Projet IMA3-4

A la découverte des commandes des drones

Kadir Tekin, Jawad Soulaimani, Selim Bensalem

Sommaire

Introduction

- Objectifs initiaux accomplis -> Nouveaux objectifs
- Architecture matérielle et logicielle du drone
- Mise en place d’un réseau de capteurs et utilisation de FreeRTOS
- COVID19 -> nouveau projet sur GODOT

Drone Crazyflie

Réseau de capteurs

-utilisation du Loco Positioning System pour obtenir la position absolue du drone

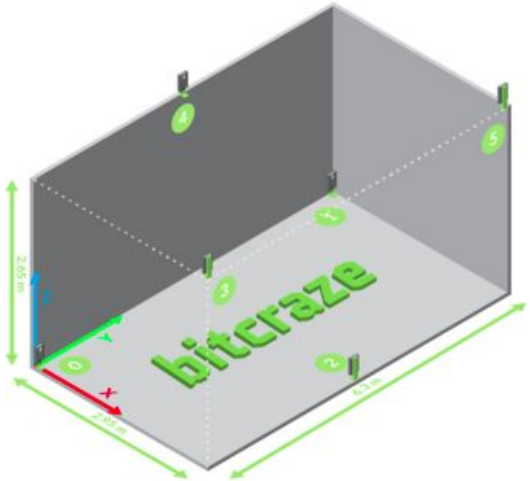
Le Loco Positioning Deck
(6 en tout)

Le Loco Positioning
Node (fixé sur le
drone)

Réseau de capteurs

Les capteurs sont répartis dans la pièce selon la configuration suivante :

Numéros attribués aux ancrages à l'aide du logiciel LPS node firmware



La position du tag drone peut se déterminer à l'aide des ancrages selon 2 types de géo-localisations différentes:

- la méthode Mesure des différences de temps d'arrivée (tdOA)
- la méthode Contraintes temporelles et mesure du temps de propagation aller-retour (TWR)

Réseau de capteurs

Mesure des différences de temps d'arrivée (TdOA):

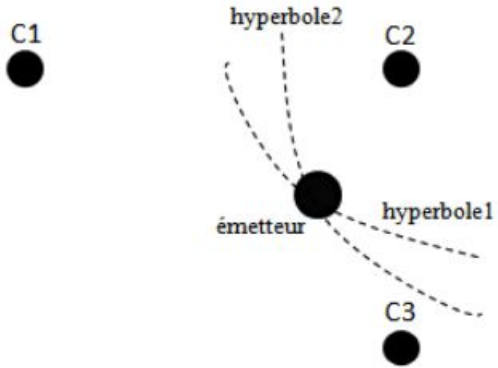
-> Méthode s'appuyant sur la différence des temps d'arrivée des signaux entre plusieurs ancrages.

-> Les ancrages sont synchronisés entre eux

-> Le tag situé sur le drone envoie des messages en broadcast à intervalles réguliers

-> Différence du temps d'arrivée du signal entre un couple d'ancrage = différence de distance entre le tag et le couple d'ancrage.

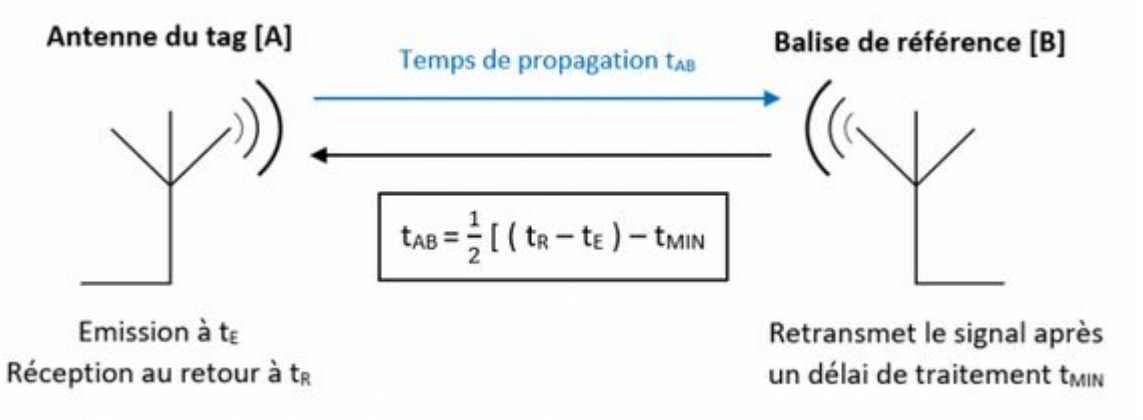
-> Tag situé sur une hyperbole ayant pour foyer le couple des récepteurs.



Réseau de capteurs

Méthode Contraintes temporelles et mesure du temps de propagation aller-retour (TWR)

-> distance obtenue à partir du temps de parcours entre un ancrage et le tag:

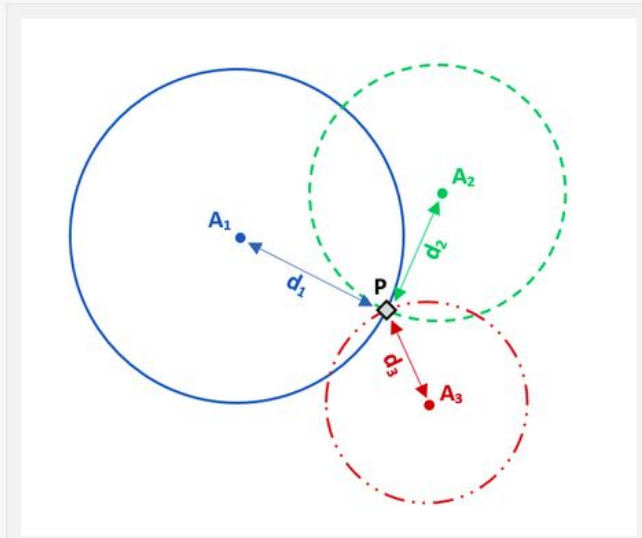


tmin : retard qui révèle tous les retards matériels et logiciels survenant lors de la transaction.

Réseau de capteurs

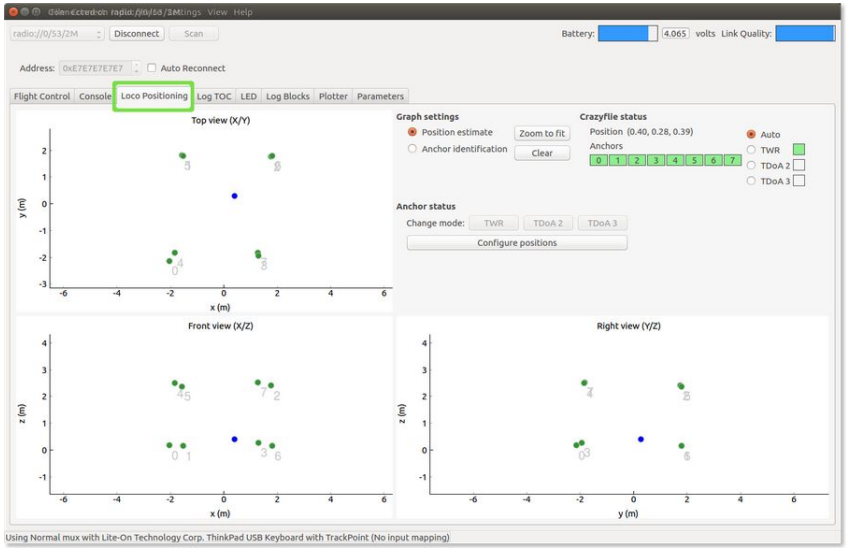
Méthode Contraintes temporelles et mesure du temps de propagation aller-retour (TWR) (suite)

-> La distance est calculée avec la formule suivante: Distance = c * tab



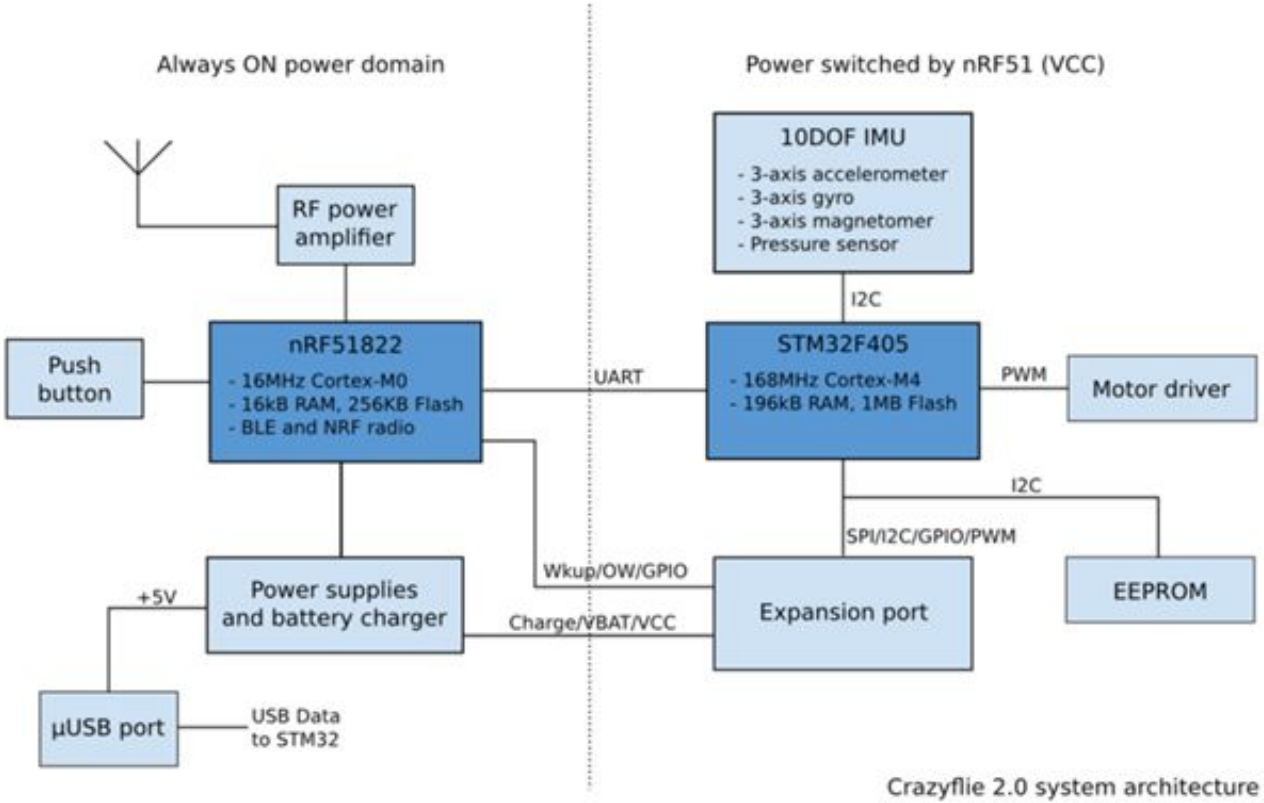
Réseau de capteurs

Les capteurs sont repérés dans le repère x,y,z de la pièce sur le logiciel :



Calculs de la précision non réalisés suite aux mesures liées aux COVID 19.

Architecture matérielle et logicielle



Crazyflie 2.0 system architecture

Premier pas en vol autonome

- Limites du système

- Problèmes rencontrés

- Fonctionnement d’une tâche de vol autonome sous freeRTOS

Projet Godot

Présentation du projet Godot

Objectifs :

- Simuler un drone sur Godot avec un moteur physique (implémenter les équations dynamiques qui régissent le drone).
- Réaliser une régulation à l'aide de Matlab.
- Etablir une communication UDP en boucle entre Matlab et Godot.

Communication UDP

Communication utilisée entre Matlab et Godot.

Communication en Boucle : Client Godot -> Serveur Matlab -> Client Matlab -> Serveur Godot....

Exemple de trame envoyée du Client Godot au Serveur Matlab:

->le mot "Start" qui indique le début d'une trame.

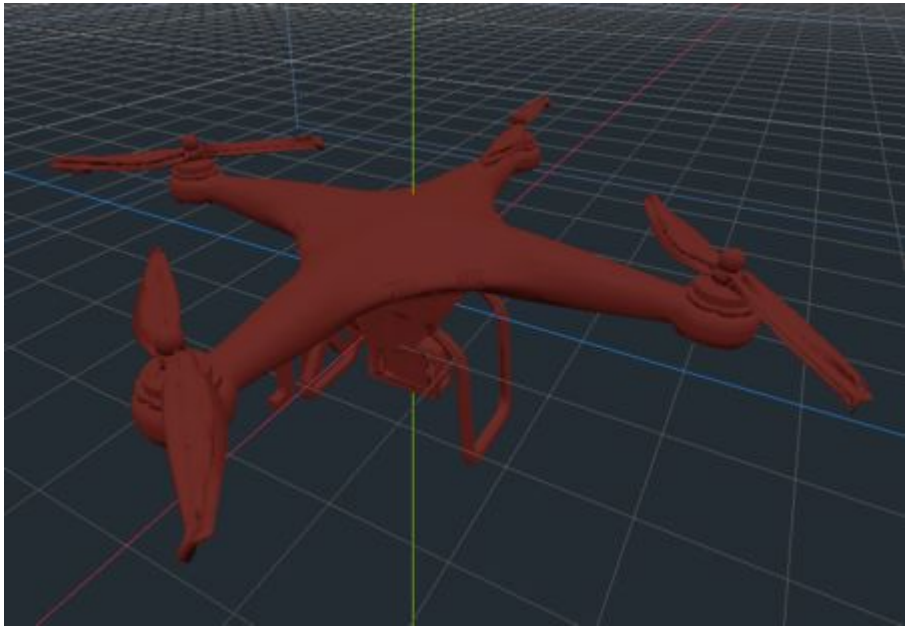
->Vitesse angulaire phi

->Vitesse angulaire psi

->Vitesse angulaire theta

->Vitesse de translation sur z

Godot : Mise en place de l’environnement graphique



Godot : Équations du modèle dynamique

$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = bl(\Omega_3^2 - \Omega_1^2) \\ \tau_y = bl(\Omega_4^2 - \Omega_2^2) \\ \tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{cases}$$

$[x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$ position linéaire et angulaire du drone dans le **repère fixe**.

$[u \quad v \quad w \quad p \quad q \quad r]^T$ **vitesse** linéaire et angulaire du drone dans le **repère mobile.**

f_t : force de poussée verticale

$\tau_{x,y,z}$: les moments selon x,y,z

l : distance moteur – centre du drone

d : facteur de traînée (drag coefficient)

b : facteur de poussée (thrust coefficient)

Ω_n : la vitesse du moteur

Godot : Équations du modèle dynamique

$$
\begin{cases}
\dot{\phi} = p \\
\dot{\theta} = q \\
\dot{\psi} = r \\
\dot{p} = \frac{\tau_x + \tau_{wx}}{I_x} \\
\dot{q} = \frac{\tau_y + \tau_{wy}}{I_y} \\
\dot{r} = \frac{\tau_z + \tau_{wz}}{I_z} \\
\dot{u} = -g\theta + \frac{f_{wx}}{m} \\
\dot{v} = g\phi + \frac{f_{wy}}{m} \\
\dot{w} = \frac{f_{wz} - f_t}{m} \\
\dot{x} = u \\
\dot{y} = v \\
\dot{z} = w
\end{cases}
$$

$\begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$ position linéaire et angulaire du drone dans le **repère fixe**.

$\begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T$ **vitesse** linéaire et angulaire du drone dans le **repère mobile.**

Perturbations du vent définies par :

$$\begin{bmatrix} f_{wx} & f_{wy} & f_{wz} & \tau_{wx} & \tau_{wy} & \tau_{wz} \end{bmatrix}^T \in \mathbb{R}^6$$

Résultats de Godot

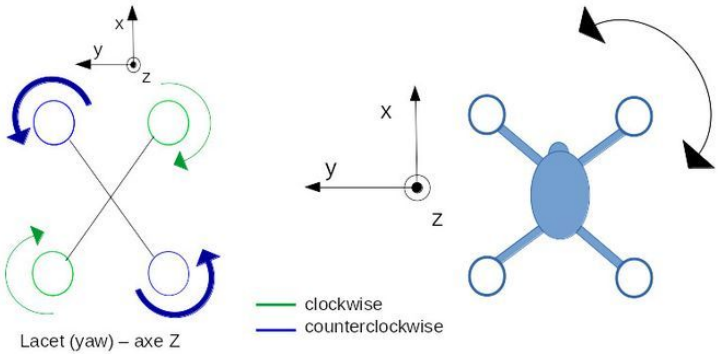
Vitesse maximale en translation verticale – axe Z						
V_{mn} (rad/s)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{z}$ (m/s)	$\dot{\phi}$ (rad/s)	$\dot{\theta}$ (rad/s)	$\dot{\psi}$ (rad/s)
$\Omega_1 = 6160$ $\Omega_2 = 6160$ $\Omega_3 = 6160$ $\Omega_4 = 6160$	0	0	-1.47	0	0	0

Vitesse maximale du Crazyflie selon Z

Résultats de Godot

Rotation – axe Z						
V_{mn} (rad/s)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{z}$ (m/s)	$\dot{\phi}$ (rad/s)	$\dot{\theta}$ (rad/s)	$\dot{\psi}$ (rad/s)
$\Omega_1 = 3000$ $\Omega_2 = 2900$ $\Omega_3 = 3000$ $\Omega_4 = 2900$	0	0	-0.34	0	0	-0.43

Ici on obtient un lacet

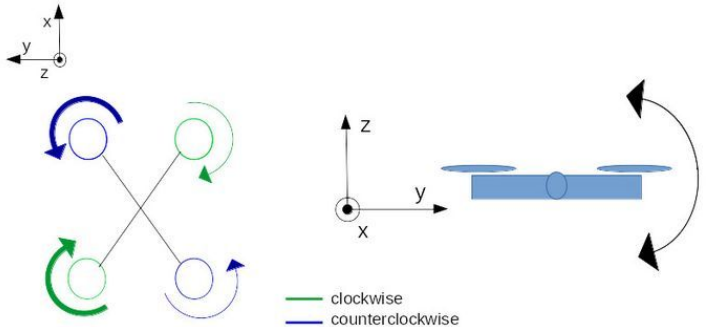


Lacet (yaw) – axe Z

Résultats de Godot

Rotation – axe X						
V_{mn} (rad/s)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{z}$ (m/s)	$\dot{\phi}$ (rad/s)	$\dot{\theta}$ (rad/s)	$\dot{\psi}$ (rad/s)
$\Omega_1 = 2980$ $\Omega_2 = 2980$ $\Omega_3 = 3000$ $\Omega_4 = 3000$	0	0	-0.35	2.73	0.1	0

Ici on obtient un roulis



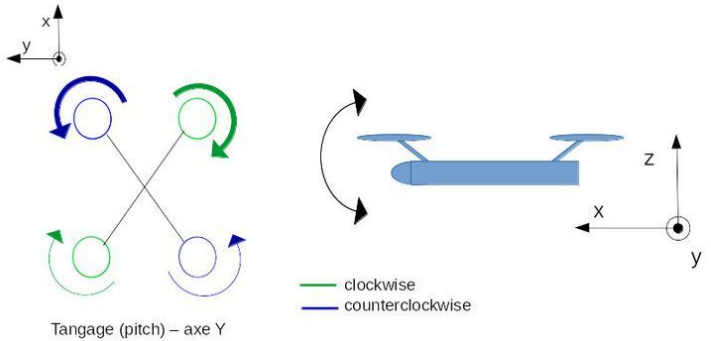
Roulis (roll) – axe X

Résultats de Godot

Rotation – axe Y						
V_{mn} (rad/s)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{z}$ (m/s)	$\dot{\phi}$ (rad/s)	$\dot{\theta}$ (rad/s)	$\dot{\psi}$ (rad/s)
$\Omega_1 = 3000$ $\Omega_2 = 2900$ $\Omega_3 = 3000$ $\Omega_4 = 3100$	0	0	-0.35	0	1.07	0

Ici on obtient un tangage

Vidéo, suivre le lien :
https://projets-ima.plil.fr/mediawiki/images/9/9e/Simulation_godot.mp4



Tangage (pitch) – axe Y

Simulink

Equations des moteurs en fonction des vitesses angulaires et de la vitesse de translation sur z :

$$\Omega_1 = \frac{\sqrt{\frac{f_t}{b}} - \sqrt{\frac{T_z}{d}}}{2} - \Omega_3$$

$$\Omega_2 = \frac{\sqrt{\frac{T_z}{b}} + \sqrt{\frac{f_t}{d}}}{2} - \Omega_4$$

$$\Omega_3 = \frac{\sqrt{\frac{T_x}{bl}} + \frac{\sqrt{\frac{f_t}{b}} - \sqrt{\frac{T_z}{d}}}{2}}{2}$$

$$\Omega_4 = \frac{\sqrt{\frac{T_y}{bl}} + \frac{\sqrt{\frac{T_z}{b}} + \sqrt{\frac{f_t}{d}}}{2}}{2}$$

$$T_x = \dot{p} \times \dot{I_x} - T_{wx}$$

$$T_y = \dot{q} \times \dot{I_y} - T_{wy}$$

$$T_z = \dot{r} \times \dot{I_z} - T_{wz}$$

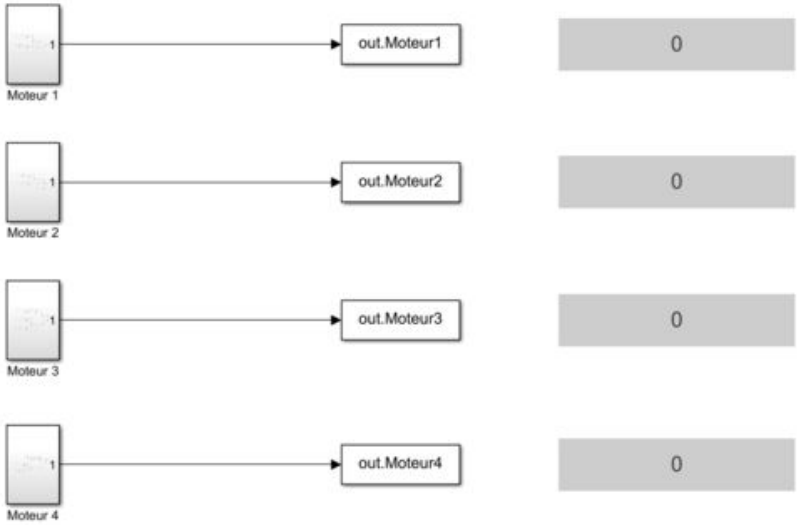
$$f_t = -(\dot{w} \times \dot{m}) + f_{wz}$$

Avec

- p, q et r les vitesses angulaires du drone
- w la vitesse de translation sur z

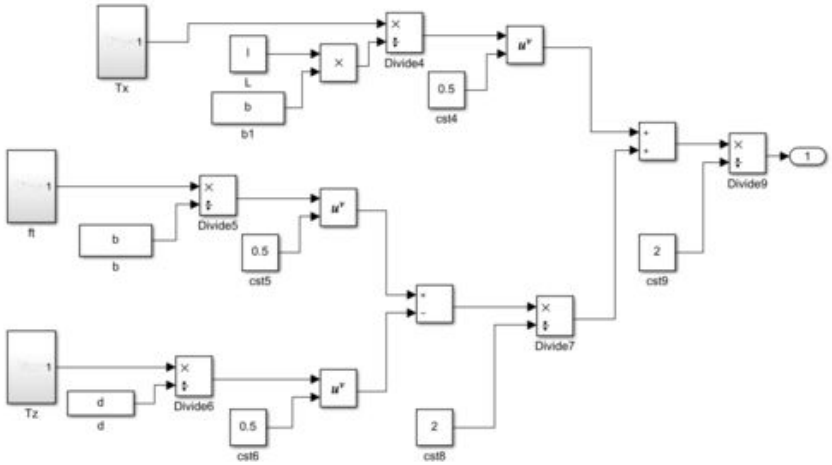
Simulink

Vue globale du simulink établie :

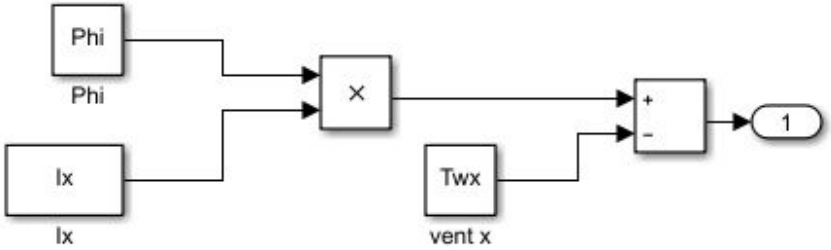


Simulink

Voici deux exemples dont l'un est le moteur numéro 3 et l'autre le moment généré par les moteurs sur l'axe x



Moteur 3



Tx

Résultats du simulink

Vitesse maximale en translation verticale sur -axe Z				
Vmoteur (rad/s)	θ (rad/s)	ϕ (rad/s)	ψ (rad/s)	$\dot{w}$ (m/s)
$\Omega_1 = 6156$	0	0	0	-1,47
$\Omega_2 = 6156$				
$\Omega_3 = 6156$				
$\Omega_4 = 6156$				

Vidéo du test simulink

Suivre le lien :

https://projets-ima.plil.fr/mediawiki/images/7/77/Simulation-drone.mp4

Régulation avec Matlab

- Suite à quelques problèmes rencontrés dans la partie simulink -> choix d'un fichier Matlab pour la régulation.

- Dans un premier temps, il faut annuler le moment du vent avec une bonne valeur des vitesses des moteurs. Pour cela nous résolvons pour chaque axe :

moment_moteur = - moment_vent, en fonction de Vm1,Vm2,Vm3,Vm4 (vitesses des moteurs)

Rappel des équations utiles :

$$\tau_x = bl(\Omega_3^2 - \Omega_1^2)$$
$$\tau_y = bl(\Omega_4^2 - \Omega_2^2)$$
$$\tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2)$$

$$\dot{\phi} = p$$
$$\dot{\theta} = q$$
$$\dot{\psi} = r$$
$$\dot{p} = \frac{\tau_x + \tau_{wx}}{I_x}$$
$$\dot{q} = \frac{\tau_y + \tau_{wy}}{I_y}$$
$$\dot{r} = \frac{\tau_z + \tau_{wz}}{I_z}$$

Régulation avec Matlab

On obtient donc un système à 3 équations :

$$\left|\begin{array}{l}\text{solve}\left(\begin{array}{l}\left(v3^2-v1^2\right)\cdot 1.32\cdot 10^{-5}=vx\\ \left(v4^2-v2^2\right)\cdot 0.56\cdot 1.32\cdot 10^{-5}=vy\\ \left(v2^2+v4^2-v1^2-v3^2\right)\cdot 1.385\cdot 10^{-6}=vz\end{array},\{v1,v2,v3,v4\}\right)\\ v1=0.003953\cdot\sqrt{-2.42375\text{E}9\cdot\left(vx+1.78571\cdot\left(vy+5.33718\cdot\left(vz-0.000003\cdot c5^2\right)\right)\right)}\ \text{and}\ v2=0.065795\end{array}\right.$$

Grâce au calculateur formel XCAS :

```
v4=max(340,378*sqrt(vy));
v1=0.003953*(-2.42375*10^9*(vx+1.78571*(vy+5.33718*(vz-0.000003*v4^2))))^(1/2);
v2=0.065795*(-3.125*10^7*(vy-0.000007*v4^2))^(1/2);
v3=194.625*(vx-1.78571*(vy+5.33718*(vz-0.000003*v4^2)))^(1/2);

% Conditions de validité
% -2.9*10^-5*(v4^2-62455*(vy+5.4*vz))≤ vx ≤ 2.9*10^-5*(v4^2-62455*(vy+5.4*vz))
% vz≤min(3*10^-6*(v4^2+34975*(vx-1.78571*vy)),3*10^-6*(v4^2-34975*(vx+1.78571*vy))) and v4-max(0.,378*√(vy))≥0. and vy≥0
% vy < 0.000007*v4^2
% v4 ≥ 378*√(vy) avec V4max=925, alors vymax=5.9
```

Vidéo de la régulation

Suivre le lien :

https://projets-ima.plil.fr/mediawiki/images/7/78/Regulation-finale.mp4

Conclusion

-> Missions réalisées

-> Compétences développées

Sources

Crazyflie's firmware
https://github.com/bitcraze/crazyflie-firmware

Design of a Trajectory Tracking Controller for a Nanoquadcopter. Technical report, Mobile Robotics and Autonomous Systems Laboratory
Luis C., & Le Ny, J., Polytechnique Montreal (August, 2016)

System Identification of the Crazyflie 2.0 Nano Quadrocopter
Julian Fôrster, ETH Zürich (2015)

Quadrotor control: modeling, nonlinear control design, and simulation
Francesco Sabatino, KTH Electrical Engineering, Stockholm Sweden (June 2015)

UWB Radiolocation Technology: Applications in Relative Positioning Algorithms for Autonomous Aerial Vehicles
Luke Beumont Barret, Murdoch University, Perth Australia (2018)