



Master Systèmes Dynamiques et Signaux

Mémoire

Robotisation d'un dirigeable dédié à la captation scénique

Auteur :

M. Thibault LANGLARD

Jury :

Pr. L. HARDOUIN

Pr. L. JAULIN

Version du
25 août 2023

Remerciements

Je tiens à remercier mon tuteur Julien Serres pour m'avoir donné l'opportunité de travailler sur le projet NOVABOT mais aussi d'avoir élargi ma culture scientifique. Je voudrais également remercier mes co-encadrants, Youssra Mansar pour sa disponibilité et ses conseils, ainsi que François Bernier pour son expertise et ses réponses à mes besoins au quotidien. Je remercie aussi mon collègue Mateo Cortes Duarte, stagiaire de 2nd année École Nationale Supérieure de l'Électronique et de ses Applications (ENSEA), pour le travail qu'il a effectué et l'assistance qu'il m'a apportée. Enfin, je tiens également à remercier Julie Elie pour son écoute, ses conseils et son support.

Table des matières

Introduction	1
1 Le dirigeable ID-Fly dans son état actuel	3
1.1 Enveloppe et structure	3
1.2 Bilan des masses et forces exercées	5
1.3 Électronique embarquée	7
1.4 État de l'art des minidirigeables	8
1.4.1 Projets similaires	9
1.5 Travaux connexes	14
2 Avancement du projet : automatisation	17
2.1 Caméra Luxonis OAK-D-IoT-40	17
2.2 Les tentatives d'installation de ROS sur Raspberry Pi zero W/2W	19
2.2.1 Sur Raspberry Pi Zero W	20
2.2.2 Sur Raspberry Pi Zero 2W	21
2.3 Modélisation de la dynamique du minidirigeable ID-Fly	23
2.4 Simulation	26
2.5 Contrôle	27
2.5.1 Objectifs du contrôle	27
2.5.2 Algorithme de contrôle	28
Conclusion	33

Table des figures

1.1	Enveloppe du minidirigeable ID-Fly.	4
1.2	Représentation schématique du minidirigeable ID-Fly.	4
1.3	Structure du minidirigeable ID-Fly avant et après mon stage	5
1.4	Bilan des forces appliquées au système volant.	6
1.5	Bilan des masses utiles du dirigeable. D’après [17].	7
1.6	Raspberry Pi Zero W avec convertisseur, shield, drivers et caméra. D’après [17].	8
1.7	Minidirigeable de Zufferey et al. [29].	10
1.8	Nacelle du minidirigeable de Zhang [28].	11
1.9	Nacelle en impression 3D du minidirigeable [6]	12
1.10	Placements possibles pour la nacelle [6] : centrée (a) et excentrée (b). C’est finalement la configuration (b) qui a été retenue.	12
1.11	Nacelle du minidirigeable de Gonzalez et al. [5].	13
1.12	Schéma du dirigeable d’Ikeda et al. [10]	13
1.13	Fonctionnement d’un micro-souffleur [10]	14
1.14	Première version de GT-MAB ¹ de Georgia Tech, USA [21]	14
1.15	Nacelle du minidirigeable GT-MAB [26]	15
1.16	Dirigeable propulsé au moyen de nageoires [14]	16
2.1	Caméra Luxonis OAK-D-IoT-40. Taille : 60 mm de large, hauteur 45 mm, largeur 25 mm ; masse : 45,5 g sur sa documentation technique.	18
2.2	Mesure d’échauffement de la caméra Luxonis au bout de 20 minutes.	18
2.3	Nouvelle batterie avec pièces de montage	20

1. Georgia Tech Miniature Autonomous Blimp

2.4	Projet de communication avec la Rasperry Pi par ROS.	20
2.5	Illustration d'un moment généré par le dirigeable penché. D'après [20]. . .	24
2.6	Suivi du minidirigeable dans une translation suivant z.	25
2.7	Scène de simulation sous v-rep.	26
2.8	Communication de la simulation avec ROS	27
2.9	Erreur de distance au cours du temps avec un contrôle par PID ²	30
2.10	Erreur de cap au cours du temps avec un contrôle par PID.	31

List of acronyms

ISM	Institut des Sciences du Mouvement - Etienne-Jules Marey (CNRS/Aix Marseille Université, ISM UMR7287)
LESA	Laboratoire d'Etudes en Sciences des Arts (Aix Marseille Université, LESA EA 3274)
PFE	Projet de Fin d'Études
PID	contrôleur Proportionnel Intégral Dérivé
PD	contrôleur Proportionnel Dérivé
ROS	Robot Operating System
ID-Fab	Innovation, Fabrication-Design
AMU	Aix Marseille Université
SLAM	Simultaneous Localisation And Mapping
UAV	Unmanned Aerial Vehicle
LTAR	Lighter Than Air Robot
Li-Po	Lithium-Polymer
RGBD	Red Green Blue Distance
GT-MAB	Georgia Tech Miniature Autonomous Blimp

Introduction

Ce PFE³ s'inscrit dans le projet doctoral de Youssra Mansar, codirigé par Julien Serres et Louis Dieuzayde, ce dernier étant maître de conférences en esthétique théâtrale au LESA⁴. Son sujet de thèse s'intitule "Acteur et drone. Dialectique du corps en scène et de son image fragmentée" [11]. Il consiste en un travail de scène mettant en interaction un comédien et un minidirigeable robotisé qui le filme et retransmet l'image sur un écran en temps réel. Ce dialogue a pour but d'étudier l'acteur en tant que sujet, de questionner sur son rapport au texte et à la diction. Dans cette idée, le texte énoncé ici est tiré de l'œuvre de Valère Novarina, auteur contemporain de théâtre dont le travail se concentre sur la question de la parole chez l'humain. Selon lui, "Faudra un jour qu'un acteur livre son corps vivant à la médecine, qu'on ouvre, qu'on sache ce qui se passe dedans, quand ça joue." Ainsi, ce projet a pour but de proposer une expérience, une sorte de dissection à distance du comédien, dont le drone serait le scalpel. Pour ce travail, un dirigeable miniaturisé présente plusieurs avantages non négligeables par rapport à un drone classique : il est notamment moins dangereux en termes d'interactions avec des humains et de vols en intérieur, et de plus il est bien plus silencieux, ce qui permet d'entendre le texte énoncé.

Mes objectifs, pour la réalisation du projet NOVABOT, sont multiples. Le principal est sa robotisation : je devais rendre le minidirigeable capable de se déplacer de manière autonome durant une partie de la représentation. Cela inclut repérer l'acteur, se rendre à sa position et le suivre en se maintenant à une distance prédéfinie (1 m). Pour cette tâche, j'avais à ma disposition une caméra intelligente Luxonis. Je devrais également étudier le minidirigeable pour trouver des moyens de l'alléger, tout en revoyant le design de certaines pièces, trop fragiles ou instables. En effet, l'objectif final serait de le miniaturiser plus encore, pour atteindre une masse de 100 grammes contre environ 600 grammes actuellement. Cela correspondrait à une enveloppe de 100 litres (minidirigeable NOVABOT), contre 600 litres actuellement (minidirigeable ID-Fly). Enfin, même si ce dirigeable est un prototype destiné à un usage de recherche, il est important de le rendre reproductible, notamment en raison de sa fragilité. Je devrais donc faire en sorte qu'il puisse être réparé et refabriqué le

3. Projet de Fin d'Études

4. Laboratoire d'Études en Sciences des Arts (Aix Marseille Université, LESA EA 3274)

plus facilement possible.

On peut également mentionner une mission annexe, qui est de donner un maximum de visibilité au projet en tant qu'œuvre artistique. Concrètement, cela veut dire participer à des représentations, des périodes de travail théâtral appelées "résidences", ou encore des événements de présentations de projets, qu'ils soient artistiques ou scientifiques. Dans cette idée, ce projet est en open-source, à la fois pour la recherche et la formation, et son dépôt git est public.

Chapitre 1

Le dirigeable ID-Fly dans son état actuel

Cette partie vise à présenter le minidirigeable ID-Fly, à décrire son état au début du projet. Cela correspond presque intégralement à son état à la fin de mon stage car, bien que des avancées aient été réalisées durant cette période, elles sont arrivées trop tardivement pour pouvoir être testées sur le minidirigeable. Malheureusement, non seulement ce dernier est très fragile et unique, mais il est aussi indispensable pour les représentations du projet NOVABOT tout au long de la thèse de Youssra Mansar. Il a donc été décidé de ne pas modifier son design ou ses programmes de peur de le rendre non fonctionnel. Néanmoins, à l'issue de mon PFE, ID-Fab sera en mesure de confectionner un clone de meilleure qualité l'année prochaine, car nous avons reçu les nouvelles enveloppes de Windreiter UG en argent Tritax. L'argent Tritax est fortement imperméable à l'hélium.

1.1 Enveloppe et structure

La majeure partie du volume du dirigeable est occupée par son enveloppe. Celle-ci est en polypropylène avec une épaisseur de 180 μm . Il s'agit d'un ellipsoïde de révolution selon l'axe x, dont les diamètres sont respectivement de 2 mètres et de 0,75 mètre (Fig. 1.1). Elle dispose de trois ailerons à l'arrière. Cette enveloppe est remplie d'hélium avec une capacité de 600 litres.

Cette enveloppe est potentiellement le point le plus problématique si l'on souhaite reproduire le dirigeable. En effet, elle avait été faite aux États-Unis, mais son fabricant ne répond plus. C'est d'autant plus problématique que l'enveloppe est fragile et déjà réparée plusieurs fois, ce qui finit par impacter ses performances (fuites d'hélium). Une mission de mon travail fut de trouver un autre fabricant en Europe pour remédier à ce problème. Windreiter UG a parfaitement répondu à nos attentes, il s'agit d'une entreprise allemande

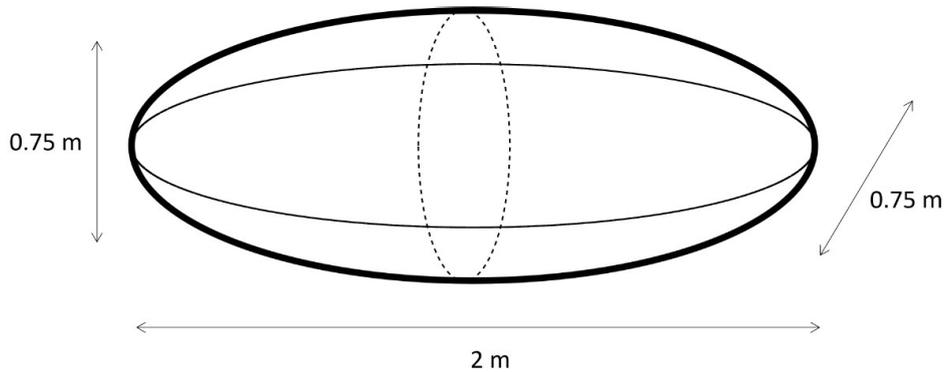


FIGURE 1.1 – Enveloppe du minidirigeable ID-Fly.

capable de fabriquer sur mesure une enveloppe de même forme, de dimensions 2,06 mètres par 0,82 mètre pour un volume de 737 litres. Cette enveloppe sera réalisée en argent Tritax.

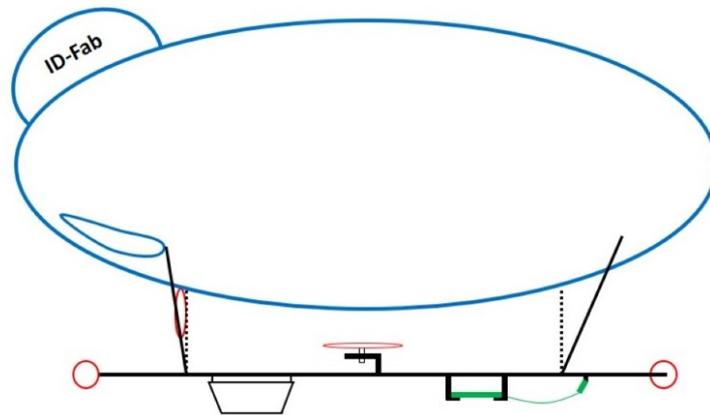


FIGURE 1.2 – Représentation schématique du minidirigeable ID-Fly.

L'enveloppe possède quatre attaches, une pour chaque tige permettant de la relier à la tige principale disposée selon l'axe x , et supportant les autres pièces de la structure (Fig. 1.2). Actuellement, la connectique est réalisée par de simples câbles placés le long de la tige et maintenus par des morceaux de gaine thermorétractable non chauffés. Outre ces câbles, les éléments placés sur cette tige sont une carte Raspberry Pi et sa caméra, quatre moteurs et une nacelle. Chacun de ces éléments possède un support pour le faire tenir, qui a été réalisé par impression 3D de résine Rigid 4000 durcie aux rayons ultraviolets à partir d'une imprimante Form 3+ de Formlabs (Fig. 1.3). Toutefois, ces éléments ont été pensés pour être le plus légers possible, parfois au détriment de la solidité. Ainsi, il est fréquent

qu'une pièce casse, même en prenant des précautions lorsqu'on manipule le dirigeable. De plus, il arrive aux pièces de glisser. Ce problème devrait être réglé en ajoutant un trou aux pièces et à la tige principale pour permettre d'ajouter une goupille.



(a) ancienne structure



(b) nouvelle structure

FIGURE 1.3 – Structure du minidirigeable ID-Fly avant et après mon stage

Le minidirigeable dispose de quatre moteurs à courant continu (Ref. Yiqigou 4Pcs RC Quadcopter A-B Moteur Positif et négatif 720 Drone Moteur Accessoires Convient pour Syma X5 / X5C / M68). Le premier est placé au centre de la tige pour exercer sa poussée selon l'axe z et ainsi contrôler l'altitude, le second à l'arrière du dirigeable pour exercer sa poussée selon l'axe x et contrôler le mouvement longitudinal, et les deux derniers aux extrémités de la tige pour contrôler le cap du robot. Leurs hélices ont été également créées spécialement par impression 3D. Celles des deux premiers moteurs sont des hélices à quatre pales avec un angle de calage de 15° , tandis que les deux latérales sont des hélices à trois pales avec un angle de calage variant.

Enfin, on trouve également une nacelle sur le dirigeable. Cette nacelle sert à porter les batteries, ainsi que d'éventuels poids pour ajuster la flottabilité du dirigeable. Elle sera retirée et remplacée par des supports pour une nouvelle batterie.

1.2 Bilan des masses et forces exercées

Pour pouvoir avoir un dirigeable fonctionnel, on doit s'assurer qu'il sera bien capable de flotter, c'est-à-dire de contrebalancer son poids avec la poussée d'Archimède (Fig. 1.4). Pour cela, on recherche sa masse maximale en considérant un minidirigeable composé d'une enveloppe de 600 L remplie d'hélium, et du reste de la structure. Dans ce cas, les forces s'appliquant au dirigeable sont son poids et la poussée d'Archimède. On pose alors les masses volumiques de l'hélium et de l'air à température ambiante (20°C) et pression atmosphérique standard (1 atm), ainsi que la masse de l'enveloppe :

$$\rho_{air} = 1,204 \text{ kg/m}^3$$

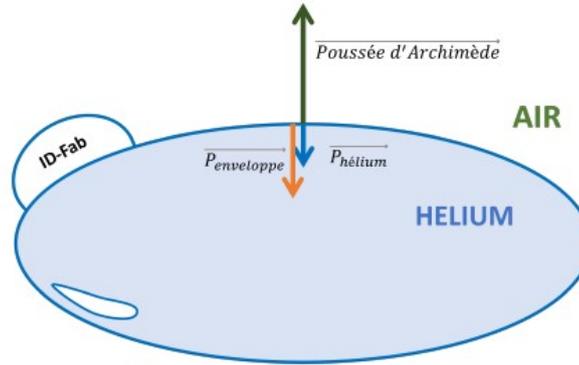


FIGURE 1.4 – Bilan des forces appliquées au système volant.

$$\rho_{helium} = 0,178 \text{ kg/m}^3$$

$$m_{enveloppe} = 0,300 \text{ kg}$$

$$g = 9,81 \text{ N/kg}$$

$$V = 600 \text{ L}$$

La masse volumique des gaz dépendra de la température du lieu d'expérimentation ainsi que de son altitude. C'est pour cela que nous avons recours à des lestes pour annuler la flottabilité du minidirigeable avant chaque représentation ou expérimentation.

En appliquant le principe fondamental de la dynamique suivant l'axe vertical z , on obtient alors :

$$a_z \cdot m_{dirigeable} = -g \cdot m_{dirigeable} + g \cdot m_{airdplac} \quad (1.1)$$

Notre but est d'avoir une masse de dirigeable telle que nous sommes à l'équilibre, c'est-à-dire avec une accélération nulle ou flottabilité nulle. On peut donc poser $a_z = 0$, d'où :

$$m_{dirigeable} + m_{airdplac} = 0 \quad (1.2)$$

Il vient :

$$m_{enveloppe} + m_{helium} + m_{utile} - m_{airdplac} = 0 \quad (1.3)$$

Avec m_{utile} la masse du dirigeable hors enveloppe et hélium, correspondant à la structure et à l'électronique embarquée. On peut alors poser :

$$m_{utile} = V \cdot \rho_{airdplac} - m_{enveloppe} - V \cdot \rho_{helium} \quad (1.4)$$

On obtient :

$$m_{utile} = 0,320 \text{ kg} \quad (1.5)$$

Par sécurité et étant donné que la masse de l'enveloppe a été mesurée de façon approximative et non vérifiée, on posera par la suite $m_{utile} = 300 \text{ g}$.

On en déduit que l'enveloppe remplie d'hélium est capable de soulever une masse de 300 grammes maximum. On effectue ensuite un bilan des masses utiles du dirigeable (Fig. 1.5) :

Matériel	nacelle (moteurs compris)	panier	batterie	Raspberry + driver	autres struc- tures	Total
Poids (grammes)	110	35	40	41	20	246

FIGURE 1.5 – Bilan des masses utiles du dirigeable. D'après [17].

La composition actuelle du dirigeable est donc bien satisfaisante. En situation de représentation théâtrale, on souhaite atteindre le point d'équilibre, ce qui est effectué d'une part en ne remplissant pas totalement l'enveloppe, et d'autre part en ajoutant des petits poids au dirigeable, dans la nacelle et le long de la tige.

1.3 Électronique embarquée

Le minidirigeable possède un ordinateur embarqué en la présence d'une Raspberry Pi Zero W pesant 9 grammes, avec Debian 10 d'installé en système d'exploitation sur une carte micro-SD de 128 Go. Cette carte est alimentée par une batterie 1800 mAh 3,7 V d'une cellule, ce qui suffit pour une représentation d'environ 20 minutes (Fig. 1.6). Cependant, la Raspberry Pi Zero W ayant une consommation de 2 A pour 5 V, on utilise un système Lipo SHIM pour effectuer l'élévation de tension d'alimentation. De plus, le modèle Zero W n'est pas prévu pour diriger quatre moteurs à la fois, ce qui doit être fait

pour commander le dirigeable. Un shield moteur lui a donc été ajouté avec deux drivers L298P, chacun permettant de contrôler deux moteurs, avec un hacheur quatre quadrants pour permettre le contrôle dans les deux sens.

Enfin, la carte est également reliée par un connecteur CSI à une caméra pour Raspberry Pi, d'une masse de 3 grammes, avec un programme python pour envoyer l'image sur une page web en temps réel. Cette page est sur un serveur web créé au démarrage de la Raspberry Pi, et permet également de contrôler le dirigeable à l'aide d'une manette.

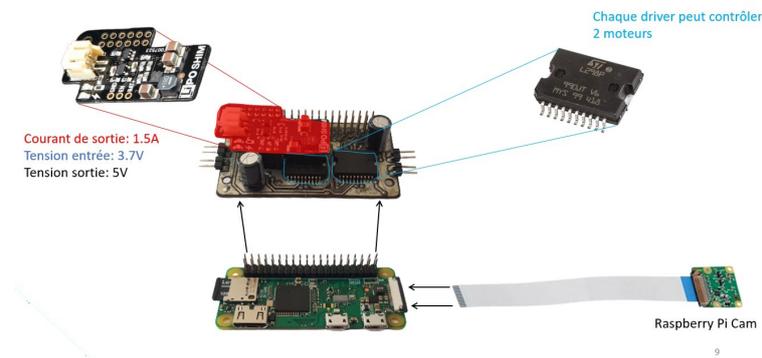


FIGURE 1.6 – Raspberry Pi Zero W avec convertisseur, shield, drivers et caméra. D'après [17].

1.4 État de l'art des minidirigeables

Les dernières années ont vu une avancée technologique particulièrement rapide dans plusieurs domaines, dont celui des drones aériens, ou UAV¹. Malgré cela, des problématiques les concernant n'ont pas encore été résolues, notamment concernant leur autonomie et la sécurité des personnes autour. C'est là qu'interviennent les dirigeables, aussi appelés LTAR². Ces drones présentent plusieurs avantages : notamment, ils nécessitent moins d'énergie pour maintenir leur altitude puisqu'ils exploitent la poussée d'Archimède, et leur taille permet à un observateur de les repérer très facilement et donc de minimiser les risques d'accident. En plus de cela, ils possèdent la mobilité de drones aériens et peuvent donc facilement explorer des zones considérées inaccessibles par voie terrestre.

Parmi ces dirigeables, on appelle minidirigeables les dirigeables dont l'enveloppe a une envergure inférieure à 2 mètres. Notre projet NOVABOT est à la limite de cette catégorie, puisqu'il mesure 2 mètres de longueur. En considérant les différents travaux effectués autour

1. Unmanned Aerial Vehicle
2. Lighter Than Air Robot

de ces robots, nous pouvons considérer qu'il y a trois points à toujours prendre en compte lors de la réalisation d'un minidirigeable. Premièrement, un LTAR doit avoir une masse particulièrement faible pour pouvoir flotter dans l'air, ce qui entraîne des contraintes sur le matériel qu'il aura embarqué. Ensuite, sa dynamique est particulière en raison de sa forme et de son inertie. Enfin, dans le cas d'un contrôle autonome, non seulement les moteurs peuvent être placés à des endroits très différents selon les modèles, mais le dirigeable est aussi soumis à des variables telles que la pression de l'air, ses mouvements et sa température. Cette partie vise à présenter différents projets de minidirigeables et différentes solutions pour ces thématiques et pour d'autres sujets jugés pertinents.

1.4.1 Projets similaires

Tout d'abord, on peut citer le projet d'Oh et al. [12], un minidirigeable avec une enveloppe longue de 2 mètres pour une masse de 1,2 kilogramme. Ce drone est pensé pour une utilisation en intérieur et a pour vocation d'avoir des images projetées sur lui. Il dispose de six capteurs à ultrasons pour évaluer les distances aux obstacles, d'une batterie, d'un micro-contrôleur et de six servomoteurs. Ces moteurs peuvent contrôler les translations selon les trois axes, ainsi que le lacet et le roulis. Le robot a donc 5 degrés de liberté, et son contrôle est un PD³ pour chacun de ces degrés. On a donc un système avec cinq erreurs pour cinq commandes indépendantes, car deux moteurs s'occupent ensemble du mouvement longitudinal. Enfin, afin de projeter des images, le minidirigeable est muni d'un marqueur infrarouge. Il est ensuite repéré par une caméra à infrarouges reliée à un ordinateur qui, d'après l'état du dirigeable, en déduit en temps réel la transformation à appliquer à l'image projetée pour qu'elle n'apparaisse pas distordue.

Zufferey et al. [29] ont, eux, conçu un minidirigeable d'une enveloppe de 110 x 60 x 60 cm et possédant trois moteurs : un pour l'altitude, un pour le mouvement longitudinal et un pour le lacet. Ses capteurs sont une caméra, un anémomètre, un capteur d'altitude et un gyroscope piézoélectrique, le tout relié à un microcontrôleur 8 bits avec un module bluetooth et alimenté par une batterie Li-Po⁴ 1200 mAh (Fig. 1.7). Il se repère avec ses informations visuelles dans une pièce dont les murs ont des motifs spécifiques, grâce à un réseau de neurones en deep learning. Il dispose également d'une autonomie de 2 à 3 heures, ce qui est assez long pour ce type de robot.

Le minidirigeable développé par Ohata et al. [13] possède des dimensions assez similaires à celles de notre minidirigeable : il mesure 170 x 60 x 60 cm pour une masse de 300 g enveloppe comprise (178 g si non comprise), mais n'utilise que trois moteurs et ne peut donc contrôler que son mouvement longitudinal, son altitude et son lacet. De plus, il est

3. contrôleur Proportionnel Dérivé

4. Lithium-Polymer

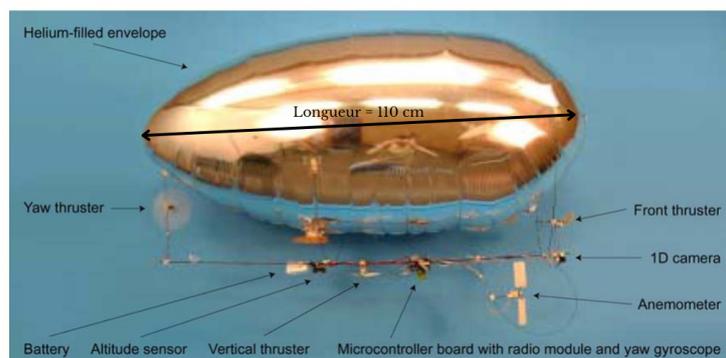


FIGURE 1.7 – Minidirigeable de Zufferey et al. [29].

téléopéré par internet, comme ce qui est fait avec NOVABOT actuellement. Un contrôleur PD a été testé en simulation avec des résultats satisfaisants, mais n'a pas été implémenté sur le robot réel.

L'équipe de Gonzalez [4] a abordé le développement d'un minidirigeable d'un autre point de vue et est partie d'un modèle trouvable dans le commerce. Le modèle sélectionné mesure 132 x 94 cm et possède, comme le précédent, trois moteurs pour contrôler l'altitude, le mouvement longitudinal et le lacet. L'équipe lui a ajouté deux capteurs ultrasoniques pour détecter les obstacles devant et sous le drone, une batterie et une carte électronique personnalisée pour contrôler les moteurs et assurer une communication avec un PC. Le contrôle de ce minidirigeable s'effectue soit par PID soit par logique floue, à ceci près que les mesures des capteurs sont corrigées avec un filtre de Kalman. La logique floue est une méthode pour aborder un problème en définissant des variables qui seraient normalement considérées comme des booléens, mais en leur attribuant une valeur entre 1 et 0. Ici, il s'agit d'une suite de if-then pour déterminer les commandes à envoyer aux moteurs en fonction de la vitesse du robot et de l'erreur de sa position par rapport à un objectif. Leurs expériences ont montré que le contrôleur PID obtenait des résultats légèrement meilleurs mais que ses paramètres étaient soumis à l'environnement, au contraire d'un contrôleur par logique floue qui sera donc plus polyvalent.

Un autre minidirigeable a été conçu par Zhang et al. [28]. Celui-ci possède une enveloppe en forme d'ellipsoïde de révolution, mais selon l'axe z , et communique lui aussi avec un ordinateur restant au sol. Il dispose en outre de cinq moteurs : deux selon l'axe x pour le mouvement longitudinal et le lacet, deux verticaux pour l'altitude et le roulis, et un pour le mouvement latéral (Fig. 1.8). Son contrôle s'effectue lui aussi par PID. Ce robot dispose également d'une caméra entraînée à détecter et suivre une main humaine par réseau de neurones. Il est capable de reconnaître un mouvement horizontal ou vertical de cette main et à tourner sur lui-même dans ce cas.

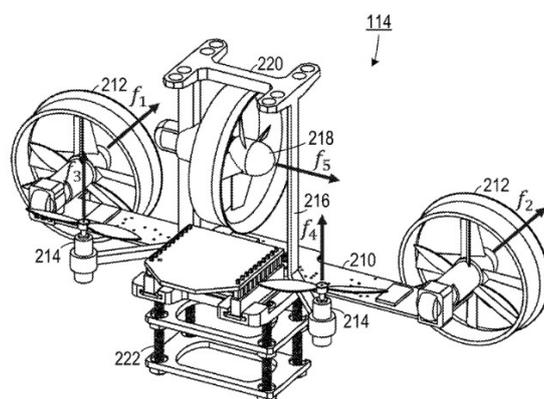


FIGURE 1.8 – Nacelle du minidirigeable de Zhang [28].

Gorjup et Taliokapis [6] se sont également intéressés au défi de construire un minidirigeable le plus économique possible. Cette équipe a d'abord sélectionné une enveloppe parmi plusieurs disponibles sur le marché, en se basant sur leur capacité à retenir l'hélium et leurs propriétés mécaniques. Bien que le matériau "Bubble" ait montré de très bons résultats, le ballon concerné ne pouvait seulement soulever que 40 grammes. C'est donc une enveloppe en microfollies de Qualatex ayant la forme d'une soucoupe de diamètre 91 cm qui a été retenue. Pour la structure du robot, la nacelle a été réalisée en impression 3D de manière à accueillir une Raspberry Pi Zero W, une batterie Li-Ion 500 mAh, trois moteurs (deux selon l'axe x, un selon l'axe z) et une caméra (Fig. 1.9). Cette équipe a cependant remarqué que placer la nacelle de manière excentrée par rapport à l'enveloppe rendait le drone plus résistant aux perturbations (Fig. 1.10). Ce minidirigeable est contrôlé par un algorithme de "carrot chasing" présenté dans [18] et communique avec un ordinateur via ROS⁵. Ce projet est open-source et possède son propre site open air ship.

Le minidirigeable conçu par Gonzalez et al. [5] est également un projet open-source et possède un git blimp SNN. Son originalité réside dans son nombre de moteurs : il n'en possède que deux, disposés selon l'axe x, mais l'équipe y a ajouté un servomoteur pour leur faire effectuer une rotation selon l'axe y, de façon à pouvoir contrôler l'altitude (Fig. 1.11). Ce drone utilise une Raspberry Pi Zero W communiquant grâce à ROS. Pour le contrôle, l'équipe a testé en simulateur et sur robot réel un contrôleur PID, un réseau de neurones artificiel et un réseau de neurones à impulsions. Bien que les trois méthodes aient donné des résultats satisfaisants, le contrôle par réseau de neurones à impulsions fait consommer moins d'énergie par les moteurs, il est donc à privilégier.

Enfin, Ikeda et al. proposent un minidirigeable assez original [10]. Celui-ci utilise comme enveloppe un ballon sphérique d'environ 80 cm de diamètre et est dirigé par un système de soufflerie, dans un objectif de vol d'intérieur le plus agréable et sécuritaire possible (Fig.

5. Robot Operating System



FIGURE 1.9 – Nacelle en impression 3D du minidirigeable [6]

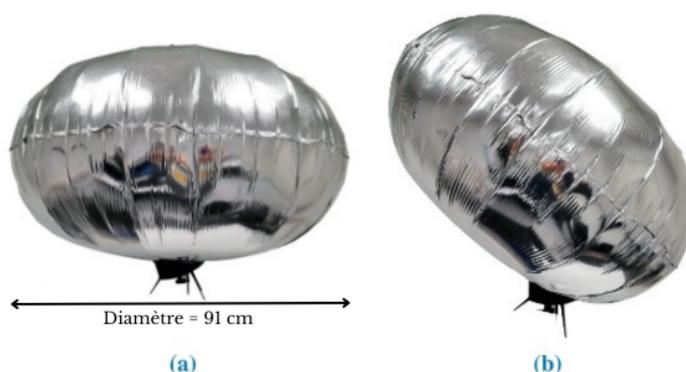


FIGURE 1.10 – Placements possibles pour la nacelle [6] : centrée (a) et excentrée (b). C'est finalement la configuration (b) qui a été retenue.

1.12). Ainsi, les micro-souffleurs utilisés fonctionnent par exemple à une fréquence de 26 kHz pour leur partie piézoélectrique de manière à ne pas être humainement entendus (Fig. 1.13). Le robot possède également une Raspberry Pi Zero et une caméra, et est alimenté par une batterie 3S 450 mAh Li-Po. Il est à noter qu'il est piloté à la main et qu'un algorithme de contrôle pour ce type de propulsion n'existe donc pas pour l'instant.

Cependant, le projet le plus abouti et le plus documenté est le minidirigeable GT-MAB de Georgia Tech [26, 2, 27, 21, 20, 19, 22, 15, 3, 7]. Il s'agit d'un minidirigeable dont l'enveloppe est un ellipsoïde de révolution selon l'axe z, de dimensions 46 x 72 x 72 cm (Fig. 1.14). Il dispose de quatre puis cinq moteurs (deux selon l'axe x, un selon l'axe y, deux selon l'axe z) et de différents capteurs dont une centrale inertielle (Fig. 1.15). Le

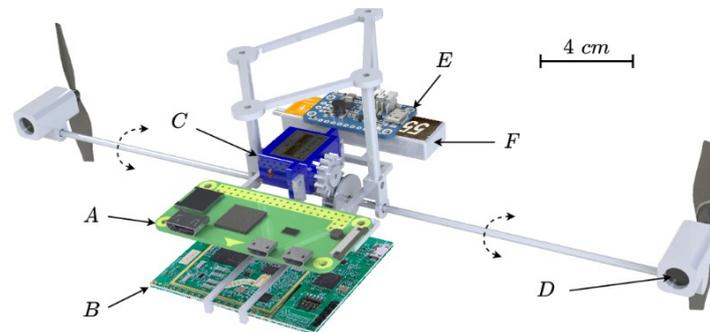


Fig. 2. Scheme of the proposed airship design. (A) Raspberry Pi W Zero; (B) 24 GHz Infineon Radar Position2Go; (C) Sub-micro Servo SG51R; (D) 8520 Coreless Motor; (E) PowerBoost 500 Basic; (F) 550mA 3.8V Li-Po Battery.

FIGURE 1.11 – Nacelle du minidirigeable de Gonzalez et al. [5].

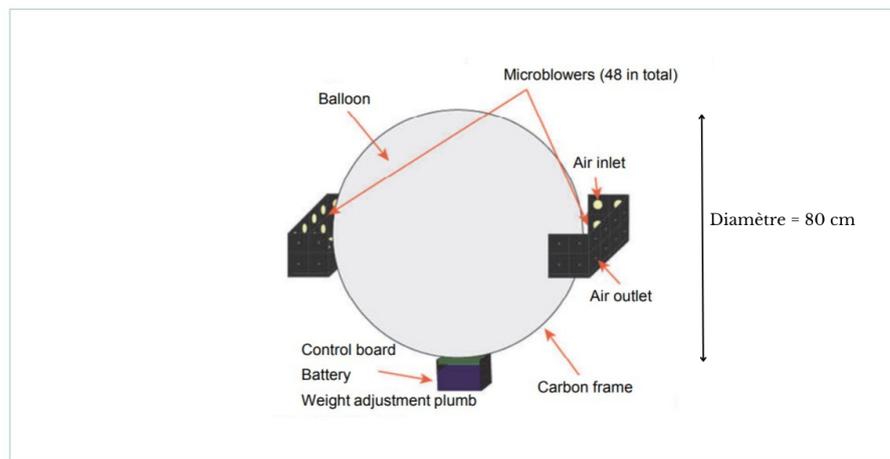


FIGURE 1.12 – Schéma du dirigeable d'Ikeda et al. [10]

contrôle s'effectue là aussi par PID. Ce qui est particulièrement intéressant avec ce robot dans notre cas est qu'il est, lui aussi, dédié aux interactions avec des humains. Ainsi, si dans un premier temps, le minidirigeable ne possédait pas de caméra, il a rapidement reçu une caméra RGBD⁶ [27] afin de détecter des visages humains par la méthode de Viola et Jones [24] et de les suivre. GT-MAB a ensuite été doté d'un réseau de neurones profond pour détecter les visages et mains humaines et réagir à certains mouvements simples [26], ainsi que d'un panneau LED pour indiquer aux utilisateurs les gestes reconnus.

De ces différents exemples, on peut retirer plusieurs points communs entre les minidirigeables. Tout d'abord, leur enveloppe est un ellipsoïde de révolution selon l'axe x ou z. Ils possèdent généralement entre trois et six moteurs et communiquent avec un ordina-

6. Red Green Blue Distance

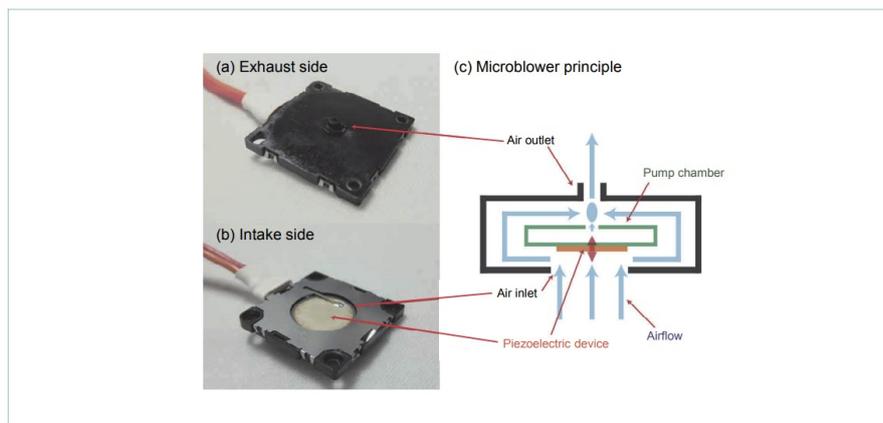


FIGURE 1.13 – Fonctionnement d'un micro-souffleur [10]



FIGURE 1.14 – Première version de GT-MAB de Georgia Tech, USA [21]

teur terrestre pour les calculs effectués par les algorithmes. Enfin, leur contrôle s'effectue généralement soit par PID, soit par réseau de neurones.

1.5 Travaux connexes

Bien évidemment, certains travaux scientifiques peuvent également s'avérer très pertinents dans notre situation sans pour autant s'agir du développement d'un minidirigeable. Par exemple, Iida s'est intéressé à un système d'odométrie visuelle applicable aux UAVs [9, 8]. Il s'est pour cela inspiré des abeilles dans une logique de biomimétisme, en utilisant un modèle de détection du mouvement de Reichardt [1] sur une caméra panoramique montée sur un rail puis sur un dirigeable de 2,3 mètres de longueur. Ses expériences ont montré des résultats satisfaisants dans un environnement intérieur à condition que le dirigeable suive

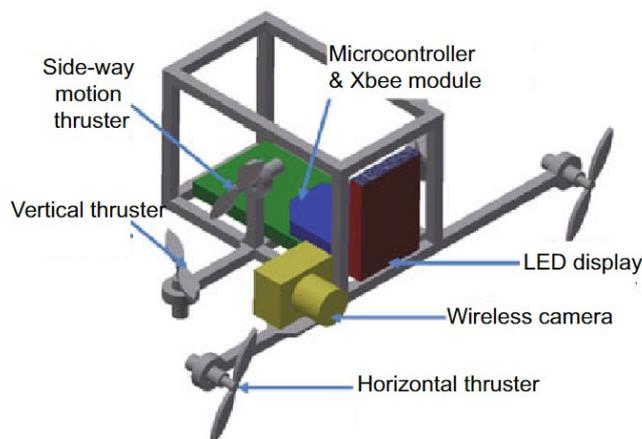


FIGURE 1.15 – Nacelle du minidirigeable GT-MAB [26]

toujours la même route. Selon Iida, ce système pourrait notamment être efficace dans le cas des atterrissages.

Toujours dans une logique de biomimétisme, Serres (mon tuteur) a effectué un état de l'art des drones aériens inspirés des insectes [16]. En plus des dirigeables y sont évoqués les aéroglisseurs, les drones à voile tournante et les microdrones. Sont explorés également des capteurs bio-inspirés pour drones aériens et différentes techniques de guidage des UAVs par différents capteurs visuels bio-inspirés.

Sur un autre sujet, Tobita s'est aussi intéressé aux dirigeables, mais avec une idée de réalité augmentée [23]. Il a donc développé un système permettant de détecter un UAV, ici un dirigeable, à partir de marqueurs placés dessus, et de fournir à un utilisateur des informations supplémentaires telles que des images projetées sur le dirigeable grâce à des lunettes adaptées. Des capteurs de pression permettraient de pousser plus loin les interactions entre l'humain et le robot en faisant réagir celui-ci lorsque l'utilisateur le touche.

Enfin, Rossouw [14] propose un état de l'art des dirigeables, bio-inspirés ou non, ainsi que des systèmes de contrôle automatisés. Elle présente également un dirigeable destiné à l'exploration et la localisation d'odeurs. Celui-ci a la forme d'une sphère de 53 cm de diamètre, et est propulsé au moyen de cinq nageoires (Fig. 1.16). Il est contrôlable manuellement ou par un algorithme de PID. Ce travail particulièrement récent peut représenter une nouvelle alternative intéressante pour des drones aériens silencieux et inoffensifs.

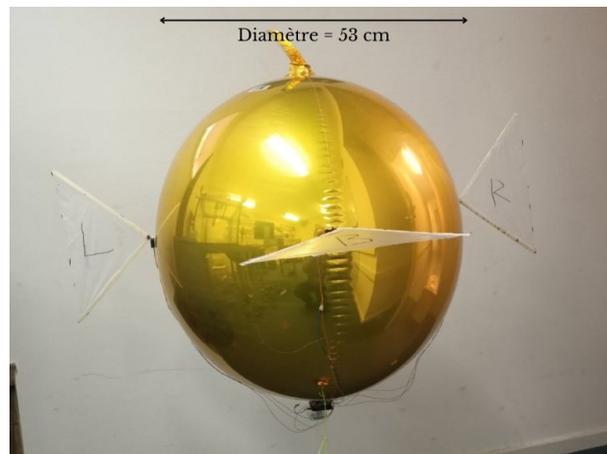


FIGURE 1.16 – Dirigeable propulsé au moyen de nageoires [14]

Chapitre 2

Avancement du projet : automatisation

2.1 Caméra Luxonis OAK-D-IoT-40

La mission principale de mon stage fut la robotisation du dirigeable pour suivre automatiquement le comédien. Cela veut dire que le robot devait être capable de repérer le comédien, de se déplacer seul pour le rejoindre et de se maintenir à une position relative prédéfinie. La première étape de cette mission est donc la reconnaissance du comédien. Pour cette tâche, je disposais d'une caméra Luxonis (Fig. 2.1) que je devais intégrer au dirigeable. Plus exactement, il s'agit de la caméra OAK-D-IoT-40. Cette caméra est constituée d'une caméra couleur et de deux caméras stéréo noir et blanc. Elle est également dotée d'une intelligence artificielle basée sur le deep learning et permettant notamment d'identifier des visages humains et de donner des informations comme leur position, leur distance à l'objectif ou encore la direction de leur regard. Pour cela, les deux caméras stéréo permettent d'avoir une vision 3D de l'environnement qui est appliquée à l'image de la caméra couleur. On peut donc s'en servir comme d'une caméra RGBD, ce qui est notre but.

Cependant, l'intégration de cette caméra pose deux difficultés. La première est sa masse : la caméra est beaucoup plus grosse qu'une caméra Raspberry Pi comme celle utilisée actuellement, et donc plus lourde. Il s'avère en effet qu'elle pèse un peu plus de 39 grammes, ce qui est contraignant étant donné que le minidirigeable actuel ne doit pas dépasser les 300 grammes de charges utiles. Pour réduire cette masse, j'ai remarqué que le radiateur de la caméra avait à lui seul une masse de presque 15 grammes. Ne disposant pas de documentation précise sur ce radiateur, j'ai recherché des radiateurs similaires et documentés, j'en ai déduit qu'il avait une résistance thermique d'environ $7\text{ }^{\circ}\text{C}/\text{W}$. J'ai décidé que je ne pouvais pas le remplacer par un ventilateur de performances équivalentes, car tous ceux que j'ai trouvés avaient une masse supérieure à 15 grammes. Je me suis donc demandé si ce radiateur était vraiment nécessaire. Pour répondre à cette question, j'ai effectué des me-



FIGURE 2.1 – Caméra Luxonis OAK-D-IoT-40. Taille : 60 mm de large, hauteur 45 mm, largeur 25 mm ; masse : 45,5 g sur sa documentation technique.

sures d'échauffement de la caméra avec radiateur à l'aide d'une caméra thermique ETS320, prévue pour les bancs d'essais et la recherche. Les mesures ont été réalisées en intérieur dans une pièce non ventilée de température ambiante de 20 °C, afin de se rapprocher des conditions d'une représentation théâtrale. J'ai pu observer un échauffement allant jusqu'à 32 °C au bout d'une minute, puis jusqu'à 65 °C au bout de 20 minutes, soit la durée prévue pour une représentation (Fig. 2.2). J'en ai donc conclu que la présence d'un radiateur est bien justifiée.



FIGURE 2.2 – Mesure d'échauffement de la caméra Luxonis au bout de 20 minutes.

La masse du dirigeable a également été augmentée par la connectique de la caméra. En effet, la caméra Raspberry Pi était connectée par une simple nappe. Cependant, la caméra Luxonis délivre ses informations par une prise USB-C, elle doit donc être connectée à la Raspberry Pi Zero par un câble USB-C micro-USB. Or le câble que j'avais à disposition possédait une masse de 34 grammes. Afin de l'alléger, nous avons enlevé sa gaine et l'avons raccourci.

La deuxième difficulté concerne l'alimentation de la caméra Luxonis, car si la caméra initiale était alimentée par la Raspberry Pi, la caméra Luxonis est bien plus gourmande en énergie. J'ai pu mesurer une consommation de 0,9 A pour 5 V. On peut donc en déduire la demande d'énergie pour une performance de 20 minutes :

$$E_{Luxonis} = 0,9 \times 5 \times \frac{1}{3} = 1,5 \text{ Wh} \quad (2.1)$$

Sachant que la batterie actuellement utilisée est une 1S 1800 mAh 3,7 V, on en déduit nos besoins en capacité énergétique de la nouvelle batterie, sachant que nous gardons une tension de 3,7 V :

$$\frac{1,5 + 1,8 \times 3,7}{3,7} \approx 2,205 \text{ Ah} \quad (2.2)$$

Après plusieurs recherches basées sur les performances demandées et la masse, j'ai choisi une nouvelle batterie de capacité 2,6 Ah. On passe donc pour la masse de la batterie de 36 grammes à 48 grammes. J'ai ensuite designé en 3D deux nouvelles pièces pour pouvoir placer cette batterie sur le dirigeable sans utiliser de panier accroché à la nacelle (Fig. 2.3).

2.2 Les tentatives d'installation de ROS sur Raspberry Pi zero W/2W

Assez tôt dans mon stage, j'ai commencé à avoir des inquiétudes sur les performances de calcul de notre micro-contrôleur. En effet, la Raspberry Pi Zero W est prévue pour être la plus compacte possible, mais au détriment de sa puissance. J'avais également pu constater certaines lenteurs lors de premiers tests que j'avais réalisés. J'ai donc décidé de remédier à ce problème en installant ROS dessus. Ma stratégie était de faire communiquer

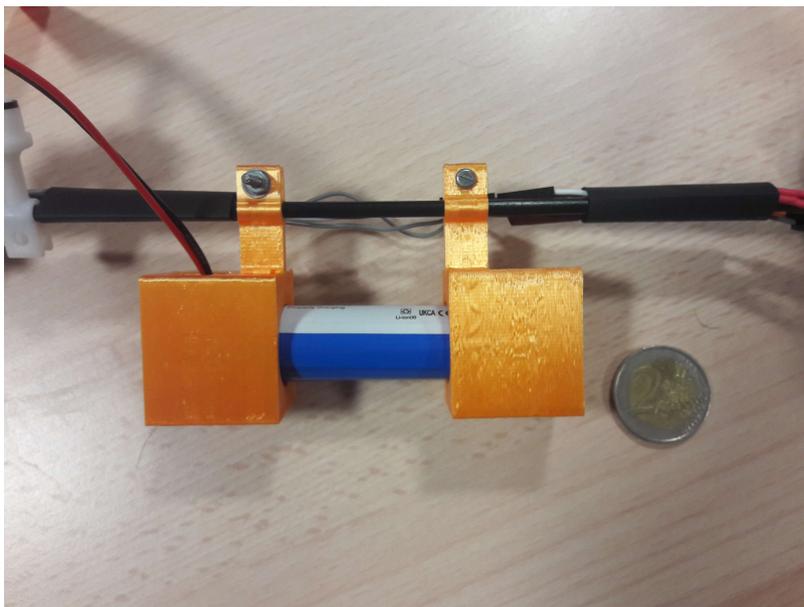


FIGURE 2.3 – Nouvelle batterie avec pièces de montage

la Raspberry avec un ordinateur plus puissant qui s'occuperait des calculs. La Raspberry aurait seulement un publisher pour les informations fournies par la caméra Luxonis et un subscriber pour les commandes à envoyer aux moteurs (Fig. 2.4). On pouvait aussi prévoir un publisher pour l'image filmée en elle-même, mais ce n'était pas nécessaire étant donné qu'elle était déjà envoyée en streaming.

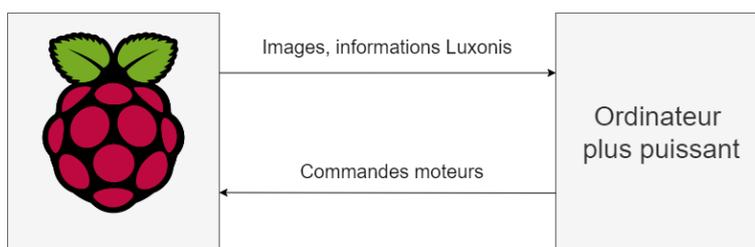


FIGURE 2.4 – Projet de communication avec la Raspberry Pi par ROS.

2.2.1 Sur Raspberry Pi Zero W

Mes premières tentatives d'installation ont été faites sur le modèle de Raspberry Pi dont je disposais et qui était présent sur le dirigeable, c'est-à-dire le modèle Zero W. Ce modèle, sorti en 2017, est prévu pour être particulièrement miniaturisé et possède une architecture armhf, sur 32 bits. J'étais relativement confiant sur la réussite de mon projet,

car j'avais trouvé des preuves de concept sur internet et dans les travaux de [5] [6]. Ces chercheurs utilisaient déjà une Raspberry Pi Zero avec ROS installé, et au vu de la date de leur publication, on peut supposer qu'il s'agissait du modèle W.

Cependant, je me suis heurté à plusieurs difficultés. La première a été une question d'obsolescence. En effet, j'ai commencé par suivre la documentation officielle pour tenter d'installer ROS Kinetic, puis d'autres versions, sur la Raspberry, et je n'arrivais pas à installer certains packages, notamment `python-rosdep` et d'autres packages en lien avec `python`. J'ai pu résoudre ce problème en remplaçant `python` par `python3`, mais dans le cas de `python-rosdep`, le package était automatiquement remplacé par `python3-rosdep2`. J'ai alors cru qu'il s'agissait d'un package prévu pour ROS2 et non pour ROS, et n'ai pas réussi à forcer l'installation de `python3-rosdep` même en précisant la version à installer. Il s'est finalement avéré que `python3-rosdep2` était bien prévu pour ROS et que le problème n'en était pas un, mais cela m'a tout de même fait perdre un temps considérable.

L'autre problème majeur que j'ai rencontré concerne les dépôts, stockés sur UBUNTU. Tout d'abord, ces dépôts n'étaient pas reconnus par la Raspberry, car ils ne possédaient pas de fichier `Release`. J'ai corrigé ce problème en fixant leur variable `TRUSTED` à `YES` dans leur fichier de déclaration, mais je ne suis pas sûr que cela ait tout résolu, car ils parcouraient alors tous les dépôts présents à l'adresse pour tous les systèmes d'exploitation, alors que j'avais bien précisé celui de la Raspberry Pi. Il faut d'ailleurs préciser que j'ai testé l'installation de ROS Noetic, Kinetic, Melodic et Humble sur des versions de Debian utilisant soit `buster`, soit `bullseye`. Je privilégiais `bullseye`, car il s'agit de la version de la version la plus récente des deux, et j'avais eu des problèmes de mise à jour avec ma première installation de `buster` lors de l'utilisation de la commande `"sudo apt-get update"`. Cependant, n'ayant pas rencontré ce problème dans mes installations suivantes de `buster` et n'ayant pas réussi à le reproduire, je ne développerai pas sur ce sujet. J'ai donc tenté des installations par le dépôt avec une version `bullseye` que je précisais grâce à `$(lsb_release -sc)`, mais j'obtenais un message d'erreur m'indiquant qu'il n'existait pas de dépôt pour cette version. J'ai alors eu l'idée de changer ma déclaration de version de `bullseye` à `buster`, et un dépôt a alors été détecté. J'ai donc poursuivi l'installation, mais je me suis heurté plus tard à des problèmes de compatibilité. J'ai alors voulu retenter l'expérience, cette fois en utilisant une version `buster` de Debian. Mais dans cette situation, le dépôt que j'avais utilisé n'était plus détecté. C'est un problème surprenant que je n'ai pas réussi à résoudre.

2.2.2 Sur Raspberry Pi Zero 2W

Après ces échecs, j'ai fait de nouveaux essais, cette fois-ci en utilisant une Raspberry Pi Zero 2W. C'est une Raspberry Pi sortie en 2021, identique au modèle W à ceci près qu'elle possède un processeur plus puissant et d'une architecture `arm64`, de 64 bits. Cela permet de

meilleures performances en termes de vitesse de calcul. Or ROS2 nécessite une architecture 64 bits pour fonctionner, cela m'ouvrait donc de nouvelles possibilités d'installation tout en réduisant les chances d'obsolescence. J'ai tout d'abord retenté les installations précédentes de ROS, sans succès. J'ai ensuite voulu installer ROS galactic puis foxy, mais là encore les dépôts de packages n'étaient pas reconnus. Et cette fois, je n'ai pas réussi à corriger le problème en passant l'argument `TRUSTED` à `YES` dans la déclaration. J'ai ensuite voulu tenter d'utiliser la Raspberry avec une version d'ubuntu, mais l'ordinateur plantait alors avant même l'écran d'accueil. J'en ai déduit qu'ubuntu était trop lourd pour une Raspberry Pi Zero et ai continué de faire des tests en me limitant à Debian.

Cependant, deux de ces tentatives ont été un demi-succès. La première consistait à installer ROS2 sur un docker. Cette méthode permet de simplifier l'installation et les problèmes de dépendance. De cette façon, j'ai pu installer et utiliser ROS Foxy sans problème. Toutefois, le docker ne se connectait pas au réseau Wi-Fi, il s'agit d'une caractéristique des dockers. Or l'intérêt d'utiliser ROS était de communiquer avec la Raspberry Pi par le réseau. Après une recherche rapide, j'ai découvert une piste pour contourner ce problème (Using ROS2, Docker & VPN), mais elle demandait notamment une inscription à un site et le temps commençait à me manquer, je n'ai donc pas poussé plus loin.

La deuxième tentative a été d'utiliser un dépôt git proposant d'installer ROS Kinetic pour une Raspberry Pi Zero. Il s'agissait d'une suite de commandes permettant de télécharger un fichier deb qui effectuait l'installation. Cependant, ce fichier deb était soumis aux problèmes d'obsolescence évoqués dans la partie 2.2.1, il a donc fallu modifier le nom des dépendances requises. Selon le modèle de Raspberry Pi Zero que j'utilisais, j'ai aussi dû modifier le nom de l'architecture, `armhf` par défaut, mais `arm64` dans le cas du modèle 2W. Après ces manipulations, l'installation a été un succès, bien que l'initialisation de ROS demande de supprimer un fichier pour le recréer alors qu'il n'aurait pas dû encore exister. Toutefois, lors de l'exécution de `roscore`, j'obtenais une erreur selon laquelle un fichier `rosout` n'existait pas ou ne remplissait pas certaines conditions. Après vérifications, le fichier `rosout` en question était bien créé et au bon endroit, mais je ne savais pas avec quoi l'ouvrir et son interprétation par l'éditeur de texte était inutilisable. Je n'ai donc pas pu le modifier et en suis resté là pour cette tentative.

Avec du recul, je me rends compte que j'ai passé beaucoup trop de temps sur cette tâche qui m'a pris trois mois. L'installation de ROS devait être une mesure de sécurité et n'était pas forcément indispensable. De plus, j'aurais pu me contenter d'accorder un espace de swap suffisamment important pour régler la plupart des problèmes de calcul. Cependant, cette tentative a fini par se transformer en un moyen pour l'équipe de recherche de savoir si, oui ou non, l'installation était possible. De mon point de vue, elle l'a été, on peut trouver beaucoup d'exemples en faisant foi ; mais si elle l'est encore aujourd'hui, c'est de façon

très compliquée. Une solution alternative à envisager serait d'utiliser un émetteur FPV si nécessaire, si la masse ajoutée n'est pas trop importante. Je n'avais tout simplement pas pensé à cette idée, elle m'a été soumise lors de ma présentation à l'équipe de recherche.

2.3 Modélisation de la dynamique du minidirigeable ID-Fly

La dynamique d'un minidirigeable est un sujet largement abordé dans les articles sur des projets similaires. Une étude a été, par exemple, détaillée par Wang et al. [25]. On peut considérer, à partir de ces travaux, d'une part, que la matrice d'inertie I associée à notre drone est diagonale, et d'autre part, que les forces à prendre en compte sont la pesanteur, la poussée d'Archimède, les poussées des moteurs et les forces de frottements. En se plaçant dans le repère inertiel du robot et en effectuant un bilan des forces sur notre minidirigeable, on peut en déduire les équations suivantes :

$$\begin{pmatrix} m \cdot a_x \\ m \cdot a_y \\ m \cdot a_z \end{pmatrix} = \begin{pmatrix} P_{long} - D_{v_x} \cdot v_x - m \cdot v_y \cdot \dot{\psi} \\ P_{latav} - P_{latar} - D_{v_y} \cdot v_y + m \cdot v_x \cdot \dot{\psi} \\ F_g + F_{pA} + P_{alt} - D_{v_z} \cdot v_z \end{pmatrix} \quad (2.3)$$

$$\begin{pmatrix} I_x \cdot \dot{\phi} \\ I_y \cdot \dot{\theta} \\ I_z \cdot \dot{\psi} \end{pmatrix} = \begin{pmatrix} -F_{pA} \cdot d_{centres} \cdot \sin(\phi) - d_{latmot} \cdot (P_{latav} - P_{latar}) - D_{w_x} \cdot \dot{\phi} \\ -F_{pA} \cdot d_{centres} \cdot \sin(\theta) - d_{longmot} \cdot P_{long} - D_{w_y} \cdot \dot{\theta} \\ \frac{L}{2} \cdot (P_{latav} + P_{latar}) + D_{w_z} \cdot \dot{\psi} \end{pmatrix} \quad (2.4)$$

Avec :

m la masse du dirigeable

a_x, a_y, a_z l'accélération du dirigeable projetée respectivement selon les axes x, y et z

v_x, v_y, v_z la vitesse du dirigeable projetée respectivement selon les axes x, y et z

I_x, I_y, I_z les coefficients respectivement (1, 1), (2, 2) et (3, 3) de la matrice d'inertie I

$P_{long}, P_{latav}, P_{latar}, P_{alt}$ les poussées respectivement exercées par les moteurs longitudinal, latéral avant, latéral arrière et vertical

$D_{v_x}, D_{v_y}, D_{v_z}$ les coefficients de frottements correspondant respectivement aux translations selon x, y et z

$D_{w_x}, D_{w_y}, D_{w_z}$ les coefficients de frottements correspondant respectivement aux rotations selon x, y et z

F_g, F_{pA} la force de gravité et la poussée d'Archimède

ϕ, θ, ψ les angles de roulis, tangage et lacet

$d_{centres}$ la distance entre le centre de masse et le centre de volume du dirigeable

d_{latmot} la distance projetée selon z entre le centre de masse et les moteurs latéraux

$d_{longmot}$ la distance projetée selon z entre le centre de masse et le moteur longitudinal

L la longueur de la tige principale de la structure, correspond au double de la distance entre le centre de masse et les moteurs latéraux projetée selon x

Suite à ce bilan, on pose deux hypothèses simplificatrices. Premièrement, le dirigeable est stable en roulis et tangage, c'est-à-dire $\dot{\phi} = \dot{\theta} = 0$. Cette hypothèse est souvent utilisée dans les modélisations de dynamique des minidirigeables et est permise par leur forme. Il est à noter que certaines études se sont intéressées aux mouvements des minidirigeables sans ces hypothèses [21, 20, 22]. On peut tout de même expliquer ce qui provoque ces deux vitesses angulaires dans un souci de meilleure compréhension. Il s'agit d'une question de positionnement par rapport au centre de masse. En effet, si la droite ayant pour vecteur directeur la poussée d'un moteur ne passe pas par le centre de masse, il en résulte un moment, ce qui est le cas avec les moteurs longitudinal et latéraux. De même, si le dirigeable est penché, les points d'application de la force de gravité et de la poussée d'Archimède, à savoir le centre de masse et le centre de volume, ne sont plus alignés selon z. Cela génère donc un couple (Fig. 2.5).

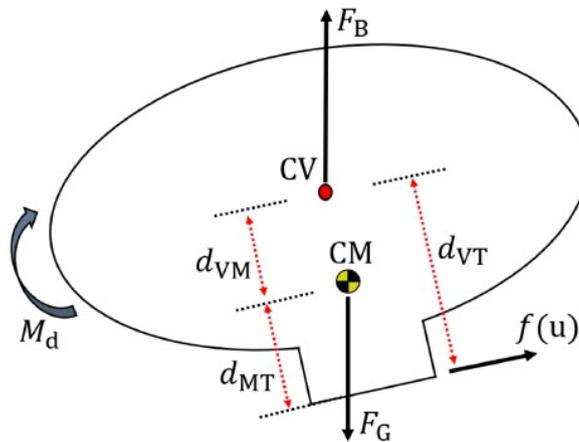


FIGURE 2.5 – Illustration d'un moment généré par le dirigeable penché. D'après [20].

La deuxième hypothèse est que la poussée d'Archimède et la gravité s'annulent : $F_g + F_{pA} = 0$. Cette hypothèse est propre à notre robot. On peut se la permettre car l'enveloppe est remplie d'hélium pour chaque vol, et la masse du dirigeable est ajustée au moyen de petits poids placés dans la nacelle. Avec ces hypothèses, on obtient le système suivant :

$$\begin{pmatrix} m \cdot a_x \\ m \cdot a_y \\ m \cdot a_z \\ I_z \cdot \dot{\psi} \end{pmatrix} = \begin{pmatrix} P_{long} - D_{v_x} \cdot v_x - m \cdot v_y \cdot \dot{\psi} \\ P_{latav} - P_{latar} - D_{v_y} \cdot v_y + m \cdot v_x \cdot \dot{\psi} \\ P_{alt} - D_{v_z} \cdot v_z \\ \frac{L}{2} \cdot (P_{latav} + P_{latar}) + D_{w_z} \cdot \dot{\psi} \end{pmatrix} \quad (2.5)$$

On obtient donc un système à quatre degrés de liberté. Il reste ensuite l'identification des paramètres. Pour la matrice d'inertie, j'ai modélisé toutes les pièces du minidirigeable dans Fusion360. Le logiciel a ensuite pu me calculer sa matrice d'inertie I (en $g \cdot mm^2$). On obtient :

$$I = \begin{pmatrix} 6,15e + 11 & 0 & 0 \\ 0 & 1,77e + 11 & 0 \\ 0 & 0 & 1,36e + 11 \end{pmatrix} \quad (2.6)$$

Pour les poussées des moteurs et les coefficients de frottement, j'ai effectué des vols filmés avec le robot, chacun en faisant effectuer une simple translation ou rotation. J'ai ensuite utilisé le logiciel Kinovea afin de faire du suivi de point pour en déduire la vitesse du dirigeable (Fig. 2.6). Cependant, le dirigeable était déséquilibré lors de ces vols, je ne suis donc pas sûr que ces données soient exploitables. Par manque de temps, je ne les ai pas exploitées. En revanche, le second stagiaire, travaillant avec moi sur le projet NOVABOT pendant quatre mois, achève son stage en mesurant la poussée des moteurs avec une autre méthode, peut-être aura-t-il des résultats exploitables.



FIGURE 2.6 – Suivi du minidirigeable dans une translation suivant z .

2.4 Simulation

Étant donné que les apports de la caméra intelligente n'ont pas pu être testés sur le minidirigeable par manque de temps, j'ai, à la place, créé une simulation pour effectuer mes essais de contrôle. J'ai pour cela utilisé le logiciel v-rep qui présente deux avantages : d'une part, il est gratuit, et d'autre part, il peut fonctionner avec ROS. J'ai tout d'abord voulu modéliser le dirigeable directement dans la simulation. J'ai donc créé un cylindre auquel j'ai lié le visuel de notre dirigeable obtenu d'après Fusion360 et j'ai défini sa matrice d'inertie I comme celle de notre minidirigeable.

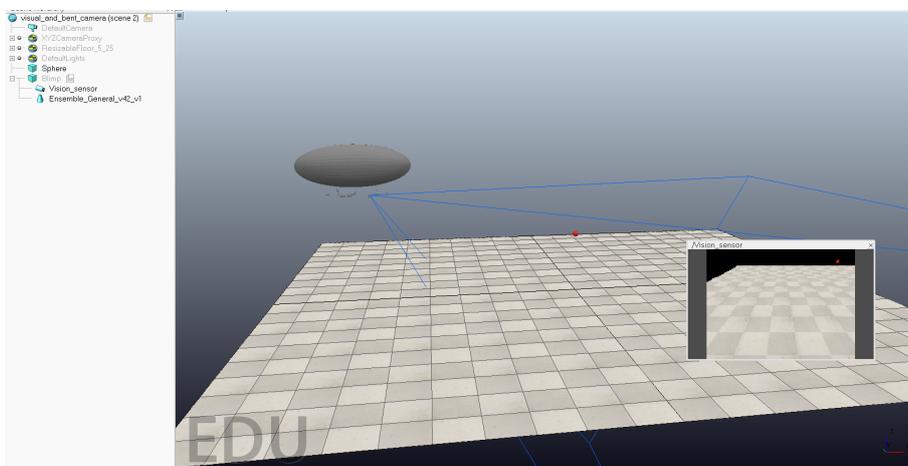


FIGURE 2.7 – Scène de simulation sous v-rep.

Cependant, le comportement que j'ai obtenu ne me convenait pas du tout et ne me semblait pas correspondre à la réalité. De plus, la modélisation des drones aériens dans v-rep ne me semblait pas répondre à nos attentes. En effet, je n'ai pu trouver qu'un seul exemple de drone aérien, faisant partie des robots proposés de base. Il s'agissait d'un drone à quatre moteurs verticaux (quadcopter) qui utilise un système de projection de particules pour calculer leur poussée. Au vu du peu de temps qu'il nous restait, j'ai alors décidé de considérer plutôt le dirigeable comme non dynamique dans v-rep et de commander directement sa position. J'ai également ajouté une caméra au dirigeable et j'ai placé dans la scène une sphère rouge de dimensions s'approchant d'une tête humaine. La caméra de la simulation possède les mêmes paramètres, notamment d'ouverture, que la caméra Luxonis. Seule la résolution diffère et est de moins bonne qualité dans le cas de la simulation, car une trop bonne résolution entrainerait un traitement d'image trop long et serait donc bloquant pour du temps réel. On dispose alors d'une scène assez simple (Fig. 2.7), mais il reste encore à faire bouger le dirigeable.

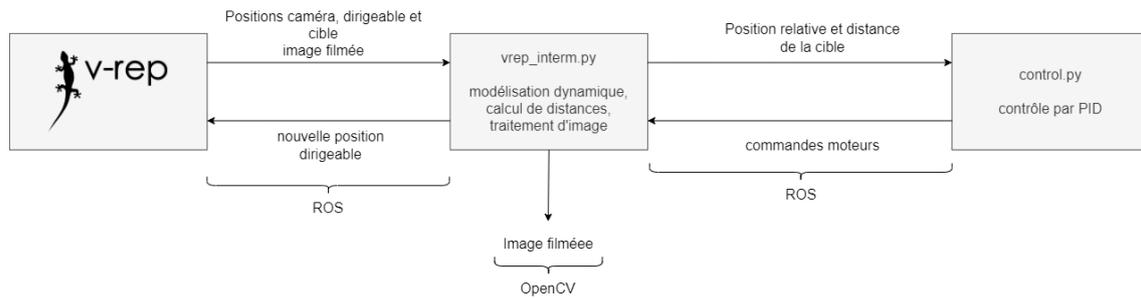


FIGURE 2.8 – Communication de la simulation avec ROS

Pour cela, j'ai fait communiquer la scène créée avec un programme python appelé `vrep_interm.py`, en utilisant ROS Noetic. Ce programme récupère d'un côté les positions des différents objets de la scène et l'image de la caméra, et d'un autre côté les commandes moteurs envoyées par l'algorithme de contrôle (Fig. 2.8). À partir de ces informations, il utilise les équations obtenues par la modélisation de la dynamique du dirigeable pour en déduire sa nouvelle position dans la scène. Ce programme effectue également le travail de la caméra Luxonis : non seulement il utilise les positions des objets pour en déduire la distance de la sphère-cible à la caméra, mais il effectue également un traitement d'image pour en déduire la position de la sphère dans la vue de la caméra. Pour cela, on passe l'image en format HSV et on recherche les pixels dont la valeur de H correspond au rouge, puis on fait une moyenne de leurs positions. Cette méthode simpliste peut être appliquée, car nous avons ici affaire à une sphère qui est le seul objet rouge de la scène et présente l'avantage de pouvoir être effectuée en temps réel. En dehors de la simulation, ce travail est réalisé par la Luxonis qui est suffisamment performante pour cela. Ces deux informations sont envoyées, toujours au moyen de ROS, à un autre programme de contrôle, qui n'a donc accès qu'aux informations que pourrait lui délivrer la caméra Luxonis.

2.5 Contrôle

2.5.1 Objectifs du contrôle

Avec ce projet, nous sommes en situation d'un robot évoluant en environnement intérieur inconnu et susceptible de changer selon les missions. Traditionnellement, l'objectif du robot serait de se repérer dans cet environnement en construisant une carte, par exemple avec une méthode de SLAM¹. Cependant, ici notre but n'est pas d'explorer l'environnement, mais de se placer à une certaine position par rapport à un acteur qu'on peut assimiler à un marqueur, puisqu'il est censé être reconnu sans problème. De plus, le dirigeable est censé être placé par une main humaine devant l'acteur, de façon à ce qu'il le détecte déjà

1. Simultaneous Localisation And Mapping

lorsque commence sa phase autonome. Notre mission n'inclut donc aucune exploration, il n'est donc pas nécessaire de chercher à se représenter l'environnement. On peut alors se placer simplement dans le repère inertiel du robot et y déplacer notre cible, qui sera la tête du comédien.

2.5.2 Algorithme de contrôle

Pour automatiser le contrôle du robot, on utilise un contrôleur PID. Comme vu après modélisation de la dynamique, on se retrouve avec un système à quatre degrés de liberté pour quatre commandes, une pour chaque moteur. On se place dans le repère du robot dans le référentiel de la cible. On choisit comme état :

$$X = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{pmatrix} \quad (2.7)$$

On peut alors écrire l'équation d'évolution du système :

$$\dot{X} = \begin{pmatrix} K_{long} \cdot u_{long} - \frac{D_{vx}}{m} \cdot \dot{x} - \dot{y} \cdot \dot{\psi} \\ K_{latav} \cdot u_{latav} - K_{latar} \cdot u_{latar} - \frac{D_{vy}}{m} \cdot \dot{y} + \dot{x} \cdot \dot{\psi} \\ K_{alt} \cdot u_{alt} - \frac{D_{vz}}{m} \cdot \dot{z} \\ \frac{L \cdot m}{2I_z} \cdot (K_{latav} \cdot u_{latav} + K_{latar} \cdot u_{latar}) + \frac{D_{wz}}{I_z} \cdot \dot{\psi} \end{pmatrix} \quad (2.8)$$

Avec :

$u_{long}, u_{latav}, u_{latar}, u_{alt}$ les commandes envoyées respectivement aux moteurs longitudinal, latéral avant, latéral arrière et vertical

$K_{long}, K_{latav}, K_{latar}, K_{alt}$ des coefficients constants propres à chaque moteur tels que :
 $m \cdot K_i \cdot u_i = P_i$

On en déduit :

$$\dot{X} = \begin{pmatrix} K_{long} & 0 & 0 & 0 \\ 0 & K_{latav} & -K_{latar} & 0 \\ 0 & 0 & 0 & K_{alt} \\ 0 & \frac{L \cdot m}{2I_z} \cdot K_{latav} & \frac{L \cdot m}{2I_z} \cdot K_{latar} & 0 \end{pmatrix} \cdot \begin{pmatrix} u_{long} \\ u_{latav} \\ u_{latar} \\ u_{vert} \end{pmatrix} + \begin{pmatrix} -\frac{D_{vx}}{m} \cdot \dot{x} - \dot{y} \cdot \dot{\psi} \\ -\frac{D_{vy}}{m} \cdot \dot{y} + \dot{x} \cdot \dot{\psi} \\ -\frac{D_{vz}}{m} \cdot \dot{z} \\ \frac{D_{wz}}{I_z} \cdot \dot{\psi} \end{pmatrix} = A \cdot U + B \quad (2.9)$$

On cherche ensuite à définir la position de notre robot. Penchons-nous sur les informations données par la caméra : il s'agit de la position de la cible dans l'image qu'on note

(y_c, x_c) et de sa distance en mètres par rapport à la caméra, notée d_c . Notons que la position est en pixels et que l'image est présentée comme un array de numpy, y_c correspond donc au numéro de la ligne et x_c à celui de la colonne. Ensuite, nous connaissons la résolution et l'ouverture de la caméra, nous pouvons donc en déduire l'erreur angulaire :

$$\begin{pmatrix} err_{\theta_h} \\ err_{\theta_v} \end{pmatrix} = \begin{pmatrix} (x_p - \frac{Res_x}{2}) \cdot HFOV \\ (y_p - \frac{Res_y}{2}) \cdot VFOV \end{pmatrix} \quad (2.10)$$

Avec :

Res_x, Res_y la résolution horizontale et verticale de la caméra

$HFOV, VFOV$ l'ouverture horizontale et verticale de la caméra

En effectuant quelques projections, on peut alors obtenir la distance de la cible à la caméra :

$$\begin{pmatrix} d_{cx} \\ d_{cy} \\ d_{cz} \end{pmatrix} = \begin{pmatrix} d_c \cdot \cos(err_{\theta_v}) \cdot \cos(err_{\theta_h}) \\ -d_c \cdot \cos(err_{\theta_v}) \cdot \sin(err_{\theta_h}) \\ -d_c \cdot \sin(err_{\theta_v}) \cdot \cos(err_{\theta_h}) \end{pmatrix} \quad (2.11)$$

Pour en déduire notre position, il faut se souvenir que la caméra est placée à l'avant du dirigeable d'une distance notée d_{cam} et penchée d'un angle noté α . On considère également que notre repère a notre cible pour origine, mais que son orientation est liée au robot et que le lacet ψ est donc constamment nul. Cela permettra plus tard de définir simplement notre ψ à atteindre comme $-err_{\theta_h}$. Notre position notée X_{int} est donc :

$$X_{int} = \begin{pmatrix} -d_{cx} \cdot \cos(\alpha) + d_{cam} - d_{cz} \cdot \sin(\alpha) \\ -d_{cy} \\ -d_{cz} \cdot \cos(\alpha) + d_{cx} \cdot \sin(\alpha) \\ 0 \end{pmatrix} \quad (2.12)$$

Il reste maintenant à en déduire l'objectif à atteindre. Pour cela, il faut prendre en compte le fait que nous ne cherchons pas à atteindre la cible mais à nous placer à une certaine distance d_{obj} . On peut à présent écrire l'objectif noté w :

$$w = \begin{pmatrix} -d_{obj} \cdot \cos(\alpha) + d_{cam} \\ 0 \\ d_{cx} \cdot \sin(\alpha) \\ -err_{\theta_h} \end{pmatrix} \quad (2.13)$$

Précisons que nous souhaitons que notre dirigeable reste immobile devant la cible, on a donc $\dot{w} = 0$ et $\ddot{w} = 0$. On applique alors le principe du PID. On définit la variable V :

$$V = (w - X_{int}) + 2 \times (\dot{w} - X) + \ddot{w} = w - X_{int} - 2X \quad (2.14)$$

Les coefficients (1, 2, 1) ont été choisis arbitrairement et peuvent être améliorés par l'expérience. On peut alors donner les commandes :

$$U = A^{-1} \cdot (V - B) \quad (2.15)$$

Ces commandes sont transmises au programme `vrep_interm.py` par `publish ROS` qui en déduit l'évolution du dirigeable et met à jour sa position. Ce contrôle a été testé uniquement en simulation avec des valeurs arbitraires pour les paramètres propres à la dynamique du dirigeable, mais les résultats obtenus ont été satisfaisants, avec une décroissance de l'erreur correspondant à ce qu'on pouvait espérer avec une vitesse cohérente pour ce minidirigeable (Figs. 2.9 et 2.10, les pics de début correspondent à l'initialisation du système).

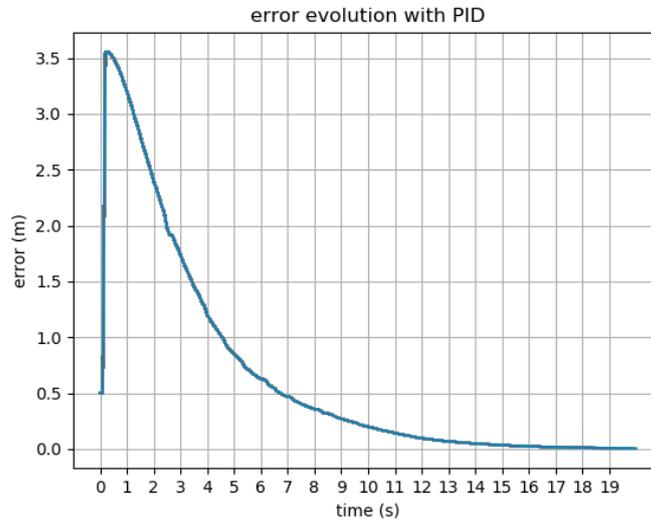


FIGURE 2.9 – Erreur de distance au cours du temps avec un contrôle par PID.

Il est à noter que cet algorithme de contrôle suppose que chaque moteur peut être commandé individuellement de façon linéaire. Cependant, ce n'est pas le cas actuellement : les drivers du dirigeable ont été écrits de façon à ce que les moteurs soient commandés en tout ou rien, et les deux moteurs latéraux sont commandés en même temps (on ne peut donc

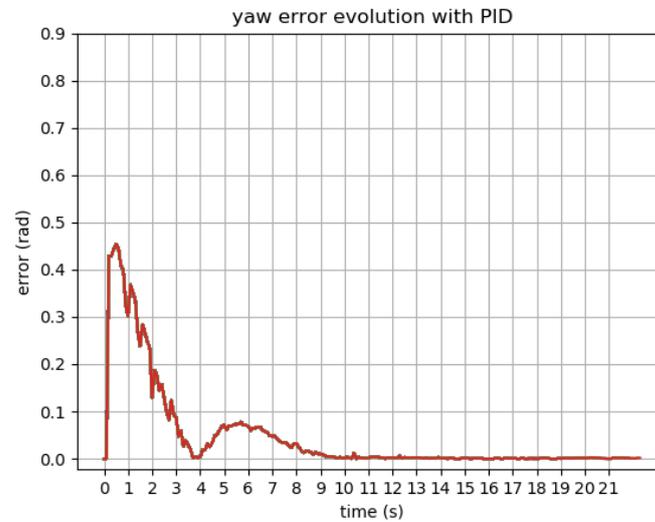


FIGURE 2.10 – Erreur de cap au cours du temps avec un contrôle par PID.

pas effectuer de translation selon y). Ce point sera à corriger lors d'un prochain stage. De plus, nous venons de découvrir que les moteurs latéraux ne se situent pas à même distance du centre de masse du robot, le couple n'est donc pas correctement estimé. Ce problème est en train d'être résolu, mais il le sera sans doute après la fin de ma contribution au projet NOVABOT.

Conclusion et perspectives

Ce rapport présente l'avancement du projet NOVABOT, qui est à la fois scientifique et artistique, de minidirigeable dédié à la captation d'acteurs de théâtre. L'objectif principal était d'automatiser le contrôle de ce robot dans le suivi du comédien par l'intégration d'une caméra intelligente reconnaissant les visages et la création d'un algorithme de contrôle. Durant ce projet, je me suis heurté à plusieurs difficultés qui m'ont empêché d'être aussi efficace que je l'aurais voulu. Au final, le design du dirigeable n'a pas été beaucoup modifié, à ceci près que la nacelle a été remplacée par deux pièces pour accueillir une nouvelle batterie. La masse du minidirigeable a également augmenté, mais ce n'est pas inquiétant sur le court terme, car nous n'avons pas encore dépassé la masse limite. De plus, une nouvelle enveloppe a été obtenue, légèrement plus grande, et pouvant donc soulever une masse plus importante.

Le principal sujet et problème de ce stage a été l'installation de ROS sur Raspberry Pi Zero W/2W. Cette étape qui aurait dû être anecdotique et optionnelle a finalement duré plus de trois mois pour ne pas fonctionner. Avec le recul, je pense que j'aurais dû l'abandonner bien plus tôt et éventuellement y revenir si elle devenait nécessaire pour permettre des calculs en temps réel. Cette mauvaise gestion de mon temps m'a empêché de me pencher sur d'autres points de la dynamique et du contrôle, et de faire des essais sur le robot réel. J'ai donc dû me contenter de tests sur une simulation que j'ai conçue. L'algorithme de contrôle créé a cependant obtenu des résultats prometteurs, et la méthode qu'il utilise a déjà fait ses preuves dans d'autres projets similaires.

Les prochains points sur lesquels travailler dans ce projet sont multiples. Le premier sera évidemment le test de l'algorithme en conditions réelles. Cela nécessitera notamment la réécriture des drivers des moteurs, ainsi que la création d'une pièce pour fixer la caméra Luxonis à la structure. Ensuite, il faudra revoir le design de cette dernière pour l'adapter à la nouvelle enveloppe. Enfin, l'intégralité de la structure devra être revue, tant sur le type de connectique utilisée que sur la fixation des différentes pièces.

references

Bibliographie

- [1] Alexander BORST. “Models of motion detection”. In : *Nature neuroscience* 3.11 (2000), p. 1168-1168. URL : https://www.nature.com/articles/nn1100_1168.
- [2] Sungjin CHO et al. “Autopilot design for a class of miniature autonomous blimps”. In : *2017 IEEE conference on control technology and applications (CCTA)*. IEEE. 2017, p. 841-846. URL : <https://ieeexplore.ieee.org/abstract/document/8062564/>.
- [3] Sungjin CHO et al. “Autopilot design of a class of miniature autonomous blimps enabled by switched controllers”. In : *International Journal of Intelligent Robotics and Applications* 6.3 (2022), p. 385-396. URL : <https://link.springer.com/article/10.1007/s41315-022-00230-6>.
- [4] Pablo GONZÁLEZ et al. “Developing a low-cost autonomous indoor blimp”. In : (2009). URL : <https://rua.ua.es/dspace/handle/10045/12579>.
- [5] Marina GONZÁLEZ-ÁLVAREZ et al. “Evolved neuromorphic radar-based altitude controller for an autonomous open-source blimp”. In : *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, p. 85-90. URL : <https://ieeexplore.ieee.org/abstract/document/9812149/>.
- [6] Gal GORJUP et Minas LIAROKAPIS. “A low-cost, open-source, robotic airship for education and research”. In : *IEEE Access* 8 (2020), p. 70713-70721. URL : <https://ieeexplore.ieee.org/abstract/document/9062572/>.
- [7] Mengxue HOU, Qiuyang TAO et Fumin ZHANG. “Human pointing motion during interaction with an autonomous blimp”. In : *Scientific Reports* 12.1 (2022), p. 11402. URL : <https://www.nature.com/articles/s41598-022-15016-w>.
- [8] Fumiya IIDA. “Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision”. In : *Proceedings of the 32nd ISR (International Symposium on Robotics)*. T. 19. 2001, p. 21-27. URL : <http://people.csail.mit.edu/iida/papers/ISR393iida.pdf>.
- [9] Fumiya IIDA et Dimitrios LAMBRINOS. “Navigation in an autonomous flying robot by using a biologically inspired visual odometer”. In : *Sensor fusion and decentralized control in robotic systems III*. T. 4196. SPIE. 2000, p. 86-97. URL : <http://people.csail.mit.edu/iida/papers/IidaLambriSPIE4196-13.pdf>.

- [10] Daizo IKEDA. “Blade-free drone”. In : *NTT DOCOMO Tech J* 21.4 (2020), p. 37-43. URL : https://www.docomo.ne.jp/english/binary/pdf/corporate/technology/rd/technical_journal/bn/vol21_4/vol21_4_006en.pdf.
- [11] Youssra MANSAR et al. “NOVABOT : Acteur et drone. Dialectique du corps en scène et de son image fragmentée”. In : *Drôles d’objets-Un nouvel art de faire*. Nancy, France, mai 2023. URL : <https://hal.science/hal-04101507/>.
- [12] Seungyong OH et al. “Flying display : Autonomous blimp with real-time visual tracking and image projection”. In : *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, p. 131-136. URL : <https://ieeexplore.ieee.org/abstract/document/4058556/>.
- [13] Yasuhiro OHATA, Satoshi USHIJIMA et Dragomir N NENCHEV. “Development of an indoor blimp robot with internet-based teleoperation capability”. In : *Proc. 13th IASTED International Conference on Robotics and Applications*. 2007, p. 186-191. URL : <https://www.actapress.com/PaperInfo.aspx?PaperID=31295&reason=500>.
- [14] Michelle ROSSOUW. “An open-source autopilot and bio-inspired source localisation strategies for miniature blimps”. Thèse de doct. UNSW, Canberra, juin 2023.
- [15] Landan SEGUIN et al. “A deep learning approach to localization for navigation on a miniature autonomous blimp”. In : *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. IEEE. 2020, p. 1130-1136. URL : <https://ieeexplore.ieee.org/abstract/document/9264514/>.
- [16] Julien SERRES, Stéphane VIOLLET et Franck RUFFIER. “Microdrones bio-inspirés - Doter nos futurs robots aériens de l’agilité des insectes”. In : *Les Techniques de l’Ingenieur* (2017), S7717-V1. URL : <https://amu.hal.science/hal-01574508/>.
- [17] Florian SICCARDI. *NOVABOT : un mini dirigeable dédié à la captation scénique*. INSA de Lyon, France, 2022.
- [18] P.B. SUJIT, Srikanth SARIPALLI et Joao Borges SOUSA. “Unmanned aerial vehicle path following : A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles”. In : *IEEE Control Systems Magazine* 34.1 (2014), p. 42-59. URL : <https://ieeexplore.ieee.org/abstract/document/6712082/>.
- [19] Qiuyang TAO, Mengxue HOU et Fumin ZHANG. “Modeling and identification of coupled translational and rotational motion of underactuated indoor miniature autonomous blimps”. In : *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE. 2020, p. 339-344. URL : <https://ieeexplore.ieee.org/abstract/document/9305371/>.
- [20] Qiuyang TAO et al. “Modeling and control of swing oscillation of underactuated indoor miniature autonomous blimps”. In : *Unmanned Systems* 9.01 (2021), p. 73-86. URL : <https://www.worldscientific.com/doi/abs/10.1142/S2301385021500060>.

- [21] Qiuyang TAO et al. "Parameter identification of blimp dynamics through swinging motion". In : *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE. 2018, p. 1186-1191. URL : <https://ieeexplore.ieee.org/abstract/document/8581376/>.
- [22] Qiuyang TAO et al. "Swing-reducing flight control system for an underactuated indoor miniature autonomous blimp". In : *IEEE/ASME Transactions on Mechatronics* 26.4 (2021), p. 1895-1904. URL : <https://ieeexplore.ieee.org/abstract/document/9408429/>.
- [23] Hiroaki TOBITA. "Augmented reality system". US Patent 9779550B2. 2017. URL : <https://patents.google.com/patent/US9779550B2/en>.
- [24] Paul VIOLA et Michael J JONES. "Robust real-time face detection". In : *International journal of computer vision* 57 (2004), p. 137-154. URL : <https://link.springer.com/article/10.1023/b:visi.0000013087.49260.fb>.
- [25] Yue WANG et al. "Disturbance compensation based controller for an indoor blimp robot". In : *Robotics and Autonomous Systems* 124 (2020), p. 103402. URL : <https://www.sciencedirect.com/science/article/pii/S0921889019304683>.
- [26] Ning-shi YAO et al. "Autonomous flying blimp interaction with human in an indoor space". In : *Frontiers of Information Technology & Electronic Engineering* 20.1 (2019), p. 45-59. URL : <https://link.springer.com/article/10.1631/FITEE.1800587>.
- [27] Ningshi YAO et al. "Monocular vision-based human following on miniature robotic blimp". In : *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, p. 3244-3249. URL : <https://ieeexplore.ieee.org/abstract/document/7989369/>.
- [28] Fumin ZHANG et al. "Miniature autonomous robotic blimp". US Patent 0258257A1. 2019. URL : <https://patentimages.storage.googleapis.com/4e/ae/06/6ddb0214de063e/US11511842.pdf>.
- [29] Jean-Christophe ZUFFEREY et al. "Flying over the reality gap : From simulated to real indoor airships". In : *Autonomous Robots* 21 (2006), p. 243-254. URL : <https://link.springer.com/article/10.1007/s10514-006-9718-8>.

Résumé — Ce rapport vise à présenter ma contribution au projet de robotisation du minidirigeable NOVABOT dédié à la captation scénique en intérieur. L'objectif du projet NOVABOT est d'automatiser le contrôle du minidirigeable lors du suivi de comédien sur scène. Pour cela, on intégrera une caméra intelligente Luxonis capable de reconnaître les visages humains avant d'utiliser un algorithme de contrôle par correction Proportionnelle Intégrale Dérivée (PID). Une tentative infructueuse d'installer ROS sur une carte Raspberry Pi Zero W et 2W à bord du robot a occupé une grande partie de mon stage. Les écueils rencontrés seront détaillés et des solutions alternatives seront proposées. Quelques tests d'identification de la dynamique du minidirigeable ont été conduits, cela aura permis d'entamer la construction d'une simulation sur v-rep compatible avec ROS. Ces résultats encourageants en simulation permettront à l'avenir d'explorer une solution exploitant ROS de manière déportée.

Mots clés : Minidirigeable, ROS, Raspberry Pi Zero, PID, caméra intelligente

Polytech Angers
62, avenue Notre Dame du Lac
49000 Angers