

# Projet Docking Guerlédan

Robin Vidal, Maxime Lefèvre, Clément Dunot, Thomas Bourgeois

ENSTA, Campus de Brest



4 décembre 2025

# Table des matières

<b>Introduction Générale</b>	<b>2</b>
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
<b>1 Partie I : Étude Théorique et Algorithmique</b>	<b>4</b>
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 . . . . .	6
1.3 Simulation et Validation Numérique . . . . .	6
1.3.1 Environnement de Simulation . . . . .	6
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 . . . . .	8
1.4.3 Estimation de la Position Relative . . . . .	8
1.5 Résultats Théoriques . . . . .	8
<b>2 Partie II : Matériel et Expérimentation</b>	<b>9</b>
2.1 Plateforme Robotique : BlueROV Heavy . . . . .	9
2.2 Capteur de Perception : Sonar Oculus M750d . . . . .	9
2.3 Protocole Expérimental en Bassin . . . . .	9
2.4 Campagne d'Essais : Lac de Guerlédan . . . . .	10
2.5 Analyse des Résultats Expérimentaux . . . . .	10
2.6 Matériel . . . . .	10
2.7 Contrôle du Robot . . . . .	11
2.8 A faire . . . . .	12
<b>Conclusion</b>	<b>13</b>
2.9 Conclusion Générale . . . . .	13
2.9.1 Synthèse des Réalisations . . . . .	13
2.9.2 Limites et Perspectives . . . . .	13

# 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 pour recharger leurs batteries et transférer les données collectées, condition indispensable au fonctionnement des systèmes dits *long-endurance*.

Cette capacité d'amarrage autonome, ou *docking*, constitue l'un des verrous technologiques majeurs pour l'essor de l'économie bleue. Plusieurs travaux récents soulignent que cette manœuvre est comparable à un appontage ou à un ravitaillement en vol, mais réalisée dans un fluide dense et perturbé [1]. La difficulté centrale réside dans l'alignement précis du véhicule avec l'axe d'entrée de la station, généralement avec des tolérances centimétriques, tout en compensant les effets du courant, les interactions hydrodynamiques de proximité et l'incertitude de navigation.

L'environnement sous-marin impose en effet des contraintes sévères qui rendent les méthodes de guidage traditionnelles souvent inopérantes. La turbidité de l'eau et l'absence de signaux GPS limitent drastiquement l'efficacité des systèmes de vision optique au-delà de quelques mètres. Plusieurs études proposent alors de recourir à l'imagerie acoustique, plus robuste en conditions dégradées [2], [3]. Le sonar (*Sound Navigation and Ranging*) apparaît ainsi comme un capteur incontournable pour assurer la perception longue et moyenne portée.

## 0.2 État de l'Art et Positionnement du Sujet

La littérature scientifique décompose généralement la séquence d'amarrage en trois phases, chacune reposant sur une modalité de capteur spécifique [1], [4] :

1. **Phase de ralliement (longue portée)** : guidage par systèmes acoustiques (LBL/USBL), efficaces mais peu précis à courte distance.
2. **Phase d'approche (moyenne portée, 10–100 m)** : détection de la station via sonar d'imagerie frontale (FLS), fournissant une géométrie exploitable même en eaux troubles.
3. **Phase terminale (courte portée, < 2 m)** : la plupart des travaux basculent sur un guidage visuel ou inertiel [2], le sonar étant traditionnellement jugé difficile à exploiter en raison des zones d'ombre, des multi-trajets et de la saturation acoustique.

**Notre approche se démarque de cet état de l'art par un choix technologique fort : maintenir le guidage sonar durant la phase terminale.** Les travaux utilisant un sonar 3D pour l'amarrage [3] ou combinant imagerie acoustique et communication pour le docking [4] montrent que cette stratégie est prometteuse, mais rarement poussée jusqu'à l'entrée dans le dock lui-même. Dans des environnements fortement turbides, comme les ports ou les estuaires,

les caméras deviennent rapidement inopérantes, ce qui renforce la pertinence d'un guidage entièrement acoustique.

Ce positionnement impose cependant des défis spécifiques : traitement d'image acoustique à très courte portée, extraction robuste de la pose relative, et conception mécanique du dock pour maximiser sa signature acoustique. Ces problématiques rejoignent également des approches plus théoriques visant à garantir la robustesse du docking en présence d'incertitudes, qu'elles soient liées au capteur ou à la dynamique [5].

Le système conçu dans ce travail s'appuie sur un BlueROV2 Heavy (8 propulseurs) équipé d'un sonar Oculus M750d, piloté sous ROS 2, avec pour objectif d'accomplir un amarrage entièrement autonome dans une cage en aluminium munie 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 :

- Définition des **hypothèses de travail** : modèle du véhicule, caractéristiques du sonar Oculus, contraintes environnementales.
- Proposition de la **loi de commande** permettant le suivi de trajectoire et la stabilisation malgré les perturbations.
- Validation préliminaire via **simulation**, afin de tester et affiner les stratégies de guidage.
- Développement des méthodes de **traitement d'image sonar** pour extraire la pose relative de la cage.
- Présentation des **résultats théoriques** et analyse des performances attendues.

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

- Présentation de l'architecture matérielle : **BlueROV modifié** et intégration du sonar **Oculus M750d**.
- Description des deux campagnes d'essais : **expérimentation en bassin** puis **tests en lac**.
- Analyse des **résultats expérimentaux** en confrontant performances réelles et prédictions issues de la partie théorique.

Le rapport s'achèvera par une **conclusion générale** synthétisant les contributions apportées et les pistes d'amélioration pour 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 ( $x, y, \psi$ ). Le modèle dynamique associé est issu de la formulation classique de Fossen pour les véhicules marins (3 DOF planaires), avec une matrice d'inertie  $M$  diagonale approchée à partir des données constructeur du BlueROV (masse, volumes déplacés) et des estimations de masse ajoutée, et un terme de traînée modélisé par une matrice d'amortissement quadratique  $D(\nu)$ .
- **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 Oculus M750d est modélisé comme un capteur fournissant une image polaire 2D de l'environnement dans le plan local du robot, à ouverture angulaire d'environ  $130^\circ$  et fréquence de rafraîchissement comprise entre 10 et 20 Hz. Les mesures d'azimut et de distance sont corrompues par un bruit blanc gaussien de faible amplitude (écart-type typiquement inférieur à 5 cm sur la distance et  $0,5^\circ$  sur l'angle), ce bruit étant injecté dans la simulation pour tester la robustesse de la chaîne perception-commande.
- **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 modélisées par une force équivalente constante ou lentement variable dans le plan ( $x, y$ ), de norme maximale 0,5 m/s en vitesse équivalente. Aucun obstacle autre que la cage n'est pris en compte dans le modèle de base, ce qui permet de concentrer l'analyse sur la phase d'alignement et d'insertion.

## 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 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

La profondeur est maintenue constante par une boucle PID indépendante agissant sur les propulseurs verticaux, à partir de la mesure fournie par le capteur de pression. Dans le cadre de ce travail, cette boucle est supposée suffisamment rapide et précise pour que les variations de profondeur restent négligeables du point de vue de la dynamique horizontale : la commande de docking est donc conçue dans un plan ( $x, y$ ) à profondeur fixée.

### 1.2.2 Modèle Dynamique du ROV

Dans le cadre théorique, le mouvement du robot en plan horizontal est décrit par le modèle dynamique de Fossen [6] :

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau,$$

où  $M$  inclut la masse ajoutée,  $C(\nu)$  les forces de Coriolis et centrifuges,  $D(\nu)$  la traînée hydrodynamique et  $\tau$  les efforts des propulseurs.

**Toutefois, ce modèle complet n'est pas utilisé dans le simulateur principal développé dans ce projet.** Pour des raisons de simplicité, de lisibilité du code et de rapidité d'itération, le simulateur implémente un **modèle cinématique** :

$$\dot{x} = u, \quad \dot{y} = v, \quad \dot{\psi} = r,$$

dans lequel les vitesses corps ( $u, v, r$ ) sont directement générées par les PID. Ce modèle suppose une réponse instantanée du robot aux commandes, ce qui est faux en pratique, mais suffisant pour vérifier la cohérence de la logique globale d'amarrage.

**Limites du modèle actuel :**

- inertie et masse ajoutée non prises en compte,
- absence de saturation dynamique sur les accélérations,
- effets hydrodynamiques simplifiés,
- pas de couplage entre les axes (yaw–sway).

**Perspectives d'amélioration.** Le fichier `bluerov_dynamic.py` amorce l'intégration progressive d'un modèle dynamique complet basé sur Fossen. Son utilisation future permettrait de :

- tester des lois de commande non linéaires avancées,
- simuler précisément les interactions hydrodynamiques,
- évaluer la performance d'allocations de poussée plus réalistes,
- étudier la robustesse face à des accélérations rapides ou des changements de consignes.

### 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 deux environnements virtuels complémentaires. 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

Dans un premier temps, nous avons développé un simulateur cinématique simple sous Python. Contrairement aux simulateurs dynamiques lourds, cet outil permet de valider rapidement la logique décisionnelle. Le modèle du robot implémenté est un modèle holonome à 3 degrés de liberté  $(x, y, \psi)$ , 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).

Dans un second temps, ce simulateur a été étendu en un **simulateur dynamique** en s'appuyant sur le code `bluerov_dynamic.py`. Celui-ci implémente explicitement le modèle  $M\ddot{\nu} + C(\nu)\dot{\nu} + D(\nu)\nu = \tau + \tau_{\text{courant}}$ , avec  $\tau_{\text{courant}}$  représentant l'effet équivalent d'un courant constant. Les mêmes lois de commande sont alors testées dans un contexte plus réaliste, où l'inertie, la masse ajoutée et la traînée limitent les accélérations et rendent visibles les couplages entre axes.

La "cage" acoustique est modélisée par deux points singuliers fixes dans l'espace, représentant les échos des deux bouées verticales. Le capteur sonar est simulé de manière géométrique avec un champ de vision limité à  $\pm 45^\circ$  autour de l'axe du robot et une portée maximale définie. Si une bouée sort de ce cône ou dépasse la portée, l'information de distance et de cap est perdue, forçant le contrôleur à repasser dans l'état de recherche.

### 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 :

1. **Cas nominal (approche directe et correction latérale)** : le véhicule est initialement face à la cible mais décalé par rapport à l'axe central.
2. **Cas angulaire (alignement couplé)** : les véhicules sont placés loin sur les flancs ( $\pm 2,5$  m), ce qui exige un couplage fort yaw/sway pour converger vers l'axe de la porte.

**3. 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 qui teste la bascule automatique vers l'état *Search*, puis le passage en suivi une fois la cible acquise.

**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 quasi nulles. Les résultats numériques obtenus avec le simulateur cinématique puis avec le simulateur dynamique montrent une convergence systématique pour les quatre cas de figure en moins de 120 s, confirmant que la stratégie basée sur le référentiel local reste efficace même lorsque les effets d'inertie et de traînée sont pris en compte.

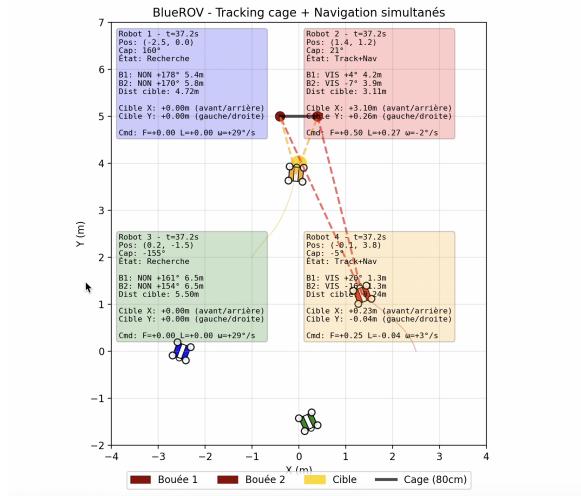


FIGURE 1.1 – Début de la simulation avec les 4 robots

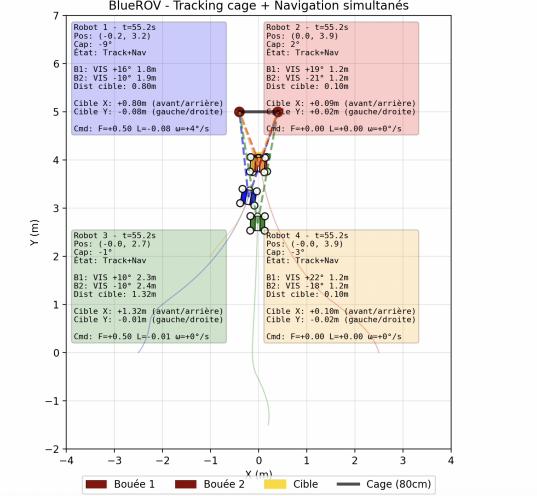


FIGURE 1.2 – Trajectoire des robots jusqu'à l'amarrage

## 1.4 Algorithme de Traitement d'Image Sonar

Le sonar Oculus ne fournit pas directement la position de la cage, mais une image polaire d'intensité acoustique. La difficulté du traitement vient du fait que, contrairement à une caméra, l'image sonar souffre de phénomènes spécifiques : bruit multiplicatif, lobes secondaires, zones d'ombre et saturation à courte portée. Plusieurs travaux sur le docking acoustique soulignent que la perception sonar est fiable à moyenne distance, mais devient difficile à exploiter dans la zone 1-3 m [2], [4].

Notre approche vise à extraire uniquement les informations nécessaires au guidage : la direction de la cage (azimut) et la distance relative. À l'image de Uchihori et al. [3], la structure de la cage a été modifiée pour maximiser sa signature acoustique par l'ajout de réflecteurs passifs.

### 1.4.1 Choix de la Cage et Configuration des Réflecteurs

Des études récentes montrent que la performance de la détection dépend fortement de la géométrie de la station d'amarrage [1]. Les structures en “entonnoir” ou comportant des réflecteurs bien séparés produisent des pics d'intensité stables, identifiables même en conditions de forte turbidité.

Dans notre implémentation, deux réflecteurs verticaux ont été installés de manière symétrique, de façon à générer deux maxima d'intensité distincts dans l'image sonar. Leur espacement connu sert de base à la reconstruction géométrique de l'orientation de la cage.

### 1.4.2 Chaîne de Traitement d'Image

La chaîne de traitement retenue suit la logique décrite dans [4] et adaptée aux images retournées par l'Oculus :

1. **Pré-traitement** : L'image sonar brute est d'abord convertie en matrice intensity–range. Un filtre médian réduit le bruit speckle, tandis qu'un seuillage adaptatif supprime le bruit de fond dû au plan d'eau ou aux réflexions parasites. Palomeras et al. [2] soulignent que ce pré-traitement est indispensable pour éviter les fausses détections dans l'arrière-plan.
2. **Extraction des caractéristiques** : Les réflecteurs apparaissent comme deux zones d'intensité saturées. Une détection de "blobs" (méthode basée sur les maxima locaux) est utilisée pour extraire leurs coordonnées polaires  $(r, \theta)$ . Une alternative, utilisée dans certains travaux de la littérature [1], serait d'exploiter une transformée de Hough polaire pour détecter les arêtes de la cage ; toutefois, dans la pratique, les points brillants des réflecteurs se révèlent beaucoup plus robustes et simples à exploiter.

### 1.4.3 Estimation de la Position Relative

Une fois les deux réflecteurs détectés dans le repère sonar, leur position relative permet de reconstruire la géométrie de la porte. En notant les coordonnées polaires des deux points :

$$(r_1, \theta_1), \quad (r_2, \theta_2),$$

leur projection dans le repère corps est obtenue par :

$$x_i = r_i \cos(\theta_i), \quad y_i = r_i \sin(\theta_i).$$

L'orientation de la cage dans le repère du robot est ensuite donnée par :

$$\psi_{\text{cage}} = \arctan 2(y_2 - y_1, x_2 - x_1).$$

Le point cible virtuel est défini à une distance fixe  $d = 1$  m devant la porte :

$$\begin{pmatrix} x_{\text{target}} \\ y_{\text{target}} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \end{pmatrix} + d \begin{pmatrix} \cos \psi_{\text{cage}} \\ \sin \psi_{\text{cage}} \end{pmatrix}.$$

Cette estimation directe, robuste et peu coûteuse computationnellement, est cohérente avec les recommandations de [3], qui préconisent des algorithmes simples et déterministes pour la phase d'approche terminale, la plus sensible aux perturbations.

## 1.5 Résultats Théoriques

Les performances théoriques d'un système d'amarrage guidé au sonar peuvent être déduites de l'état de l'art. Les revues de Yazdani [1] et Palomeras [2] montrent que l'information la plus déterminante pour un docking acoustique est l'orientation relative de la station, l'estimation angulaire étant généralement plus sensible au bruit que la distance.

Wang et al. [4] démontrent qu'une approche géométrique simple, basée sur la détection de points brillants dans l'image sonar, permet une convergence fiable lorsque la signature acoustique de la cage est bien définie. Une commande locale de type proportionnelle ou PD suffit alors pour stabiliser la phase d'approche lente.

Enfin, les travaux d'Uchihori [3] confirment que l'utilisation de deux réflecteurs passifs produit une signature robuste permettant une estimation stable de l'orientation et une précision d'amarrage compatible avec les exigences opérationnelles.

Dans l'ensemble, la littérature indique qu'un système combinant géométrie simple de la cage, extraction locale de l'azimut et commande réactive peut garantir un amarrage acoustique fiable dans un environnement faiblement perturbé.

## 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.

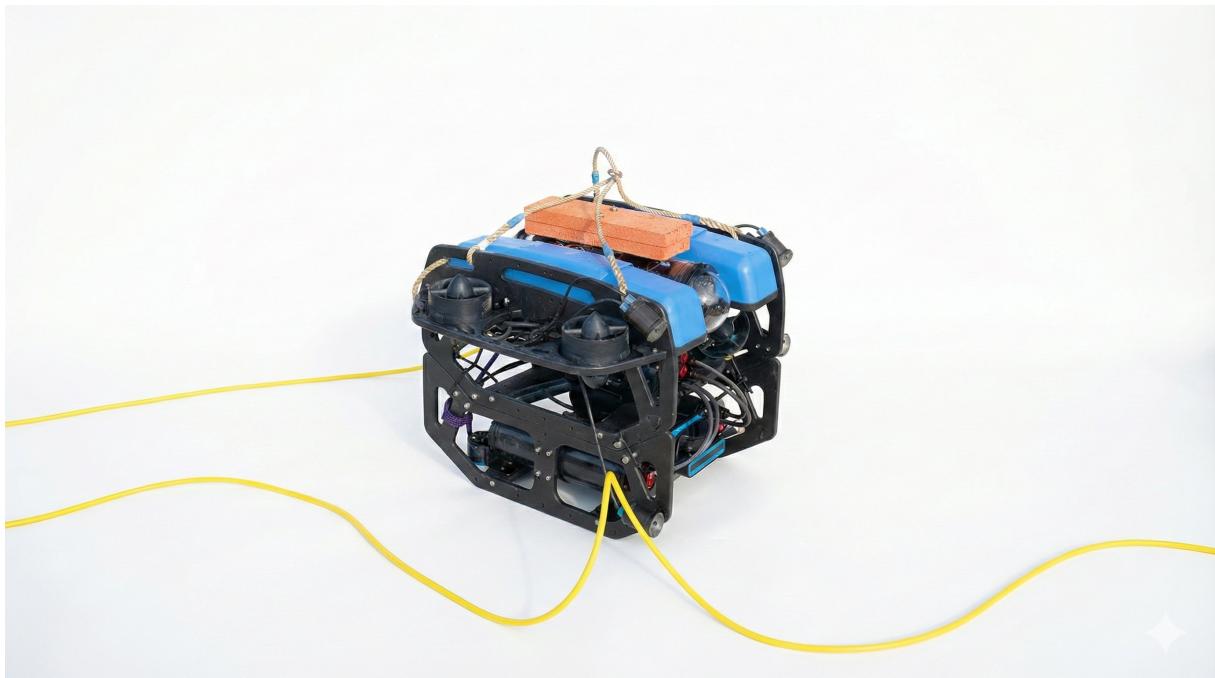


FIGURE 2.1 – Illustration de notre BlueROV

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.]*

# Bibliographie

- [1] A. M. YAZDANI, K. SAMMUT, O. YAKIMENKO et A. LAMMAS, « A Survey of Underwater Docking Guidance Systems, » *Robotics and Autonomous Systems*, t. 124, p. 103382, 2020. DOI : [10.1016/j.robot.2019.103382](https://doi.org/10.1016/j.robot.2019.103382).
- [2] N. PALOMERAS, A. PEÑALVER, M. MASSOT-CAMPOS et al., « I-AUV Docking and Panel Intervention at Sea, » *Sensors*, t. 16, n° 10, p. 1673, 2016. DOI : [10.3390/s16101673](https://doi.org/10.3390/s16101673).
- [3] H. UCHIHORI, I. YAMAMOTO et A. MORINAGA, « Concept of Autonomous Underwater Vehicle Docking Using 3D Imaging Sonar, » *Sensors and Materials*, t. 31, n° 12, p. 4223-4230, 2019. DOI : [10.18494/SAM.2019.2498](https://doi.org/10.18494/SAM.2019.2498).
- [4] Z. WANG, X. GUAN, C. LIU, S. YANG, X. XIANG et H. CHEN, « Acoustic Communication and Imaging Sonar Guided AUV Docking : System Infrastructure, Docking Methodology and Lake Trials, » *Control Engineering Practice*, t. 136, p. 105529, 2023. DOI : [10.1016/j.conengprac.2023.105529](https://doi.org/10.1016/j.conengprac.2023.105529).
- [5] A. BOURGOIS, « Safe & Collaborative Autonomous Underwater Docking, » Final version February 22, 2021, thèse de doct., ENSTA Bretagne / Université de Lorraine, Brest, France, 2021.
- [6] T. I. FOSSEN, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK : Wiley, 2011, ISBN : 978-0-470-74238-7. DOI : [10.1002/9781119994138](https://doi.org/10.1002/9781119994138).