

Projet Docking Guerlédan

Robin Vidal, Maxime Lefèvre, Clément Dunot, Thomas Bourgeois

ENSTA, Campus de Brest



4 décembre 2025

Table des matières

Introduction Générale	2
0.1 Contexte Stratégique et Nécessité du Docking	2
0.2 État de l'Art et Positionnement du Sujet	2
0.3 Annonce du Plan	3
1 Partie I : Étude Théorique et Algorithmique	4
1.1 Hypothèses de Travail et Domaine d'Emploi	4
1.2 Modélisation et Loi de Commande	5
1.2.1 Contrôle en profondeur	5
1.2.2 Modèle Dynamique du ROV	5
1.2.3 Synthèse de la Loi de Commande	5
1.3 Simulation et Validation Numérique	5
1.3.1 Environnement de Simulation	5
1.3.2 Scénarios de Test et Résultats	6
1.4 Algorithme de Traitement d'Image Sonar	7
1.4.1 Choix de la cage et Configuration des Réflecteurs	7
1.4.2 Chaîne de Traitement d'Image	7
1.4.3 Estimation de la Position Relative	7
1.5 Résultats Théoriques	7
2 Partie II : Matériel et Expérimentation	8
2.1 Plateforme Robotique : BlueROV Heavy	8
2.2 Capteur de Perception : Sonar Oculus M750d	8
2.3 Protocole Expérimental en Bassin	8
2.4 Campagne d'Essais : Lac de Guerlédan	9
2.5 Analyse des Résultats Expérimentaux	9
2.6 Matériel	9
2.7 Contrôle du Robot	10
2.8 A faire	11
Conclusion	12
2.9 Conclusion Générale	12
2.9.1 Synthèse des Réalisations	12
2.9.2 Limites et Perspectives	12

Introduction Générale

0.1 Contexte Stratégique et Nécessité du Docking

L'évolution de la robotique sous-marine marque une transition fondamentale, passant de missions expéditionnaires ponctuelles à des déploiements persistants de systèmes résidents. Dans ce nouveau paradigme, les Véhicules Sous-marins Autonomes (AUV) et les Véhicules Télécommandés (ROV) hybrides ne sont plus systématiquement récupérés en surface après chaque intervention. Ils doivent au contraire être capables de s'amarrer de manière autonome à des infrastructures immergées (stations d'accueil, garages, cages) pour recharger leurs batteries et transférer les données collectées.

Cette capacité d'amarrage autonome, ou « *docking* », constitue le verrou technologique majeur pour l'essor de l'économie bleue, qu'il s'agisse de la surveillance environnementale à long terme ou de l'inspection des infrastructures offshore. L'amarrage est une manœuvre délicate, comparable à un ravitaillement en vol ou à un appontage, mais effectuée dans un fluide visqueux et perturbé. La difficulté réside dans l'alignement précis du véhicule avec l'axe d'entrée de la station, nécessitant souvent des tolérances centimétriques tout en luttant contre les courants marins et les effets hydrodynamiques de proximité.

Cependant, l'environnement sous-marin impose des contraintes sévères qui rendent les méthodes de guidage traditionnelles inopérantes. La turbidité de l'eau, l'absence de signaux GPS et la variabilité des conditions d'éclairage limitent drastiquement l'efficacité des systèmes de vision optique au-delà de quelques mètres. C'est ici que la technologie sonar (*Sound Navigation and Ranging*) devient indispensable, offrant une capacité de perception robuste à travers les eaux troubles et sur de longues distances.

0.2 État de l'Art et Positionnement du Sujet

Dans la littérature scientifique actuelle, la séquence d'amarrage se décompose généralement en trois phases distinctes, chacune faisant appel à une modalité de capteur spécifique :

1. **Phase de ralliement (Longue portée)** : Guidage par systèmes acoustiques (LBL/USBL).
2. **Phase d'approche (Moyenne portée, 10-100m)** : Utilisation prédominante du sonar d'imagerie frontale (FLS) pour détecter la signature de la station.
3. **Phase terminale (Courte portée, < 2m)** : La majorité des travaux existants bascule sur un guidage visuel (caméras) ou inertiel (DVL) pour l'entrée finale dans la cage, en raison de la difficulté d'interprétation des images sonar à très courte distance (zones d'ombre, trajets multiples, saturation).

Notre approche se démarque de cet état de l'art par un choix technologique fort : le maintien du guidage sonar durant la phase terminale. L'hypothèse centrale de ce projet est que, dans des conditions de turbidité élevée (ports, estuaires, eaux profondes) où les caméras deviennent aveugles, le sonar doit rester le capteur primaire jusqu'à l'amarrage complet. Ce choix soulève des défis spécifiques ignorés par les approches hybrides sonar/vision :

il nécessite un traitement d'image acoustique avancé et une conception mécanique de la cible (la cage) adaptée pour maximiser sa signature acoustique à courte portée.

Notre projet met ainsi en œuvre un BlueROV Heavy (8 moteurs) équipé d'un sonar Oculus M750d et piloté via ROS2, avec pour objectif de réaliser un amarrage autonome dans une cage en aluminium instrumentée de réflecteurs passifs.

0.3 Annonce du Plan

Ce rapport détaille la méthodologie et les résultats de cette démarche à travers deux parties principales.

Partie I : Étude Théorique et Algorithmique Cette première partie posera les fondements du système développé :

- Nous commencerons par définir les **hypothèses de travail**, en précisant le domaine d'emploi de nos travaux (type de véhicule holonome, caractéristiques du sonar Oculus, environnement statique).
- Nous détaillerons ensuite la **loi de commande** élaborée pour assurer le suivi de trajectoire et la stabilisation face aux perturbations.
- Ces concepts seront validés par une étape de **simulation**, permettant de tester la logique de contrôle sans risque matériel.
- Un chapitre sera consacré au **traitement d'image sonar**, expliquant comment nous extrayons la position de la cage à partir des données acoustiques brutes.
- Cette partie se conclura par la présentation des **résultats théoriques** attendus avant le passage au réel.

Partie II : Matériel et Expérimentation Cette seconde partie se concentrera sur la mise en œuvre pratique :

- Nous présenterons l'architecture matérielle, incluant le **BlueROV** modifié et l'intégration du sonar **Oculus M750d**.
- Nous détaillerons les deux campagnes d'essais réalisées : d'abord en environnement contrôlé lors de l'**expérience en bassin**, puis en conditions réelles lors de l'**expérience au lac de Guerlédan**.
- Nous analyserons enfin les **résultats expérimentaux** obtenus, en confrontant les performances réelles aux prédictions théoriques.

Le rapport s'achèvera par une **Conclusion** qui synthétisera les objectifs atteints et identifiera les pistes d'amélioration pour les verrous restants afin de fiabiliser totalement l'amarrage autonome par sonar.

Chapitre 1

Partie I : Étude Théorique et Algorithmique

Cette partie a pour objectif d'établir le cadre formel du projet. Elle détaille les outils mathématiques utilisés pour modéliser le comportement du robot, les algorithmes de perception développés pour interpréter les données sonar, ainsi que les lois de commande garantissant l'amarrage.

1.1 Hypothèses de Travail et Domaine d'Emploi

Avant de définir les modèles, il est nécessaire de circonscrire le domaine de validité de nos travaux. Ces hypothèses simplificatrices permettent de rendre le problème traitable tout en restant représentatives des conditions réelles d'opération.

- **Le Véhicule** : Ce système de navigation est spécifiquement dimensionné pour des véhicules sous-marins de type ROV (ici un BlueROV Heavy) disposant d'une configuration de propulsion vectorielle, autorisant une manœuvrabilité holonome dans le plan horizontal. Nous postulons un découplage des dynamiques verticales, supposant une stabilisation de la profondeur et de l'assiette (tangage et roulis) assurée passivement par la conception hydrostatique ou par une boucle de régulation distincte. Le problème de contrôle est ainsi réduit à trois degrés de liberté (x, y, ψ). *[Instruction manquante : Confirmer ici la matrice d'inertie approximative si connue, ou préciser que le modèle de traînée est supposé quadratique.]*
- **Le Capteur Sonar** : Le système de perception est considéré comme la seule source de vérité pour la navigation relative (pas de positionnement absolu global). Le sonar est modélisé comme un capteur fournissant une projection 2D de l'environnement dans le plan local du robot, extrayant l'azimut et la distance des amers (bouées). *[Instruction manquante : Préciser ici les spécifications du sonar Oculus (ex : ouverture angulaire 130°, fréquence de rafraîchissement en Hz) et le modèle de bruit retenu pour la simulation (ex : bruit blanc gaussien sur la mesure de distance).]*
- **L'Environnement** : L'évolution du robot est restreinte à un plan horizontal à profondeur constante. La "cage" cible est supposée statique ou quasi-statique. Les perturbations environnementales, telles que les courants marins ou la tension du câble ombilical, sont considérées comme des forces constantes ou lentement variables que le contrôleur devra rejeter. *[Instruction manquante : Quantifier l'intensité maximale du courant simulé (ex : < 0.5 m/s) et préciser si des obstacles autres que les bouées sont pris en compte.]*

1.2 Modélisation et Loi de Commande

Pour asservir la position du robot par rapport à la cage, nous avons synthétisé un contrôleur adapté aux contraintes de la perception locale.

La stratégie de contrôle repose sur une architecture réactive à deux états qui privilégie exclusivement la perception locale du robot, sans recours au GPS. Initialement, si le champ de vision ne capture pas les deux bouées, le robot active un mode de recherche en pivotant sur lui-même jusqu'à l'acquisition des cibles. Une fois le contact visuel établi, l'algorithme reconstruit géométriquement la "porte" formée par les bouées directement dans le référentiel du robot (où ce dernier est toujours à l'origine [0, 0]) pour calculer un point de consigne virtuel situé un mètre devant cette porte.

Le pilotage est alors découplé en trois axes indépendants (avance, dérive latérale et cap) régulés par des boucles PID complètes. Dans cette configuration, l'action **Proportionnelle** transforme directement les coordonnées locales de la cible (x et y) en commandes de vitesse pour assurer une réactivité immédiate, tandis que le terme **Integral** s'avère indispensable pour compenser les perturbations constantes, telles que les courants marins ou la traînée du câble, qui créeraient sinon une erreur stationnaire. Enfin, la composante **Dérivée** amortit la réponse du système.

[Image of PID control loop block diagram] pour stabiliser le robot face à la forte inertie et aux effets hydrodynamiques. Cette méthode permet au robot de "glisser" vers sa destination en corrigeant les erreurs de position relative en temps réel, garantissant une approche fluide et précise même si le robot est initialement décentré.

1.2.1 Contrôle en profondeur

[Instruction : Expliquez brièvement comment la profondeur est maintenue constante (ex : PID sur la profondeur avec un capteur de pression).]

1.2.2 Modèle Dynamique du ROV

[Instruction : Présentez ici les équations du mouvement de Fossen sous forme matricielle $M\ddot{\nu} + C(\nu)\dot{\nu} + D(\nu)\nu + g(\eta) = \tau$. Détaillez brièvement ce que contient chaque matrice (Masse ajoutée, Coriolis, Amortissement quadratique) pour le BlueROV.]

1.2.3 Synthèse de la Loi de Commande

[Instruction : C'est le cœur de la commande. Présentez la structure de votre contrôleur (PID vectoriel, LQR, Commande Prédictive MPC, ou Mode Glissant). Écrivez l'équation de la commande $\tau = \dots$ et expliquez le rôle de chaque terme (terme proportionnel pour l'erreur, intégral pour le courant, etc.). Justifiez ce choix par rapport aux capacités de calcul du robot.]

1.3 Simulation et Validation Numérique

Avant le déploiement expérimental en bassin, la loi de commande a été validée dans un environnement virtuel. Cette étape est cruciale pour vérifier la convergence des algorithmes de guidage et la robustesse de la machine à états (basculement entre recherche et suivi) sans mettre en péril le matériel onéreux.

1.3.1 Environnement de Simulation

Pour le prototypage rapide de la loi de commande, nous avons développé un simulateur cinématique simple sous le langage Python. Contrairement aux simulateurs dynamiques lourds, cet outil permet de valider la logique décisionnelle en temps réel accéléré. Le modèle du robot

implémenté est un modèle holonome à 3 degrés de liberté (x, y, ψ), où les commandes de vitesse (u, v, r) sont intégrées directement pour obtenir la position, avec une saturation réaliste des actionneurs (fixée à 0.5 m/s et 0.5 rad/s dans nos tests).

La "cage" acoustique est modélisée par deux points singuliers fixes dans l'espace, représentant les échos des deux bouées verticales. Le capteur (sonar ou caméra) est simulé de manière géométrique avec une contrainte majeure : un champ de vision (FOV) limité à $\pm 45^\circ$. Si une bouée sort de ce cône ou dépasse la portée maximale, l'information de distance et de cap est perdue, forçant le contrôleur à réagir.

1.3.2 Scénarios de Test et Résultats

Pour éprouver la robustesse du contrôleur PID local, nous avons défini un scénario multi-robots impliquant quatre configurations initiales distinctes, visant à tester trois classes de manœuvres critiques :

- Cas Nominal (Approche Directe et Correction Latérale)** : Ce scénario valide l'approche initiale et la correction latérale simple en positionnant le véhicule face à la cible mais avec un décalage par rapport à l'axe central.
- Cas Angulaire (Alignement Couplé)** : Les véhicules sont placés loin sur les flancs ($\pm 2.5\text{m}$). Ce scénario exige un couplage fort *Yaw/Sway* pour que le contrôleur puisse aligner le cap tout en convergeant simultanément vers l'axe central de la porte.
- Cas "Aveugle" (Déclenchement de la Recherche)** : Le véhicule est orienté de telle sorte que les bouées sont initialement hors de son champ de vision. Ce cas teste le déclenchement immédiat de l'état "Search", la rotation autonome pour l'acquisition de la cible, puis le verrouillage (Tracking).

Critère de succès : La simulation est considérée comme validée lorsque le robot stabilise sa position à moins de 10 cm du point de consigne virtuel (situé 1 m devant le plan des bouées) et que les vitesses résiduelles sont nulles. Les résultats numériques montrent une convergence systématique pour les quatre cas de figure en moins de 120 secondes, confirmant que la stratégie basée sur le référentiel local est suffisante pour accomplir la mission sans positionnement absolu.

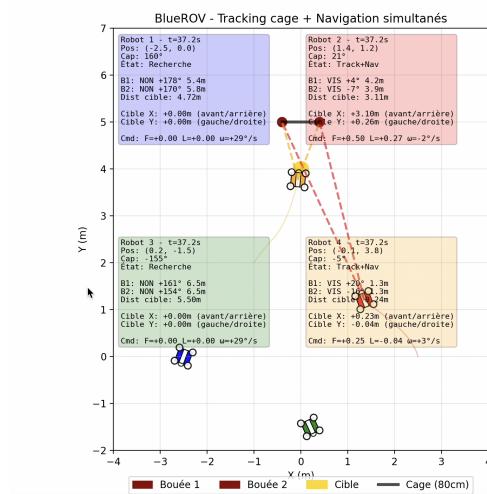


FIGURE 1.1 – Début de la simulation avec les 4 robots

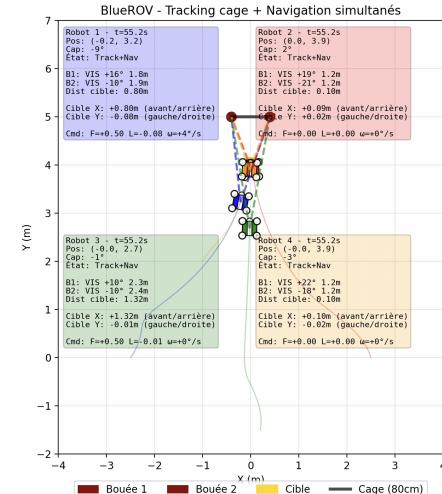


FIGURE 1.2 – Trajectoire des robots jusqu'à l'amarrage

1.4 Algorithme de Traitement d'Image Sonar

Le sonar ne fournit pas directement la position de la cage, mais une image d'intensité acoustique brute. Cette section détaille la chaîne de traitement permettant d'extraire les coordonnées (x, y, ψ) de la cible.

1.4.1 Choix de la cage et Configuration des Réflecteurs

1.4.2 Chaîne de Traitement d'Image

1. **Pré-traitement et Filtrage :** *[Instruction : Expliquez comment vous nettoyez l'image (filtre médian, seuillage, suppression du bruit de fond).]*
2. **Extraction de Caractéristiques (Feature Extraction) :** *[Instruction : Détaillez comment l'algorithme "trouve" la cage. Est-ce par détection de blobs (zones brillantes des réflecteurs), par détection de lignes (Transformée de Hough), ou par Machine Learning ?]*

1.4.3 Estimation de la Position Relative

[Instruction : Une fois les points d'intérêt trouvés dans l'image, expliquez la formule trigonométrique ou l'algorithme (ex : PnP) qui convertit les pixels (u, v) en mètres (x, y) dans le repère du robot.]

1.5 Résultats Théoriques

Cette section présente les performances attendues du système complet (Perception + Commande) en simulation.

[Instruction : Insérez ici une description des graphiques attendus. Par exemple : "La Figure X montre la décroissance de l'erreur de position en fonction du temps. On observe une convergence en moins de 30 secondes avec un dépassement nul." Discutez de la robustesse théorique : comment le système réagit-il si on ajoute du bruit sur les mesures sonar dans la simulation ?] // ...existing code...

Chapitre 2

Partie II : Matériel et Expérimentation

La mise en œuvre pratique de l'amarrage autonome nécessite une intégration matérielle rigoureuse et une validation progressive. Cette partie décrit la plateforme robotique, le capteur principal, ainsi que les deux phases expérimentales menées pour valider nos algorithmes.

2.1 Plateforme Robotique : BlueROV Heavy

Le vecteur utilisé pour nos travaux est une version modifiée du ROV commercial BlueROV2.

[Instruction : Décrivez la configuration "Heavy" à 8 propulseurs (4 verticaux, 4 horizontaux) et l'avantage de cette configuration vectorielle pour le contrôle à 6 degrés de liberté (6-DoF). Mentionnez l'électronique embarquée (Pixhawk pour le bas niveau, Raspberry Pi 4 ou NVIDIA Jetson pour le haut niveau/ROS2). Précisez l'autonomie de la batterie et les modifications mécaniques éventuelles (ajout de flottabilité, support sonar).] [Instruction : Expliquez comment ROS2 est installé sur le ROV, et comment la communication entre le Pixhawk (via MAVLink) et le PC embarqué est assurée. Mentionnez l'utilisation de mavros pour interfaçer ROS2 avec le contrôleur de vol. Expliquer les commandes de la manette.]

2.2 Capteur de Perception : Sonar Oculus M750d

L'élément central de la boucle de perception est le sonar multi-faisceaux (Multibeam Imaging Sonar).

[Instruction : Donnez les spécifications techniques clés du M750d : double fréquence (750kHz / 1.2MHz), ouverture angulaire (130° horizontal / 20° vertical), résolution en portée et en azimut. Expliquez pourquoi ce modèle est adapté au docking (haute résolution à courte portée). Décrivez son montage mécanique sur le ROV (inclinaison "tilt" vers le bas ou alignement horizontal) et l'interface logicielle utilisée pour récupérer les données (driver ROS, SDK Blueprint Subsea).]

2.3 Protocole Expérimental en Bassin

La première phase de validation a été réalisée en environnement contrôlé afin de caractériser les performances du système sans perturbations externes majeures.

[Instruction : Décrivez les dimensions du bassin (longueur, largeur, profondeur). Expliquez la mise en place de la cage (fixée au fond ou suspendue). Détaillez le scénario de test : position de départ du ROV, procédure de lancement de l'autonomie, critères de succès (entrée complète dans la cage). Mentionnez les outils de vérification utilisés si disponibles (ex : caméras externes pour filmer l'essai).]

2.4 Campagne d'Essais : Lac de Guerlédan

Afin de valider la robustesse du système en conditions réelles, une campagne a été menée au lac de Guerlédan.

[Instruction : Décrivez les conditions environnementales spécifiques : turbidité de l'eau (visibilité réduite), présence potentielle de courants, profondeur d'opération. Expliquez les défis logistiques (mise à l'eau depuis un ponton ou un bateau). Comparez la signature acoustique de la cage dans cet environnement "bruité" par rapport au bassin (réverbération du fond, objets parasites).]

2.5 Analyse des Résultats Expérimentaux

Cette section synthétise les données collectées lors des deux campagnes d'essais.

[Instruction : Comparez les trajectoires réelles du ROV avec celles de la simulation. Présentez des statistiques de réussite (ex : "Sur 20 tentatives, 15 ont abouti à un amarrage réussi"). Analysez les échecs : est-ce un problème de perte de la cible au sonar (perception) ou une incapacité des propulseurs à contrer une perturbation (commande) ? Incluez des images sonar réelles montrant la cage vue par le robot à différentes distances.]

2.6 Matériel

Dans cette section, nous présentons le matériel utilisé pour la réalisation de notre projet de docking via notre BlueROV.

Nous présenterons ses spécificités techniques dans les grandes lignes.

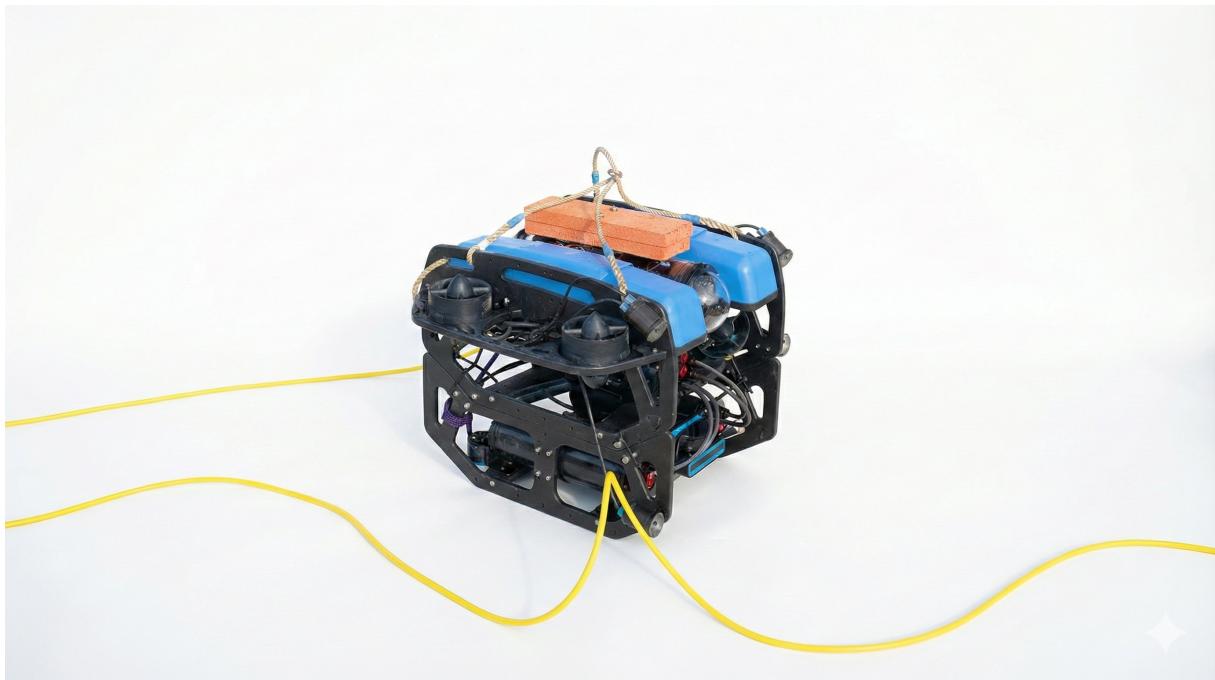


FIGURE 2.1 – Illustration de notre BlueROV

Sur la figure ci-dessus, on peut observer notre BlueROV modifié pour le projet d'amarrage autonome.

Pour la perception et le docking, le ROV embarque le sonar multifaisceaux Oculus M750d (Blueprint Subsea). Exploité en haute fréquence (1.2 MHz), il offre la résolution nécessaire aux manœuvres fines : 2.5 mm en portée et 0.6° en azimut. Son ouverture (130° horizontal / 20° vertical) impose un montage incliné vers le bas pour optimiser la visibilité de la cible. L'acquisition

des données acoustiques est assurée en temps réel via un driver ROS, permettant leur intégration directe dans la boucle de contrôle autonome.

Nous disposons de deux cordes permettant de remonter le ROV à la surface, ainsi que d'un long câble jaune assurant le contrôle à distance et l'alimentation électrique. Des mousses orange supplémentaires pour assurer la flottabilité.

L'architecture propulsive du BlueROV Heavy, articulée autour de huit propulseurs (quatre horizontaux et quatre verticaux), confère au véhicule un caractère holonome complet. En effet, cette configuration vectorielle permet de commander indépendamment et simultanément les six degrés de liberté : les trois translations et les trois rotations (lacet, roulis, tangage). L'avantage majeur de cette disposition réside dans sa capacité à offrir une autorité totale sur la stabilisation de l'assiette (roulis et tangage) tout en assurant le maintien en profondeur. Cette omnidirectionnalité garantit une manœuvrabilité fine, permettant au ROV d'effectuer des déplacements latéraux ou verticaux tout en conservant une orientation angulaire fixe — une caractéristique indispensable pour la précision des relevés et la stabilité des interventions en milieu perturbé.

2.7 Contrôle du Robot

Dans cette partie, nous expliquons la configuration et l'utilisation de la manette. Voici une illustration de celle-ci :



FIGURE 2.2 – Image de notre manette

Pour pouvoir commander le robot via la manette, il faut d'abord l'armer avec la touche « Start ».

Il s'agit d'une sécurité qui permet aussi de désarmer le robot avec la même touche. Si une mission se déroule mal, il est donc possible de désarmer le ROV : aucune commande ne lui sera alors envoyée.

Les lumières sont contrôlées par le bouton « logitech » au centre de la manette. Initialisées à une valeur PWM de 1000, les lumières sont éteintes. Un appui sur la touche augmente la puissance de 100 jusqu'à la valeur maximale de 2000 ; un nouvel appui la ramène à 1000, éteinte.

Pour pouvoir contrôler les lumières, il faut au moins avoir établi une connexion SSH entre l'ordinateur et le robot une fois, afin d'autoriser les échanges entre le Raspberry Pi (sur lequel sont connectées les lumières) et les codes.

Théoriquement, il devrait être possible de commander l'inclinaison de la caméra avec les flèches « haut / bas » à gauche de la manette, mais des problèmes électriques nous empêchent actuellement d'utiliser cette fonctionnalité.

Concernant le contrôle directionnel, le robot peut se déplacer en translation avec le joystick gauche (L3).

L'orientation (cap) est contrôlée par le joystick droit (R3) de gauche à droite.

L'altitude du robot est commandée par le joystick droit (R3) poussé vers l'avant/arrière : en le poussant vers l'avant, le ROV remonte ; en le poussant vers l'arrière, il descend.

Un régulateur PID a été implémenté pour maintenir une profondeur donnée. Ce mode peut être activé en appuyant sur « X » : lorsqu'on navigue et que le joystick R3 vient d'être relâché, la profondeur actuelle est mémorisée et le PID stabilise le ROV à cette profondeur. Il faut environ une dizaine de secondes pour obtenir une stabilisation quasi-parfaite, l'action intégrale du PID contribuant à éliminer l'erreur résiduelle.

Un second régulateur PID permet la stabilisation du cap. Il fonctionne de la même façon que l'asservissement de profondeur : lorsque le joystick responsable de l'orientation est relâché, le PID maintient le cap courant.

2.8 A faire

Mentionnez l'électronique embarquée (Pixhawk pour le bas niveau, Raspberry Pi 4 ou NVIDIA Jetson pour le haut niveau/ROS2) Précisez l'autonomie de la batterie Expliquez comment ROS2 est installé sur le ROV, et comment la communication entre le Pixhawk (via MAVLink) et le PC embarqué est assurée. Mentionnez l'utilisation de mavros pour interfacer ROS2 avec le contrôleur de vol.

Conclusion

2.9 Conclusion Générale

Ce rapport a permis d'explorer la faisabilité technique de l'amarrage autonome d'un ROV d'inspection en utilisant uniquement un sonar d'imagerie acoustique pour la phase terminale.

2.9.1 Synthèse des Réalisations

[Instruction : Résumez ici les jalons techniques validés ("ce qui est fait"). Mentionnez que la chaîne logicielle complète (ROS2 + Driver Oculus + Algorithme de détection) est opérationnelle. Soulignez la réussite de l'intégration mécanique et électronique sur le BlueROV Heavy. Rappelez les résultats clés de la simulation et des essais en bassin qui valident l'approche théorique. Si l'amarrage a réussi partiellement ou totalement à Guerlédan, mettez-le en avant comme une preuve de concept majeure en milieu réel turbide.]

2.9.2 Limites et Perspectives

Malgré les résultats encourageants, plusieurs verrous subsistent pour atteindre une autonomie industrielle fiable.

[Instruction : Discutez honnêtement des "choses non faites" ou des limites actuelles. Par exemple : la détection est-elle robuste si la cage est vue sous un angle critique ? La loi de commande actuelle permet-elle de lutter contre des courants marins violents ? Proposez des pistes pour la suite : fusionner le sonar avec un DVL pour stabiliser la vitesse, implémenter une commande plus avancée (type MPC), ou utiliser des techniques d'apprentissage profond (Deep Learning) pour améliorer la reconnaissance de la cage dans des environnements acoustiques complexes.]