_\phi c_\psi) \\ \ddot{z} &= -g + \frac{T}{m} c_\phi c_\theta \end{cases} \tag{11}$$

En considérant le modèle non linéaire 11, on identifie l'état $\xi = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$ et le vecteur de contrôle $u = \begin{bmatrix} \phi & \theta & T \end{bmatrix}^T$

$$\dot{\xi} = f(\xi, u)$$

Linéarisation autour de l'équilibre en vol stationnaire, $\xi_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ et $u_0 = \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T$.

$$\dot{\xi} \simeq \xi_0 + \left.\frac{\partial f}{\partial \xi}\right|_{\xi_0, u_0} (\xi - \xi_0) + \left.\frac{\partial f}{\partial u}\right|_{\xi_0, u_0} (u - u_0) \tag{12}$$

Définisions $\tilde{\xi} = \xi - \xi_0$ et $\tilde{u} = u - u_0$.
On arrive alors au système linéaire suivant :

$$\begin{cases} \begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \\ \ddot{\tilde{x}} \\ \ddot{\tilde{y}} \\ \ddot{\tilde{z}} \end{bmatrix} &= \begin{bmatrix} 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ g * s_\psi & g * c_\psi & 0 \\ -g * c_\psi & g * s_\psi & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{T} \end{bmatrix} \\ &= \begin{bmatrix} 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{T} \end{bmatrix} \\ &= A \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix} + Bf(\psi) \begin{bmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{T} \end{bmatrix} \end{cases}$$

Le système est linéaire en $\tilde{\xi}$ et $v = f(\psi) \begin{bmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{T} \end{bmatrix}$, avec $f(\psi)$ une matrice de rotation 3D, "projetant" le contrôle dans le repère du drone. On propose alors par exemple un contole par retour d'état :

$$\begin{aligned} v &= -K(\tilde{\xi} - \tilde{\xi}_{ref}) \\ &= -\begin{bmatrix} K_p^{3,3} & K_d^{3,3} \end{bmatrix} * (\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix} - \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix}_{ref}) \end{aligned}$$

Enfin, on retrouve u par

$$\begin{bmatrix} \tilde{\phi} \\ \tilde{\theta} \\ \tilde{T} \end{bmatrix} = f^{-1}(\psi)v = f^{T}(\psi)v = -\begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} * K * (\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix} - \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix}_{ref} \quad)$$