

# Travaux Pratiques de modélisation et commande de robots sous-marins

Vincent Creuze<sup>1</sup>, Vincent Hugel<sup>2</sup>, Sabine Seillier<sup>2,3</sup>

[vincent.creuze@umontpellier.fr](mailto:vincent.creuze@umontpellier.fr) (Auteur correspondant), [vincent.hugel@univ-tln.fr](mailto:vincent.hugel@univ-tln.fr) et [sabine.seillier@univ-tln.fr](mailto:sabine.seillier@univ-tln.fr)

Adresses :

<sup>1</sup>LIRMM, Université de Montpellier, CNRS, Montpellier, France

<sup>1</sup>Polytech Montpellier, Université de Montpellier, France.

<sup>2</sup>COSMER, Université de Toulon, France

<sup>3</sup>SeaTech, Université de Toulon, France

**RESUME :** L'année universitaire 2021-2022 a vu démarrer à l'Université de Toulon la première promotion du Master Erasmus Mundus MIR (Master Sciences en robotique marine et maritime intelligente). Parmi les enseignements proposés en première année, un module a pour vocation la modélisation et la commande des robots sous-marins. Dans ce domaine, les phénomènes non modélisables sont nombreux et le recours à l'expérimentation est indispensable pour donner aux étudiants à la fois un certain nombre de savoir-faire expérimentaux très spécifiques (en particulier pour l'identification du modèle) et pour leur donner un aperçu de la complexité de la commande des systèmes fortement non linéaires que sont les véhicules sous-marins. Dans cet article, nous décrivons deux séances de travaux pratiques au cours desquelles les étudiants manipulent pour la première fois des robots sous-marins et effectuent l'identification de certains paramètres du modèle, mais aussi la commande selon deux degrés de liberté (profondeur et cap). L'article se conclut par l'évaluation de cet enseignement par les étudiants.

**Mots clés :** robotique sous-marine, automatique, modélisation, commande, travaux pratiques.

## 1 INTRODUCTION

Le master Sciences en robotique marine et maritime intelligente (MIR) [1] combine de façon innovante la robotique et l'intelligence artificielle dans le contexte du développement des sciences marines et maritimes et de leurs applications technologiques. Il vise à façonner des compétences permettant des avancées dans les domaines liés à la mer, en particulier dans la mise en œuvre de robots autonomes marins et sous-marins, dotés de capacités de téléopération, et dans l'utilisation de l'intelligence artificielle dans ces domaines.

Le consortium MIR est composé de quatre partenaires académiques principaux, l'Université norvégienne des sciences et technologies (NTNU), l'Université Jaume I (UJI, Espagne), l'Instituto Superior Técnico (IST-UL, Portugal) et l'Université de Toulon (UTLN, France) en tant que coordinateur. Le réseau inclut plus d'une cinquantaine de partenaires industriels et académiques associés, spécialisés en sciences marines/maritimes, en intelligence artificielle et en robotique.

La première année du master s'effectue en France à l'UTLN, afin d'acquérir des bases solides en sciences marines, robotique et intelligence artificielle, et de suivre des séminaires spécialisés dans l'industrie. En deuxième année, les étudiants choisissent une spécialisation en Robotique d'intervention sous-marine à l'UJI, en Opérations sous-marines autonomes au NTNU, ou en Robotique coopérative pour véhicules marins à l'IST.

Le processus de sélection des étudiants de M1 (Master 1<sup>ère</sup> année) a lieu chaque année au mois de janvier, pour former une promotion d'une vingtaine d'étudiants, issus de nombreux pays (Allemagne, Autriche, Bangladesh, Colombie, Egypte, Espagne, Etats-Unis,

France, Indonésie, Mexique, Nigéria, Pakistan, Thaïlande, Tunisie, Turquie...), dont la plupart perçoivent une bourse Erasmus Mundus. Les enseignements sont tous donnés en anglais.



*Figure 1. Le véhicule sous-marin BlueROV2 de la société Blue Robotics, utilisé pour les séances de travaux pratiques du module « Marine Mechatronics » du master MIR. Quatre propulseurs verticaux sont placés sur les côtés du robot. Le tube central contient la carte Raspberry Pi et le reste de l'électronique (Pixhawk, modem CPL, variateurs ESC...).*

*Crédit : BlueRobotics.*

Parmi les modules d'enseignement du master MIR, figure le module « Marine Mechatronics », dans lequel la modélisation et la commande des robots sous-marins sont abordées. Cette partie du module est détaillée dans cet article. Elle est essentiellement expérimentale, puisqu'elle est constituée d'un cours magistral d'une heure, suivi d'une séance de Travaux Dirigés de

deux heures, et deux séances de Travaux Pratiques (3h + 2h). La partie travaux pratiques de ce module est montée conjointement par l'Université de Toulon et l'Université de Montpellier, ayant toutes deux des laboratoires spécialisés en robotique sous-marine (respectivement le COSMER : CONception de Systèmes Mécaniques et Robotiques et le LIRMM : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier). C'est en particulier grâce au COSMER qu'il a été possible de mobiliser trois robots sous-marins de type BlueROV2 (Blue Robotics, USA) (Figure 1) et l'architecture logicielle associée. Ces robots sont utilisés par de nombreux laboratoires dans le monde et leur prise en main présente donc aussi un intérêt à long terme pour les étudiants.

Dans cet article, nous présentons tout d'abord la structure et le contenu du module « *Marine Mechatronics* », puis nous présentons le matériel utilisé et les différentes étapes des TP. Enfin, nous présentons les résultats d'un sondage réalisé en fin de module auprès de l'ensemble des étudiants (20) des deux premières promotions. Il existe peu d'articles antérieurs consacrés à des enseignements de robotique sous-marine, mais les lecteurs intéressés pourront se référer aux publications proposées par l'université de Nice dans les années 2000 [2][3].

## 2 DESCRIPTION DE L'ENSEIGNEMENT

### 2.1 Module *Marine Mechatronics*

Le module « *Marine Mechatronics* » est composé de trois séries de 8 heures d'enseignement chacune (1h CM, 2h TD et 5h TP), soit un total de 24 heures représentant une équivalence de 2,5 ECTS. La première série vise à mettre en œuvre les bases de la modélisation et de la commande des véhicules sous-marins pour les asservir en cap et en profondeur. C'est cette série que nous décrivons dans cet article. La deuxième série concerne l'asservissement visuel appliqué aux véhicules sous-marins et implémenté à l'aide de bibliothèques. La troisième série a pour objectif la programmation d'un comportement autonome permettant au robot sous-marin de réaliser une trajectoire avec évitement d'obstacles en bassin. Les enseignements sont prodigués dans l'ordre CM, TD, puis TP. Les TP sont précédés d'une période de préparation incluant l'installation des logiciels ad hoc sur l'ordinateur portable des étudiants, le test des programmes de base sur les robots sous-marins, et la réflexion sur les questions posées dans les sujets de travaux pratiques. Les programmes de base fournis aux étudiants permettent le pilotage à distance de l'engin au moyen d'une manette (type Logitech F310), l'envoi d'ordres aux propulseurs et la récupération des données des capteurs du robot (IMU et capteur de profondeur).

Le module « *Marine Mechatronics* » a lieu au deuxième semestre de la première année et a pour vocation de confronter les étudiants à des situations réelles où il s'agit de faire fonctionner un véhicule sous-marin à l'aide de programmes, en utilisant les connaissances

acquises lors des modules théoriques du premier semestre, consacrés aux systèmes mécatroniques et à l'automatique linéaire. La mise en œuvre de solutions sur engins réels est essentielle afin d'asseoir les compétences des étudiants dans les domaines de la robotique marine et sous-marine. La simple acquisition de connaissances théoriques ne suffit pas pour développer le sens pratique, savoir appliquer les lois de commande, et acquérir un savoir-faire opérationnel (équilibrage du robot, réglage des lois de commande...).

Les travaux pratiques ont lieu au sein du bassin à houle de l'École d'ingénieurs Seatech de l'Université de Toulon (Figures 2 et 3). Les dimensions du bassin sont de 3 mètres sur 10 mètres, avec une profondeur maximale de 1,5 mètre. Les parois latérales du bassin sont partiellement vitrées, ce qui permet aux étudiants d'observer directement le comportement des robots. Le bassin dispose d'une pente à l'une de ses extrémités et le générateur de houle peut produire des vagues de 10 cm crête à crête à différentes fréquences. Le nombre maximal d'étudiants par séance est de 14, sous l'encadrement de deux enseignants.

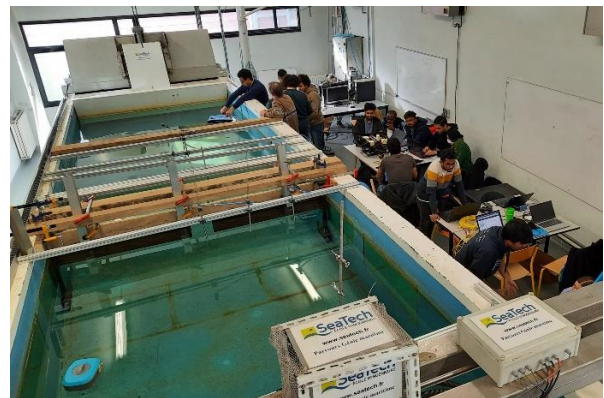


Figure 2. Le bassin à houle de l'école Seatech (Toulon) au cours d'une des séances de travaux pratiques du master MIR.

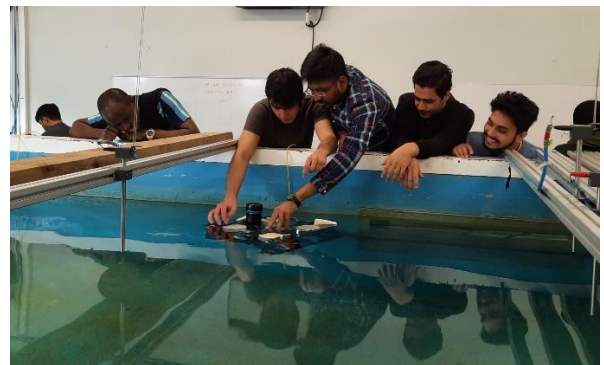


Figure 3. L'accès immédiat au bassin permet aux étudiants de manipuler le robot dans l'eau, par exemple pour l'équilibrer.

## 2.2 Objectifs pédagogiques des séances de TP

Au cours des séances de travaux pratiques, outre la prise en main des engins et des programmes de base, les étudiants doivent appréhender les spécificités du milieu aquatique et de son impact sur des corps rigides statiques ou en mouvement (flottabilité, hydrostatique, forces de traînée, etc.) et faire le lien avec les équations modélisant les phénomènes hydrostatiques et hydrodynamiques.

Les étudiants doivent également être capables de mettre en œuvre les engins du point de vue matériel et du point de vue logiciel. Ils doivent aussi analyser les résultats expérimentaux afin d'identifier les problèmes rencontrés et tenter d'y apporter des solutions. Enfin, ils sont amenés à rédiger un compte-rendu de leurs travaux au fur et à mesure de leur avancée. Ce compte-rendu est utilisé pour l'évaluation du module. L'ensemble du module se déroule en anglais.

## 3 EQUIPEMENT ET LOGICIEL

### 3.1 Robot sous-marin BlueROV2

Les étudiants sont répartis en trois groupes de quatre et chaque groupe dispose d'un robot sous-marin de type BlueROV2, commercialisé par la société BlueRobotics (Fig. 1) [4]. Ces robots comportent tous une caméra frontale, un capteur de pression pour la profondeur, un sonar monofaisceau sur la partie avant, et une centrale inertielle intégrée dans une carte à microcontrôleur de type *Pixhawk*<sup>TM</sup>, exploitée à l'aide de la suite logicielle libre *ArduSub* [5].

Les robots comportent huit propulseurs, quatre sont orientés verticalement, disposés aux quatre coins du robot, tandis que les quatre autres sont orientés horizontalement, en configuration X symétrique, appelée également configuration vectorielle. Les quatre moteurs verticaux permettent donc de contrôler les mouvements de translation verticale, mais aussi les mouvements de rotation en roulis (rotation autour de l'axe longitudinal du robot) et tangage (rotation autour de l'axe transversal du robot). Les moteurs horizontaux autorisent tous les mouvements de translation dans le plan horizontal du robot, ainsi que la rotation autour de l'axe vertical du robot (cap ou angle de lacet). Les six degrés de liberté du robot sont donc actionnés et contrôlables indépendamment. Cependant, durant les séances de travaux pratiques, les étudiants ne contrôleront que la profondeur d'immersion et le cap du robot (angle lacet). Le roulis et le tangage sont passivement stables (centre de gravité du robot plus bas que son centre de volume), tandis que les translations horizontales seront contrôlées manuellement par les étudiants, avec la manette Logitech, simplement pour éviter que le robot ne touche les parois du bassin, ce qui modifierait son comportement.

En termes d'alimentation, chaque robot est équipé d'une batterie connectée à un petit testeur sonore à LEDs permettant d'en visualiser le niveau de charge. La batterie et son testeur sont placés dans un caisson

cylindrique étanche et transparent, en acrylique, placé sous le caisson principal du ROV. Les batteries utilisées sont de type LiPo, de 11,1 V (i.e. 3S) ou de 14,8 V (i.e. 4S) et ont une capacité comprise entre 2000 et 2400 mAh selon le modèle. Chaque batterie permet une autonomie d'environ 2 heures pour une utilisation moyenne des propulseurs. Chaque groupe dispose d'un jeu de 2 batteries et le remplacement d'une batterie prend moins de 5 minutes.

Le robot est relié à l'ordinateur des étudiants grâce à un ombilical de flottabilité très légèrement positive, suffisamment fin et léger pour que les efforts qu'il crée sur le robot soient négligeables (Fig. 4). Les données sont véhiculées sur une paire torsadée de l'ombilical en utilisant la technologie des Courants Porteurs en Ligne (CPL). Un modem CPL dans le robot et un modem CPL en surface assurent la conversion en Ethernet.

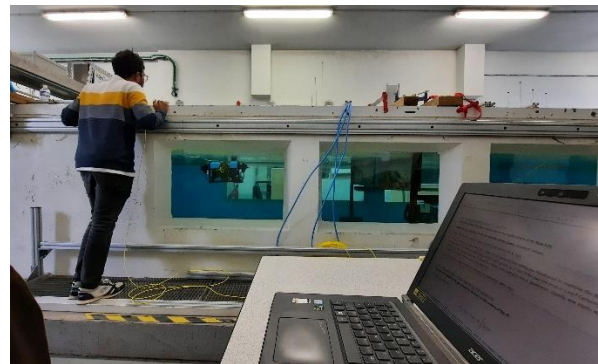


Figure 4. Le robot est relié directement à un ordinateur portable par l'intermédiaire d'un fin ombilical jaune que l'on aperçoit aux pieds de l'étudiant. Les vitres du bassin permettent d'observer le comportement du robot au cours des expérimentations.

### 3.2 Architecture matérielle et logicielle

Le cœur de l'architecture matérielle est une carte Raspberry Pi 3 connectée par liaison USB à une carte à microcontrôleur de type *Pixhawk*<sup>TM</sup>. La carte Raspberry Pi 3 est reliée en Ethernet au PC portable des étudiants via l'ombilical et deux modem CPL. La caméra frontale du robot est soit une caméra USB grand angle, soit une caméra de type Raspicam, directement connectée à la carte Raspberry. Le capteur de profondeur (par mesure de pression) est relié par liaison I<sup>2</sup>C à la carte *Pixhawk*<sup>TM</sup>. Le capteur sonar monofaisceau est connecté à la carte Raspberry par un adaptateur série-USB, mais n'est pas utilisé dans cette série de TP.

La carte Raspberry fonctionne sous linux avec une distribution ROS-compatible. L'intérêt de disposer de ROS sur la carte embarquée est de pouvoir installer des pilotes de périphériques supplémentaires (caméras, sonars, etc.). L'ordinateur à distance est lui aussi exploité sous ROS, ce qui permet de définir des programmes d'un côté, de l'autre, ou des deux. Les programmes pilotes des périphériques caméra et microcontrôleur sont lancés au démarrage de la carte Raspberry

sous la forme de threads ROS. La communication avec la carte à microcontrôleur *Pixhawk*<sup>TM</sup> se fait en utilisant le protocole *Mavlink*. Le code libre source du thread *mavros* en charge des échanges avec la carte à microcontrôleur a été légèrement modifié afin de prendre en compte la mesure brute du capteur de pression.

Sur leur ordinateur portable, les étudiants disposent de paquetages ROS avec du code source C++ et Python (pilotage à la manette et réception des données des capteurs). L'ordinateur est configuré avec une adresse IP fixe, ce qui permet la connexion au robot. Les premiers travaux à réaliser au début de la séance de TP consistent à modifier les scripts python existants afin d'envoyer directement des consignes aux propulseurs du robot.

## 4 DETAIL DES SEANCES DE TP

### 4.1 Modélisation des propulseurs

La première tâche effectuée par les étudiants est la caractérisation des propulseurs. Chaque propulseur est actionné par un moteur brushless, piloté par un variateur (ESC - Electronic Speed Controller) du même type que ceux utilisés pour les drones aériens. L'ESC est contrôlé avec un signal du même type que celui utilisé pour commander un servo-moteur. Il s'agit donc d'impulsions d'amplitude 5 V et de largeur variable, dont la durée de l'état haut vaut 1500  $\mu\text{s}$  lorsque le moteur est arrêté et est comprises entre 1100  $\mu\text{s}$  (vitesse de rotation maximale dans un sens) et 1900  $\mu\text{s}$  (vitesse de rotation maximale dans l'autre sens). Les étudiants disposent d'un relevé de valeurs de poussée du propulseur en fonction de la largeur d'impulsion de commande de l'ESC et doivent dessiner la caractéristique poussée/largeur d'impulsion (abusivement appelée PWM par les fabricants d'ESC) (Fig. 5, courbe bleue). Les étudiants doivent ensuite interpoler cette courbe, soit par deux fonctions polynomiales, soit par deux segments de droite qu'ils pourront ajuster en utilisant les valeurs extrêmes de la courbe réelle (Fig. 5, courbe rouge) ou en utilisant une méthode d'approximation de type moindres-carrés. Il n'y a pas de meilleure méthode et, pour chacune d'elle, l'enseignant discute avec l'étudiant des avantages et inconvénients dans le contexte particulier des propulseurs sous-marins.

Cette interpolation est calculée avant le TP par les étudiants et sera utilisée dans la loi de commande pour générer la largeur d'impulsion correspondant à la poussée désirée.

### 4.2 Identification de la flottabilité

La première manipulation du TP consiste à déterminer expérimentalement la flottabilité du robot, c'est-à-dire la différence entre son poids et la poussée d'Archimède qu'il subit [6]. Les robots sous-marins sont quasiment toujours flottants, pour revenir seuls en surface en cas de panne et pour ne pas remuer les sédiments du fond marin (les flux des propulseurs agissant principalement vers le haut pour compenser la flottabilité).

Pour mesurer la flottabilité, nous proposons aux étudiants une méthode bien adaptée aux expérimentations en bassin. Elle consiste à déposer sur le dessus du robot une bouteille en plastique remplie d'eau.

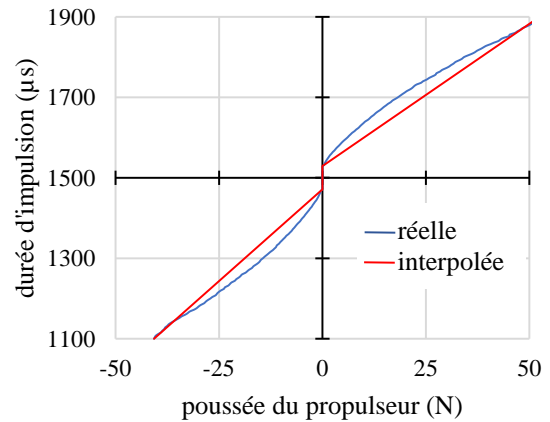


Figure 5. Caractéristique d'un propulseur Blue Robotics T200 contrôlé par un ESC. En bleu, la caractéristique réelle du propulseur et en rouge une interpolation linéaire possible.

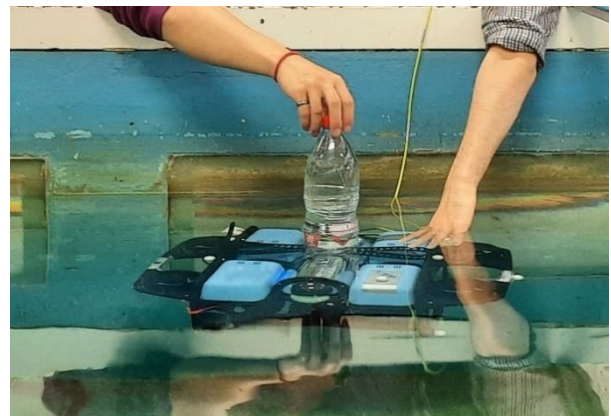


Figure 6. Pour mesurer la flottabilité du robot, les étudiants posent délicatement une bouteille pleine d'eau sur le dessus de l'engin et observent le niveau d'enfoncement de la bouteille lorsque le robot atteint un état d'équilibre. Ici les mains des étudiants servent juste à stabiliser horizontalement la bouteille et le robot mais n'agissent pas verticalement.

Les étudiants considèrent que le plastique de la bouteille a une flottabilité neutre et que son poids est négligeable par rapport au poids de l'eau contenue dans la bouteille. Sous le poids de la bouteille, le robot s'enfonce jusqu'à atteindre un état d'équilibre (Fig. 6). Dans cet état, la partie immergée de la bouteille n'agit pas puisque la poussée d'Archimède y est rigoureusement égale au poids de l'eau, tandis que le poids de la partie émergée de la bouteille compense exactement la flottabilité du robot. Il ne reste alors plus qu'à tracer un trait sur la bouteille au niveau de l'eau du bassin, puis à peser l'eau contenue au-dessus de ce trait. Cela se fait par exemple en versant l'eau dans un verre doseur.

La valeur de la flottabilité est notée et servira plus tard pour la commande du robot.

### 4.3 Vérification du modèle de la caractéristique des moteurs

La manipulation suivante consiste à vérifier que l'interpolation de la caractéristique des propulseurs et la mesure de la flottabilité sont correctes. Pour cela, les étudiants attachent au-dessus du robot une bouteille vide et bouchée (Fig. 7). Une fois immergée totalement, cette bouteille exercera une force verticale de 15 N vers le haut. A cette force s'ajoutera celle de la flottabilité du robot. Les étudiants sont invités à additionner ces deux forces et à utiliser la caractéristique interpolée des propulseurs pour générer une poussée équivalente vers le bas. Le robot est alors censé s'enfoncer dans l'eau jusqu'à atteindre un état de quasi-neutralité (il descend très doucement vers le fond ou reste en surface en laissant dépasser de l'eau une très petite partie de la bouteille). Cette expérimentation permet de valider les étapes précédentes, mais surtout de mettre en évidence les limites de l'interpolation linéaire. En effet, le robot n'a presque jamais le mouvement attendu et cela permet d'expliquer aux étudiants qu'il faudrait appliquer exactement la valeur de largeur d'impulsion lue sur la caractéristique réelle du propulseur pour obtenir la poussée désirée. L'interpolation n'est pas parfaite, mais l'écart entre la valeur obtenue par interpolation et la valeur réelle est suffisamment faible pour considérer qu'il sera absorbé par la commande comme une légère perturbation.

*Remarque : une bouteille exerçant une force de 15 N est utilisée car la flottabilité de certains robots peut être extrêmement faible et il est souhaitable de ne pas faire ce test aux extrémités de la caractéristique (sans la bouteille, on risquerait d'être trop proche de 0 N).*

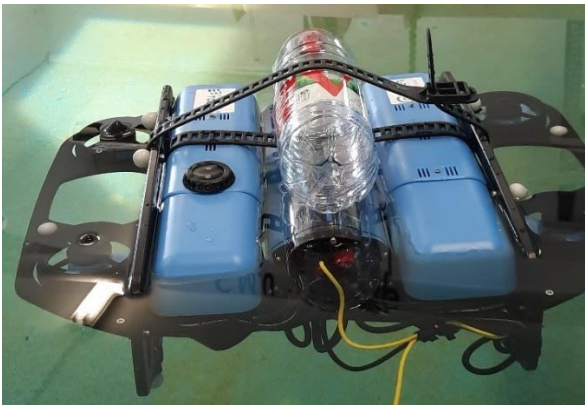


Figure 7. Bouteille vide et bouchée attachée au-dessus du robot et créant une force de 15 N une fois qu'elle est totalement immergée.

### 4.4 Commande du robot en profondeur

Dans cette partie du TP, les étudiants doivent implémenter une commande proportionnelle simple (1) pour asservir le robot à une profondeur désirée constante, notée  $z_{des}$ . La loi de commande est la suivante :

$$f_z = K_p \times \varepsilon = K_p \times (z_{des} - z) \quad (1)$$

avec  $f_z$  la force verticale à créer (consigne des propulseurs),  $K_p$  le gain de la commande proportionnelle et  $z$  la profondeur mesurée par le capteur de pression du robot. La consigne est un échelon de profondeur  $z_{des}$ . Les étudiants procèdent à quelques essais pour régler  $K_p$ , mais comprennent rapidement qu'ils ne pourront annuler l'erreur statique. L'enseignant leur explique alors que la flottabilité est l'origine principale de cette erreur car la commande ne dispose pas de terme intégral. Il les invite à ajouter à la commande un terme de compensation de flottabilité. La commande devient alors :

$$\tau_z = K_p(z_{des} - z) + flottabilité \quad (2)$$

Si la mesure de flottabilité a été bien effectuée au début du TP, les étudiants parviennent à régler beaucoup plus facilement le gain  $K_p$  et l'erreur statique se réduit considérablement. L'enseignant montre alors, expériences à l'appui, l'effet de la valeur de  $K_p$  sur la rapidité et la stabilité du système. Quel que soit le soin apporté au réglage de la commande, l'erreur statique ne s'annule jamais complètement. Cela est dû principalement à l'interpolation de la caractéristique des moteurs et à leur variabilité. Il est alors temps d'ajouter un terme intégral à la commande, pour obtenir une erreur statique nulle.

La commande proportionnelle-intégrale (PI) avec compensation de flottabilité a la forme suivante :

$$\tau_z = K_p \tilde{z} + K_i \int_0^t \tilde{z} dt + flottabilité \quad (3)$$

où  $\tilde{z} = z_{des} - z$  et  $K_i$  est le gain du terme intégral.

Les étudiants implémentent cette commande et se réjouissent alors d'obtenir une erreur statique nulle, mais ils ne parviennent pas à obtenir une réponse temporelle sans dépassement. Cela est dû à la trajectoire de consigne, qui est un échelon d'amplitude  $z_{des}$ . Ce type de trajectoire est incompatible avec la dynamique du véhicule en raison de la discontinuité de la dérivée première. Nous proposons alors de générer une « trajectoire » (au sens d'une évolution temporelle de la consigne de profondeur) plus douce, que le robot pourra suivre et qui permettra à la fois d'éviter le dépassement et de simplifier le réglage des gains du correcteur.

### 4.5 Trajectoire compatible avec la dynamique du véhicule

La valeur de la profondeur initiale de la trajectoire de consigne (i.e. à  $t = 0$  s.) est notée  $z_{init}$ . Dans un

souci de continuité, la valeur de  $z_{init}$  correspond à la valeur mesurée par le capteur de profondeur du robot lorsqu'il flotte au repos en surface, juste avant l'application de la trajectoire de consigne. La valeur de la profondeur à atteindre au terme de la trajectoire, c'est-à-dire à un instant noté  $t_{final}$ , est notée  $z_{final}$ .

La trajectoire de consigne  $z_{des}$  appliquée au robot est ensuite construite en utilisant un polynôme de degré 3, de la façon suivante :

- si  $t < t_{final}$ , alors  $z_{des} = z_{init} + a_2 t^2 + a_3 t^3$

$$\text{avec } a_2 = \frac{3(z_{final}-z_{init})}{t_{final}^2} \text{ et } a_3 = \frac{-2(z_{final}-z_{init})}{t_{final}^3}$$

- si  $t \geq t_{final}$ , alors  $z_{desired} = z_{final}$

De façon analogue, la dérivée  $\dot{z}_{des}$  de la trajectoire désirée, est calculée de la façon suivante :

- si  $t < t_{final}$ , alors  $\dot{z}_{des} = 2a_2 t + 3a_3 t^2$
- si  $t \geq t_{final}$ , alors  $\dot{z}_{des} = 0$

Les courbes ainsi générées sont représentées sur la figure 8.

Une fois la génération des trajectoires implémentée dans leur code et vérifiée graphiquement, les étudiants sont invités à les tester sur leur robot. Ils constatent les améliorations en termes de non-dépassement de la consigne et de simplification du réglage des gains.

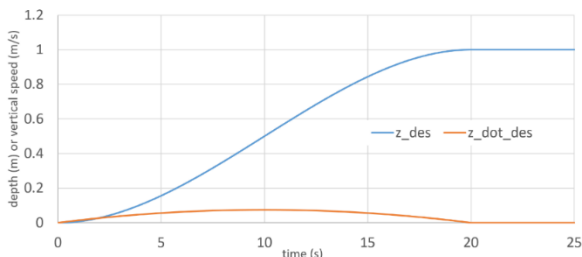


Figure 8. Trajectoires de consigne pour la profondeur  $z_{des}$  et sa dérivée  $\dot{z}_{des}$ , compatibles avec la dynamique du véhicule, améliorant les performances de suivi de trajectoire et de facilitant le réglage des gains.

#### 4.6 Pour aller plus loin...

Les étapes précédentes sont habituellement tout juste accomplies durant les 3+2 heures de TP. Pour les groupes les plus rapides, des approfondissements sont proposés. Le premier consiste à tester la robustesse de leur commande en profondeur en fixant une bouteille vide au robot pour en modifier la flottabilité. Il leur est ensuite demandé de coder un observateur de la vitesse verticale du robot, puis d'utiliser le résultat dans le terme Dérivé d'un correcteur PID avec compensation de flottabilité. Enfin, les étudiants peuvent implémenter un correcteur

PI pour asservir le cap du véhicule, en reproduisant la méthode utilisée pour asservir la profondeur.

## 5 BILAN DES SEANCES DE TP

### 5.1 Le point de vue des enseignants

Ces TP nécessitent un investissement très important de la part des enseignants et de la technicienne : préparation des machines, vérification de l'installation des codes de base sur les ordinateurs portables des étudiants en amont du TP, mobilisation du matériel sur le lieu des expérimentations, très nombreuses interventions auprès des étudiants durant les séances de TP. Cependant, ce travail est très largement récompensé par la satisfaction de voir les étudiants s'enthousiasmer par l'obtention de résultats sur un véritable robot sous-marin (Fig. 9).

C'est aussi l'occasion de leur transmettre un savoir-faire pratique difficilement trouvable dans les livres et d'essayer de leur faire ressentir les aspects physiques du système, pour mieux les assimiler. Le réglage des lois de commande d'un robot sous-marin est la tâche de base qui sera le préalable indispensable à toute autre application durant leur carrière de roboticiens sous-marins, qu'il s'agisse d'applications de perception, de localisation, de navigation, ou d'inspection...

### 5.2 Le point de vue des étudiants

Les enseignements du master MIR font l'objet d'évaluations systématiques de la part des étudiants sous Moodle. Ces séances de travaux pratiques font partie des enseignements qui ont reçu les meilleures évaluations. Cela n'est pas surprenant dans la mesure où ces travaux pratiques leur apportent la satisfaction réelle de voir leurs programmes fonctionner sur un robot qui accomplit les tâches prévues. Ils expliquent aussi que cela leur permet de faire le lien avec la théorie, parfois abstraite. Les étudiants sont demandeurs d'un nombre plus important d'heures de travaux pratiques de ce type. Quelques séances ont été ajoutées au cas par cas, mais sous réserve de la disponibilité des enseignants et du bassin. Le succès de ces séances a entraîné des demandes de la part d'autres formations. Des mutualisations sont envisagées afin d'y répondre. Cela nécessite également l'acquisition de matériels supplémentaires.

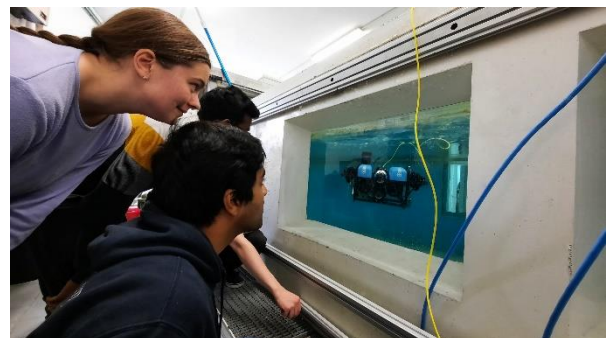


Figure 9. Les étudiants apprécient beaucoup l'implémentation sur des systèmes réels, qui plus est dans le cœur de métier de leur master.

## 6 CONCLUSION

Cet article présente deux séances de travaux pratiques de robotique sous-marine démarrées en 2021-2022 dans le module Marine Mechatronics, en première année du master Sciences en robotique marine et maritime intelligente (MIR). Durant ces séances, les étudiants découvrent concrètement les principaux éléments du modèle hydrostatique et du modèle d'actionnement d'un véhicule sous-marin de type ROV. Ils implémentent ensuite plusieurs lois de commande, du simple proportionnel (P) au Proportionnel Intégral avec compensation de flottabilité (PI+g), afin d'asservir la profondeur d'immersion du robot. Les groupes les plus rapides étudient également le PID+g. Durant ces TP, les étudiants sont sensibilisés à l'intérêt d'utiliser des trajectoires de consigne compatibles avec la dynamique du véhicule à commander. Ces TP nécessitent une logistique assez complexe et un investissement important de l'équipe pédagogique et technique, largement récompensés par la façon dont cet enseignement est apprécié par les étudiants.

### Bibliographie

- [1] Site web du master MIR : <https://www.master-mir.eu/>
- [2] Jean-Pierre Folcher et Maria-João Rendas, « Une plateforme expérimentale pour l'automatique : le robot sous-marin Phantom 500 », CETSIS 2001, Clermont-Ferrand, pp.149-152, novembre 2001.
- [3] Jean-Pierre Folcher, « Commande fréquentielle en cap d'un robot sous-marin », J3eA, Vol. 4 - Hors-Série n°2, 2005.
- [4] Site web du fabricant de robots sous-marins BlueRobotics : <https://bluerobotics.com/>
- [5] Site web du logiciel libre ArduSub : <https://www.ardusub.com/>
- [6] Vincent Creuze, « Robots marins et sous-marins. Perception, modélisation, commande », deuxième édition mise à jour, *Techniques de l'Ingénieur, Collection Robotique, Editions T.I., fév. 2024.*

### Auteurs

**Vincent Creuze** est professeur au département Electronique et Informatique Industrielle (EII) de Polytech Montpellier (Université de Montpellier) et mène ses recherches en robotique sous-marine au Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM).

**Vincent Hugel** est professeur à l'Université de Toulon et dirige le laboratoire Cosmer (Conception de systèmes mécaniques et robotiques), où il mène des recherches sur les robots autonomes et les véhicules sous-marins.

**Sabine Seillier** occupe un poste d'Assistant Ingénieur (ASI) à l'Université de Toulon et au laboratoire Cosmer (Conception de systèmes mécaniques et robotiques). Elle est spécialisée dans l'électronique et la fabrication additive. Elle assure notamment la maintenance et l'évolution de la flotte de robots sous-marins du Cosmer.