



Master Systèmes Dynamiques et Signaux

Mémoire

---

# Conception et réalisation d'un AUV hybride flotteur et planeur

---

*Auteur:*  
Gabriel BETTON

*Jury:*  
Pr. L. HARDOUIN  
Pr. L. JAULIN

Version du  
25 août 2024

## Résumé

Dans le cadre de la validation de mon diplôme d'ingénieur ENSTA Bretagne et de mon master Systèmes Dynamiques et Signaux, j'ai choisi de réaliser un stage avec l'UMR CNRS 6285 Lab-STICC à l'ENSTA Bretagne.

Le contexte de recherche est le projet Électro-Karst. Coordonné par Lionel LAPIERRE et financé par L'Agence Nationale de la Recherche, le projet vise à développer des systèmes robotiques pour explorer et réaliser la cartographie des réseaux karstiques, de grandes cavités souterraines remplies d'eau douce [1].

De nombreuses problématiques émergent de cette initiative, notamment la conception d'un système robotisé capable de se déplacer dans un environnement confiné et non structuré.

C'est dans cette direction que mon stage s'est orienté, avec en première approche la maîtrise de la technologie du ballast pour contrôler sa profondeur, puis avec le développement d'un AUV (Autonomous Underwater Vehicle) minimaliste mêlant flotteur et planeur.

L'objectif de ce stage était d'apprendre à travailler en laboratoire dans une logique de recherche. Mon sujet était une problématique de départ à partir de laquelle j'évoluerais pour explorer les pistes de développement les plus intéressantes.

## Abstract

As part of the validation of my ENSTA Bretagne engineering degree and my master's degree in Dynamic Systems and Signals, I have chosen to do my internship with the UMR CNRS 6285 Lab-STICC located at ENSTA Bretagne.

The research context is the Electro-Karst project. Coordinated by Lionel LAPIERRE and funded by the French National Research Agency (ANR), the project aims to develop robotic systems for exploring and mapping karst networks, which are large underground cavities filled with fresh water [1].

Many challenges and issues are raised with this project, including the design of a robotic system capable of moving in a confined and unstructured environment such as karst.

It was in this direction that my internship was directed, starting with the mastery of ballast technology for depth control, then with the development of a minimalist AUV (Autonomous Underwater Vehicle) combining a float and a glider.

This internship was about learning how to work in a laboratory in a research environment, with an initial subject as a starting point from which I would evolve to explore the most interesting avenues.

## Remerciements

Je tiens à remercier tout particulièrement Lionel LAPIERRE et Luc JAULIN qui m'ont accompagné pour toute la durée du stage. Leurs nombreuses explications et conseils m'ont été précieux.

Je remercie également Mathis RIERA, un ami et collègue stagiaire qui m'a épaulé pendant toute cette période. Merci également à Thomas LE MEZO pour son expertise sur les flotteurs et Loïck DEGORRE pour sa maîtrise de la commande sans modèle.

Je remercie enfin tous les membres du laboratoire avec qui j'ai pu discuté et qui m'ont conseillé aussi bien sur la réalisation de mon stage que pour mon parcours professionnel.

# Table des matières

<b>1. Introduction</b> .....	<b>5</b>
1.1. Objectifs .....	5
1.2. Planification .....	5
1.3. État de l'art .....	5
1.3.1. Flotteur .....	5
1.3.2. Planeur .....	6
<b>2. Réalisation</b> .....	<b>7</b>
2.1. Flotteur simple .....	7
2.1.1. Structure .....	7
2.1.2. Composants .....	8
2.2. Flotteur planeur .....	11
2.2.1. Structure .....	11
2.2.2. Composants .....	12
<b>3. Contrôle de la profondeur</b> .....	<b>16</b>
3.1. Modèle dynamique .....	16
3.2. Linéarisation par bouclage .....	17
3.2.1. Calcul de la commande .....	17
3.2.2. Estimation du coefficient de frottement et de la masse ajoutée ..	18
3.2.3. Estimation du volume .....	19
3.2.4. Essais .....	21
3.2.5. Résultats .....	25
3.3. Commande sans modèle .....	25
3.3.1. Théorie .....	25
3.3.2. Implémentation .....	26
3.3.3. Résultats .....	27
<b>4. Déplacement dans l'espace</b> .....	<b>28</b>
4.1. Avance .....	28
4.2. Contrôle du cap .....	29
4.2.1. Dynamique du volant d'inertie .....	29
4.2.2. Commande par bouclage linéarisant .....	29
4.2.3. Commande en cycle .....	30
4.2.4. Commande hybride .....	33
4.2.5. Résultats .....	34
<b>5. Conclusion</b> .....	<b>35</b>
5.1. Résultats .....	35
5.2. Publications .....	35
<b>Bibliographie</b> .....	<b>36</b>

# 1. Introduction

## 1.1. Objectifs

Le point de départ du stage était la maîtrise de la technologie du ballast comme moyen de contrôler la profondeur d'un robot sous-marin. D'un point de vue commande pour développer une loi de commande adaptée mais aussi matériel pour valider l'utilisation d'un ballast acheté sur étagère.

Une fois le contrôle en profondeur maîtrisé, l'objectif suivant était d'ajouter au système la capacité de se déplacer dans l'espace, de manière minimaliste et sans ajouter d'actionneurs à l'extérieur de l'enceinte étanche. Les avantages visés sont la maîtrise du coût en évitant les composants les plus onéreux, la diminution des perturbations extérieurs (sonores et fluides) et la possibilité d'aller en eau profonde plus facilement.

Concrètement, la solution étudiée a été d'une part d'ajouter des ailes au flotteur pour générer une avance lors des phases de montée et de descente, et de l'autre d'intégrer un volant d'inertie pour contrôler le cap du robot. Ainsi le système serait capable de se déplacer dans l'espace en n'utilisant que des composants passifs et des actionneurs inertiels confinés dans l'enceinte étanche.

## 1.2. Planification

Les objectifs du stage s'étant développés au fur et à mesure du stage, écrire un planning prévisionnel était impossible. Le diagramme proposé Fig. 1 rend compte du temps passé sur chacune des tâches réalisées.

Section	Titre	Mars	Avril	Mai	Juin	Juillet	Août
Simulateur Physique 3D	Essais simulation physique 2D						
	Création simulateur 3D						
	Rendu 3D et interface utilisateur						
Flotteur	Établissement du modèle dynamique						
	Loi de commande en profondeur						
	Estimateur en simulation						
	Réalisation du prototype						
	Implémentation commande et estimateur						
	Essais et améliorations						
	Développement commande sans modèle						
	Implémentation et essais						
Flotteur Planeur Hybride	Étude du volant d'inertie						
	Loi de commande en cap						
	Implémentation et essais						
Autres	Essais des ailes						
	Rédaction						

Fig. 1. – Planification des tâches

## 1.3. État de l'art

### 1.3.1. Flotteur

L'utilisation d'un ballast pour contrôler la profondeur d'un AUV est bien présente dans la littérature, particulièrement dans le contexte du développement des flotteurs océanographiques, des instruments de mesures sous-marins qui suivent des profils de profondeur et mesurent les propriétés physiques et chimiques de l'eau. Argo [2] est le principal programme de mesures océanogra-

phiques utilisant une flotte de ces flotteurs pour couvrir la majeure partie des océans. De nombreux chercheurs utilisent les données de ces flotteurs pour leurs publications.

Le programme Argo permet de comprendre le besoin et les cas d'utilisation des flotteurs mais ne présente pas en détail les technologies et les stratégies de commande utilisées. Pour mieux saisir l'état de l'art du contrôle il faut se tourner vers [3] qui passe en revue les différentes approches de conception des flotteurs.

Le ballast est l'élément central du flotteur. Il sert à modifier le rapport entre son poids (proportionnel à sa masse) et la poussée d'Archimède (proportionnelle à son volume).

Différentes technologies de ballast existent. Lorsque le flotteur doit atteindre de grandes profondeurs [4], c'est une pompe qui gonfle une vessie flexible avec de l'huile afin de modifier le volume. Lorsque l'étude océanographique vise sur des eaux moins profondes, le ballast est souvent un vérin hydraulique se remplissant et se vidant d'eau pour augmenter sa masse [5]. Il peut aussi être un vérin sortant et rentrant une tige pour augmenter son volume [6].

Dans les deux cas, l'actionneur permet de modifier la densité du flotteur, les différentes loi de commandes présentes dans la littérature peuvent donc s'appliquer sur l'ensemble des flotteurs.

Les contrôleurs classiques PID ou PD présentent des performances peu satisfaisantes, de nombreuses publications développent donc des contrôleurs plus adaptés à ce système. La stratégie la plus commune semble être la linéarisation par bouclage [5], [6], cette méthode se base sur un modèle dynamique du système donc plus le modèle est précis meilleure sera la réponse. Ainsi, l'utilisation d'une commande par bouclage linéarisant implique souvent l'estimation de paramètres difficiles à mesurer, tels que le volume, le coefficient de frottement ou simplement la vitesse. Une autre approche se basant également sur un modèle est le contrôle par mode glissant [7] souvent utilisé dans le contrôle des véhicules sous-marins. Mais le mode glissant entraîne généralement des fréquences élevées dans l'actionnement, augmentant alors la consommation énergétique et l'usure. Pour palier à ce dernier problème des méthodes améliorent l'approche par mode glissant avec des contrôleurs à logique floue (*fuzzy control*) [8] ou utilisant un contrôle actif de rejet des perturbations (ADRC) [9].

### 1.3.2. Planeur

Le planeur sous-marin le plus commun est un engin profilé muni d'aile et d'ailerons qui déplace des masses internes pour naviguer [10]. De nombreuses publications existent sur le contrôle, la navigation et l'optimisation du design de ces AUVs.

Pour notre AUV, nous avons utilisé un volant d'inertie pour contrôler l'orientation du robot. Cette technique est très commune pour les satellites [11] mais à notre connaissance n'a jamais été utilisé dans un système avec frottement fluide comme notre robot sous-marin.

## 2. Réalisation

La réalisation du démonstrateur a progressé au fur et à mesure du stage avec l'évolution de nos objectifs. Le but de la première version était un système minimaliste qui permettrait de valider le contrôle en profondeur, les versions suivantes ont été réalisées pour convertir le flotteur en planeur hybride.

### 2.1. Flotteur simple

#### 2.1.1. Structure



Fig. 2. – Le flotteur

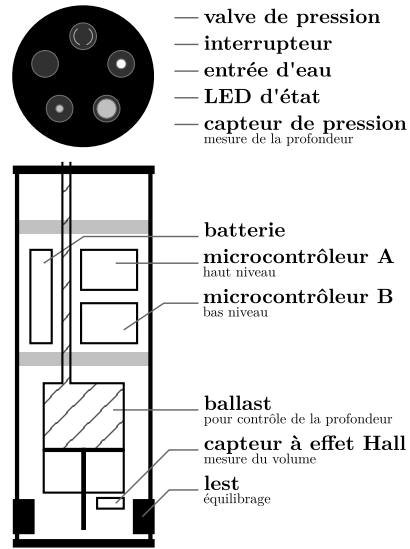


Fig. 3. – Schéma descriptif du flotteur

Tous les composants du démonstrateur (Fig. 3) sont contenus dans une enceinte étanche.

Le ballast, seul actionneur du système occupe le bas du flotteur. Il est équipé d'un capteur permettant d'obtenir la masse d'eau dans le ballast ainsi que le débit.

Le ballast permet de faire varier sa masse  $m_b$  entre 0 et  $m_{b,\max}$ . Pour que cette variation maximise la dynamique verticale du système, il faut ajuster la masse à vide  $m_0$  avec du lest de sorte à ce que le poids  $F_g$  et la poussée d'Archimède  $F_a$  se compensent lorsque le ballast est à moitié rempli:

$$\begin{aligned}
 F_g + F_a &= 0 \\
 \Leftrightarrow \left( m_0 + \frac{m_{b,\max}}{2} \right) g - \rho V g &= 0 \\
 \Leftrightarrow m_0 + \frac{m_{b,\max}}{2} &= \rho V
 \end{aligned}$$

Le reste du tube est occupé par l'électronique, comprenant la batterie, le variateur de vitesse et deux microcontrôleurs.

Le premier microcontrôleur traite les opérations de bas niveau, il permet de piloter le ballast et renvoie la mesure de la masse d'eau dans le ballast  $m_b$ . L'arrêt en butée du ballast est géré par le programme qui doit donc être suffisamment rapide. C'est pour cette raison que le reste des calculs est fait par un deuxième microcontrôleur.

Ce dernier calcule les filtres d'estimation et les lois de commandes pour le contrôle en profondeur à partir des mesures. Il enregistre également toutes les données de navigation.

L'interface avec l'extérieur se fait à l'extrémité haute du tube, elle comprend l'entrée d'eau pour le ballast, un interrupteur d'alimentation, une LED d'indication, une valve de pression permettant de garantir l'étanchéité du tube et le capteur de pression pour mesurer la profondeur.

### 2.1.2. Composants

Afin de conserver une approche minimaliste et bas coûts, nous avons cherché à utiliser le plus de composants standards et disponibles sur étagère.

#### Enceinte étanche



Fig. 4. – Enceintes étanches Blue Robotics

Le tube, les brides, les tapes et les connectiques étanches viennent du fabricant Blue Robotics (Fig. 4). Nous avons choisi d'utiliser un tube en acrylique de diamètre intérieur de 100mm, le diamètre minimal par rapport à la taille du ballast, et de longueur 400mm, la longueur maximale pour ce diamètre. L'intégrité pour ces caractéristiques de tube en acrylique est garantie jusqu'à une profondeur de 60m. Pour atteindre de plus grandes profondeurs, une amélioration simple serait d'utiliser les mêmes tubes en aluminiums garantis jusqu'à 800m.

#### Ballast

La technologie de ballast choisie est un vérin permettant de remplir un volume avec l'eau environnante pour augmenter la masse du système. La question s'est posée de réaliser notre propre ballast mais la conception de système d'étanchéi-



té peut s'avérer compliquer à mettre en œuvre et en cas de casse long à remplacer.

Nous nous sommes donc tourné vers un choix sur étagère avec le constructeur *Alexander Engel KG*. Le ballast est actionné par un moteur à courant continu 6V entraînant un train d'engrenages, la rotation est converti en translation pour le piston avec un système vis-écrou (Fig. 5).

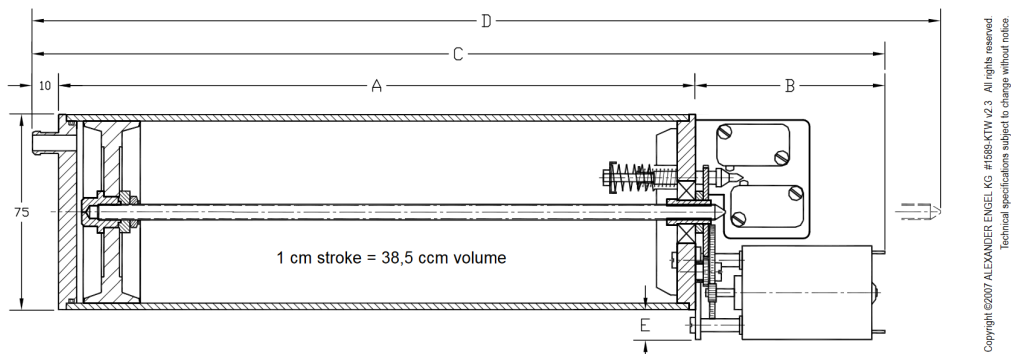


Fig. 5. – Dessin technique du Ballast Engel

Des aimants sont fixés sur un des pignons et permet grâce à un capteur à effet Hall de compter le nombre de tours pour remonter au déplacement du vérin et donc au volume d'eau dans le ballast et au débit.

### Capteur de pression

Le fabricant Blue Robotics propose également un ensemble de composants adaptés au montage sur les tapes. Pour les capteurs de pressions, trois choix sont disponibles :

- Bar02 :
  - fonctionnement jusqu'à 10m
  - résolution de 0.16mm
  - précision à  $\pm 2$ cm dans l'eau
- Bar30 :
  - fonctionnement jusqu'à 300m
  - résolution de 2mm
  - précision à  $\pm 20$ cm dans l'eau
- Bar100 :
  - fonctionnement jusqu'à 1020m
  - résolution de 3mm
  - précision à  $\pm 5.1$ m dans l'eau

Notre premier choix a été d'utiliser le capteur Bar30, car son fonctionnement jusqu'à 300m nous permettrait de réaliser sans crainte des essais en bassin et en extérieur. Mais nous avons été induit en erreur par la résolution de 2mm mise en avant sur le site du constructeur. En effet, la résolution est le plus petit écart de pression mesurable par le capteur, la précision quant à elle quantifie l'écart entre la mesure et la vérité. La documentation technique nous indique

une précision à  $\pm 204\text{cm}$ , donc à priori pas adaptée pour des essais en bassin de 3m de profondeur.

Il semble tout de même important de noter que la précision à  $\pm 204\text{cm}$  ne signifie pas que le capteur renvoie une valeur aléatoire entre  $-204\text{cm}$  et  $+204\text{cm}$ . De fait, nous avons utilisé le capteur pendant plusieurs essais relativement satisfaisants avant de se rendre compte de la faible précision du capteur. En effet, le capteur présente une dérive lente de la mesure avec le temps. Pour mettre en évidence ce comportement, nous avons placé le flotteur au fond du bassin et nous avons enregistré les mesures du capteur pendant une heure (Fig. 6). La profondeur réelle du capteur a été mesurée et était de  $2.5\text{m} \pm 0.1\text{m}$ .

On constate bien une dérive des mesures du capteur qui se stabilise au bout d'une vingtaine de minutes vers une profondeur dans les marge de précision du capteur. Bien que la précision absolue du capteur ne soit pas adaptée à nos essais, les variations de pressions étaient bien détectées et nous permettait de nous stabiliser autour d'une profondeur variant lentement.

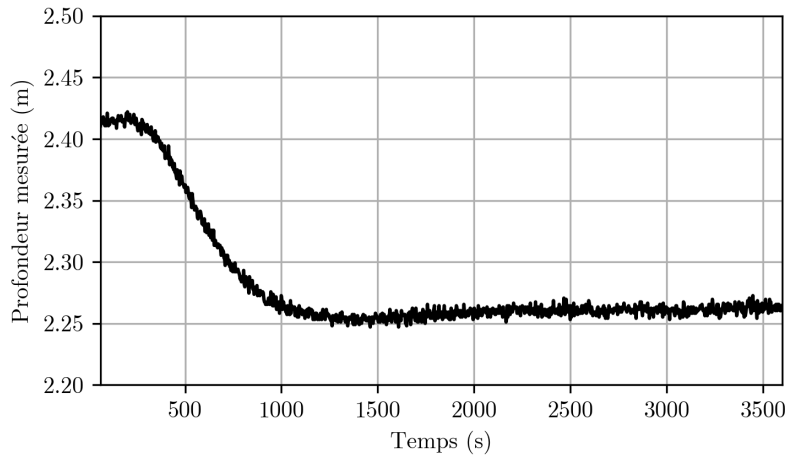


Fig. 6. – Évolution de la profondeur lors d'un essai du capteur Bar30 immobile

Fort de ces observations nous avons finalement fait le choix d'utiliser le capteur Bar02 et nous limiter à des profondeurs entre 0 et 10m. À titre de comparaison nous avons réalisé le même essai (Fig. 7), le capteur présente une dérive à l'allure similaire mais cette fois-ci contenue dans une marge d'environ 1cm. Une précision bien plus satisfaisante pour des essais dans un bassin de 3m de profondeur.

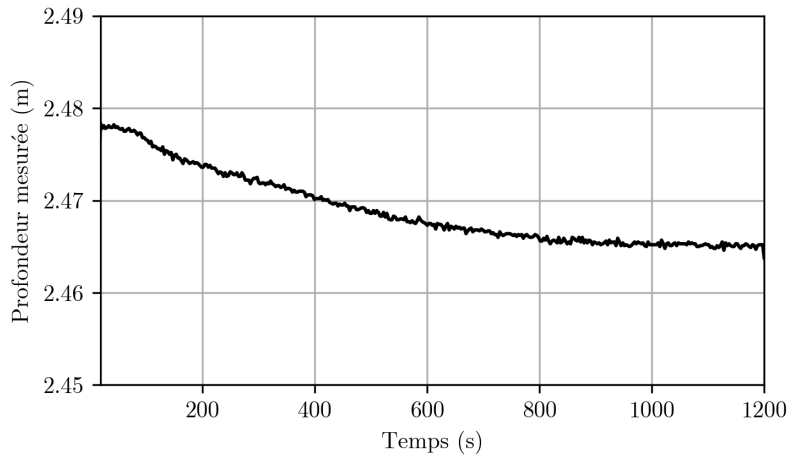


Fig. 7. – Évolution de la profondeur lors d'un essai du capteur Bar02 immobile

## Microcontrôleurs

Les deux microcontrôleurs sont des Arduino 33 BLE, ce choix a été fait pour la simplicité de mise en œuvre et au vu des composants disponibles au laboratoire. Le microcontrôleur de haut niveau calculant les estimateurs et les lois de commandes aurait pu être remplacé par un micro ordinateur type Raspberry Pi, plus énergivore mais plus polyvalent.

## 2.2. Flotteur planeur

### 2.2.1. Structure

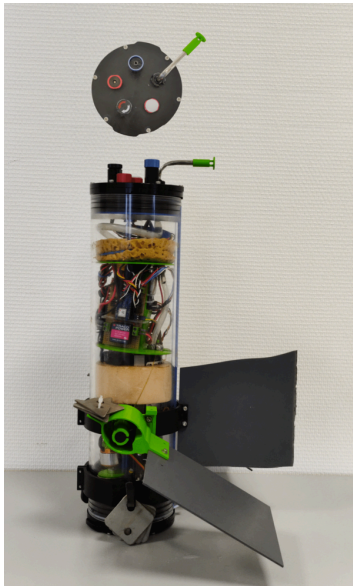


Fig. 8. – Le flotteur planeur

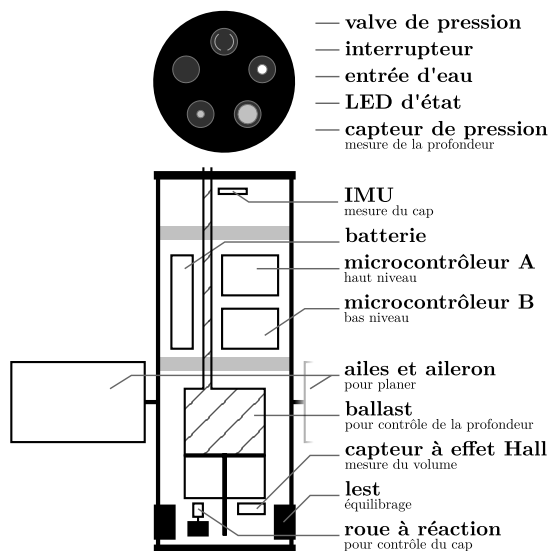


Fig. 9. – Schéma descriptif du flotteur planeur

Jusque là le flotteur se déplaçait sur une dimension, la verticale. Pour lui permettre de se déplacer dans une dimension supplémentaire, nous avons couplé l'avancement vertical à l'avancement horizontale en ajoutant des ailes (Fig. 10). Enfin pour atteindre le déplacement dans les trois dimensions, nous avons ajouté un volant d'inertie pour contrôler le cap.

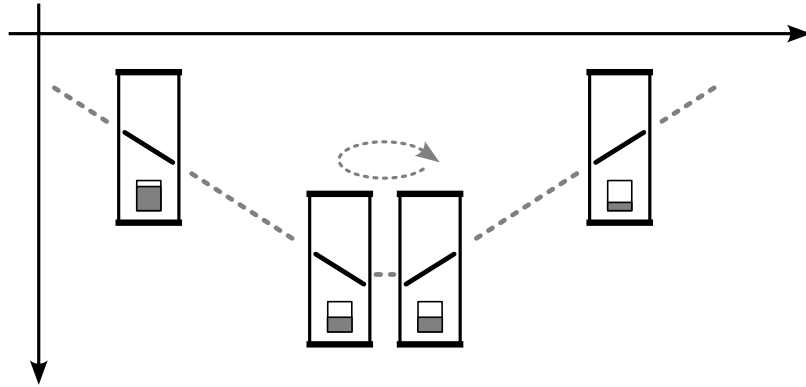


Fig. 10. – Déplacement d'un flotteur planeur

### 2.2.2. Composants

#### Ailes

Les ailes sous l'eau fonctionnent de la même manière que dans l'air, la force d'avancement est la force de portance que crée le fluide sur l'aile. Cette force peut s'écrire  $F_p = \frac{1}{2}\rho v^2 S C_x$  avec  $\rho$  la masse volumique du fluide,  $v$  sa vitesse,  $S$  la surface apparente et  $C_x$  le coefficient de portance dépendant de la forme de l'aile.

La masse volumique de l'eau étant environ 1000 fois supérieure à celle de l'air, la surface d'aile nécessaire pour faire planer le système dans l'eau est relativement petite. Les études hydrodynamiques étant hors de notre domaine de prédilection nous n'avons pas voulu nous attarder sur un dimensionnement précis des ailes. Nous nous sommes inspirés des planeurs existants pour choisir une surface raisonnable et avons découpé dans une plaque de PVC deux ailes planes de 17x15cm.

Avec ces ailes nous avons cherché quel serait l'angle de montage qui maximiserait la finesse du robot, c'est-à-dire le rapport entre la distance parcourue et la hauteur perdue. Une approche analytique aurait nécessité une connaissance précise des efforts hydrodynamiques en jeu et des paramètres physiques tels que les coefficients de portance et de traînée, nous avons donc préféré une approche expérimentale.

L'expérience consistait à lâcher le robot à la bordure du bassin avec un ballast rempli d'un volume fixe (Fig. 11) puis de relever la distance horizontale parcourue lorsqu'il atteint le sol (Fig. 12).

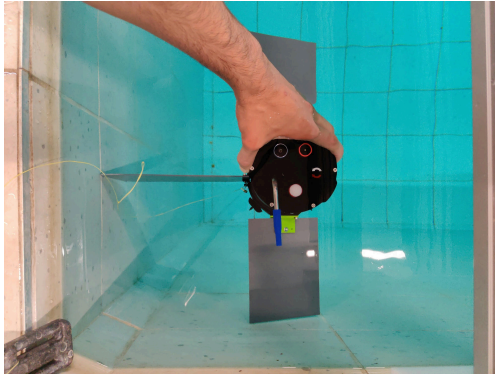


Fig. 11. – Lâcher du robot

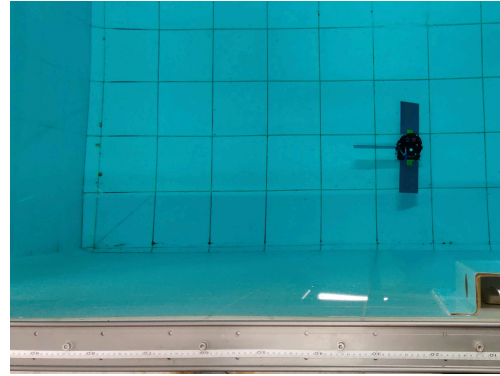


Fig. 12. – Mesure de la distance parcourue

Quatre lâchers ont été effectués avec des ailes inclinées à  $12^\circ$ ,  $20^\circ$ ,  $30^\circ$  et  $45^\circ$  (Fig. 13). La finesse maximale semble être atteinte pour un angle des ailes proche de  $30^\circ$ .

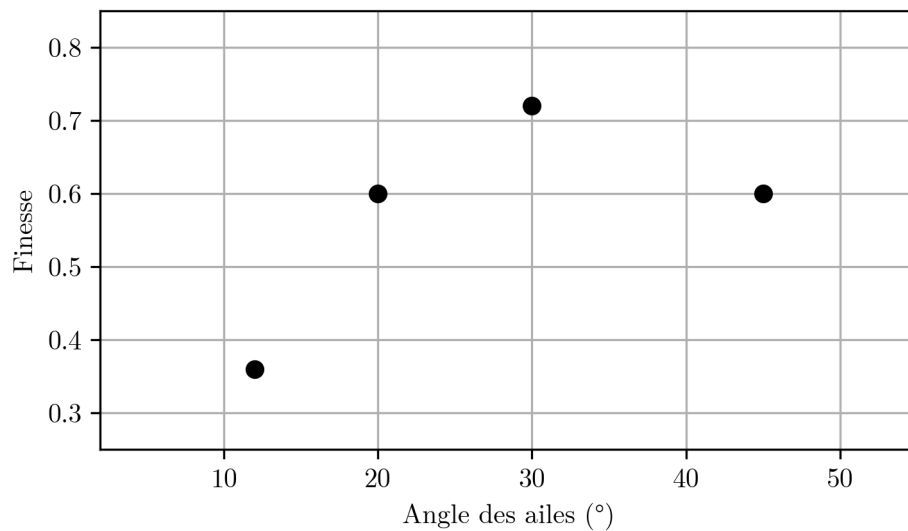


Fig. 13. – Mesure de la distance parcourue

Les premiers essais de validation du prototype ont été réalisés avec un montage des ailes fixes. Le robot avance donc lors de la descente mais doit se retourner avant d'entamer la remontée (Fig. 10), en plus d'une perte de temps et d'énergie, la deuxième conséquence et l'impossibilité d'ajouter un aileron de stabilisation. En effet, lors de la remontée l'aileron serait à l'avant du robot et rendrait le système instable.

Par conséquent, nous avons conçu des ailes mobiles, inclinées vers le bas lors de la descente et vers le haut lors de la remontée pour éviter le retournement. La solution que nous avons choisi a été de rendre flexible la liaison entre les ailes et le robot afin de créer une pseudo liaison pivot (Fig. 14). L'assemblage a été réalisé en impression 3D et en utilisant deux matériaux différents, du PLA pour la partie rigide et du TPU pour le lien flexible.

À l'arrêt et sous l'eau, les ailes sont équilibrés à l'horizontale pour ne pas freiner la rotation. Lors de la descente, l'eau applique un effort sur les ailes suffisant pour les plaquer en butée haute, lors de la remontée le même effort s'applique dans l'autre sens et plaque les ailes en butée basse.

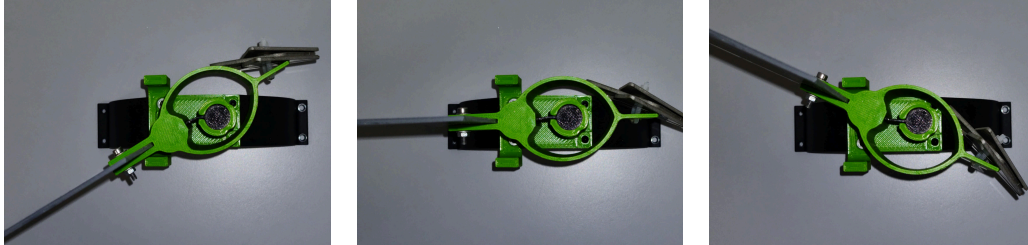


Fig. 14. – Position des ailes mobiles en montée, à l'arrêt et en descente

### Volant d'inertie

Pour contrôler le cap du robot nous avons ajouté au robot un volant d'inertie, composé d'un moteur entraînant une masse en rotation. Le choix du moteur et de son contrôleur de vitesse s'est fait parmi les composants disponibles, nous avons décidé d'utiliser un moteur brushless de 1400KV. 1 KV équivaut à 1 (tr/min)/V, donc en alimentant avec une batterie LiPo 3s (11.1V), le moteur peut à priori atteindre une vitesse de  $11.1 \times 1400 = 15540$  tr/min.

Pour le choix de la masse en rotation, le facteur limitant était l'encombrement. Nous avons donc cherché à fixer au moteur la masse la plus élevée pour l'espace disponible. Nous avons donc conçu un système pour attacher des rondelles en acier au moteur (Fig. 15)

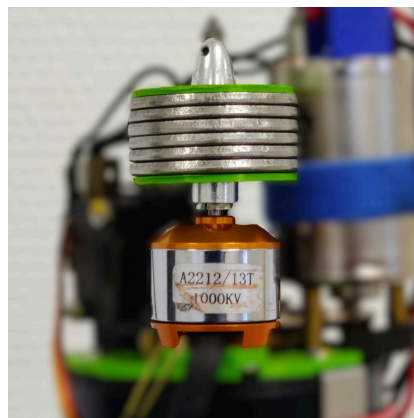


Fig. 15. – Volant d'inertie

### Centrale inertielle

Pour mesurer le cap du robot, nous avons choisi d'utiliser une centrale inertielle comprenant un accéléromètre, un gyroscope et un magnétomètre. Un filtrage avec fusion de ces données permet d'obtenir de meilleurs résultats qu'un magnétomètre simple.

Notre choix s'est porté vers la carte Sparkfun OpenLog Artemis. Abordable et peu encombrante elle présente aussi l'intérêt d'avoir un micrologiciel de fusion de données open source [12].

Reposer sur la centrale inertielle et son magnétomètre est presque le seul moyen d'obtenir son cap sous l'eau. Cette solution est très satisfaisante en milieu naturel, mais nous réalisons nos essais dans un bassin artificiel qui se trouve être particulièrement perturbé magnétiquement. Pour utiliser au mieux le capteur nous réalisons une calibration automatique à chaque début d'essai: le robot descend au milieu du bassin et réalise un tour complet grâce au volant d'inertie tout en relevant l'intensité du champ magnétique sur les deux axes du plan horizontal.

En traçant les valeurs du champ sur un axe en fonction de l'autre, nous obtenons un cercle non centré et de taille non unitaire, en supposant que les points de mesure sont équitablement réparti sur le cercle on calcule la moyenne sur chaque axe pour recentrer le cercle ainsi que les valeurs minimales et maximales pour le rendre unitaire (Fig. 16). Le cap est ensuite obtenu en utilisant la fonction atan2.

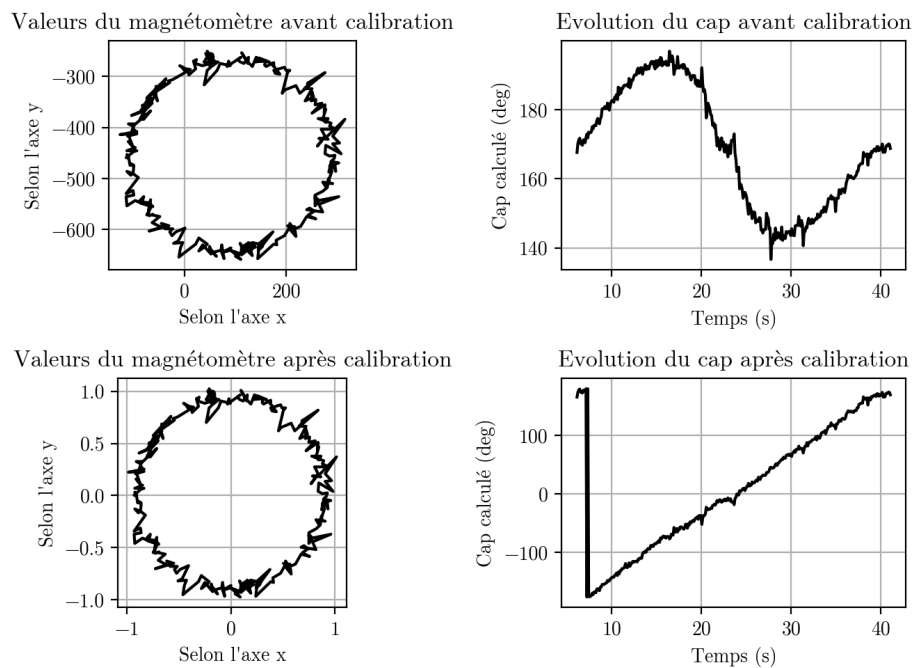


Fig. 16. – Calibration du magnétomètre

### 3. Contrôle de la profondeur

#### Hypothèses

- toutes les forces sont alignés à la verticale: le robot ne possède qu'un degré de liberté en translation selon  $e_z$

#### Notations

- $e_z$  : vecteur unitaire vertical descendant
- $g$  : accélération gravitationnelle
- $\rho$  : masse volumique de l'eau
- $m_0$  : masse du robot à vide (sans eau dans le ballast)
- $m_b$  : masse d'eau dans le ballast
- $m_a$  : masse ajoutée du robot
- $V$  : volume du robot
- $u$  : débit volumique en sortie du ballast

#### 3.1. Modèle dynamique

Le flotteur peut être modélisé par un solide soumis à trois force :

- le poids  $F_p = (m_0 + m_b)g.e_z$
- la poussée d'Archimède  $F_a = -\rho V g.e_z$
- le frottement  $F_v = -C_f \dot{z} |\dot{z}|.e_z$

Avec  $C_f = \frac{1}{2}c_x \rho S$ , où  $c_x$  est le coefficient de traînée dépendant de la forme du robot et  $S$  la surface de référence.  $C_f$  est constant et sera par la suite estimé expérimentalement.

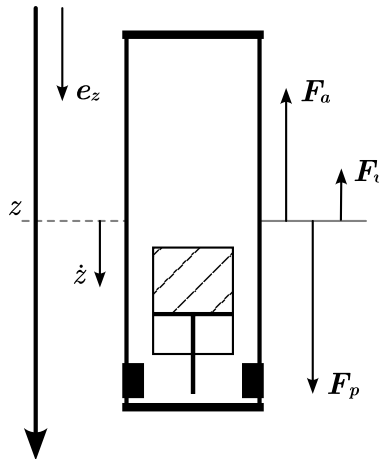


Fig. 17. – Schéma des forces simplifié

La masse ajoutée du robot  $m_a$  est une masse ajoutée à l'inertie du système pour rendre compte du déplacement de fluide par le robot. Ce paramètre, difficilement calculable mais constant sera aussi estimé expérimentalement.

La dynamique du système peut alors être écrite :



$$\begin{aligned}
(m_0 + m_b + m_a)\ddot{z} &= (\mathbf{F}_p + \mathbf{F}_a + \mathbf{F}_v) \cdot \mathbf{e}_z \\
\Leftrightarrow (m_0 + m_b + m_a)\ddot{z} &= (m_0 + m_b)g - \rho Vg - C_f \dot{z}|\dot{z}| \\
\Leftrightarrow \ddot{z} &= \frac{(m_0 + m_b)g - \rho Vg - C_f \dot{z}|\dot{z}|}{m_0 + m_b + m_a}
\end{aligned} \tag{1}$$

La commande  $u$  est le débit volumique en sortie du ballast tel que  $\dot{m}_b = \rho u$ . Elle apparaît dans la troisième dérivée de  $z$  :

$$\ddot{\ddot{z}} = \frac{(m_0 + m_b + m_a)(\rho u g - 2C_f \dot{z}|\dot{z}|) - \rho u g((m_0 + m_b)g - \rho Vg - C_f \dot{z}|\dot{z}|)}{(m_0 + m_b + m_a)^2} \tag{2}$$

Le degré relatif de ce système est de trois, le système se comporte comme un triple intégrateur.

## 3.2. Linéarisation par bouclage

### 3.2.1. Calcul de la commande

La commande doit permettre de contrôler la profondeur  $z$  du robot selon une profondeur désirée  $z_d$ , on peut écrire l'erreur  $e$  :

$$e = z - z_d \tag{3}$$

Le degré relatif étant de trois, on cherche à contrôler la troisième dérivée  $\ddot{\ddot{z}}$ , qu'on note  $\tilde{u}$  et qui vérifie l'équation :

$$\begin{aligned}
(\tilde{u} - \ddot{\ddot{z}}_d) + \alpha_2 \ddot{e} + \alpha_1 \dot{e} + \alpha_0 e &= 0 \\
\Leftrightarrow \tilde{u} - \ddot{\ddot{z}}_d &= -\alpha_2 \ddot{e} - \alpha_1 \dot{e} - \alpha_0 e \\
\Leftrightarrow \tilde{u} &= \ddot{\ddot{z}}_d + \alpha_2(\ddot{\ddot{z}}_d - \ddot{\ddot{z}}) + \alpha_1(\dot{\ddot{\ddot{z}}}_d - \dot{\ddot{\ddot{z}}}) + \alpha_0(z_d - z)
\end{aligned} \tag{4}$$

Où  $\alpha_0$ ,  $\alpha_1$  et  $\alpha_2$  sont choisis pour générer un polynôme caractéristique stable adapté à la dynamique du système.

À partir de l'Équation 2 on peut écrire la commande  $u$  :

$$u = \frac{\tilde{u}(m_0 + m_b + m_a)^2 + 2C_f \dot{z}|\dot{z}|(m_0 + m_b + m_a)}{\rho(m_a g + \rho g V + C_f \dot{z}|\dot{z}|)} \tag{5}$$

Avec  $\tilde{u}$  calculée avec l'Équation 4 et  $\ddot{\ddot{z}}$  calculée avec l'Équation 1. Les grandeurs  $z$  et  $m_b$  sont mesurées grâce aux capteurs du robot, et  $z$  est dérivée numériquement pour obtenir  $\dot{z}$ .

Cette commande cherche à linéariser le comportement du système à partir du modèle. Seulement, le modèle ne décrit pas parfaitement la réalité car certains phénomènes sont trop complexes à modéliser précisément (modèles de frottement, interaction fluide-solide). De plus pour les phénomènes considérés nous ne connaissons pas exactement leurs paramètres physiques. Nous avons donc intérêt à estimer correctement ces paramètres difficilement calculables.

### 3.2.2. Estimation du coefficient de frottement et de la masse ajoutée

Nous avons cherché à estimer le coefficient  $C_f$  et la masse ajoutée  $m_a$  expérimentalement, pour cela, nous avons réalisé des essais en « chute libre », c'est-à-dire à masse fixée pour que le robot coule et atteigne sa vitesse finale.

Le poids  $\mathbf{F}_p$  et la flottabilité  $\mathbf{F}_a$  sont constants pour chaque essai et on peut mesurer leur somme grâce à un peson. On note cette somme pour le  $i$ -ème essai  $F_i = (\mathbf{F}_p + \mathbf{F}_a) \cdot \mathbf{e}_z$ . La masse totale du robot est également constante et est mesurée avec une balance, on la note  $m_i$ .

La dynamique du système peut alors être simplifiée par rapport à l'Équation 1 :

$$(m_i + m_a)\ddot{z} = F_i + \mathbf{F}_v \cdot \mathbf{e}_z \quad (6)$$

On cherche alors à formuler l'équation sous la forme  $AX = B$  pour utiliser la méthode des moindres carrés et estimer  $C_f$  et  $m_a$  :

$$\begin{aligned} (m_i + m_a)\ddot{z} &= F_i - C_f \dot{z} |\dot{z}| \\ \Leftrightarrow m_a \ddot{z} + C_f \dot{z} |\dot{z}| &= F_i - m_i \ddot{z} \\ \Leftrightarrow (\ddot{z} \quad \dot{z} |\dot{z}|) \begin{pmatrix} m_a \\ C_f \end{pmatrix} &= F_i - m_i \ddot{z} \end{aligned} \quad (7)$$

Lors de nos essais nous mesurons la profondeur dix fois par seconde, on peut noter  $z_k$  la profondeur au  $k$ -ième pas de temps et calculer numériquement la vitesse  $\dot{z}_k$  et l'accélération  $\ddot{z}_k$ . On peut alors noter :

$$a_{i,k} = (\ddot{z}_k \quad \dot{z}_k |\dot{z}_k|), \quad X = \begin{pmatrix} m_a \\ C_f \end{pmatrix} \text{ et } b_{i,k} = F_i - m_i \ddot{z}_k \quad (8)$$

Et écrire la concaténation des  $a_{i,k}$  et des  $b_{i,k}$  calculés pour toutes les mesures de tous les essais effectués :

$$A = \begin{pmatrix} a_{0,0} \\ \dots \\ a_{i,k} \\ \dots \end{pmatrix} \text{ et } B = \begin{pmatrix} b_{0,0} \\ \dots \\ b_{i,k} \\ \dots \end{pmatrix} \quad (9)$$

On cherche alors  $X$  qui résout au mieux l'équation  $AX = B$ , en utilisant la méthode des moindres carrés.

Nous avons réalisé quatre essais en modifiant entre chaque la masse  $m_i$  et donc la force  $F_i$ . En appliquant la méthode des moindres carrés aux mesures on obtient :

$$C_f = 48.3 \text{ kg.m}^{-1} \text{ et } m_a = 5.70 \text{ kg} \quad (10)$$

Pour s'assurer de la cohérence des résultats obtenus nous avons calculé l'évolution du système avec le modèle et les paramètres calculés (Fig. 18).

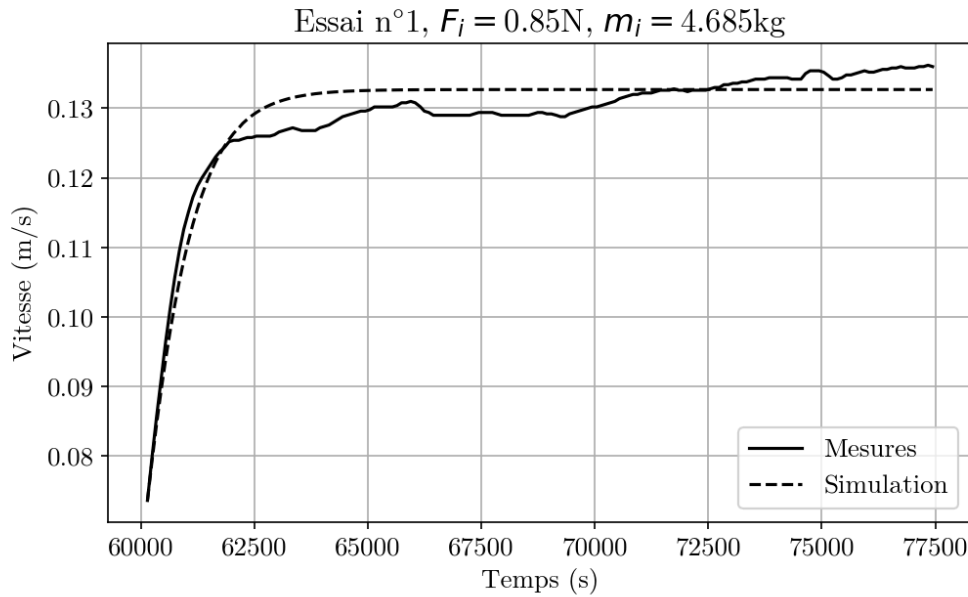


Fig. 18. – Comparaison de la vitesse mesurée et simulée pour un essai de chute libre

### 3.2.3. Estimation du volume

Contrairement au coefficient de frottement et à la masse ajoutée, le volume peut varier au cours de la mission à cause de la compressibilité de l'enceinte étanche ou de la présence de bulles d'air qui se délogent ou se dissolvent dans l'eau pendant la mission.

C'est pour cette raison que l'estimation du volume est faite en temps réel en se basant sur le modèle et sur les mesures du système. Le filtre utilisé pour y parvenir est un filtre de Kalman étendu au cas non-linéaire.

On peut écrire l'état de notre système avec un vecteur d'état  $x$  et un vecteur d'observation  $y$ :

$$x = \begin{pmatrix} z \\ \dot{z} \\ m_b \\ V \end{pmatrix} \text{ et } y = \begin{pmatrix} z \\ \dot{z} \\ m_b \end{pmatrix}$$

On peut alors décrire le comportement du robot avec des équations d'état de la forme :

$$\begin{cases} \dot{x} = f_c(x, u) \\ y = g(x) \end{cases}$$

Avec :

$$f_c(x, u) = \begin{pmatrix} \dot{z} \\ \ddot{z} \\ \dot{m}_b \\ \dot{V} \end{pmatrix} = \begin{pmatrix} \dot{z} \\ \frac{(m_0+m_b)g-\rho Vg-C_f\dot{z}|\dot{z}|}{m_0+m_b+m_a} \\ \rho u \\ 0 \end{pmatrix}$$

$$g(x) = \begin{pmatrix} z \\ \dot{z} \\ m_b \end{pmatrix}$$

On considère ici que le volume est constant. En réalité, le robot est compressible et le volume diminue avec la profondeur. Dans le filtre de Kalman, pour permettre à l'estimateur de prédire la valeur du volume, on définit un bruit d'évolution non nul qui permettra à la valeur de volume d'évoluer.

La fonction d'évolution  $f_c$  est bien non-linéaire. On peut cependant réduire la non-linéarité en négligeant l'influence de la variation de la masse du ballast  $m_b$  sur la masse inertielle  $m_0 + m_b + m_a$ . En effet, la masse du ballast varie entre 0 et 0.250kg, en comparaison la somme  $m_0 + m_a$  est de l'ordre de 10kg.

La fonction d'évolution devient :

$$f_c(x, u) = \begin{pmatrix} \dot{z} \\ \ddot{z} \\ \dot{m}_b \\ \dot{V} \end{pmatrix} = \begin{pmatrix} \dot{z} \\ \frac{(m_0+m_b)g-\rho Vg-C_f\dot{z}|\dot{z}|}{m_0+m_a} \\ \rho u \\ 0 \end{pmatrix}$$

On cherche à discrétiser le système avec la méthode d'Euler et formaliser le problème pour utiliser le filtre de Kalman étendu comme décrit dans [13] :

$$\begin{cases} x_{k+1} = x_k + dt f_c(x_k, u_k) = f(x_k, u_k) \\ y_k = g(x_k) \end{cases}$$

Pour linéariser le système, on suppose qu'on a une estimation  $\hat{x}$  de l'état, alors :

$$\begin{cases} x_{k+1} \simeq f(\hat{x}_k, u_k) + \frac{\partial f(\hat{x}_k, u_k)}{\partial x}(x_k - \hat{x}_k) \\ y_{k+1} \simeq g(\hat{x}_k) + \frac{\partial g(\hat{x}_k)}{\partial x}(x_k - \hat{x}_k) \end{cases}$$

En posant :

$$A_k = \frac{\partial f(\hat{x}_k, u_k)}{\partial x} \text{ et } C_k = \frac{\partial g(\hat{x}_k)}{\partial x} \quad (11)$$

On a :

$$\begin{cases} x_{k+1} \simeq A_k x_k + \underbrace{f(\hat{x}_k, u_k) - A_k \hat{x}_k}_{v_k} \\ \underbrace{y_k - g(\hat{x}_k) + C_k \hat{x}_k}_{z_k} \simeq C_k \hat{x}_k \end{cases}$$

Et on peut alors écrire l'approximation :

$$\begin{cases} x_{k+1} = A_k x_k + v_k + \alpha_k \\ z_k = C_k x_k + \beta_k \end{cases}$$

Avec  $\alpha_k$  le bruit d'évolution de matrice de covariance  $\Gamma_\alpha$  et  $\beta_k$  le bruit de mesure de matrice de covariance  $\Gamma_\beta$ . On peut appliquer le filtre Kalman étendu [13].

On explicite les matrices de transition  $A_k$  et d'observation  $C_k$  (Équation 11):

$$A_k = \begin{pmatrix} 1 & dt & 0 & 0 \\ 0 & 1 - \frac{2 dt C_f |\dot{z}|}{m_0 + m_a} & \frac{dt g}{m_0 + m_a} & -\frac{dt \rho g}{m_0 + m_a} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$C_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Nous avons implémenté les équations du filtre de Kalman étendu avec ces matrices de transition et d'observation, puis nous avons réalisé des essais pour trouver un réglage satisfaisant des matrices de covariances  $\Gamma_\alpha$  et  $\Gamma_\beta$  ainsi que des valeurs initiales de la matrice de covariance de  $x$ , noté  $\Gamma_x$ .

### 3.2.4. Essais

#### Commande par bouclage linéarisant avec un volume estimé constant

Nous avons réalisé un premier essai de la loi de commande par bouclage linéarisant sans estimation continue du volume par le filtre de Kalman. On considère donc que le volume est une constante.

Pour estimer cette constante approximativement, nous avons équilibré le flotteur dans l'eau puis mesuré sa masse. On a alors l'équilibre des forces  $\rho V g = m_{\text{eq}} g$ , donc le volume estimé est  $V = \frac{m_{\text{eq}}}{\rho}$ .

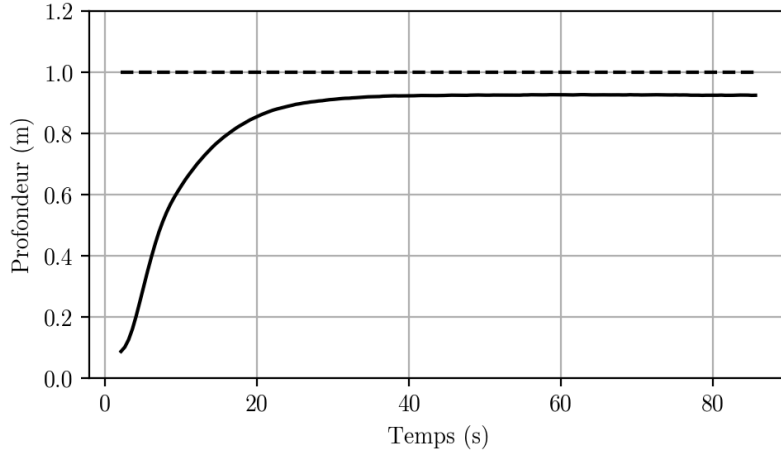


Fig. 19. – Évolution de la profondeur  $z$  pour une consigne  $z_d = 1\text{m}$ , avec un volume estimé constant,  $\alpha_2 = 1.5, \alpha_1 = 0.71, \alpha_0 = 0.105$

Nous avons fixé une consigne constante à 1m, le système évolue comme un système linéaire sans dépassement mais avec une erreur statique (Fig. 19). Essayons d'expliquer cette erreur:

On considère le système en régime établi, le robot est alors à l'équilibre donc :

$$\begin{cases} \dot{z} = 0 \\ u = 0 \\ m_0 + m_b = m_{\text{eq}} \end{cases} \quad (12)$$

Or selon l'Équation 5 on a:

$$u = \frac{\tilde{u}(m_0 + m_b + m_a)^2 + \overbrace{2C_f \dot{z} |\dot{z}| (m_0 + m_b + m_a)}^0}{\rho(m_a g + \rho g V + C_f \dot{z} |\dot{z}|)} \\ \Rightarrow \tilde{u} = 0$$

Donc en reprenant l'Équation 4, on peut écrire:

$$\tilde{u} = \ddot{z}_d + \alpha_2(\ddot{z}_d - \ddot{z}) + \alpha_1(\dot{z}_d - \dot{z}) + \alpha_0(z_d - z)$$

Avec  $z_d$  constant donc  $\ddot{z}_d = \dot{z}_d = z_d = 0$ , l'équation devient:

$$\alpha_2 \ddot{z} + \alpha_0 \overbrace{(z_d - z)}^e = 0 \\ \Leftrightarrow e = \frac{\alpha_2}{\alpha_0} \ddot{z}$$

La dérivée seconde  $\ddot{z}$  n'est pas dérivé numériquement comme  $\dot{z}$  mais calculé à partir du modèle grâce à l'Équation 1 :

$$\ddot{z} = \frac{\overbrace{(m_0 + m_b)}^{m_{\text{eq}}} g - \rho V g - \overbrace{C_f \dot{z}}^0 |\dot{z}|}{\underbrace{(m_0 + m_b) + m_a}_{m_{\text{eq}}}}$$

On a donc:

$$e = \frac{\alpha_2 (m_{\text{eq}} - \rho V) g}{\alpha_0 m_{\text{eq}} + m_a}$$

Si notre estimation de volume est exacte alors  $V = \frac{m_{\text{eq}}}{\rho}$  et l'erreur statique s'annule :

$$e = \frac{\alpha_2 (m_{\text{eq}} - \rho \frac{m_{\text{eq}}}{\rho}) g}{\alpha_0 m_{\text{eq}} + m_a} = 0$$

Mais en réalité la valeur du volume  $V$  n'est pas exacte d'où l'erreur statique

### Commande par bouclage linéarisant avec estimation continue du volume

Nous avons donc ensuite réalisé le même essai avec l'estimation continue du volume grâce au filtre de Kalman.

Le système parvient à se stabiliser et annuler l'erreur statique (Fig. 20) mais présente un dépassement. Ce n'est pas dû à la commande mais au filtre de Kalman, en effet, le filtre se base sur un modèle dynamique du robot qui ne décrit pas parfaitement la réalité, particulièrement en dynamique où les interactions entre le fluide et le robot sont très complexes et nécessiterait une étude plus poussée d'hydrodynamique.

En statique, le système ne subit plus que son poids et la poussée d'Archimède. Ces deux efforts sont bien modélisés et permettent lorsque la vitesse diminue de converger vers la bonne valeur de volume (Fig. 20).

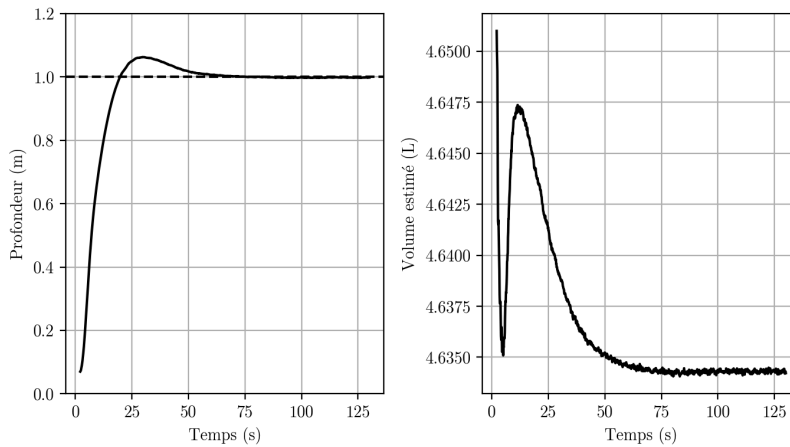


Fig. 20. – Évolution de la profondeur  $z$  et de l'estimation du volume pour une consigne  $z_d = 1\text{m}$ ,  $\alpha_2 = 1.5$ ,  $\alpha_1 = 0.71$ ,  $\alpha_0 = 0.105$

## Observation de la compressibilité du robot

Pour cette expérience nous avons réalisé des paliers de profondeurs avec le robot. Pour chaque palier, le système se stabilise en déterminant la valeur du volume à cette profondeur.

Nous observons bien une variation du volume avec la profondeur, pour des profondeurs de 0.5m, 1.0m, 1.5m et 2.0m le volume du robot converge respectivement vers  $4.6332 \pm 2.10^{-4}$  L,  $4.6325 \pm 2.10^{-4}$  L,  $4.6318 \pm 2.10^{-4}$  L et  $4.6312 \pm 2.10^{-4}$  L. C'est une variation d'environ 1.3mL/m sur cette plage de profondeur.

Bien que ces résultats nous montrent la tendance de l'évolution du volume avec la profondeur, ce ne sont pas de vraies mesures de volume. En effet, en régime établi, l'estimateur converge vers une valeur de volume tel que  $\rho g V = (m_0 + m_b)g$ , donc la valeur estimée du volume dépend de la mesure de  $m_0$ , si elle est mauvaise la valeur du volume ne représentera pas la réalité. Malgré tout, le système convergera vers sa valeur de consigne avec un volume virtuel.

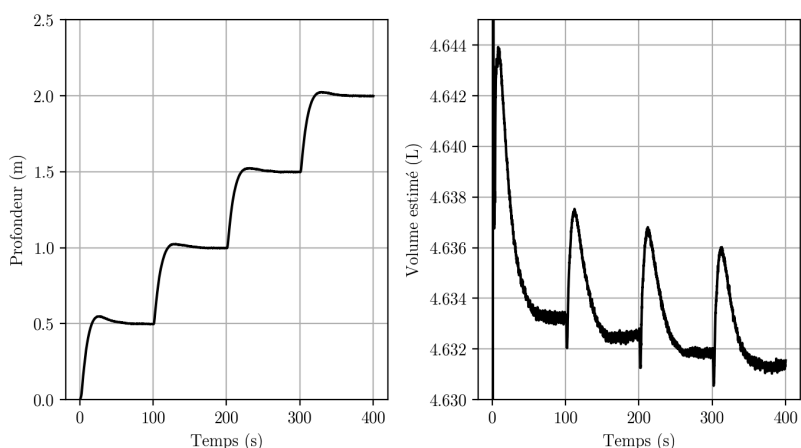


Fig. 21. – Évolution de la profondeur  $z$  et de l'estimation du volume pour des consignes en paliers  $z_d = [0.5\text{m}, 1.0\text{m}, 1.5\text{m}, 2.0\text{m}]$ ,  
 $\alpha_2 = 1.5, \alpha_1 = 0.71, \alpha_0 = 0.105$

## Perturbation par ajout de masse

Pour éprouver la robustesse de la commande à une mauvaise mesure de  $m_0$  ou à une perturbation pendant la mission nous avons modifié la masse du robot pendant la mission.

Une fois le système stabilisé à 1m, nous avons placé une masse de  $50 \pm 2.5\text{g}$  à 220s. Naturellement, le robot a coulé puis s'est stabilisé de nouveau à 1m et le volume estimé a convergé vers une nouvelle valeur. La différence entre l'ancien volume et le nouveau est de  $4.6332 - 4.5843 = 0.0489\text{L}$ , ce volume d'eau correspond à une masse de  $0.0489 \times 997 = 48.9\text{g}$  en accord avec la masse du poids ajouté.



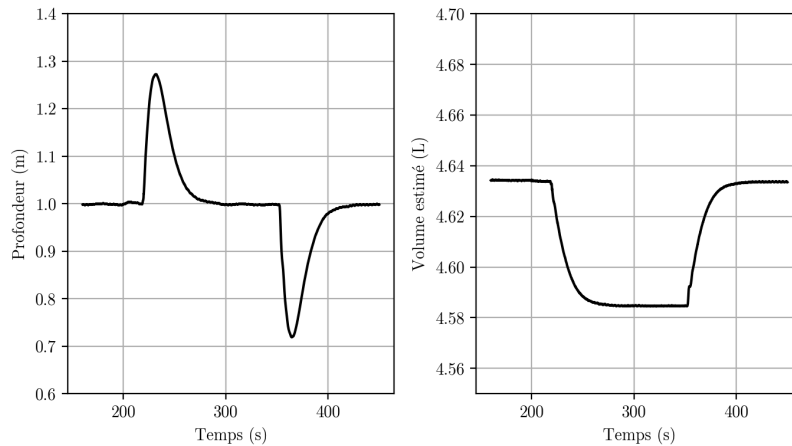


Fig. 22. – Évolution du système pour une consigne à 1m, avec perturbation,  $\alpha_2 = 1.5, \alpha_1 = 0.71, \alpha_0 = 0.105$

### 3.2.5. Résultats

Le contrôle en profondeur du robot avec la commande par bouclage linéarisante est satisfaisant et robuste face aux perturbations. L'estimation en temps réel du volume grâce au filtre de Kalman étendu compense les erreurs de mesures ou les variations de la masse et du volume permettant de ne pas avoir à mesurer ou équilibrer précisément la flottabilité du robot.

## 3.3. Commande sans modèle

### 3.3.1. Théorie

La commande sans modèle [14] est une contribution aux correcteurs PID « intelligents ». Elle repose sur une modélisation ultra-locale de la relation entre la grandeur d'entrée et de sortie et une fonction d'estimation chargée de compenser les non-linéarités, les perturbations extérieures et tous les termes non modélisés dans une seule variable.

Le comportement d'un système à une seule entrée et une seule sortie respecte une équation différentielle de la forme  $E(y, \dot{y}, \dots, y^{(a)}, u, \dot{u}, \dots, u^{(b)})$ . Le modèle ultra-local peut être écrit :

$$y^{(n)} = F(t) + \alpha u(t) \quad (13)$$

avec

- l'ordre de dérivation  $n$ , selon [14] cet ordre est différent du degré du système déterminé au Chapitre 3.1 et peut être choisi égal à 1 ou 2 pour tous les systèmes
- le coefficient alpha, choisi pour que  $u$  et  $y$  soient du même ordre de grandeur
- la variable  $F$ , estimé avec grâce à  $y$  et  $u$  est la variable qui compense les non-linéarités, les perturbations extérieures et tous les termes non modélisés

### 3.3.2. Implémentation

Le système se comporte comme un triple intégrateur, l'Équation 2 montre qu'il y a trois degrés de dérivation entre la sortie et l'entrée. La théorie de la commande sans modèle annonce qu'il n'y a pour autant pas besoin de choisir l'ordre de dérivation  $n$  égal à 3 mais qu'un ordre de 1 ou 2 suffit.

Il n'existe dans la littérature pas d'exemple réel de système triple intégrateur commandé par une méthode sans modèle. Nous l'avons donc implémenté sur le flotteur pour l'expérimenter grâce au code écrit par Emmanuel DELALEAU, un professeur ayant contribué au développement de la commande sans modèle.

En fixant l'ordre de dérivation  $n$  à 1 et 2, aucune tentative de réglage n'a montré de résultat satisfaisant. Ce résultat peut paraître étonnant par rapport aux annonces des publications sur le sujet mais est plus en accord avec les théories de l'automatique classique.

Nous avons donc fixé l'ordre de dérivation à 3 pour implémenter un correcteur PDD « intelligent » de la forme :

$$u = \frac{\ddot{z}_d + K_{dd}(\ddot{z}_d - \ddot{z}) + K_d(\dot{z}_d - \dot{z}) + K_p(z_d - z) - F}{\alpha} \quad (14)$$

Pour le premier essai  $F$  a été fixé à 0 pour observer l'évolution du système avec un correcteur PDD traditionnel (Fig. 23). Après quelques itérations, un réglage satisfaisant a été trouvé pour diminuer les oscillations et converger vers une valeur de profondeur proche de la consigne. Mais dans tous les cas le système présente une erreur statique.

Celle-ci peut s'expliquer par une non-linéarité du système. En effet, l'actionneur présente une zone morte, il existe un débit minimum en dessous duquel l'actionneur ne peut pas atteindre la consigne. Donc lorsque l'erreur statique devient trop faible, le ballast reçoit des entrées trop faibles et ne s'actionne plus.

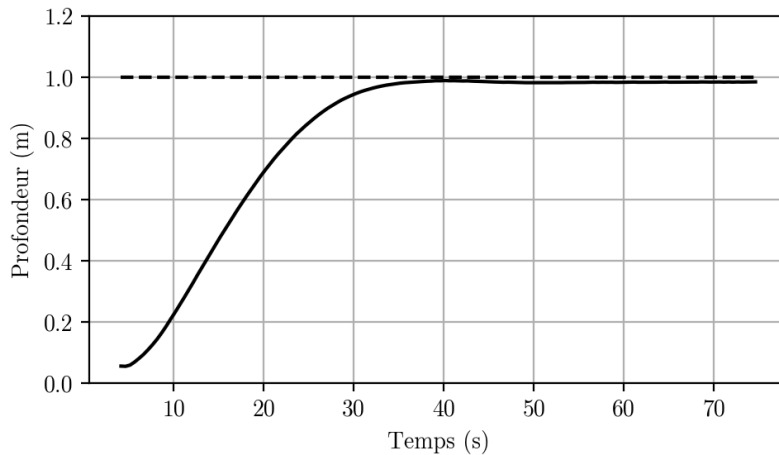


Fig. 23. – Évolution du système pour une consigne à 1m, correcteur PDD  
 $K_p = 0.021, K_d = 0.3, K_{dd} = 0.5, \alpha = 1000$

L'ajout de l'estimateur de  $F$  devrait compenser cette non-linéarité de l'actionneur. Nous avons réalisé le même essai en calculant  $F$  à chaque pas de temps grâce à l'estimateur (Fig. 24). Le résultat est satisfaisant, l'erreur statique est annulée.

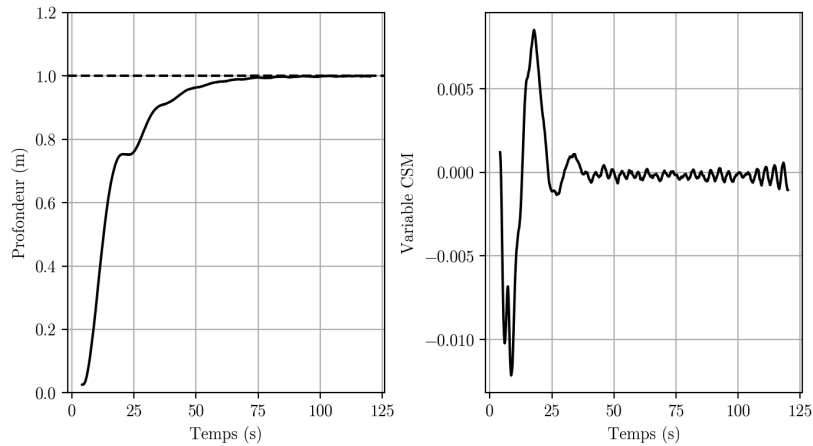


Fig. 24. – Évolution du système pour une consigne à 1m, correcteur PDD et estimation sans-modèle  $K_p = 0.021$ ,  $K_d = 0.3$ ,  $K_{dd} = 0.5$ ,  $\alpha = 1000$

### 3.3.3. Résultats

Nos expérimentations ont commencé à montrer que la commande sans-modèle était utilisable pour un système triple intégrateur en fixant l'ordre de dérivation à 3.

Pour l'instant il s'agit d'observations expérimentales et des travaux sont en cours pour mieux cerner les limites de la commande sans modèle et l'influence de l'ordre de dérivation pour ce genre de système.

## 4. Déplacement dans l'espace

Avec le contrôle de la profondeur maîtrisé, nous avons voulu ajouter au robot la possibilité de se déplacer dans l'espace, en contrôlant son cap avec un volant d'inertie et en couplant sa vitesse verticale et horizontale avec l'ajout d'ailes.

### 4.1. Avance

La conception des ailes est détaillé dans le Chapitre 2.2.2, les essais nous ont montré que les ailes permettent de faire avancer le robot en alternant les descentes et les montées (Fig. 25).

Contrairement à la profondeur mesurée par le capteur de pression et le cap mesuré par la centrale inertielle, la mesure de la position est plus délicate. Nous avons envisagé plusieurs solutions:

- capteur GPS: seulement utilisable en extérieur et en surface pour recalibrer la position puis naviguer à l'estime avec une centrale inertielle et un capteur de vitesse
- suivi optique et communication: en installant une caméra au dessus du bassin, la position peut facilement être obtenue, mais la complexité réside dans l'envoi des données au robot sous l'eau [15]
- sonar: pour se positionner en connaissant la géométrie du bassin d'essai [16], l'intégration aurait nécessité de nombreuses modifications sur le flotteur et le traitement relativement complexe
- capteur infrarouge: même stratégie que le sonar avec moins de portée mais plus simple à intégrer au flotteur, une ceinture de capteurs infrarouge pourraient être ajoutés à l'intérieur de l'enceinte étanche pour mesurer des distances aux murs

Aucune solution n'étant simple à tester, nous avons décidé de rester en boucle ouverte pour l'avance et se concentrer sur le contrôle en profondeur et en cap.

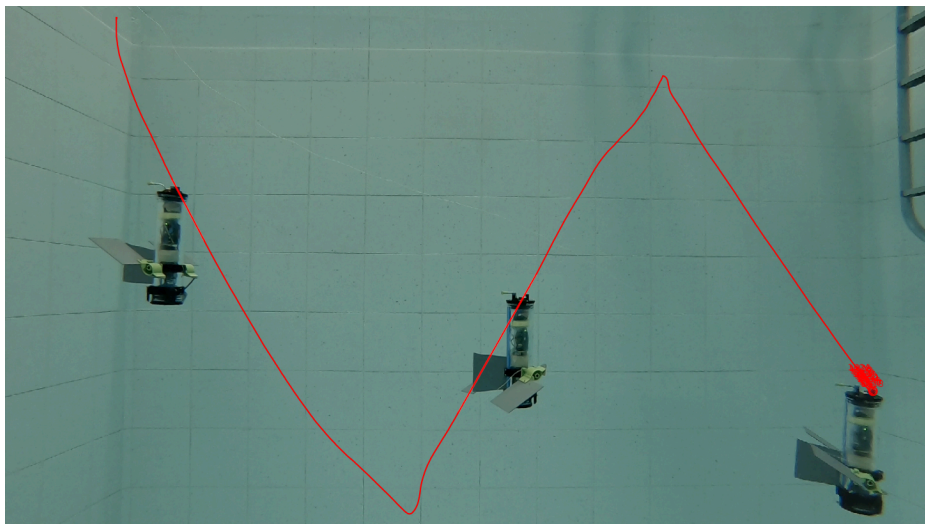


Fig. 25. – Trajectoire du flotteur planeur

## 4.2. Contrôle du cap

### Hypothèses

- le robot ne possède qu'un degré de rotation selon l'axe  $e_z$

### Notations

- $e_z$  : vecteur unitaire vertical descendant
- $\theta_r$  : cap du robot
- $\omega_r$  : vitesse angulaire du robot
- $\omega_v$  : vitesse relative du volant d'inertie par rapport au robot
- $J_r$  : moment d'inertie du robot sans le volant
- $J_v$  : moment d'inertie du volant
- $C_\omega$  : coefficient de frottement en rotation
- $u$  : accélération du volant  $u = \dot{\omega}_v$

#### 4.2.1. Dynamique du volant d'inertie

Le volant d'inertie est mis en mouvement par le moteur fixé au robot, la vitesse absolue du volant est donc  $\omega_r + \omega_v$ .

Le robot est soumis à deux actions :

- le couple de frottement  $C_f = -C_\omega \omega_r |\omega_r| e_z$
- le moment du volant sur le robot  $C_v = -J_v (\dot{\omega}_r + \dot{\omega}_v)$

On peut alors écrire le principe fondamental de la dynamique en rotation :

$$\begin{aligned}
 J_r \dot{\omega}_r &= (C_f + C_v) \cdot e_z \\
 \Leftrightarrow J_r \dot{\omega}_r &= -C_\omega \omega_r |\omega_r| - J_v (\dot{\omega}_r + \dot{\omega}_v) \\
 \Leftrightarrow J_r \dot{\omega}_r &= -C_\omega \omega_r |\omega_r| - J_v (\dot{\omega}_r + u) \\
 \Leftrightarrow (J_r + J_v) \dot{\omega}_r &= -C_\omega \omega_r |\omega_r| - J_v u
 \end{aligned} \tag{15}$$

#### 4.2.2. Commande par bouclage linéarisant

La commande doit permettre de contrôler le cap du robot  $\theta_r$  selon un cap désiré  $\theta_{r,d}$ , on peut écrire l'erreur  $e$ :

$$e = \theta_r - \theta_{r,d} \tag{16}$$

Selon l'Équation 15 la commande  $u$  apparaît à la deuxième dérivée de  $\theta_r$ , le degré relatif est de deux, on cherche à imposer  $\ddot{\theta}_r = \dot{\omega}_r$ , qu'on note  $\tilde{u}$  et qui vérifie l'équation :

$$\begin{aligned}
 (\tilde{u} - \ddot{\theta}_{r,d}) + \alpha_1 \dot{e} + \alpha_0 e &= 0 \\
 \Leftrightarrow \tilde{u} = \ddot{\theta}_{r,d} + \alpha_1 (\dot{\theta}_r - \dot{\theta}_{r,d}) + \alpha_0 (\theta_r - \theta_{r,d})
 \end{aligned} \tag{17}$$

Où  $\alpha_0$ ,  $\alpha_1$  et  $\alpha_2$  sont choisis pour générer un polynôme caractéristique stable adapté à la dynamique du système.

À partir de l'Équation 15 on peut écrire la commande  $u$  en fonction de  $\tilde{u}$  :

$$u = -\frac{(J_r + J_v)\tilde{u} + C_\omega\omega_r|\omega_r|}{J_v} \quad (18)$$

Avec  $\tilde{u}$  calculé à partir de l'Équation 17 et  $\omega_r$  dérivée numériquement des mesures du magnétomètre ou mesurée avec le gyroscope de la centrale inertielle.

À noter que la commande du système est l'accélération du volant, pour l'implémentation il convient donc d'intégrer la commande à chaque pas de temps car le contrôle du moteur se fait en vitesse.

Cette loi de commande fonctionne tant que le moteur n'arrive pas à saturation en vitesse de rotation. Dans la réalité, cette saturation est atteinte relativement vite donc cette loi de commande seule ne permet donc pas de contrôler entièrement le cap du robot.

Cette limitation est bien visible dans le cas d'un maintien de cap constant avec perturbation (Fig. 26). Dans cet exemple, le robot est stabilisé autour d'un cap constant et nous perturbons trois fois dans le même sens le robot. Dans les deux premiers cas le cap se rétablit correctement grâce à la loi de commande mais on remarque que pour revenir au même cap le moteur a accéléré. La vitesse maximale du moteur est de 14000 tr/min et effectivement, lors de la troisième perturbation le moteur l'atteint et ne peut plus accélérer davantage pour corriger la perturbation.

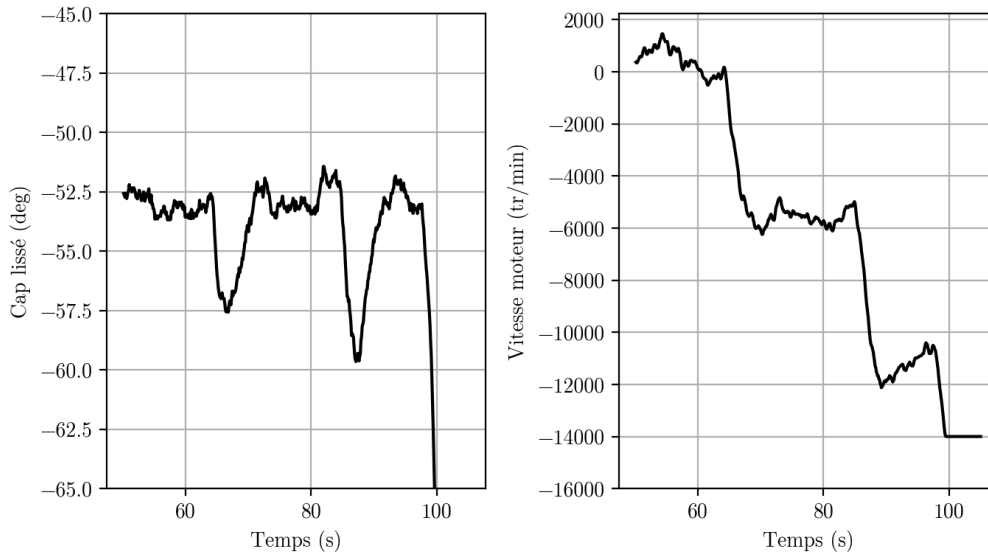


Fig. 26. – Maintien en cap avec perturbation et saturation de la vitesse moteur

#### 4.2.3. Commande en cycle

Nous avons donc cherché une stratégie de commande moins traditionnelle pour parvenir à atteindre tous les angles en tenant compte de la vitesse maximale du moteur du volant d'inertie.

## Comportement libre

Tout d'abord, observons le comportement sans actionnement d'un solide en rotation dans l'eau, on peut écrire l'Équation 15 avec une commande nulle :

$$\dot{\omega}_r + \underbrace{\frac{C_\omega}{J_r + J_v}}_K \omega_r |\omega_r| = 0$$

On se place dans le cas où la vitesse de rotation initiale  $\omega_{r,0} > 0$ , on conçoit que sans actionnement la vitesse de rotation diminuera mais ne changera pas de signe. On trouve alors une solution pour l'équation différentielle dans le cas  $\omega_r > 0$  :

$$\omega_r(t) = \frac{1}{Kt + c_1}$$

Avec  $c_1$  tel que :

$$\begin{aligned}\omega_r(t=0) &= \omega_{r,0} = \frac{1}{c_1} \\ \Leftrightarrow c_1 &= \frac{1}{\omega_{r,0}}\end{aligned}$$

Pour calculer l'évolution de l'angle de rotation, on cherche à intégrer de 0 à  $t$  la vitesse de rotation :

$$\theta_r(t) = \frac{1}{K} \ln(Kt + c_1) + c_2$$

Avec  $c_2$  tel que :

$$\begin{aligned}\theta_r(t=0) &= \theta_{r,0} = \frac{1}{K} \ln(c_1) + c_2 \\ \Leftrightarrow c_2 &= \theta_{r,0} - \frac{1}{K} \ln(c_1)\end{aligned}$$

En rotation libre avec des frottements visqueux, l'angle du robot évolue selon un logarithme népérien. Or,  $\lim_{t \rightarrow +\infty} \ln(t) = +\infty$ , donc en donnant une impulsion au robot, il tournera à l'infini.

## Effet d'une consigne en créneau

Nous avons simulé l'effet d'un créneau sur la vitesse du volant. Cela correspond à tourner à vitesse constante pendant une certaine durée puis s'arrêter (Fig. 27).

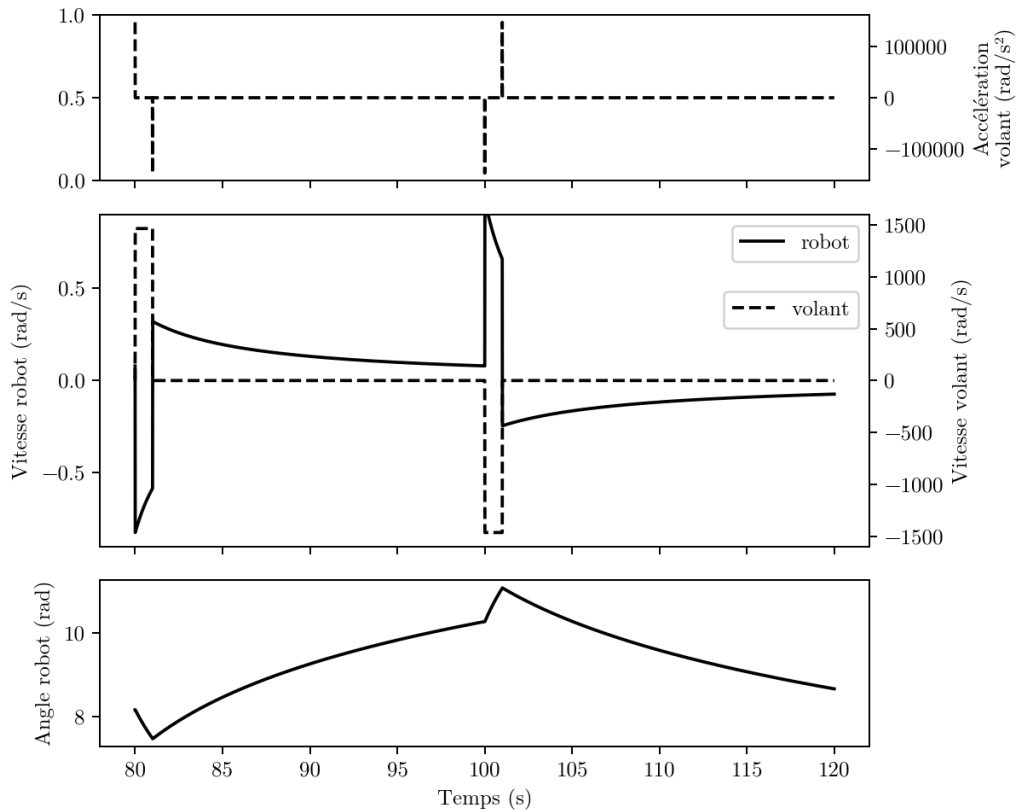


Fig. 27. – Simulation de l'effet d'une consigne en créneau

Décomposons l'effet du créneau sur le comportement du robot :

- À  $t = 80$ s, le volant est accéléré par un Dirak, le robot accélère à l'opposé et proportionnellement au rapport des inerties (Équation 15)
- Entre  $t = 80$ s et  $t = 81$ s, l'accélération du volant est nulle, l'angle du robot décroît avec une évolution en logarithme
- À  $t = 81$ s, le volant subit un Dirak opposé et de même valeur qu'au démarrage. La vitesse du volant étant resté constante, le volant s'arrête. Le robot subit également l'accélération opposée et de même valeur qu'au démarrage (Équation 15). Seulement, le robot a ralenti avec les frottements, il ne s'arrête donc pas et change de sens de rotation.
- Entre  $t = 81$ s et  $t = 100$ s, l'accélération du volant est nulle, l'angle du robot croît avec une évolution en logarithme
- À partir de  $t = 100$ s, un créneau opposé est appliqué, on observe le même effet dans l'autre sens

Une telle consigne permet donc de tourner dans les deux sens sans être limité par la saturation en vitesse du volant.

### Loi de commande

En se basant sur ces observations, nous avons réalisé une première loi de commande simple pour contrôler le cap.



Le principe est de répéter périodiquement la consigne en créneau en changeant le sens du créneau selon le signe de l'erreur en cap.

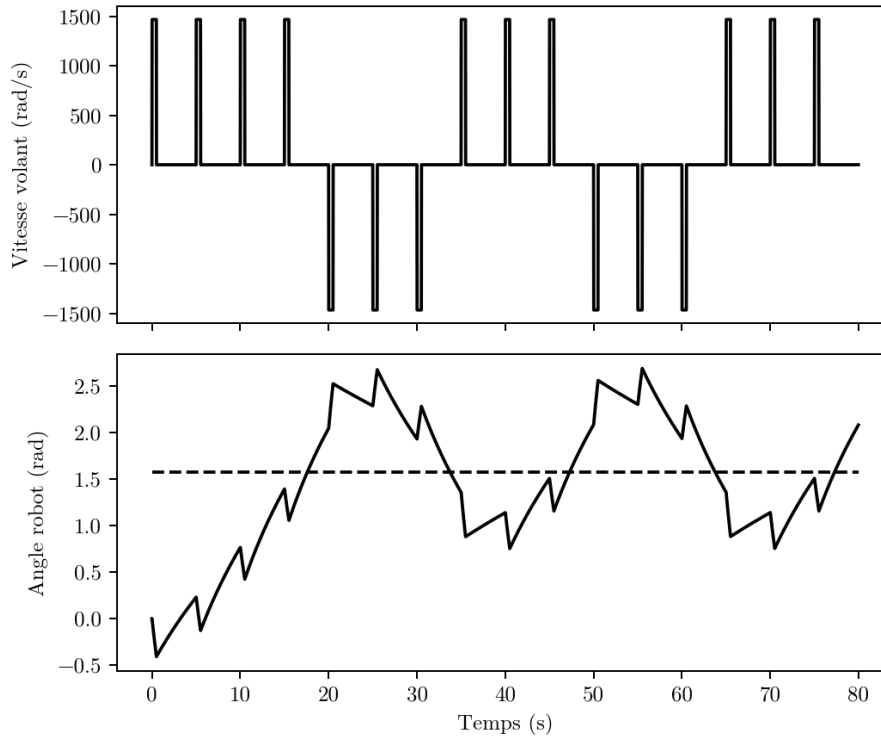


Fig. 28. – Loi de commande en cycle

Cette commande permet d'osciller autour de la consigne. Les oscillations sont inévitables notamment à cause de la structure de la loi de commande qui impose des temps de cycle fixe.

#### 4.2.4. Commande hybride

Nous avons donc combiné la loi de commande en cycle avec la loi de commande par bouclage linéarisant.

La première permet depuis n'importe quelle position de s'approcher suffisamment de la consigne pour que la loi de commande par bouclage linéarisant puisse réguler correctement sans atteindre la saturation en vitesse du moteur.

Pour observer ce comportement nous avons donné une consigne de cap fixe au robot puis une fois que ce dernier était stabilisé, nous l'avons perturbé en le tournant à la main (Fig. 29). Pour se stabiliser de nouveau, le volant d'inertie suit la loi de commande en cycle pour se rapprocher de la valeur consigne puis suit la loi de commande par bouclage linéarisant lorsque l'erreur en cap est suffisamment faible.

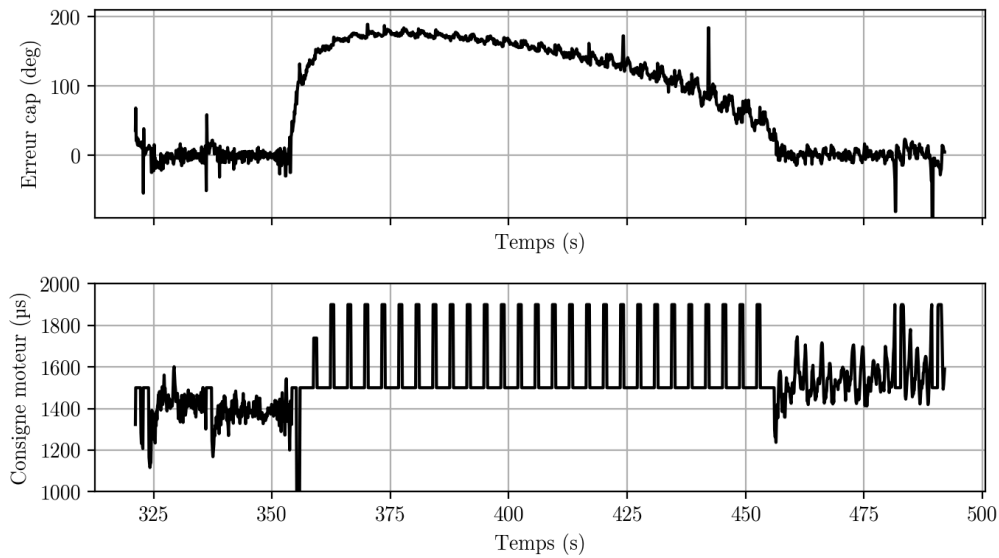


Fig. 29. – Évolution de l'erreur en cap et de la consigne moteur avec perturbation

#### 4.2.5. Résultats

La commande hybride du cap permet d'atteindre n'importe quelle position et permet donc d'utiliser un volant d'inertie pour contrôler le cap du robot. C'est une solution facile à mettre en œuvre et économe car les composants sont communs et n'ont pas besoin d'être étanchéifié. De plus, ils sont relativement petits et peuvent facilement être intégrés à un système existant.

Cette solution est évidemment moins rapide et moins robuste que des propulseurs mais présente un compromis intéressant dans certaines situations qui demandent par exemple le moins de perturbations possibles ou qui nécessitent de descendre en profondeur à bas coûts.

## 5. Conclusion

### 5.1. Résultats

Ce rapport présente le travail réalisé durant des ces six mois de stage de fin d'étude. Il en émerge plusieurs résultats intéressants.

Plusieurs lois de commande ont été développées pour contrôler la profondeur d'un AUV. Une loi classique par bouclage linéarisant avec estimation du volume en continue et une loi basée sur la théorie de la commande sans modèle. Les deux ont été expérimentalement validées.

Par la même occasion, ces expérimentations ont permis de valider l'achat de ballast sur étagère pour la conception des futurs AUVs.

Un prototype d'aile mobile passive a également été conçu pour coupler la descente verticale d'un flotteur à l'avance horizontale sans changer d'orientation. Des travaux complémentaire seront nécessaire pour mesurer l'avance et la contrôler précisément.

Enfin, une stratégie de commande a été développée pour utiliser un volant d'inertie comme actionneur pour contrôler le cap d'un AUV. Des expérimentations ont été conduites et valident les résultats théoriques.

D'un point de vue personnel, ce stage m'a permis de me conforter dans le choix de continuer en tant que doctorant au laboratoire de l'ENSTA Bretagne.

### 5.2. Publications

Une publication est en cours pour ICRA (IEEE International Conference on Robotics and Automation). Le sujet de l'article est l'utilisation de la commande sans-modèle à l'ordre 3, en utilisant le flotteur comme démonstrateur.

## Bibliographie

- [1] « Electro-Karst ». Consulté le: 8 juillet 2024. [En ligne]. Disponible sur: <https://anr.fr/Projet-ANR-22-CE33-0005>
- [2] « Argo ». Consulté le: 9 juillet 2024. [En ligne]. Disponible sur: <https://argo.ucsd.edu/>
- [3] Y. Bai, R. Hu, Y. Bi, C. Liu, Z. Zeng, et L. Lian, « Design and Depth Control of a Buoyancy-Driven Profiling Float », *Sensors*, vol. 22, n° 7, 2022, doi: 10.3390/s22072505.
- [4] S. L. Reste *et al.*, « “Deep-Arvor”: A New Profiling Float to Extend the Argo Observations Down to 4000-m Depth », *Journal of Atmospheric and Oceanic Technology*, vol. 33, n° 5, p. 1039-1055, 2016, doi: 10.1175/JTECH-D-15-0214.1.
- [5] B. McGilvray et C. Roman, « Control system performance and efficiency for a mid-depth Lagrangian profiling float », in *OCEANS'10 IEEE SYDNEY*, 2010, p. 1-10. doi: 10.1109/OCEANSSYD.2010.5603906.
- [6] T. Le Mézo, « Bracketing largest invariant sets of dynamical systems : an application to drifting underwater robots in ocean currents », 2019. [En ligne]. Disponible sur: <https://theses.hal.science/tel-03174429>
- [7] H. Zhou *et al.*, « Adaptive robust tracking control for underwater gliders with uncertainty and time-varying input delay », *Ocean Engineering*, vol. 240, p. 109945-109946, 2021, doi: <https://doi.org/10.1016/j.oceaneng.2021.109945>.
- [8] W. M. Bessa, E. Kreuzer, J. Lange, M.-A. Pick, et E. Solowjow, « Design and Adaptive Depth Control of a Micro Diving Agent », *IEEE Robotics and Automation Letters*, vol. 2, n° 4, p. 1871-1877, 2017, doi: 10.1109/LRA.2017.2714142.
- [9] Y. Bai, R. Hu, Y. Bi, C. Liu, Z. Zeng, et L. Lian, « Design and Depth Control of a Buoyancy-Driven Profiling Float », *Sensors*, vol. 22, n° 7, 2022, doi: 10.3390/s22072505.
- [10] N. A. A. Hussain, M. R. Arshad, et R. Mohd-Mokhtar, « Underwater glider modelling and analysis for net buoyancy, depth and pitch angle control », *Ocean Engineering*, vol. 38, n° 16, p. 1782-1791, 2011, doi: <https://doi.org/10.1016/j.oceaneng.2011.09.001>.
- [11] Z. Ismail et R. Varatharajoo, « A study of reaction wheel configurations for a 3-axis satellite attitude control », *Advances in Space Research*, vol. 45, n° 6, p. 750-759, 2010, doi: <https://doi.org/10.1016/j.asr.2009.11.004>.
- [12] F. L. Bars, « razor-9dof-ahrs ». [En ligne]. Disponible sur: <https://github.com/lebarsfa/razor-9dof-ahrs/tree/master>

- [13] L. Jaulin, *La robotique mobile: Cours et exercices*. in Collection Robotique. ISTE Editions Limited, 2015. [En ligne]. Disponible sur: <https://books.google.fr/books?id=4g5mDwAAQBAJ>
- [14] M. Fliess et C. Join, « Commande sans modèle et commande à modèle restreint », *e-STA Sciences et Technologies de l'Automatique*, vol. 5, n° 4, p. 1-23, 2008, [En ligne]. Disponible sur: <https://inria.hal.science/inria-00288107>
- [15] C. Gussen, P. Diniz, M. Campos, W. Martins, F. Costa, et J. Gois, « A Survey of Underwater Wireless Communication Technologies », *Journal of Communication and Information Systems*, vol. 31, p. 242-255, 2016, doi: 10.14209/jcis.2016.22.
- [16] J. Kim, R. Pearce, et N. Amato, « Robust geometric-based localization in indoor environments using sonar range sensors », in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, p. 421-426. doi: 10.1109/IRDS.2002.1041426.