

SAPHIRE Module Transverse

Projet Cobra - Equipe 2

Compte rendu

Jérémy Coulson - Simon Chardin - Nicolas Detzen
Bourdyot Mathys - Ulysse Poulain - Ahmed Nachawati
Fantine Mpacko Priso

Professeurs encadrants :
Bruno Denis - Bruce Anglade - Anthony Juton
Bastien Durand - Fabien Adam

– 11 juin 2024 –

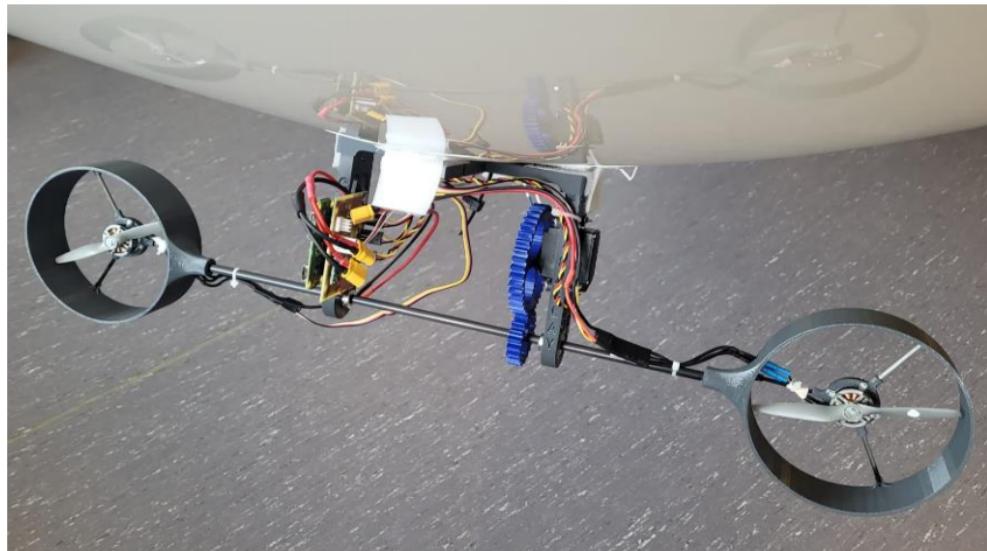


Table des matières

I	Introduction	3
II	Structure mécanique - Jérémy Couslon, Simon Chardin, Mathys Bourdyot	4
II.1	Prototype - Jérémy Couslon, Simon Chardin	4
II.2	Implémentation des fonctions mécaniques sur la nacelle - Mathys Bourdyot	5
II.3	Conception finale - Jérémy Couslon, Simon Chardin	7
III	Électronique embarquée - Fantine Mpacko Priso	10
III.1	Composants à intégrer	10
III.2	Contraintes électriques	10
III.2.1	Conception théorique de la carte	10
III.3	Gravure de la carte	11
III.4	Contraintes mécaniques	12
IV	Asservissement - Nicolas Detzen	13
IV.1	Mise en contexte et définition	13
IV.2	Stratégie d'asservissement	14
IV.3	Implémentation du code	15
IV.4	Résultats expérimentaux	18
V	Tests et validation - Ulysse Poulin - Ahmed Nachawati	19
V.1	Objectif des tests	19
V.2	Tests des composants électroniques	19
V.3	Tests d'asservissement	19
V.4	Tests de la structure mécanique	19
V.4.1	Test de rigidité et de résistance	20
V.4.2	Évaluation du poids	20
V.4.3	Tests d'assemblage et de montage	20
V.4.4	Tests en conditions réelles	20
V.4.5	Conclusion des tests de la structure mécanique	21
VI	Conclusion et perspectives	21

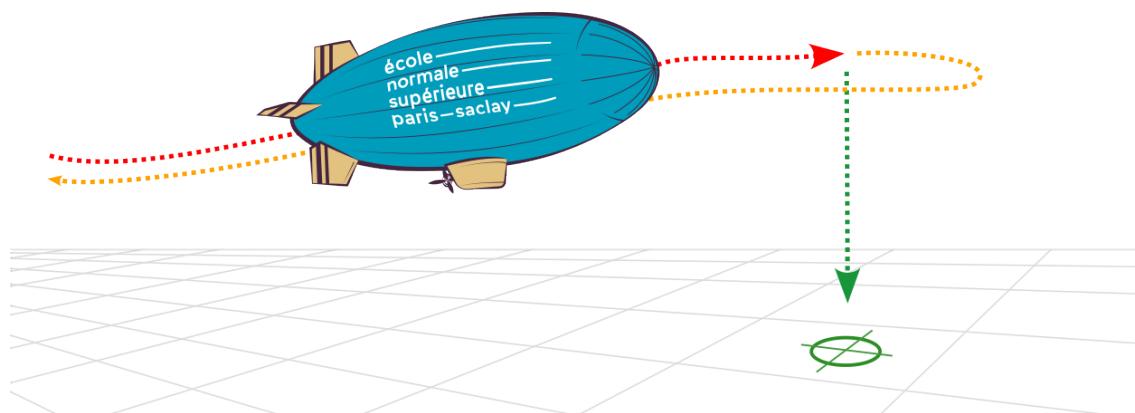
I Introduction

Ce projet de conception et de fabrication de la nacelle du dirigeable vise à développer un prototype fonctionnel capable de se déplacer de manière autonome tout en respectant des contraintes strictes de poids, de taille et de performances.

Après les premiers tests, plusieurs améliorations ont été nécessaires. La nacelle a été divisée en trois pièces pour faciliter la maintenance. Des emplacements pour la carte de commande, les capteurs et la batterie ont été ajoutés. L'emplacement du servomoteur a été ajusté en hauteur, et les supports moteurs ont été redimensionnés pour réduire le poids. La conception assistée par ordinateur (CAO) a été réalisée avec Catia V5, et les paramètres d'impression optimisés pour alléger la structure.

L'intégration électronique a été effectuée avec le logiciel EAGLE, en tenant compte des contraintes de poids et de taille. Les composants intégrés comprennent un accéléromètre, une Raspberry Pi, un servopilote PWM, un régulateur de tension et des solutions d'alimentation pour les moteurs. Un correcteur PID a été utilisé pour l'asservissement du dirigeable, contrôlant les angles de lacet et de tangage ainsi que la hauteur de vol.

Ce rapport présente les étapes de conception, de réalisation et d'optimisation du dirigeable, en détaillant les solutions techniques adoptées pour répondre aux contraintes et améliorer les performances du prototype.



II Structure mécanique - Jérémie Couslon, Simon Chardin, Mathys Bourdyot

L'objectif de cette partie est la réalisation et conception d'une structure nommée nacelle permettant le mouvement du dirigeable, respectant diverses exigences :

1. la masse doit être minimisée
2. la structure doit être suffisamment rigide
3. la structure doit être montable/démontable facilement étant donné le partage des dirigeables entre groupes
4. la structure doit permettre la fixation des divers capteurs utilisés
5. la structure doit faire preuve d'un esthétisme suscitant un sentiment d'admiration de la part des spectateurs, tel qu'il suffit à lui seul à élire vainqueur l'équipe 2

II.1 Prototype - Jérémie Couslon, Simon Chardin

Voici la conception qui nous sert de base :

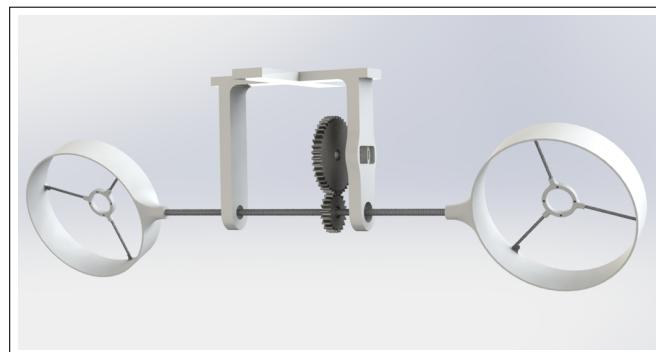


FIGURE 1 – Prototype de base

Sur cette base, nous réalisons une première nacelle après ajustements (perçages, découpages, limages) visible figure 2 :

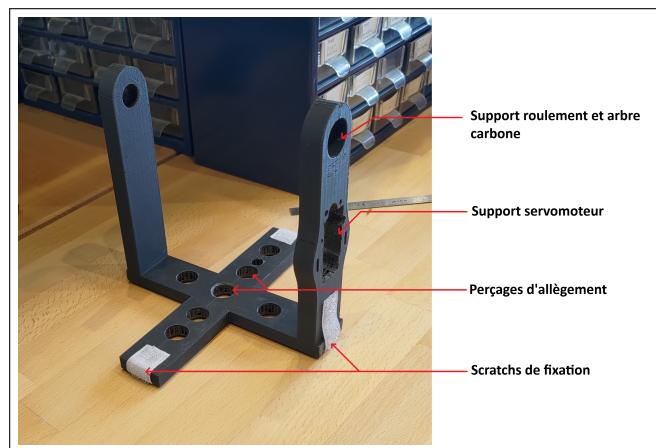


FIGURE 2 – Prototype de nacelle

Les paramètres d'impression n'ayant pas été optimisés pour respecter le critère 1, des trous ont été percés pour alléger la structure.

II.2 Implémentation des fonctions mécaniques sur la nacelle - Mathys Bourdyot

Dans le cadre de notre projet de dirigeable, nous avons pris grand soin dans l'implémentation des fonctions mécaniques et électriques afin de garantir un fonctionnement optimal et une performance de vol sans faille.

La fixation de la nacelle au dirigeable a été réalisée à l'aide de velcros. Cette solution permet de fixer n'importe quelle forme de nacelle sur le dirigeable, d'autant plus que ce dernier est facilement déformable (cf figure 3).

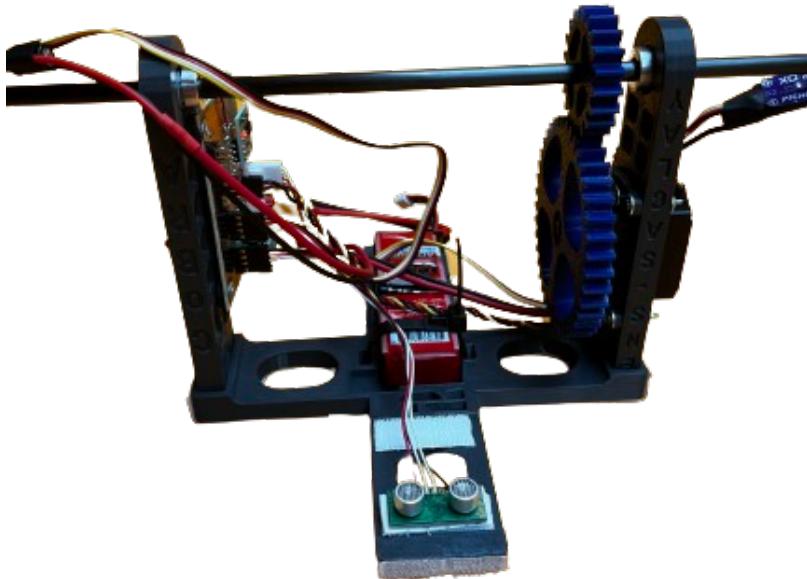


FIGURE 3 – Photographie de la base de la nacelle

Dans un second temps, nous devons placer les différents capteurs et actionneurs sur le dirigeable, ainsi que la carte de commande. Il est important de les placer de manière symétrique afin de placer le centre de gravité au centre du dirigeable. En effet, vu la disposition des hélices (bras de levier plutôt faible et pas d'hélices sur la queue), il est important de placer le centre de masse sur l'axe du dirigeable et de concentrer les masses afin de maximiser les différentes rotations autour du pivot et de minimiser les éventuelles rotations involontaires lorsque les hélices sont tournées vers le haut. C'est pourquoi l'emplacement la batterie est placée sur le centre de la nacelle, dans le même axe que le dirigeable.

En outre, la carte électronique placée sur un bras permet de minimiser le déséquilibre causé par le système servomoteurs + roues dentées, placé sur l'autre bras.

Les mises en position de la batterie, et du servomoteur sont réalisés par des liaisons prismatiques. Le maintien en position de la batterie est fait par des cerflex pour leur simplicité de mise en oeuvre. Et celui du servo par des vis, vissées dans des lumières afin de régler la translation entre les deux roues dentées.

Il faut enfin réaliser la liaison pivot de l'axe supportant les moteurs, cette liaison est assurée par le montage hyperstatique de deux roulements, un sur chaque bras monté en force. L'arrêt en translation du bras est assuré par une vis de pression (cf figure 4).

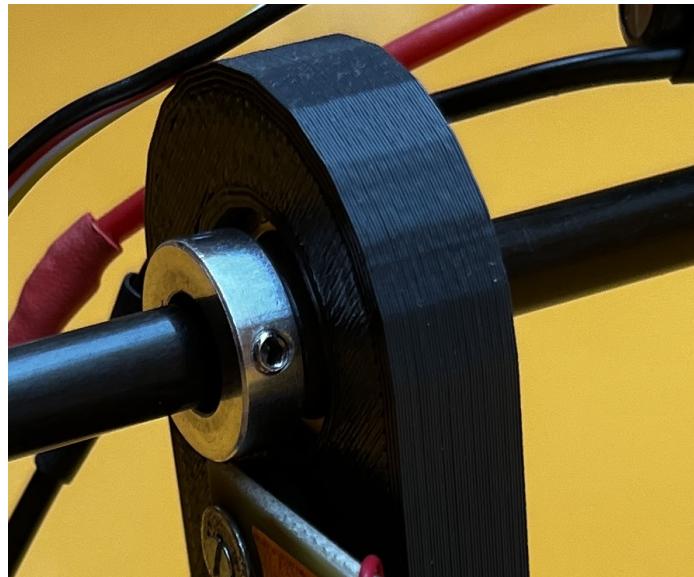


FIGURE 4 – Photographie d'un roulement et de son arrêt en translation

Enfin, pour la liaison entre les moteurs et l'axe : l'axe est mis en butée dans la carène extérieure du moteur, et monté serré à l'intérieur, puis collé (minimise la masse du montage). Ensuite, le moteur est vissé sur un cerclage supporté par 3 bras en carbone détachables. En cas de panne, cela permet de changer en toute simplicité le moteur en les déclipsant.

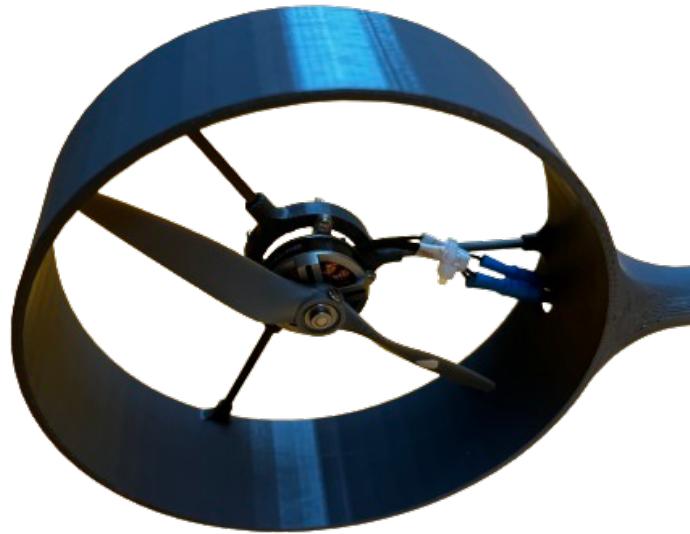


FIGURE 5 – Photographie du moteur et de sa liaison à l'axe

Nous verrons dans la partie suivante comment le fichier CAO de la nacelle est fait à l'aide de CATIA V5.

II.3 Conception finale - Jérémy Couslon, Simon Chardin

Après avoir réalisé des tests avec le prototype, beaucoup de changements ont été jugés nécessaire tel que :

1. Séparer la nacelle en 3 pièces pour pouvoir corriger de petits défauts sans devoir tout réimprimer
2. Créer des emplacements de fixations pour les divers composants de commande tels que la carte de commande, les capteurs et la batterie.
3. refaire l'emplacement du servomoteur pour pouvoir ajuster sa hauteur
4. redimensionner les supports moteurs pour diminuer le poids en étant plus adaptés à la taille des hélices.

Nous avons utilisé le logiciel Catia V5 pour réaliser la CAO de la conception finale. Nous obtenons ainsi la nacelle dans son état final :

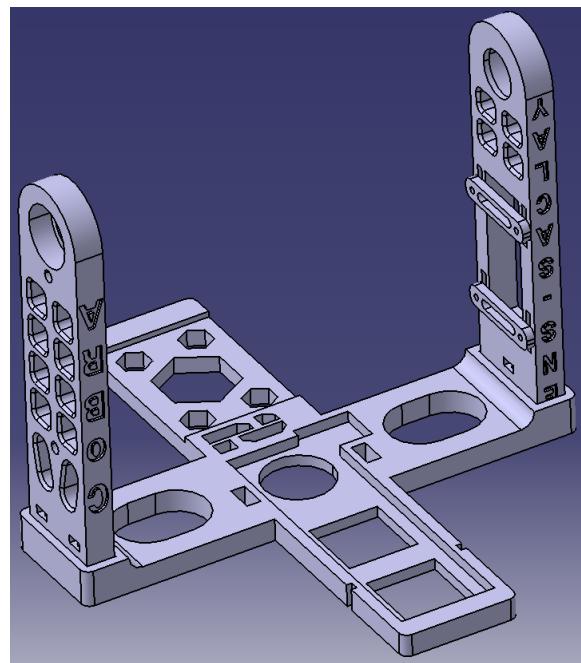


FIGURE 6 – Nacelle assemblée dans le logiciel CatiaV5

Voyons ensemble les 3 composants de cette nacelle :

1. Le premier composant est la croix de fixation sur lequel a été créé un emplacement pour la batterie et un emplacement à l'avant pour le capteur ultrason Marvelmind. Deux emplacements de fixation ont été créé pour les bras de la nacelle. Des trous ont été créé pour alléger la nacelle

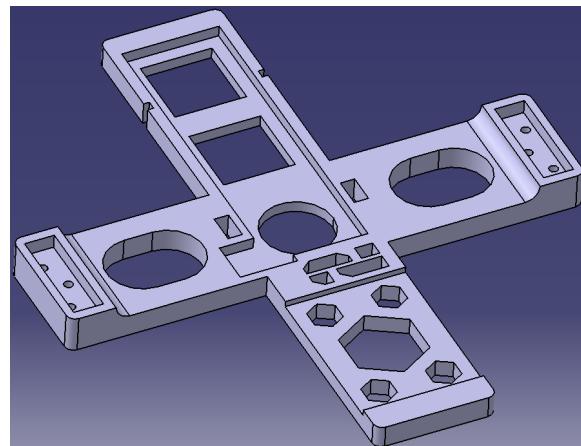


FIGURE 7 – Croix de fixation

2. Le deuxième composant est le bras gauche sur lequel est fixée le servo-moteur. Sa position peut être réglée en hauteur dans le cas où le système d'engrènement entre le servo-moteur et l'axe de fixation des moteurs change. Le maintien en place du bras sur la croix est assurée par trois vis situées sous le bras et par les écrous insérés sur les côtés du bras. Enfin "ENS-SACLAY" a été gravé des deux coté du bras dans un but d'esthétisme.

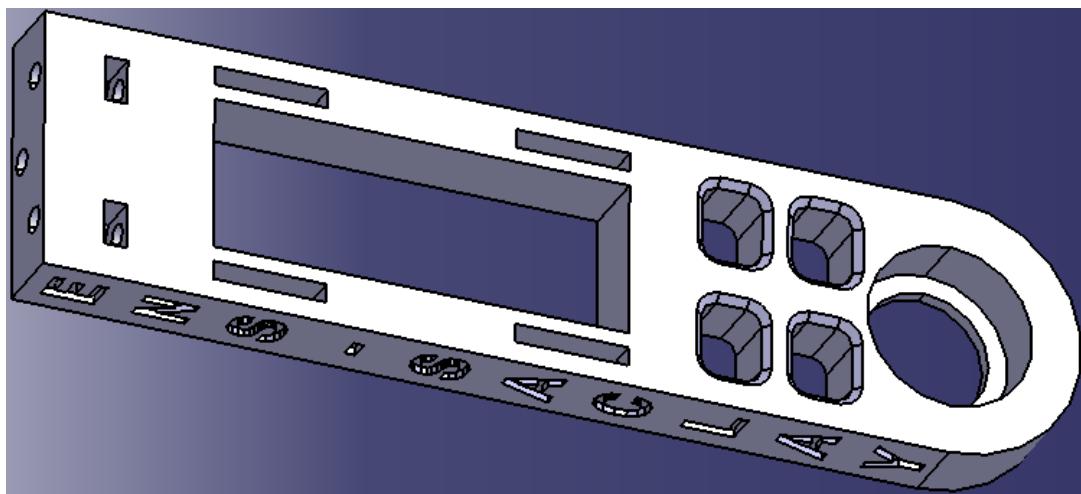


FIGURE 8 – Bras gauche

3. Le troisième composant est le bras droit sur lequel est fixée la carte de commande. Sa position est fixe et est assurée par les deux vis présentes au centre du bras. Le maintien en place du bras sur la croix est identique à celui du bras gauche. Enfin "COBRA" a été gravé des deux coté du bras dans un but d'esthétisme.

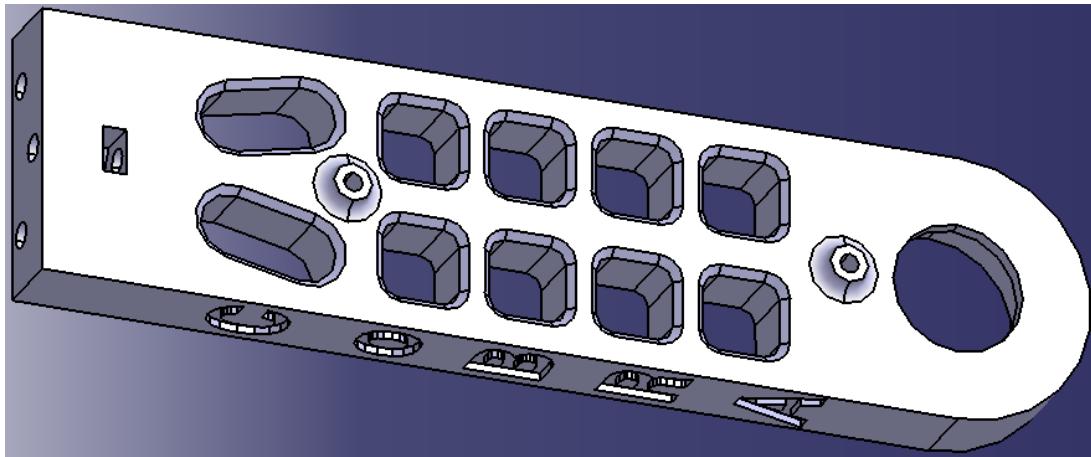


FIGURE 9 – Bras gauche

Afin de respecter l'exigence 2, les paramètres d'impression ont été choisis comme ci-dessous :

Pièce	Epaisseur de coque [mm]	Taux de remplissage (%)
Bras	2	22
Corps (croix)	1.5	17

De plus, le diamètre des cloches support moteurs a été réduit. Au final, nous obtenons en terme de masse :

Masse de la nacelle sans batterie	Masse de la batterie	Masse totale
528 g	133 g	661 g

III Électronique embarquée - Fantine Mpcko Priso

III.1 Composants à intégrer

Le cahier des charges de l'électronique à intégrer a d'abord été réalisé communément avec des membres de tous les groupes, pour réaliser un système qui convienne et s'adapte à chaque solution technique proposée. Les éléments à intégrer sont :

- un **accéléromètre** permettant la récupération de données sur la position du dirigeable
- une **Raspberry pi** permettant de gérer la commande des moteurs et d'analyser les images fournies par une caméra
- un **servopilote PWM** contrôlable en I2C avec lequel travaillera l'équipe commande et asservissement
- un **régulateur de tension** qui délivre en sortie une alimentation adéquate à la raspberry pour l'entrée d'une batterie déterminée .
- des solutions d'alimentation et de connexion pour les moteurs d'hélices du dirigeable

On choisira alors la chaîne d'énergie ci-dessous, permettant avec la même batterie d'alimenter les servomoteurs et moteurs en 5V, ainsi la raspberry en 5V, elle-même alimentant les composants en 3V3.

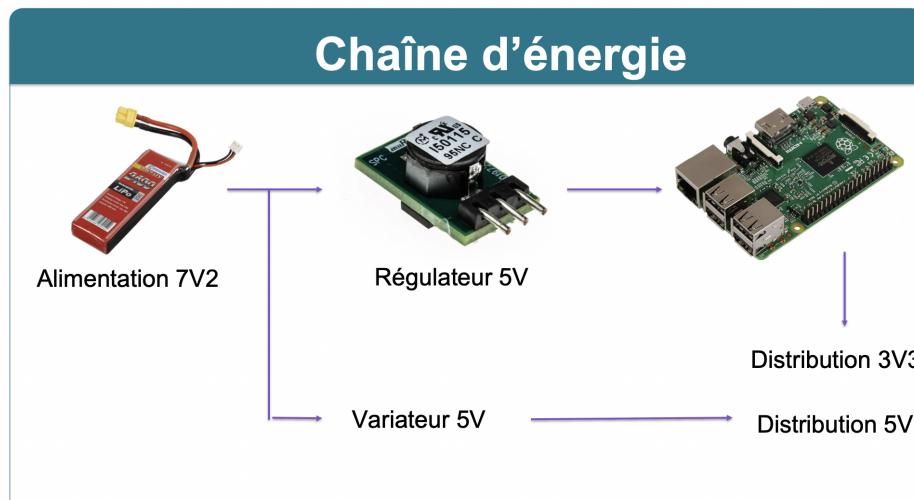


FIGURE 10 – Schéma de principe de la chaîne d'énergie

Note : Lors de la réalisation de la carte, nous avons malheureusement oublié de séparer l'arrivée de l'alimentation 7V de la batterie en 2, ce qui nous a fait perdre 1h de travail, car ce pont était indispensable pour le variateur 5V et donc la distribution de l'alimentation des moteurs.

III.2 Contraintes électriques

III.2.1 Conception théorique de la carte

Nous avons appris à l'occasion de ce défi Saphire à utiliser le logiciel EAGLE, permettant de concevoir des cartes électroniques, et donc appris à nos dépens des règles à ne

pas violer. Par exemple, nous avons fait l'erreur de ne pas assez agrandir les pistes transportant des tensions "importantes", ce qui crée un effet résistif dans les connexions et tend à échauffer la piste, ainsi qu'à mal transmettre le signal. De même, nous avions mal relié toutes les masses avec le plan de masse, nous obligeant par la suite à rajouter des fils sur la carte.

De plus, nous avons disposé par sécurité des capacités de découplages et des résistances de tirages autour des composants sensibles (le répéteur I2C, le servopilote, ...).

Comme nous n'avions accès qu'au top et au bottom de la carte (pas de couche intermédiaire car fabrication "à la main" de la carte), nous avons du faire le routage en intégrant un certain nombre de vias (connexion entre le top et le bottom d'une carte), comme on peut le voir sur la figure 10. Notons dans cette même problématique de top et de bottom que nous avons eu des difficultés à visualiser avant d'avoir la vraie carte entre les mains la position des éléments, si les pistes devaient leur arriver par le top ou le bottom selon qu'ils étaient CMS ou traversants. Enfin, grâce aux conseils des professeurs, nous avons appris à optimiser le positionnement des éléments sur la carte, selon leur fonctionnement, et gagné quelques centimètres qui, nous le verrons plus tard, sont primordiaux.

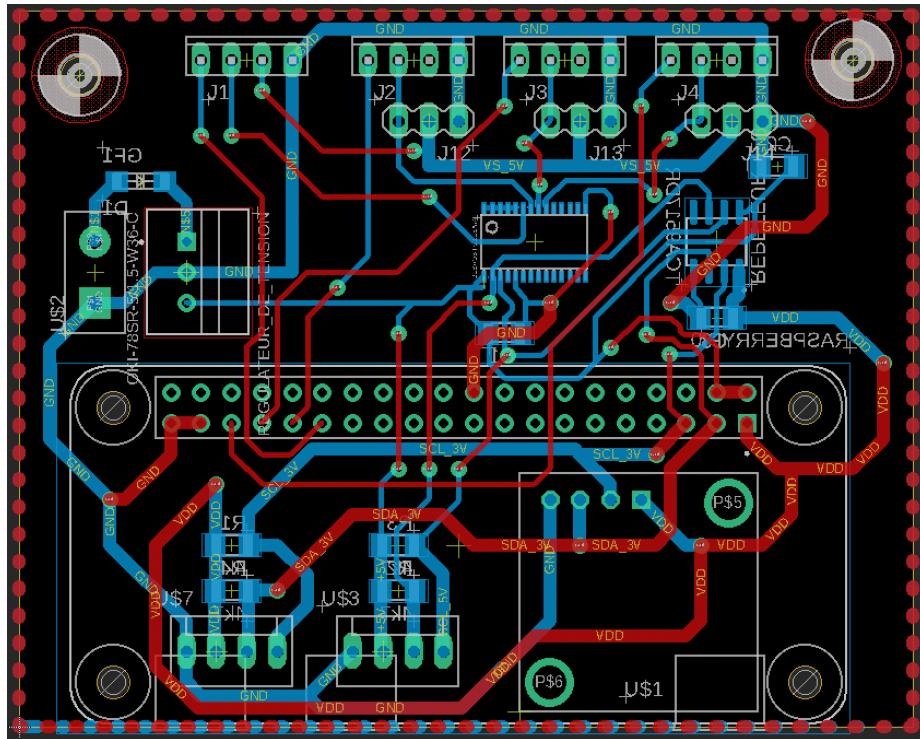


FIGURE 11 – Routage de la carte électronique sous EAGLE

III.3 Gravure de la carte

Grâce à Christophe notre technicien, nous avons pu imprimer notre carte électronique nous-même ; en protégeant une plaque avec un typon, on dissout certaines zones d'une résine photosensible disposée partout sur la carte, exposant des zones de conduction. Après traitement, on se retrouve avec un circuit gravé, qu'on doit percer pour faire passer nos vias et également insérer nos composants traversants.

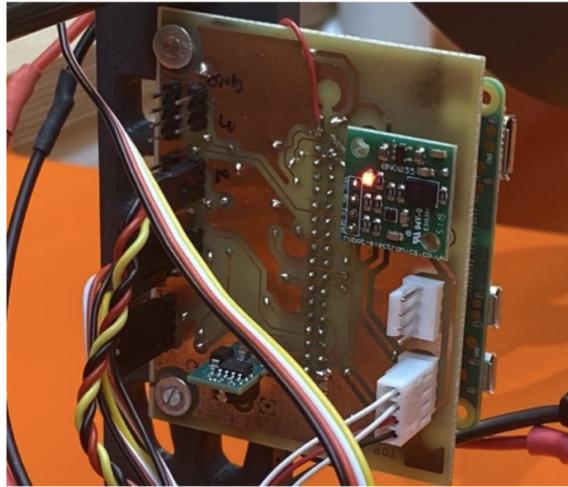


FIGURE 12 – Photo de notre carte électronique une fois les composants assemblés

Le travail de soudure sur les 3 cartes nous a ensuite pris 3 séances, de par notre manque d'expérience en la matière, mais lors des tests, hormis quelques cafouillages de masses et de courts-circuits, nous étions très fiers de notre travail.

III.4 Contraintes mécaniques

Nous avons créé la carte sur le logiciel EAGLE, avec pour objectif de limiter au maximum sa taille et son poids, deux critères très importants dans le cas précis du dirigeable. En effet, les groupes mécaniques nous ont demandé de ne pas dépasser les 70g ; pour ce faire, nous avons privilégié des composants CMS (plus petits et plus légers), et les versions les plus "light" des composants.

Nous avons également utilisé toutes les fixations possibles pour mettre la carte et fixer ses composants, notamment à cause de la présence de l'accéléromètre, dont la précision de mesure était essentielle.

Nous avons également dû tenir compte des différentes connectiques nécessitées (est-ce que la raspberry a une connectique male ou femelle ?, la batterie est-elle en XT60 ?, la caméra a-t-elle un câble assez long ?, etc...) Les différences de solution technique entre les groupes nous ont forcés à faire un travail global et en équipe avec des membres de chacun des groupes.

IV Asservissement - Nicolas Detzen

IV.1 Mise en contexte et définition

Commençons tout d'abord par définir les notations dont nous aurons besoin par la suite :

- γ : l'angle correspondant au lacet, son signe est défini sur la figure suivante
- θ : l'angle du servomoteur dans le référentiel de la nacelle
- α ($\in [0,1]$) : coefficient de proportionnalité pour la somme des vitesses de rotation des deux moteurs
- ω_{sum} : somme des vitesses de rotation des deux moteurs
- ω_{diff} : différence des vitesses de rotation des deux moteurs ($\omega_{diff} = \omega_{gauche} - \omega_{droit}$)
- $p \in [0,1]$, consigne envoyée aux moteurs, pour $p = 1$, un moteur tourne à sa vitesse maximale

Les mesures se font à l'aide :

- d'une centrale inertie retournant le roulis, le lacet et le tangage
- capteur ultrason retournant la distance entre le sol et la nacelle.

Par ailleurs, nous avons observé que l'angle de roulis du dirigeable reste stable (la nacelle reste parallèle au sol). Nous supposerons donc que la distance renvoyée par le capteur correspond directement à la mesure de la hauteur.

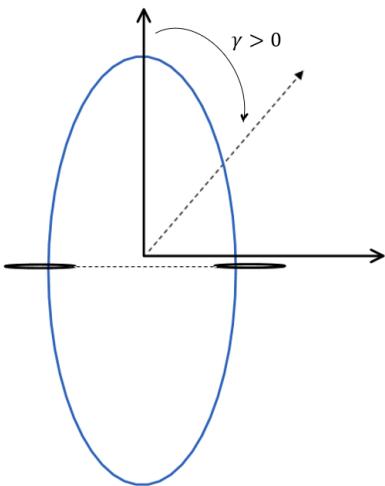


FIGURE 13 – Dirigeable vu du haut

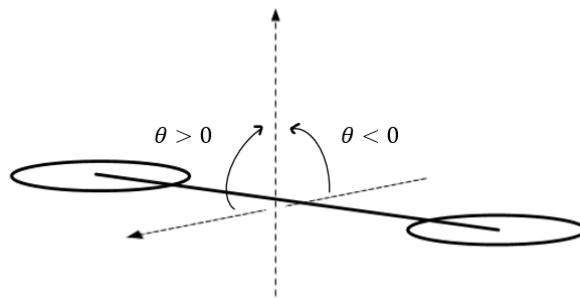


FIGURE 14 – Schéma de la nacelle pour $\theta = 90^\circ$

Le contrôle de la trajectoire du dirigeable se fera uniquement à l'aide de la vitesse de rotation des deux moteurs ainsi que de l'angle du servomoteur. La trajectoire que nous souhaitons réalisée est la suivante :

- Phase 1 - Tout droit : $\gamma = 0^\circ$, $hauteur_{min} = 100$ cm, $\theta = 20^\circ$, durée = 10 s
- Phase 2 - Rotation : $\gamma = 180^\circ$, $hauteur_{min} = 100$ cm, $\theta = 20^\circ$, durée = 10 s
- Phase 3 - Tout droit : $\gamma = 180^\circ$, $hauteur_{min} = 100$ cm, $\theta = 20^\circ$, durée = 10 s

Le parcours sera défini en python sous le format d'une liste, où un élément sera du type : [lacet, hauteur, angle servomoteur, durée].

IV.2 Stratégie d'asservissement

Pour pouvoir simplifier le contrôle de la trajectoire du dirigeable, nous allons décorréler l'asservissement du lacet de l'asservissement de la hauteur. Le maintien de la hauteur sera privilégié devant celui du lacet. La stratégie peut être représentée de la façon suivante :

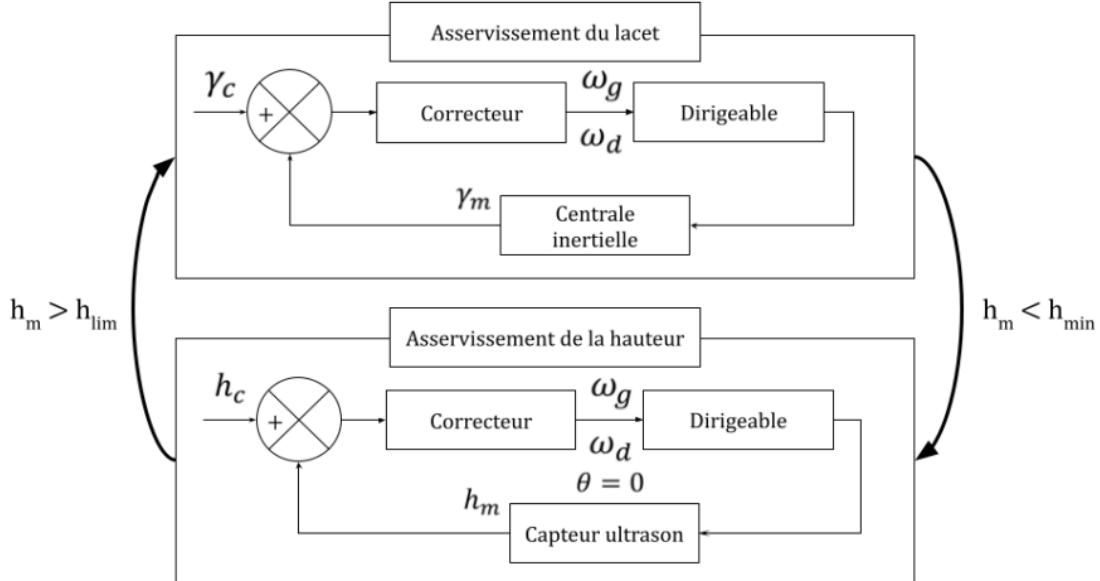


FIGURE 15 – Stratégie d'asservissement

Pour assurer une hauteur minimale à ne pas descendre en dessous, nous orientons alors le servomoteur à $\theta = 90^\circ$ afin que la poussée des moteurs soit entièrement utilisée pour contrôler la hauteur de la nacelle. Une fois que cette dernière à dépasser une hauteur limite, la correction se fera à nouveau sur le lacet.

Pour l'asservissement, nous utiliserons un correcteur PID dont le code python sera donné par la suite.

La différence de la vitesse des moteurs entre le moteur gauche et le moteur droit doit nous permettre de contrôler (avec ici un correcteur proportionnel) le lacet, suivant la manière suivante :

$$\begin{cases} \omega_{\text{diff}} = K(\gamma_{\text{consigne}} - \gamma_{\text{mesure}}) = \omega_g - \omega_d \\ \omega_{\text{sum}} = 2\alpha\omega_{\text{max}} = \omega_g + \omega_d \end{cases}$$

Nous obtenons donc finalement :

$$\begin{cases} \omega_g = \frac{\omega_{\text{sum}} + \omega_{\text{diff}}}{2} \\ \omega_d = \frac{\omega_{\text{sum}} - \omega_{\text{diff}}}{2} \end{cases}$$

Remarque : Le paramètre α nous permet ici de moduler la vitesse totale des moteurs (vitesse moyenne). Nous avons par exemple constater que pour $\alpha = 1$, le ballon se déplaçait trop rapidement. Nous avons pris après plusieurs tests $\alpha = 0.2$.

IV.3 Implémentation du code

Le code pour les capteurs et le contrôle des moteurs n'étant pas le travail de notre équipe, il ne sera pas détaillé dans le code suivant. Nous obtenons les mesures à l'aide des fonctions suivantes :

- *sensor.euler[-1]* : renvoie la mesure du lacet dans le référentiel de la nacelle
- *C_hauteur.get_data()* : renvoie la mesure de la distance entre la nacelle et le sol.
Cette distance est ici supposée égale à la hauteur

Le contrôle des moteurs et du servomoteur se fait à l'aide des fonctions suivantes :

- *moteur_gauche.commande_puissance()* : envoie de la consigne au moteur gauche
- *moteur_droit.commande_puissance()* : envoie de la consigne au moteur droit
- *servo.commande_angle()* : envoie de la commande angulaire au servomoteur

Voici l'ensemble des fonctions et classe utilisées pour réaliser l'asservissement :

```
1 class CorrecteurPID:  
2     def __init__(self, kp, ki, kd, setpoint):  
3         self.kp = kp # Terme proportionnel  
4         self.ki = ki # Terme integral  
5         self.kd = kd # Terme derive  
6         self.setpoint = setpoint # Valeur de consigne  
7         # Pris a zero ici pour les tests avec le correcteur  
8         # proportionnel  
9         self.error_prior = 0 # Erreur precedente  
10        self.integral = 0 # Terme integral  
11  
12    def compute(self, feedback):  
13        error = self.setpoint - feedback # Calcul de l'erreur actuelle  
14  
15        # Terme proportionnel  
16        proportional = self.kp * error  
17  
18        # Terme integral  
19        self.integral += error  
20        integral = self.ki * self.integral  
21  
22        # Terme derive  
23        derivative = self.kd * (error - self.error_prior)  
24        self.error_prior = error  
25  
26        # Calcul de la commande de correction  
27        correction = proportional + integral + derivative  
28  
29        return correction  
30  
31    def sequence_asservissement(parcours_inverse, alpha, ky, kyi, kdy):  
32        """  
33            Applique la strategie de commande.  
34  
35            Parametres :  
36            - parcours_inverse (liste) : liste inversee du parcours defini (la  
              liste est inversee du fait du fonctionnement de .pop())  
            - alpha (float) : coefficient utilisee pour la somme des vitesses
```

```

    des moteurs.
37 - ky (float) : coefficient du correcteur proportionnel
38 - kyi (float) : coefficient du correcteur integral
39 - kyd (float) : coefficient du corrrecteur derivee
40
41 Effectue :
42 - donne la consigne de vitesse aux moteurs
43 - donne la consigne d'angle au servomoteur
44 """
45
46 yaw_ref = sensor.euler[-1] #angle de reference = origine du repere
47 wmax = 13431 # tr/min, vitesse de rotation maximale des moteurs
    pour p = 1
48
49 while len(parcours_inverse) != 0:
50
51     if keyboard.is_pressed('esc'):
52         # Assure l'arret si l'on presse la touche echap du clavier
53         off() # fonction d'arret des moteurs (n'est pas donnee ici
                car ne presente pas d'interet majeur)
54         return None
55
56     else:
57         yaw_parcours, hauteur_limite, theta, duree =
58             parcours_inverse.pop()
59         t0 = time.time()
60
61         Correction_yaw = CorrecteurPID(ky, kiy, kdy, yaw_parcours)
62
63         servo.commande_angle(theta)
64
65         while time.time() - t0 <= duree:
66
67             try:
68                 # Mesure du lacet
69                 yaw_mesure = sensor.euler[-1] - yaw_ref
70                 yaw_liste.append(yaw_mesure)
71
72             except:
73                 yaw_mesure = yaw_liste[-1]
74
75             # Mesure de la hauteur
76             hauteur_mesure = C_hauteur.get_data()
77
78             # Si jamais la hauteur est corrigee, on ne veut pas que
79             # la duree de cette correction vienne impacter la
80             # duree de l'etape
81
82             height_corrected = False
83
84             start_height_correction = time.time()
85             while hauteur_mesure < hauteur_limite + 20:
86                 theta = 90
87                 servo.commande_angle(theta)
88                 time.sleep(0.01)
89                 p = 0.6

```

```

87         moteur_gauche.commande(p)
88         moteur_droit.commande(p)
89         hauteur_mesure = C_hauteur.get_data() # Verifie
90             hauteur apres commande
91             height_corrected = True
92         end_height_correction = time.time()
93
94         servo.commande_angle(theta)
95
96     if height_corrected:
97         # Ajuste le temps t0 pour le temps passe a corriger
98             la hauteur
99
100    t0 += (end_height_correction -
101        start_height_correction)
102
103    if yaw_parcours == 180:
104        # si l'on veut faire demi-tour, la consigne n'est plus
105            un echelon mais une rampe
106
107        yaw_consigne = (yaw_parcours / duree) *
108            (time.time() - t0)
109        Correction_yaw = CorrecteurPID(ky, kyi, kdy,
110            yaw_consigne)
111
112        wdiff = Correction_yaw.compute(yaw_mesure)
113
114        wsum = 2 * alpha * w_max
115
116        wg = (wsum + wdiff) / 2
117        wd = (wsum - wdiff) / 2
118
119        if wg > wmax:
120            wg = wmax
121
122        if wg < 0:
123            wg = 0
124
125        if wd > wmax:
126            wd = wmax
127
128        if wd < 0 :
129            wd = 0
130
131        pg = wg/wmax
132        pd = wd/wmax
133
134        # Envoi de la consigne de vitesse aux moteurs
135        moteur_gauche.commande_puissance(pg)
136        moteur_droit.commande_puissance(pd)
137        servo.commande_angle(theta)
138
139        time.sleep(0.01)
140
141 off()

```

IV.4 Résultats expérimentaux

Nous avons quelques essais en portant à bout de bras la nacelle. Nous avons fait varier le lacet entre -90 et 90 degrés et implémenté l'asservissement avec un correcteur proportionnel :

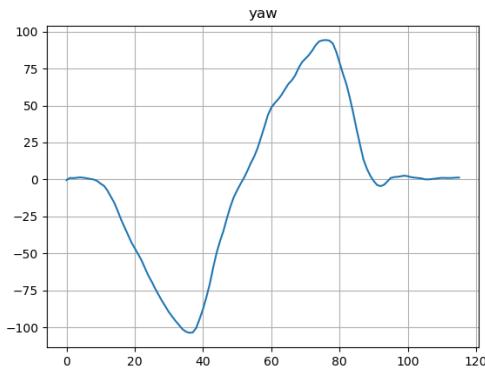


FIGURE 16 – Evolution du lacet

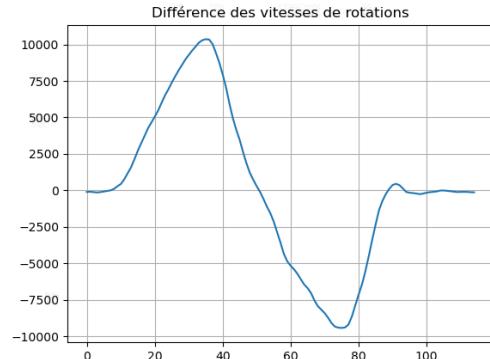


FIGURE 17 – Evolution de ω_{diff}

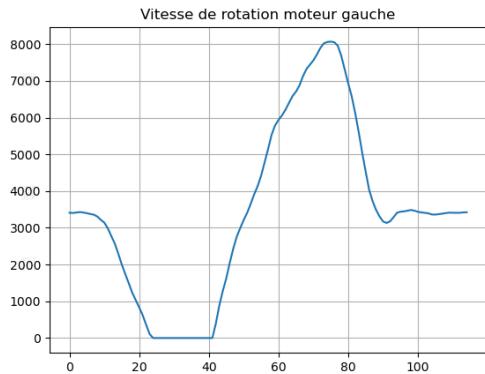


FIGURE 18 – Vitesse de rotation du moteur gauche

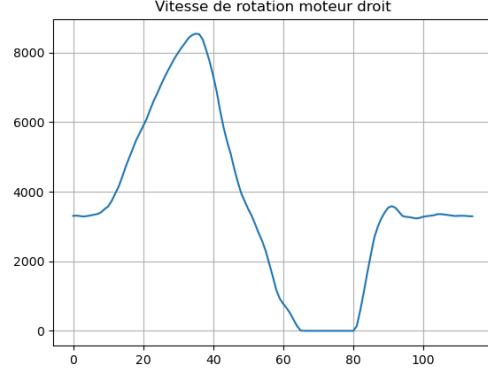


FIGURE 19 – Vitesse de rotation moteur droit

L'évolution de la correction, i.e ω_{diff} est cohérente par rapport à l'évolution du lacet. L'asservissement fonctionne. Sans modèle théorique, la détermination des coefficients du correcteur se fera expérimentalement.

Constat expérimental

L'asservissement fonctionne mais dans une certaine limite. L'utilisation d'un correcteur proportionnel ne suffit pas pour empêcher le dirigeable de dériver, l'inertie du dirigeable étant trop importante. Plusieurs solutions seraient envisageables :

- développer le modèle théorique pour pouvoir régler finement les coefficients du correcteur dérivé et intégral
- augmenter la distance entre les moteurs pour augmenter le couple et donc corriger plus rapidement

V Tests et validation - Ulysse Poulain - Ahmed Nachawati

V.1 Objectif des tests

L'objectif des tests est de vérifier le bon fonctionnement des composants électroniques, de l'asservissement et de la structure mécanique de la nacelle du dirigeable. Ces tests permettent d'identifier les éventuels dysfonctionnements et de valider les choix techniques effectués lors de la conception.

V.2 Tests des composants électroniques

Les tests des composants électroniques incluent l'évaluation des performances de l'accéléromètre, de la Raspberry Pi, du servopilote PWM, et du régulateur de tension. Chaque composant a été testé individuellement avant d'être intégré au système global.

- **Accéléromètre** : Les données de position recueillies par l'accéléromètre ont été vérifiées pour s'assurer de leur précision.
- **Raspberry Pi** : La capacité de la Raspberry Pi à gérer les commandes des moteurs et à analyser les images de la caméra a été testée en conditions réelles.
- **Servopilote PWM** : Le servopilote a été contrôlé via I2C pour vérifier la réactivité et la précision des mouvements.
- **Régulateur de tension** : Le régulateur de tension a été testé pour garantir une alimentation stable et adéquate à la Raspberry Pi et aux autres composants électriques.

V.3 Tests d'asservissement

Les tests d'asservissement se sont concentrés sur le contrôle des angles de lacet et de tangage, ainsi que sur la stabilité en vol.

- **Lacet et tangage** : Des essais en vol ont permis de vérifier les réactions du dirigeable aux variations de commandes de lacet et de tangage. Les résultats montrent une bonne évolution du lacet avec des variations cohérentes par rapport aux commandes envoyées.
- **Stabilité en vol** : Des tests de stabilité ont été réalisés pour observer le comportement du dirigeable en vol stationnaire et lors de mouvements programmés. L'asservissement proportionnel a montré ses limites, nécessitant des ajustements pour réduire l'inertie du dirigeable et améliorer la réactivité aux commandes.

V.4 Tests de la structure mécanique

Les tests de la structure mécanique de la nacelle sont cruciaux pour garantir la fiabilité et la performance du dirigeable. Ces tests visent à valider la rigidité, la légèreté, et la capacité de la structure à supporter les composants électroniques et les moteurs en conditions réelles.

V.4.1 Test de rigidité et de résistance

La rigidité et la résistance de la structure ont été évaluées par plusieurs méthodes :

- **Essais de flexion** : La structure a été soumise à des forces de flexion pour simuler les conditions réelles de vol et les impacts potentiels. Ces essais ont permis de vérifier que la nacelle ne se déforme pas sous des charges normales.
- **Tests de vibrations** : Des tests de vibrations ont été réalisés pour s'assurer que la structure peut absorber les oscillations et les secousses dues aux mouvements du dirigeable sans se détériorer. Cela inclut des tests en vol avec des mouvements brusques et des changements rapides de direction.
- **Essais de charge statique** : La structure a été chargée statiquement avec un poids supérieur à celui des composants embarqués pour vérifier qu'elle peut supporter une surcharge ponctuelle sans déformation permanente.

V.4.2 Évaluation du poids

L'évaluation du poids de la structure a été réalisée pour s'assurer qu'elle est suffisamment légère pour permettre une bonne flottabilité tout en étant assez robuste pour supporter tous les composants nécessaires.

- **Pesée de la structure** : La structure a été pesée avec et sans les composants électroniques pour vérifier que le poids total reste dans les limites acceptables pour garantir la flottabilité du dirigeable.
- **Optimisation du design** : Des ajustements ont été faits pour alléger la structure, comme l'utilisation de matériaux composites ou l'optimisation de la répartition des charges.

V.4.3 Tests d'assemblage et de montage

Les tests d'assemblage et de montage des composants électroniques et des moteurs sur la structure mécanique sont essentiels pour vérifier la facilité d'assemblage et la robustesse des fixations.

- **Montage des composants** : Chaque composant (carte de commande, capteurs, batterie, moteurs) a été monté sur la nacelle pour vérifier la compatibilité et la solidité des fixations. Les positions ont été ajustées pour optimiser l'équilibre et la répartition du poids.
- **Tests de démontage** : Les tests de démontage ont permis de s'assurer que chaque composant peut être facilement retiré et remplacé sans endommager la structure. Cela est crucial pour les opérations de maintenance et les mises à jour futures.

V.4.4 Tests en conditions réelles

Des tests en conditions réelles ont été effectués pour simuler les environnements et les situations auxquels le dirigeable pourrait être confronté.

- **Vols d'essai** : Des vols d'essai ont été réalisés pour observer le comportement de la nacelle en conditions de vol réelles. Ces tests incluaient des manœuvres de montée, de descente, et des déplacements latéraux pour évaluer la réaction de la structure aux commandes.

- **Simulation de pannes** : Des scénarios de panne ont été simulés pour vérifier la robustesse de la structure en cas de défaillance d'un ou plusieurs composants. Ces tests ont permis de s'assurer que la nacelle peut continuer à fonctionner ou atterrir en toute sécurité même en cas de problème technique.

V.4.5 Conclusion des tests de la structure mécanique

Les tests de la structure mécanique ont globalement validé la solidité, la légèreté, et la facilité d'assemblage de la nacelle. Les essais en conditions réelles ont montré que la structure peut supporter les forces et les contraintes rencontrées en vol sans compromettre la stabilité ou la performance du dirigeable. Quelques ajustements mineurs ont été identifiés pour optimiser encore davantage la structure, mais les résultats sont globalement très positifs. La structure est prête pour une utilisation intensive et pour des environnements de test variés.

VI Conclusion et perspectives

Après plusieurs expérimentations, nous avons pu observer que :

- La structure de la nacelle permet l'inclusion de tous les composants nécessaires
- Grâce à sa masse suffisamment faible, la flottabilité est assurée sans apport d'énergie (si le dirigeable est correctement gonflé).
- Les vitesses relatives des hélices permettent une bonne orientation du dirigeable et une compensation de sa chute selon le mode de consigne.
- L'inertie importante du dirigeable rend son asservissement par un simple correcteur proportionnel limité, notamment en raison du retard créé.

L'architecture de la nacelle permet le contrôle d'une trajectoire simple. Cependant, le contrôle de l'ensemble des degrés de liberté sans un rotor de queue complexifie l'asservissement. Le contrôle de la position du dirigeable dépend en outre de la vitesse de rotation des moteurs ainsi que de l'angle du servomoteur. Un asservissement plus complexe nécessiterait de développer le modèle dynamique du dirigeable ou l'ajout d'un rotor de queue.

Remerciements

Nous tenons à remercier nos professeurs encadrants qui nous ont accompagnés tout au long de ce travail. Leur aide et leurs conseils nous ont été d'une grande importance pour l'aboutissement de ce travail.