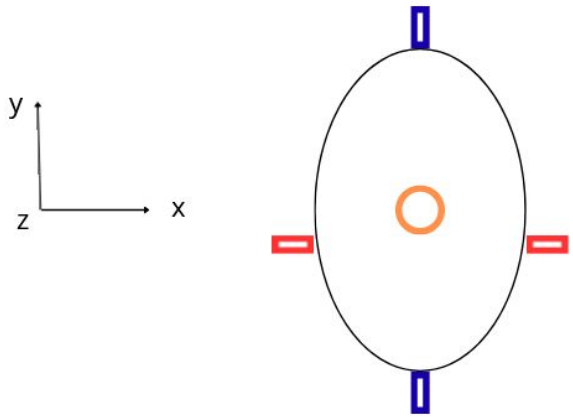
Asservissement

Projet Cobra

Ally–Platon Solenn
Dangmann Eloi
Guevel Xavier
Hugon Iliann
Bouteruche Martin

Chronologie :

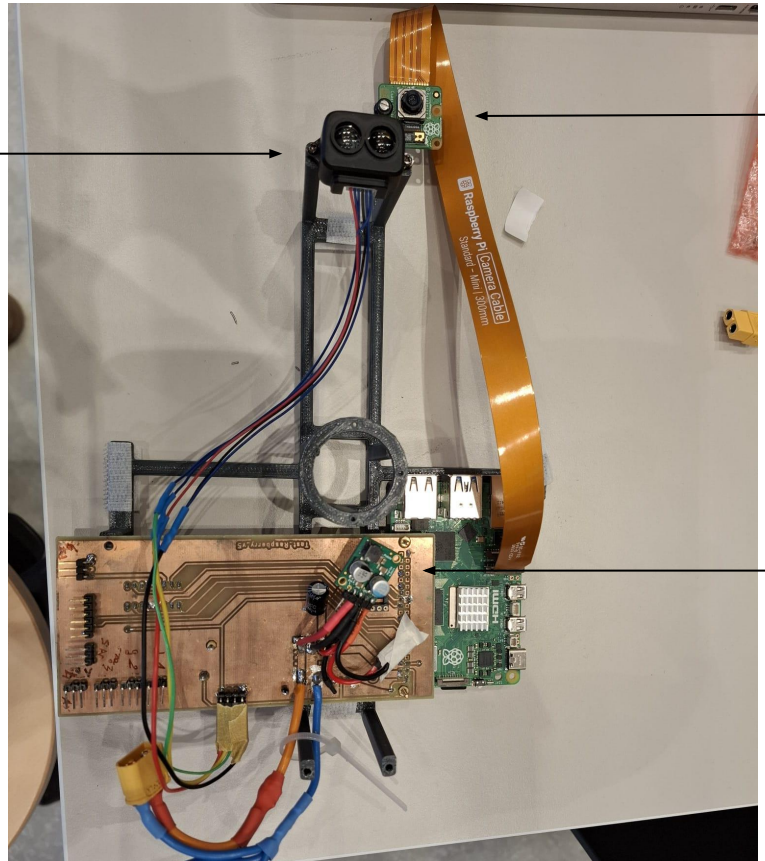
Positionnement des moteurs : vue du dessous



- Altitude z
- Déplacement y
- Lacet sur z
- Déplacement y

Capteurs pour l’asservissement

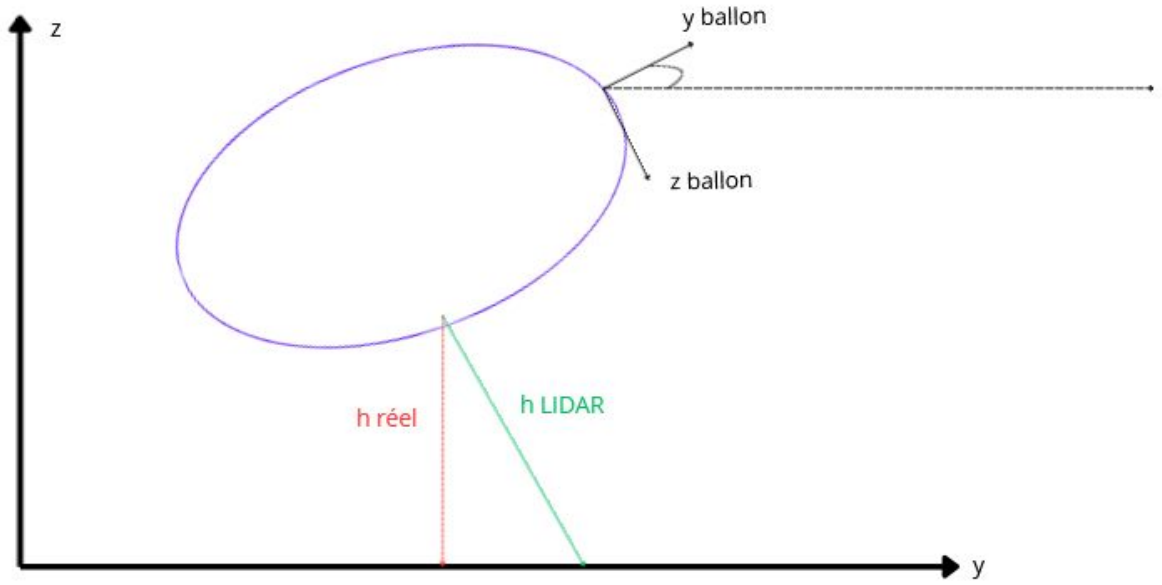
LIDAR :
Hauteur z



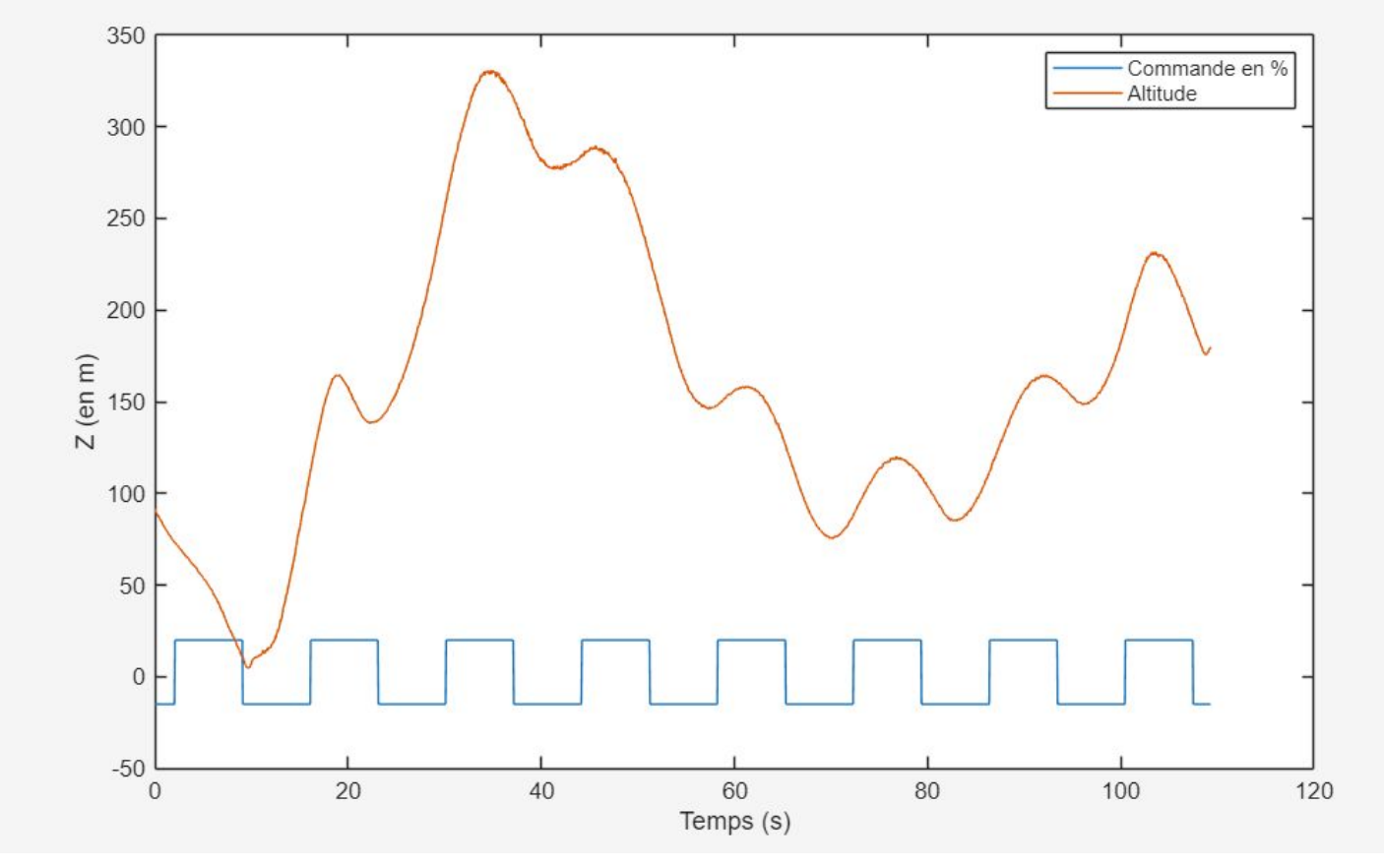
Caméra :
Position X et Y

Centrale inertielle :
Angles d’Euler

Hauteur mesurée et hauteur réelle



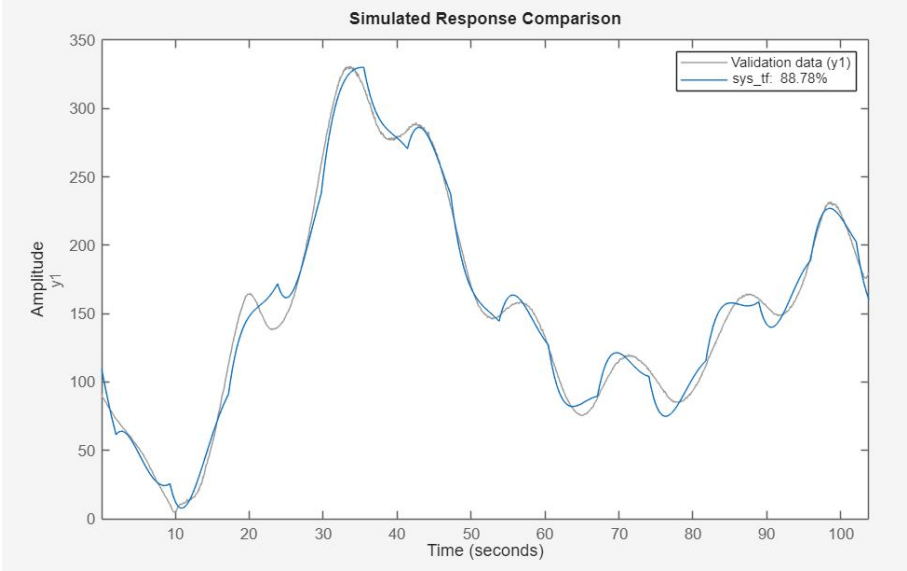
Asservissement V1: essai indiciel en altitude



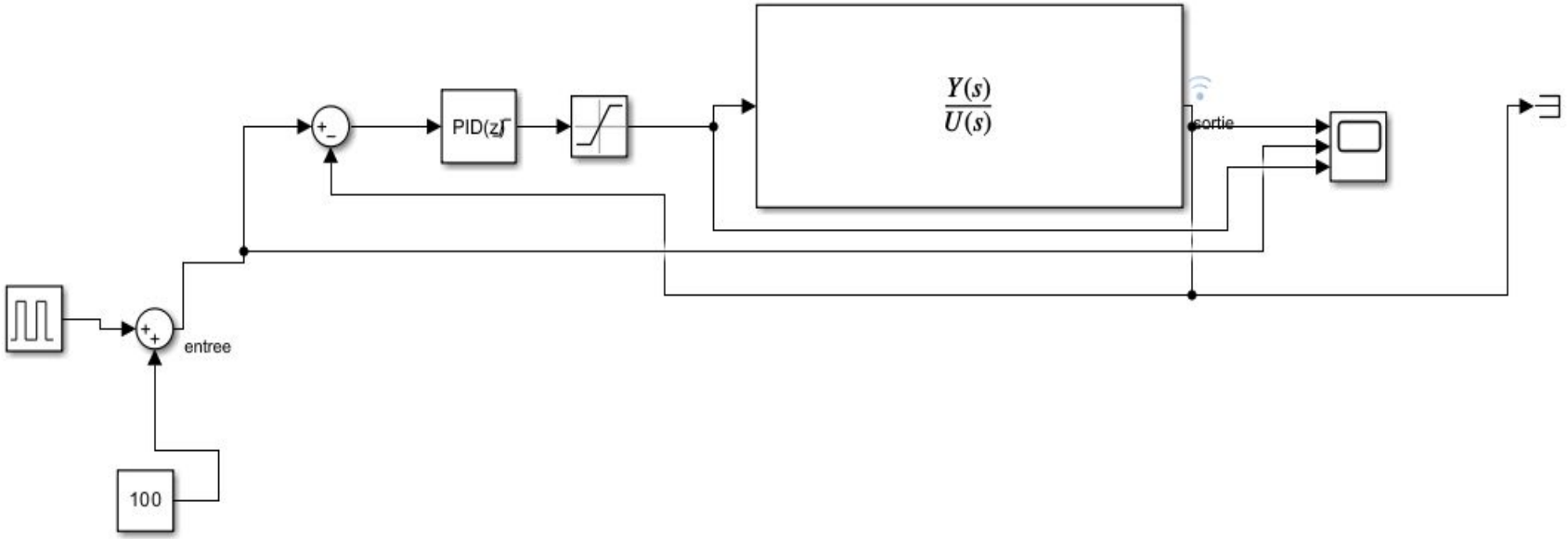
Asservissement V1 : Etablissement du modèle

Fonction de transfert du modèle :

$$\frac{0.8268\,s^4 - 0.2527\,s^3 + 0.1396\,s^2 - 0.01979\,s + 0.00236}{s^5 + 0.2269\,s^4 + 0.08403\,s^3 + 0.008173\,s^2 + 0.0008344\,s + 3.635e\text{-}05}$$



Asservissement V1: Schéma-bloc



Asservissement V1: Tuning du PID

$$P + I\cdot\frac{T_s}{2}\frac{z+1}{z-1} + D\frac{N}{1+N\cdot\frac{T_s}{2}\frac{z+1}{z-1}}$$

Main | Initialization | Saturation | Data Types | State Attributes

Controller parameters

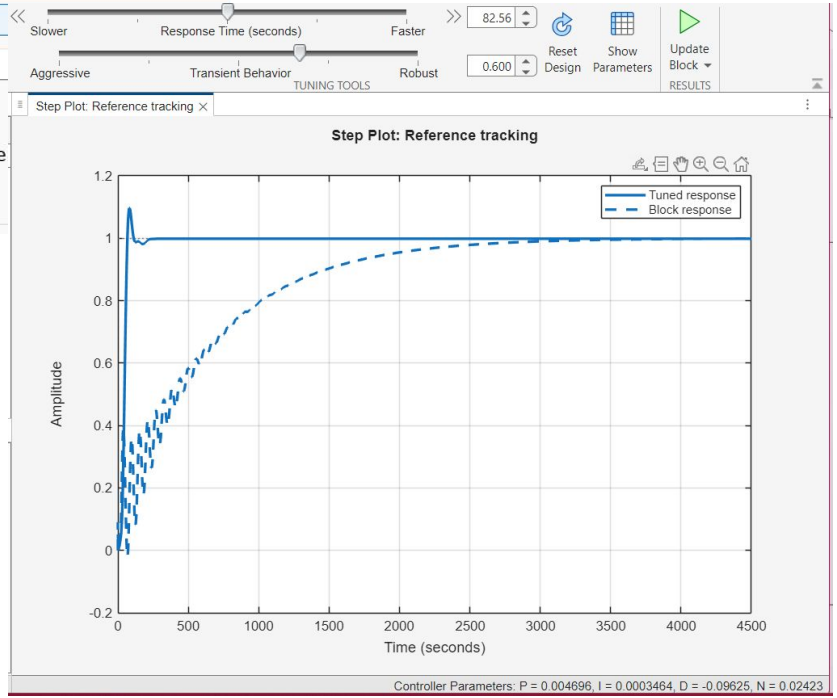
Source: internal

Proportional (P): 0.002

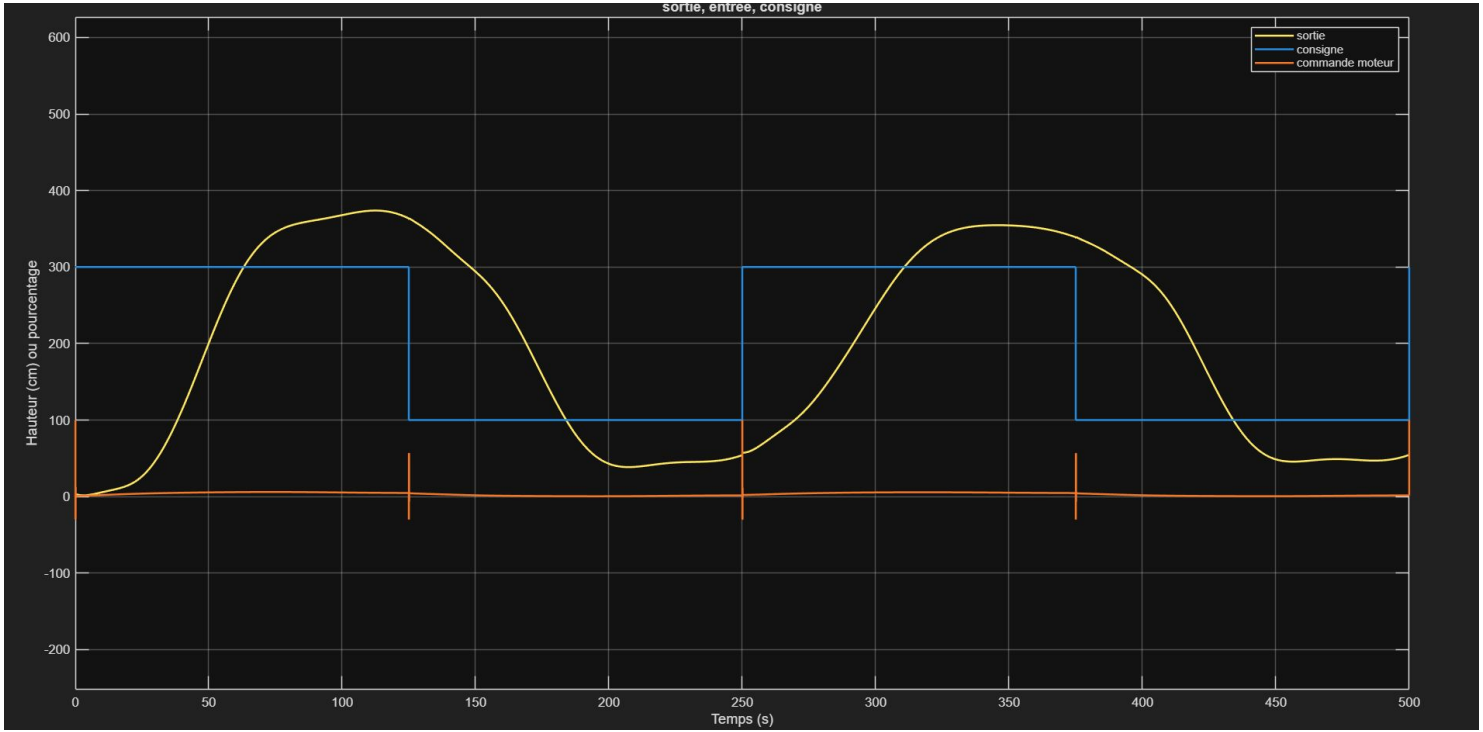
Integral (I*Ts): 0.000025 — Use I*Ts (optimal for codegen)

Derivative (D): 0.1 — Use externally sourced derivative

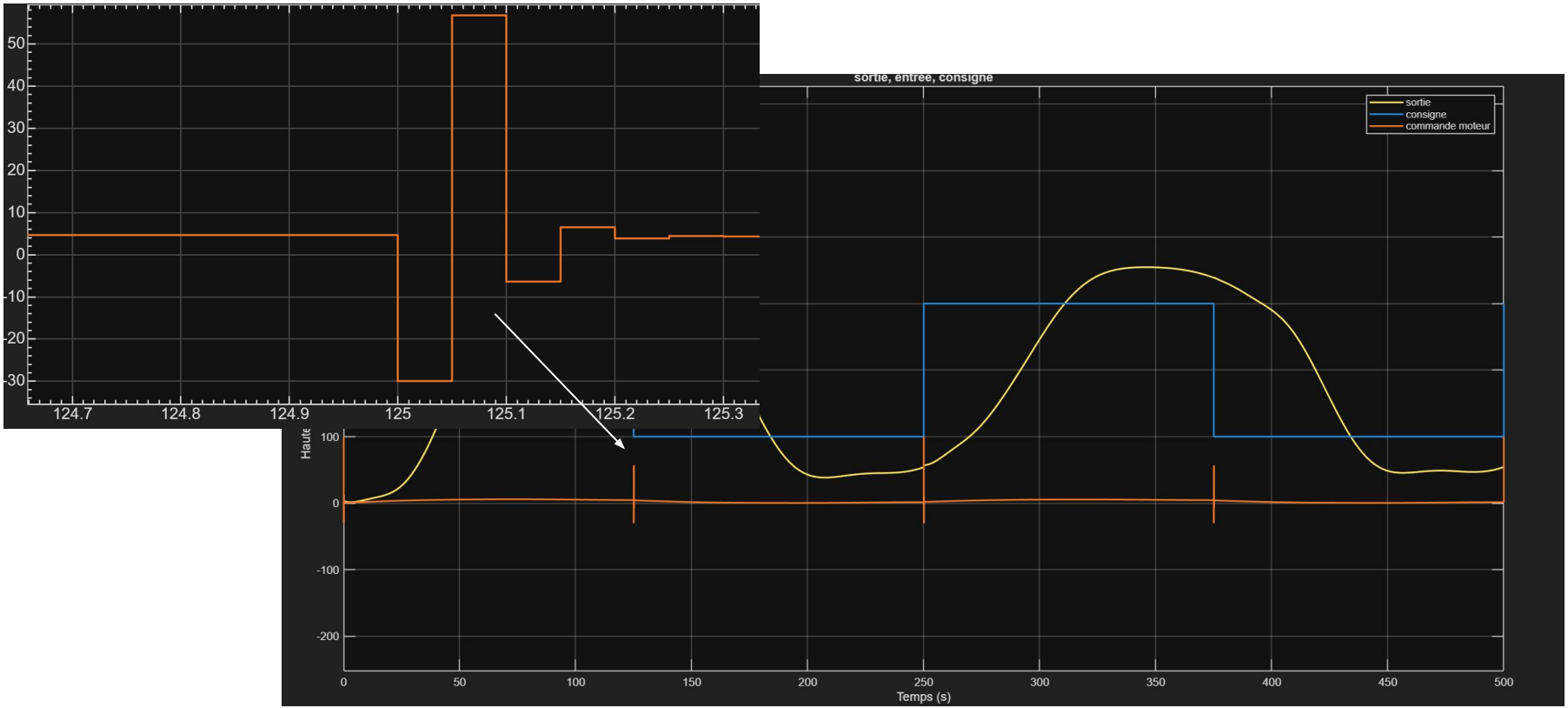
Filter coefficient (N): 50 — Use filtered derivative



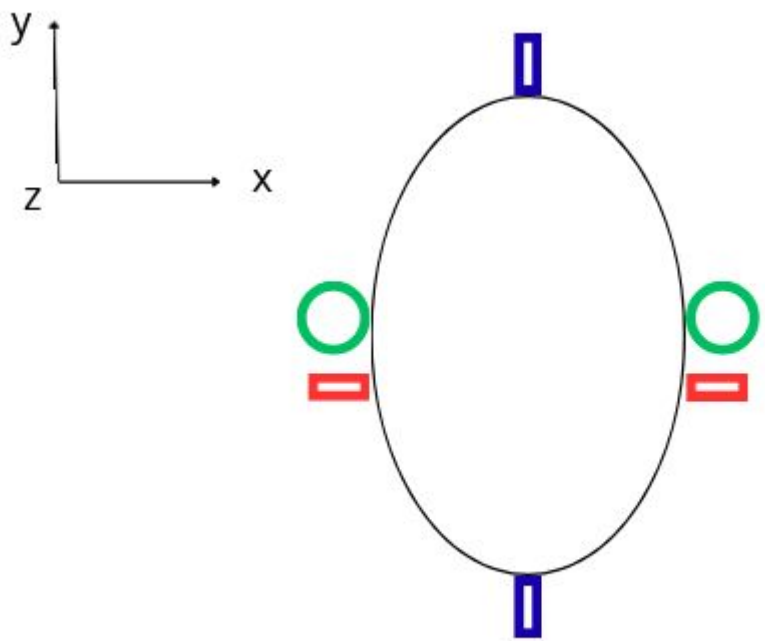
Asservissement V1: résultat de la simulation



Asservissement V1: résultat de la simulation

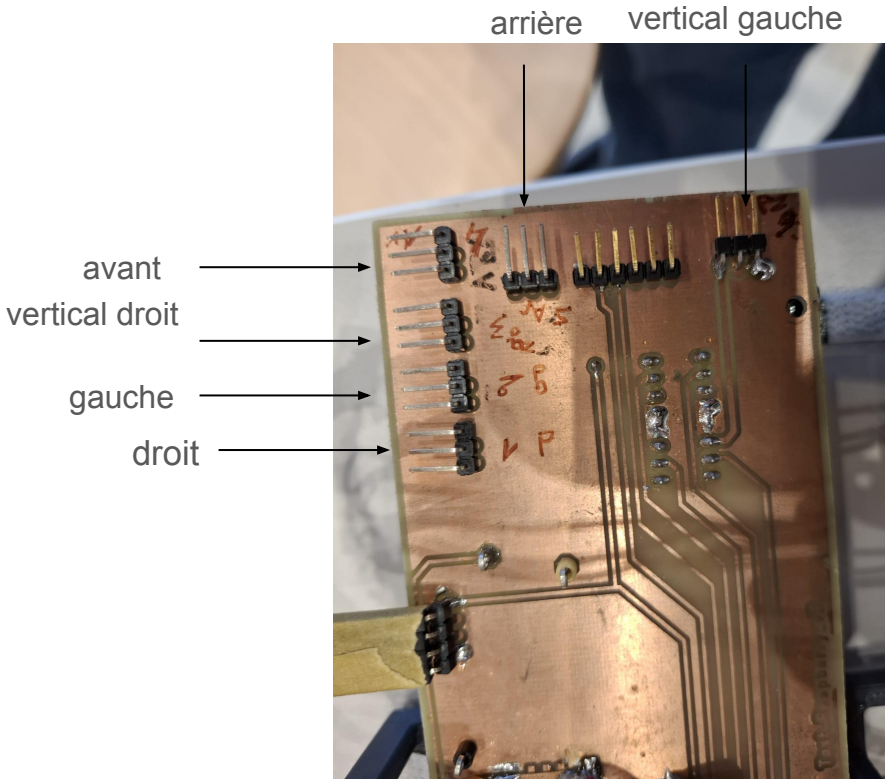


Disposition des nouveaux moteurs : vue du dessus



- Altitude z
- Déplacement y
- Lacet sur z

Carte électronique et branchement des moteurs



Asservissement du Cobra en pratique

Avant d’aborder l’asservissement multivariable, nous avons mis en œuvre des asservissements indépendants en altitude et en lacet.

Le correcteur PID a été recodé en intégrant un **filtre passe-bas** sur le terme dérivé, ce qui a permis d’obtenir une commande moteur plus lisse.

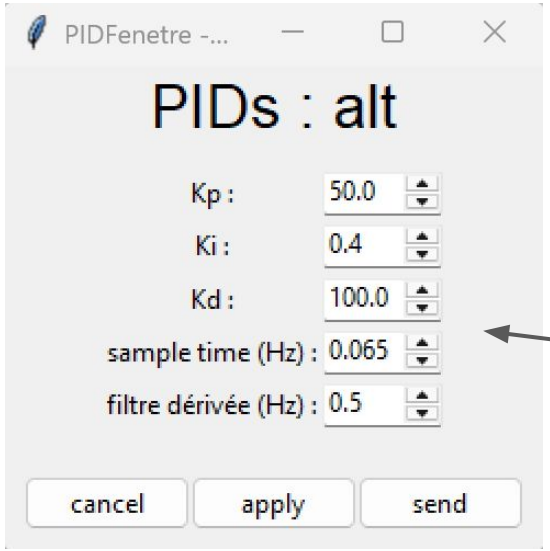
Les PID ont été réglés **empiriquement**, rendant le Cobra **globalement stable**.

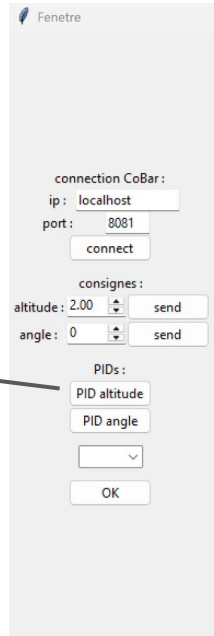
En revanche, l’asservissement en **position** n’a **pas** pu être **implémenté** de manière satisfaisante.

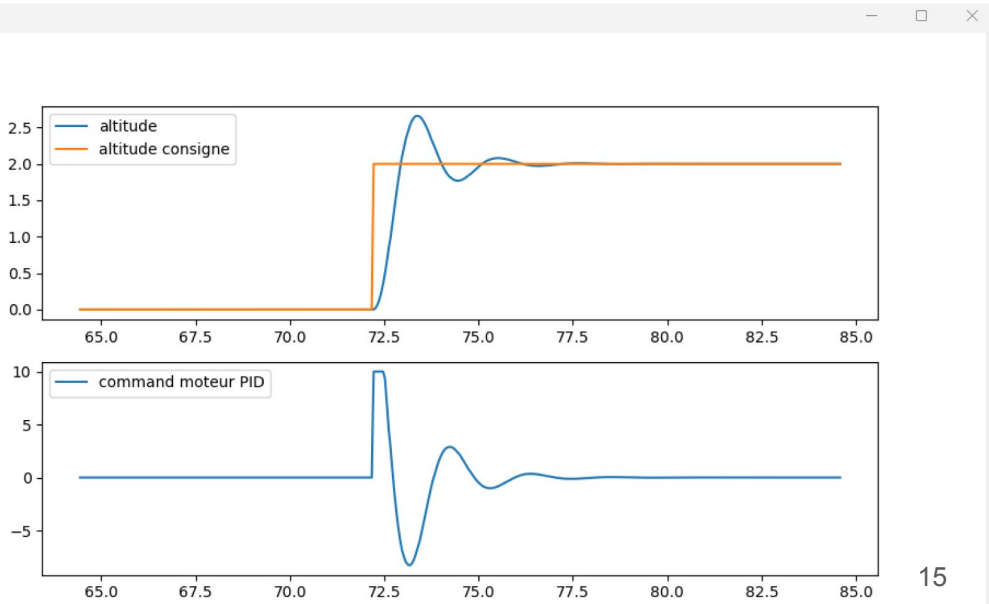
Télémesure en temps réel

La **principale** difficulté rencontrée lors du réglage empirique des PID était **l'impossibilité d'analyser** correctement les **données**.

Pour y remédier, nous avons développé une **interface graphique** en Python permettant de **visualiser les courbes en temps réel** :

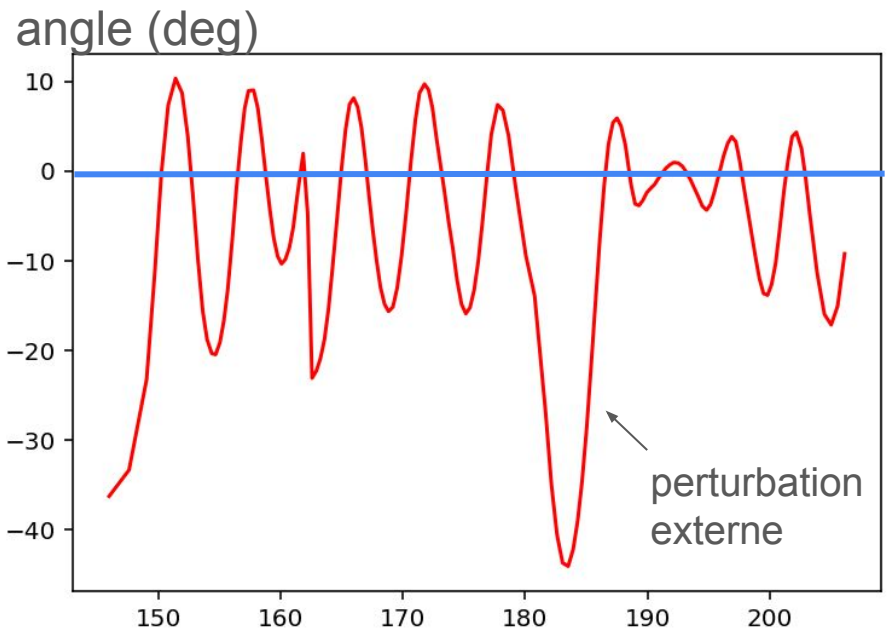




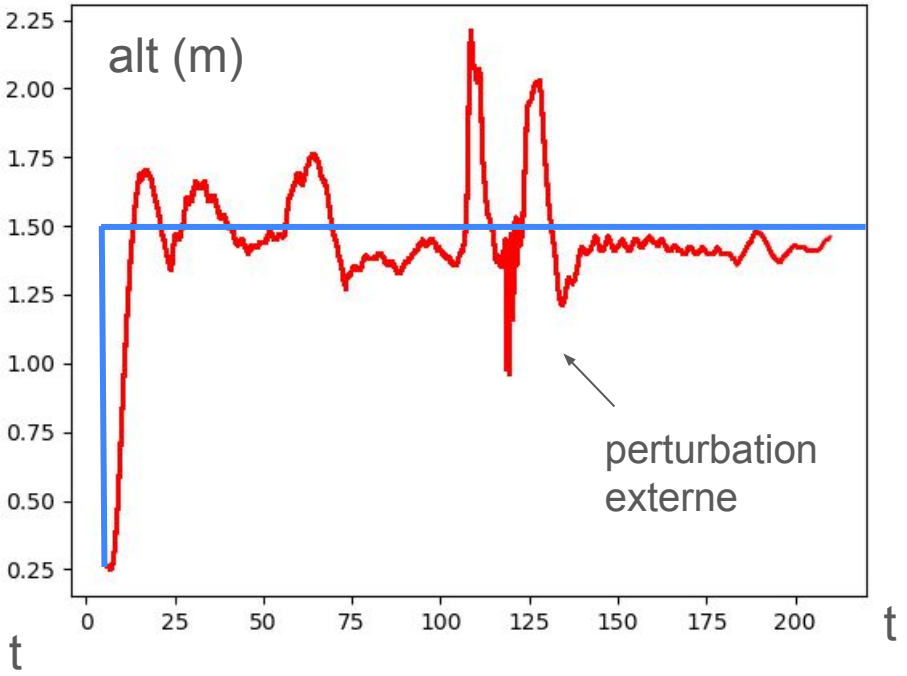


Exemples d'asservissements

en angle, avec une consigne de 0 degré:



en altitude, avec une consigne de 1.5 mètre:



La suite

– Correction des bugs de l’interface graphique

– Amélioration des correcteurs PID, entre autres avec prise en compte de la masse variable du ballon

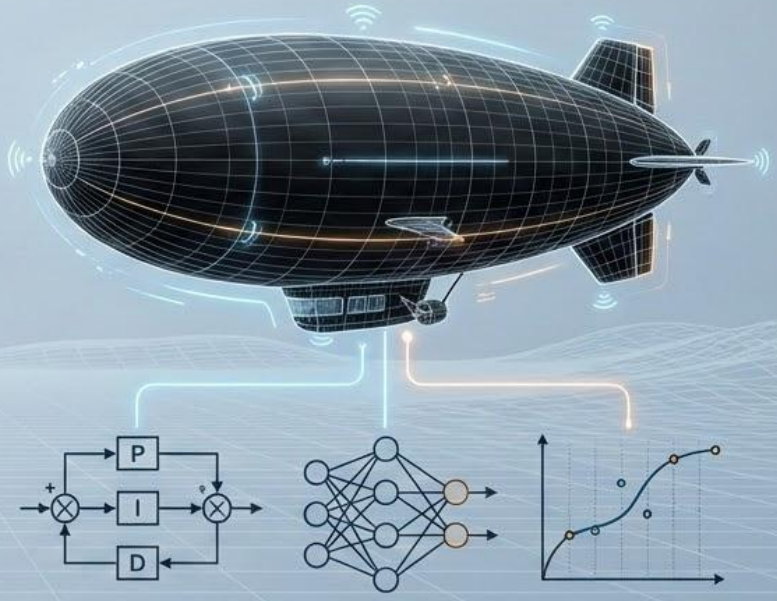
– Ajout des commandes en position

– Mise en œuvre de l’asservissement multivariable

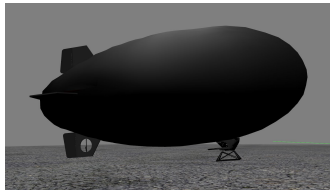
Simulation du Cobra

(Xavier et Iliann)

Objectif: Faire une simulation réaliste du ballon dirigeable pour pouvoir tester facilement les paramètres de correcteur PID et explorer une solution d'asservissement par apprentissage par renforcement

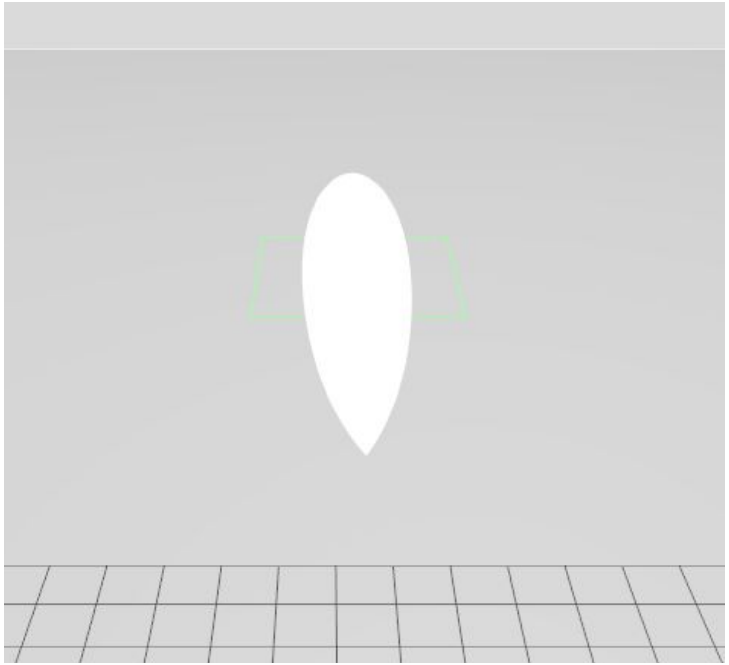


Bilan sur les logiciels de simulation

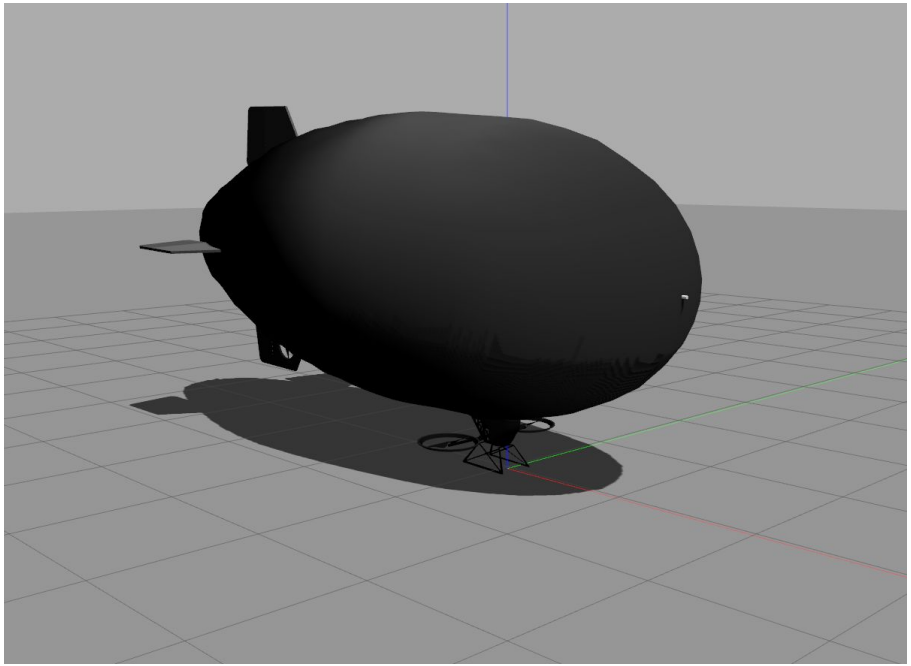


	Webots	Gazebo	Airsim	FlightGear
Avantages	Logiciel connu de Mr. Juton Projet d'objets volants existants + existence module physique Faible demande en ressource informatique	Projet existant similaire open-source Interface GUI compréhensible Fort potentiel de modification/adaptation	Bonne documentation (dev. par Microsoft) Intégration dans des logiciels modernes Intégration simple avec les systèmes de contrôle (ex: PX4) Initialisation facile	Modèles aero matures, robustes complets Possibilité de contrôle externe Modèles de dirigeables existants Projets de ML existants
Inconvénients	Modélisation peut être trop simple	Complexité d'installation - Installation impossible sur Windows - Versions spécifiques d'Ubuntu - Mises à jour des plugins Mal maintenu	Fait pour la modélisation d'hélices Fonctions haut niveau -> difficile à adapter	Initialement pour le JV -> difficile à détourner

Gazebo :

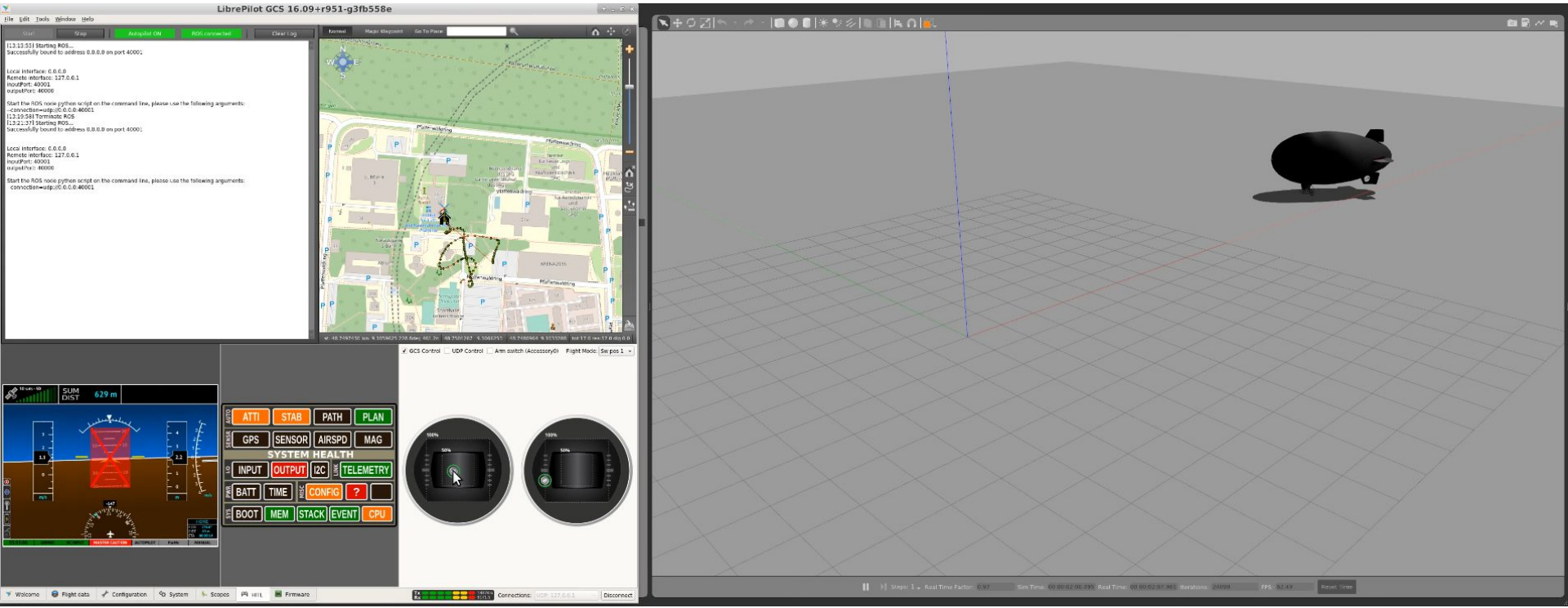


Modèle déjà présent

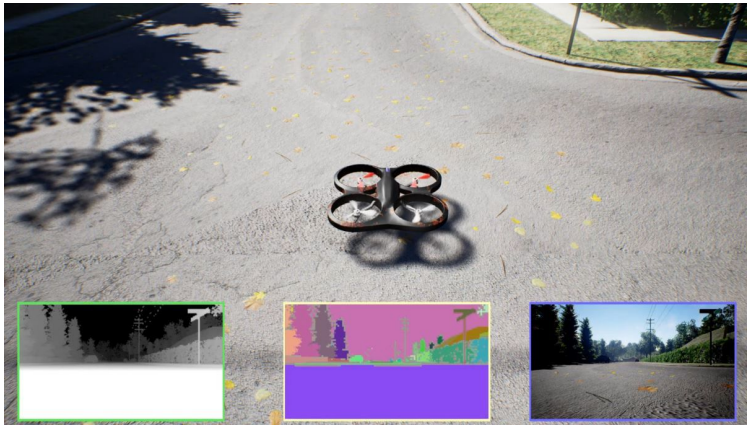


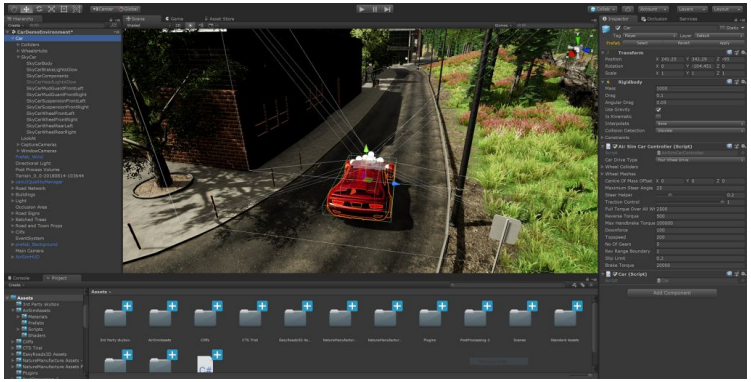
Modèle proposé par l’article

Gazebo :


LibrePilot GCS 16.09+r951-g3fb558e
ATTI
STAB
PATH
PLAN
GPS
SENSOR
AIRSPD
MAG
SYSTEM HEALTH
INPUT
OUTPUT
I2C
TELEMETRY
BATT
TIME
CONFIG
BOOT
MEM
STACK
EVENT
CPU

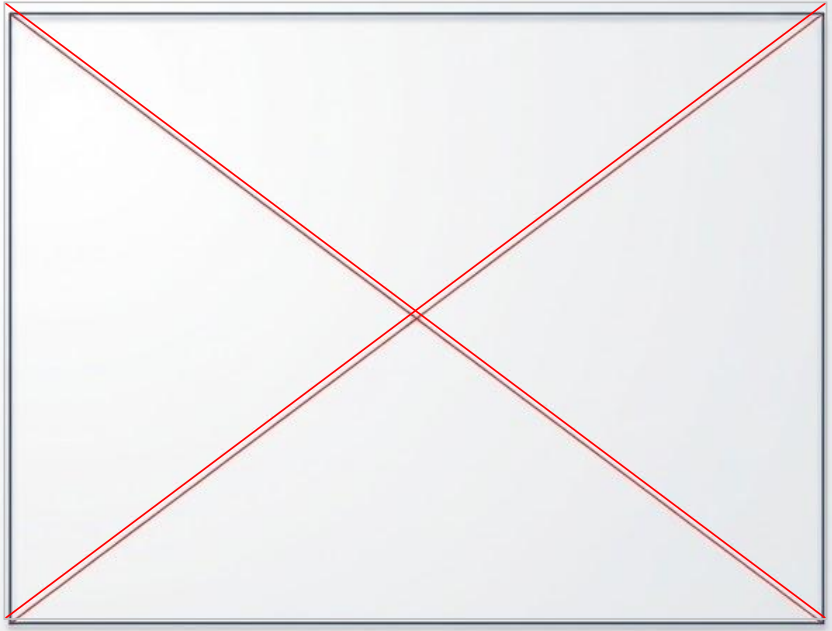
Airsim / FlightGear





WEBOTS

- Logiciel sur lequel le projet est actuellement le plus abouti.
- Le modèle est fonctionnel et facilement exploitable.
- Le ballon peut se déplacer dans toutes les directions à l'aide des touches du clavier.

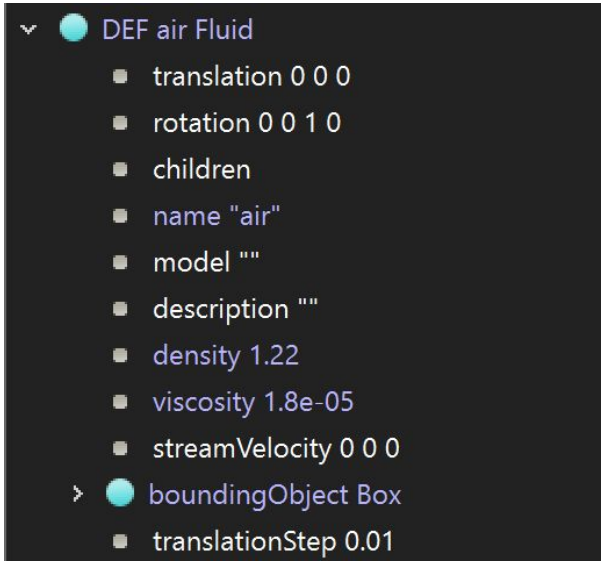


Travail sur webots

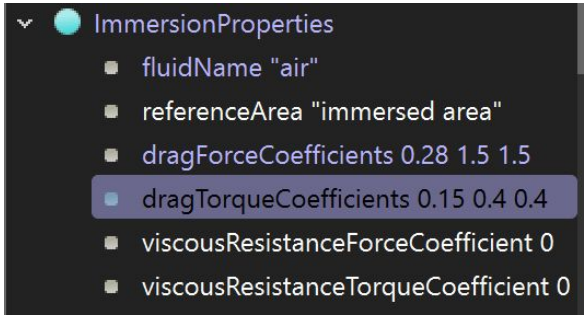
- **Mise en place de la physique** : utilisation d'un fluide de type "air" afin de reproduire un comportement de vol réaliste.
- Ballon dirigeable modélisé avec des **dimensions** proches de la réalité, garantissant un placement cohérent des moteurs.
- **Gestion des collisions** à l'aide d'un système de trois sphères, permettant d' éviter les conflits de collision.
- Estimation rapide des **propriétés physiques** de vol : coefficients de traînée, suppression des effets de viscosité, centre de masse, matrice d'inertie et masse globale.
- Utilisation de **moteurs "de drones DJI"** pour la propulsion
- Modélisation des **moteurs** intégrant des valeurs de poussée réalistes (refaire mesure de vitesse moteur car résultats de simulation étonnants)
- Un peu de code pour relier consigne clavier et consigne des moteurs

Choix pour la physique du système

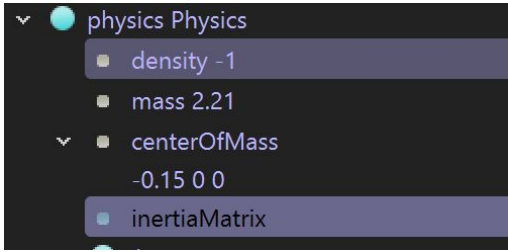
Immersion dans un fluide avec les propriété de l'air



Drag coefficient: https://www.engineeringtoolbox.com/drag-coefficient-d_627.html

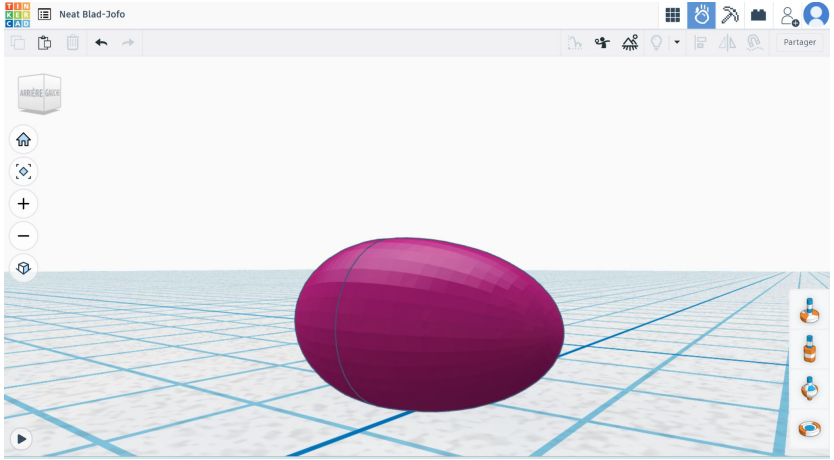


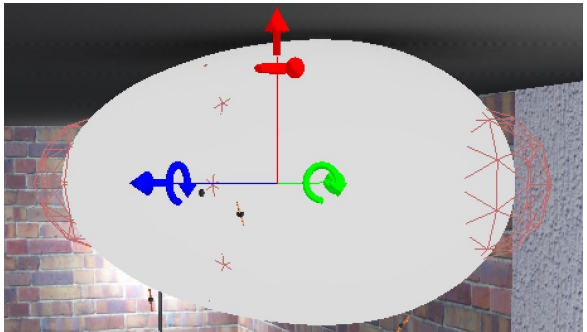
Positionnement du centre de masse et auto-compilation de la matrice d'inertie



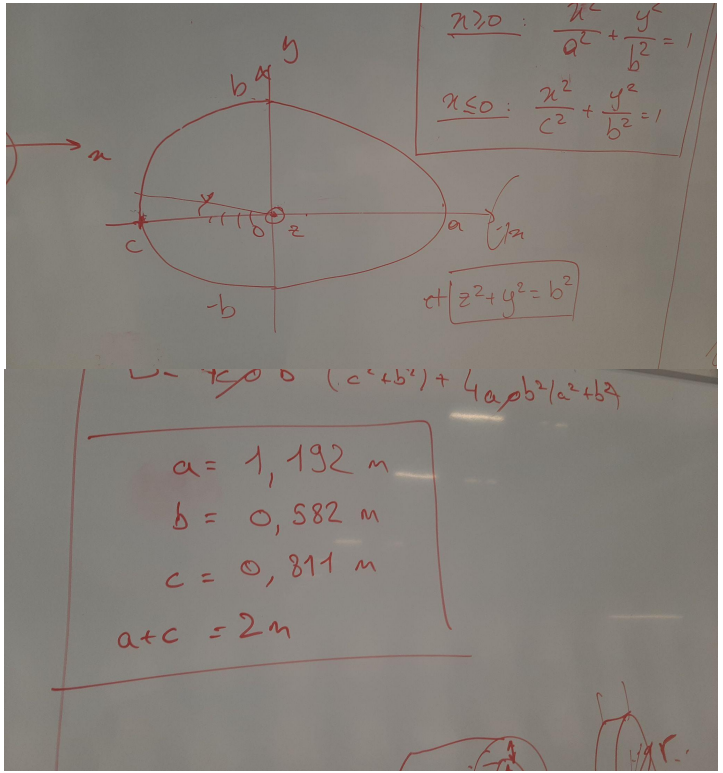
Dimensions et gestions des collisions

Fabrication de l’enveloppe sur Tinkercad + Blender

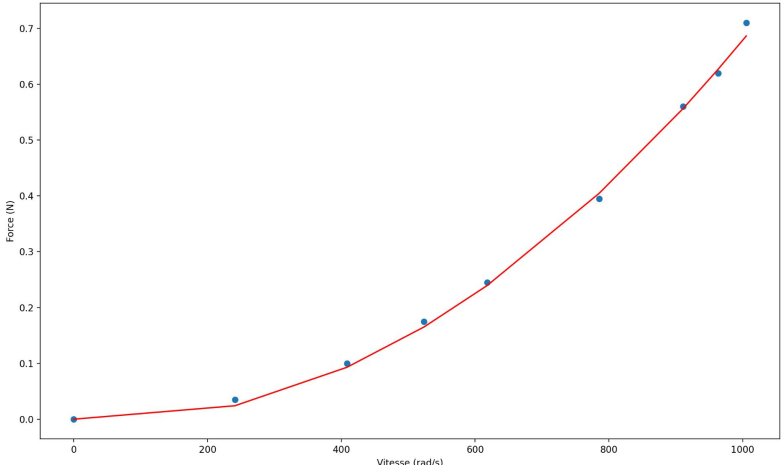




Dimension réel du ballon:



Calcul de la puissance des moteurs

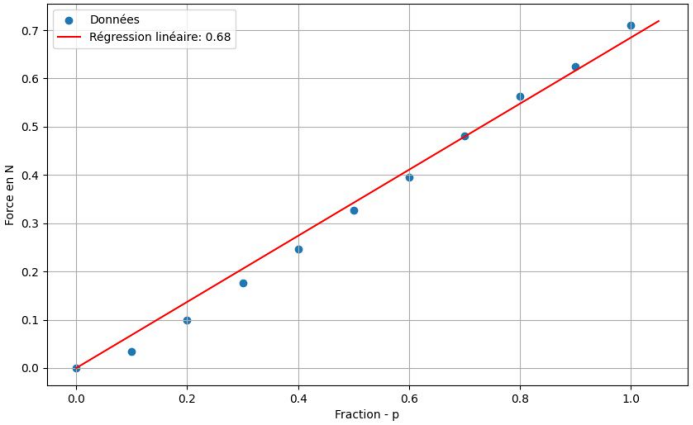


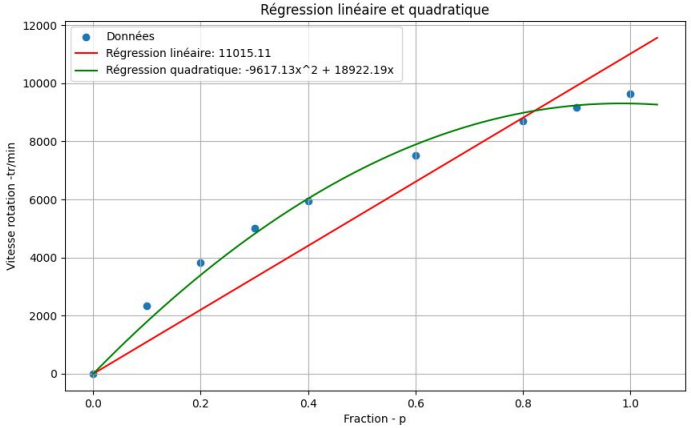
Régression polynomiale d'ordre 2 sur la force en fonction de la vitesse de rotation pour determiner les coefficients t1 et t2

```
T = t1 * |omega| * omega - t2 * |omega| * V
```

```
Q = q1 * |omega| * omega - q2 * |omega| * V
```

Données sur les moteurs





Évolutions à venir:

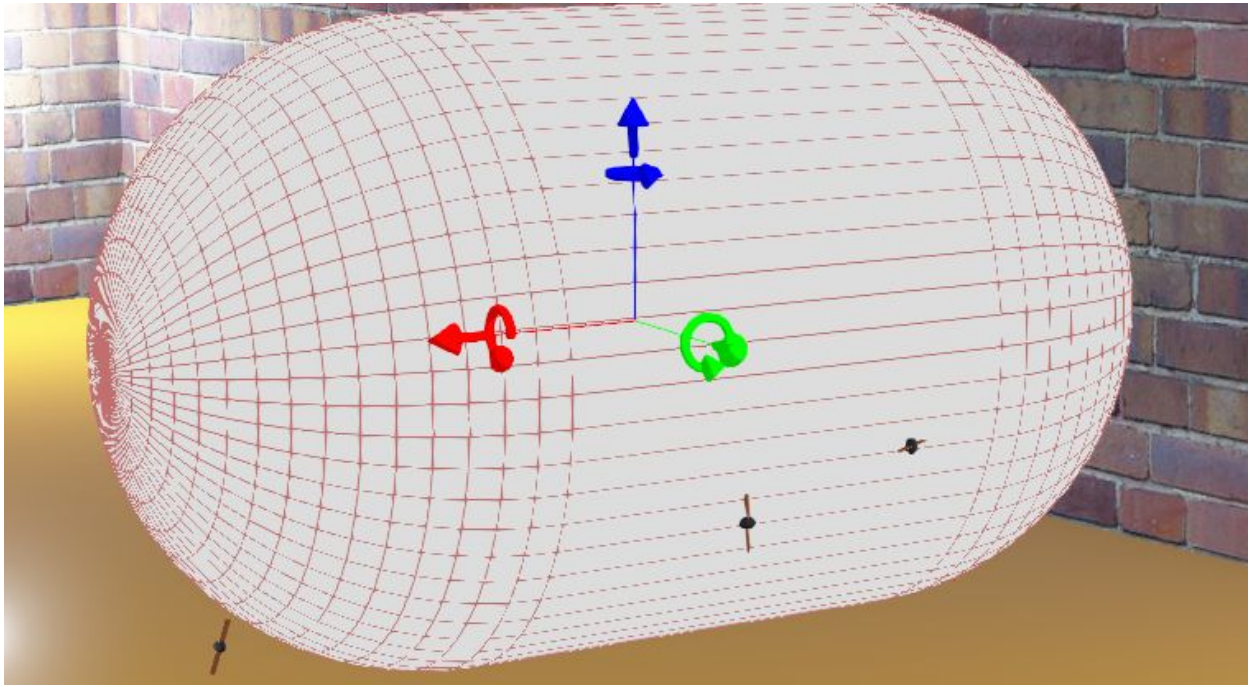
Evolution court terme:

Mise en place du PID dans Webots

Evolution long terme (plus difficile):

- Mettre en place un asservissement multiparamètres
- Explorer une solution d'un modèle d'asservissement par **apprentissage par renforcement**

Annexe :



Autre représentation 3D