



# Réalisation d'un AUV : Autonomous Underwater Vehicle

Rapport de stage de spécialité

À l'attention de  
Nicolas MALANDAIN  
Benoît CLÉMENT  
Michel BENOIT  
Luc JAULIN

Étudiant  
Simon ROHOU

Été 2013



# Sommaire

Remerciements .....	4
Introduction.....	5
<b>I Contexte</b>	<b>6</b>
<b>I. La robotique sous-marine</b> .....	<b>7</b>
1. Le concept du drone sous-marin .....	7
2. Mon intérêt pour cette discipline .....	8
<b>II. Le stage de spécialité : un tremplin vers ma carrière</b> .....	<b>9</b>
1. L'ENSTA Bretagne et son pôle STIC .....	9
2. Sujet du stage .....	10
3. Planning - déroulement des activités .....	10
<b>II Préparation du robot</b>	<b>11</b>
<b>III. Mise en place du hardware et des interfaces logicielles</b> .....	<b>12</b>
1. La base du CISCREA .....	12
2. La propulsion vectorielle : intérêts et implémentation .....	13
3. Régulation en cap et en profondeur .....	14
4. Contenu de la boîte en aluminium .....	15
5. Un sonar pour cartographier .....	17
6. Un échosondeur pour donner une échelle .....	18
7. Un modem pour communiquer avec un autre AUV .....	18
8. Deux caméras pour percevoir l'environnement .....	19
<b>IV. L'architecture logicielle</b> .....	<b>21</b>
1. L'architecture MOOS : un middleware du MIT .....	21
2. La communication par Modbus .....	24
<b>III Le concours SAUC'E</b>	<b>25</b>
<b>V. Règles et organisation</b> .....	<b>26</b>
1. Le Student AUV Challenge Europe .....	26
2. Les règles du concours .....	27
<b>VI. Les épreuves</b> .....	<b>28</b>
1. Les qualifications : passage à travers une gate .....	28
2. Suivi de mur et détection d'anomalie .....	29
3. Inspection et suivi d'une structure immergée .....	32
4. À la recherche d'une boîte noire .....	33
5. Cartographie et localisation .....	33
6. Conclusion sur les résultats obtenus .....	34
<b>IV Mission autonome</b>	<b>35</b>
<b>VII. Faire le tour d'une île de manière autonome</b> .....	<b>36</b>
1. L'île des Morts, dans la rade de Brest .....	36
2. L'algorithme de suivi de waypoints GPS .....	37
3. L'expérience du 24 juillet 2013 .....	39
<b>Conclusion</b> .....	<b>41</b>
<b>Références</b> .....	<b>42</b>
<b>Annexes</b> .....	<b>44</b>

# Remerciements

Ce stage de spécialité a été une magnifique expérience et je ne saurais rédiger ce rapport sans remercier à nouveau les personnes qui ont contribué à faire de ces 9 semaines un période très enrichissante.

Ainsi, je tiens particulièrement à remercier :

- **Benoit Clément** : maître de stage et enseignant-chercheur, pour m’avoir accepté en stage à l’ENSTA et pour m’avoir donné de nombreux détails sur la faisabilité d’une thèse ; de quoi semer un doute positif dans mon esprit... ;
- **Vincent Drevelle** : postdoctorant dans l’équipe OSM<sup>1</sup> de l’ENSTA, avec qui j’ai passé de nombreuses journées à travailler sur le robot *Vici* ;
- **Mohamed Saad Ibn Seddik** : doctorant en robotique sous-marine, pour m’avoir présenté avec enthousiasme le middleware MOOS et pour sa grande motivation pendant les épreuves du concours SAUC-E ;
- **Fabrice Le Bars** : enseignant-chercheur dans l’équipe OSM, pour son expérience sur les robots et son investissement dans l’inscription des participants au concours ;
- **Luc Jaulin** : enseignant-chercheur dans l’équipe OSM, pour m’avoir ouvert une porte sur le monde de la recherche et sans qui je ne me serais sûrement pas autant posé de questions sur la possibilité de faire une thèse dans ce domaine ;
- **Benoit Zerr** : directeur du pôle OSM et enseignant-chercheur, pour la véritable passion scientifique qu’il sait communiquer ;
- **Irvin Probst** : enseignant dans l’équipe OSM, pour ses coups de main et ses explications pendant la préparation des robots ;
- **Jesus Arturo Monroy Anieva** et **Eduardo Campos Mercado** : deux doctorants mexicains ayant traversé l’Atlantique pour participer au concours avec l’ENSTA. Je ne pensais pas autant m’améliorer en espagnol pendant ce stage. Je les remercie aussi pour la bonne ambiance qu’ils ont su communiquer dans l’équipe ;
- **Yvon Gallou** : enseignant à l’ENSTA, pour son aide précieuse lors de la préparation matérielle des robots ;
- **Aymeric Bethencourt** : doctorant en robotique sous-marine, pour sa participation en Italie en tant que *team-leader* ;
- les secrétaires **Michèle Hofmann** et **Annick Billon-Coat**, qui m’ont facilité toute la partie administrative lors de mon arrivée dans les locaux.

Enfin, je remercie grandement **Michel Benoit**, ancien chercheur du LIRMM<sup>2</sup>, pour m’avoir fait découvrir le monde de la recherche en robotique, bien avant ce stage. Voilà bien un domaine que je ne suis pas prêt de quitter.

---

1. Ocean Sensing and Mapping

2. Laboratoire d’Informatique, de Robotique et de Microélectronique de Montpellier

# Introduction

Le stage de spécialité de fin de quatrième année est une période qui s'inscrit parfaitement dans mon cursus INSA : c'est pour moi l'occasion de mettre en pratique les connaissances acquises pendant mes cours dans le département Architecture des Systèmes d'Information de l'INSA de Rouen, tout en approfondissant ma technique et mon savoir faire dans un domaine de mon choix. Ce stage est d'autant plus important qu'il doit me permettre de décider du domaine dans lequel je réaliserai mon stage ingénieur, une étape primordiale entre ma formation et mon entrée dans la vie active.

Mes projets personnels m'ont naturellement poussé à regarder du côté du monde de la robotique, un secteur au cœur de l'innovation et où se croisent de nombreuses technologies. La robotique est un domaine qui me passionne, et particulièrement lorsqu'elle devient sous-marine. Je ne pouvais donc pas espérer mieux pour cet été qu'un stage à l'ENSTA de Bretagne, dans un laboratoire disposant de robots sous-marins. Je n'imaginais pas non plus avoir la chance de pouvoir participer à un concours européen de drones sous-marins comme celui organisé en Italie dans les locaux d'un centre de recherche de l'OTAN.

Ce rapport évoque les travaux réalisés pendant les préparatifs de ce concours ainsi que les épreuves auxquelles j'ai pu participer en Italie. Enfin, le concours ne clôturant pas cette période de stage, d'autres expérimentations ont été faites avec les robots dans la rade de Brest. L'une d'entre elles fait l'objet de la quatrième partie de ce rapport.

# Première partie

## Contexte

## I | La robotique sous-marine

### 1. Le concept du drone sous-marin

Un AUV (Autonomous Underwater Vehicle) est un robot sous-marin autonome dont la vocation est la réalisation de missions en mer en autonomie complète. Une fois la mission lancée, le sous-marin décide de lui-même des mouvements qu'il doit faire et des données qu'il doit récupérer.

Les ROV (Remotely Operated Vehicle), bien plus répandus que les AUV, sont des engins télé-guidés qui ne peuvent pas répondre aux mêmes problématiques que leurs équivalents autonomes. Les missions s'effectuant sur de longues distances ou à de grandes profondeurs sont difficilement réalisables avec des ROV pour des raisons matérielles liées au câble de commande (longueur, encombrement, débit). De même, le remplacement de ce câble par une communication sans fil n'est pas envisageable en raison du fort taux d'absorption de l'eau. C'est avec ce constat que les AUV entrent en jeu<sup>3</sup>, avec les nouvelles problématiques qui leurs sont liées.

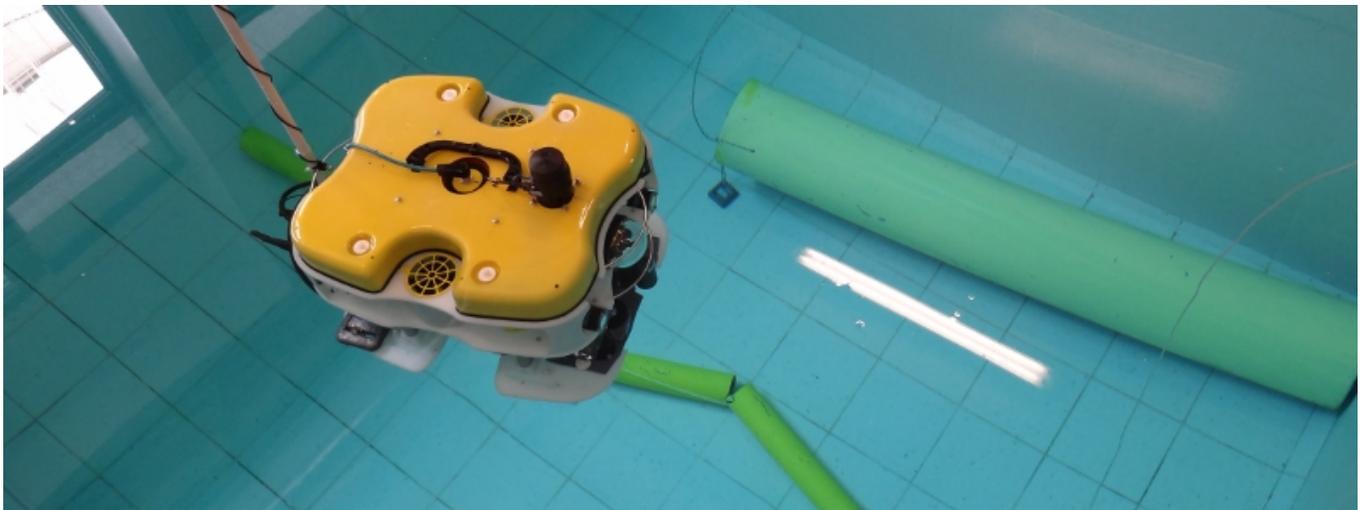


FIGURE 1 – L'AUV *Vici* sur lequel j'ai travaillé pendant ce stage

Cette nouvelle discipline est complexe et technique : comme les ROV, les AUV évoluent dans un milieu hostile (humidité, fortes pressions<sup>4</sup>, eau salée, isolement, obscurité, inaccessibilité pour l'homme, etc.). Un AUV doit donc se doter d'une bonne perception visuelle, physique et sonore de l'environnement ainsi que d'une connaissance précise de son état (vitesse, position, mais aussi taux de charge des batteries, température interne, détection de fuites...).

---

3. en 1987, on imagine le concept d'AUV pour la première fois dans le journal Ocean Industry

4. dans l'eau, la pression augmente d'environ 1 bar tous les 10 mètres

Les AUV entrent en jeu dans plusieurs domaines [1]. S'ils sont principalement tournés vers le civil, les applications militaires ne sauraient tarder. Ce sont les collaborations de robots autonomes qui intéressent de plus en plus les armées. En attendant, les AUV sont utilisés pour le désamorçage de mines sous-marines et en tant que navires éclaireurs pour la navigation de bâtiments importants. Les ingénieurs cherchent ici à produire en série des robots de très faible coût car il n'y a pour le moment pas d'autre solution que de perdre l'AUV en désamorçant la mine. Ces robots autonomes sont aussi utilisés pour de l'inspection et de la réparation de structures immergées comme des pipelines ou des barrages fluviaux. Enfin, ils sont aussi utilisés pour de la recherche et de l'entretien d'épaves (exploration, désensablage) ainsi que pour une étude de la faune, de la flore et de la géologie des fonds marins (recherche de pétrole, écoutes sismiques).

## 2. Mon intérêt pour cette discipline

Le virus de la robotique sous-marine m'a pris lorsque j'étais au collège. Voilà maintenant huit ans que je travaille sur la construction d'un sous-marin radio-commandé (type ROV). Travaillant en autodidacte, j'apprends beaucoup de cette construction où plusieurs domaines interviennent : mécanique, électronique, informatique, pneumatique, propulsion. La robotique est un véritable carrefour des technologies où il est difficile de se lasser puisqu'il est facile de varier les plaisirs.

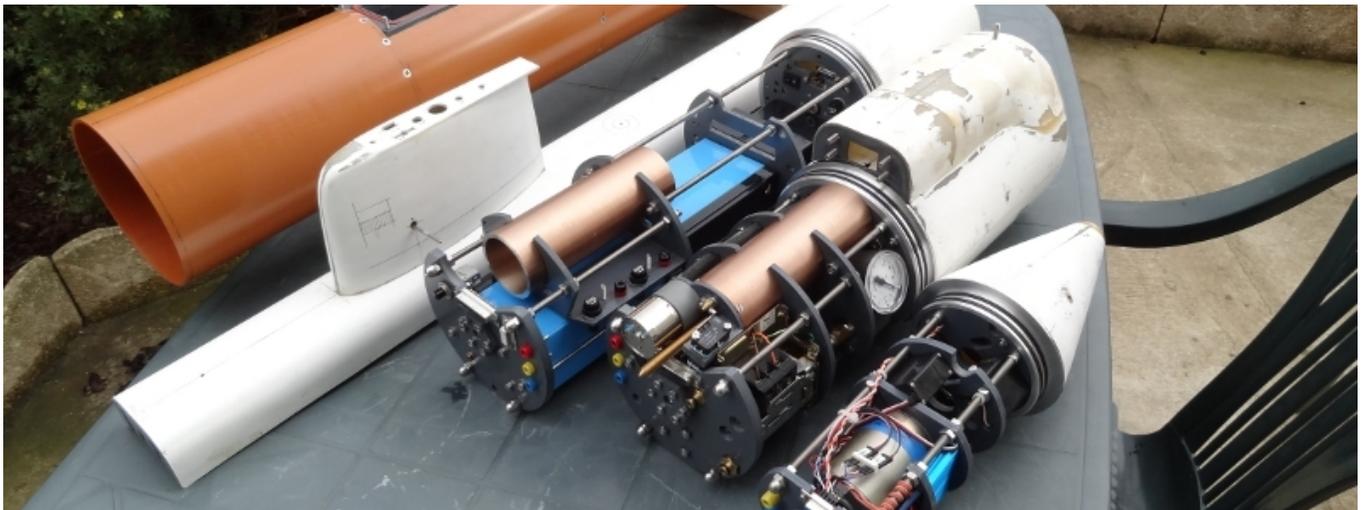


FIGURE 2 – Le *Collins*, une réalisation personnelle de type ROV

Aujourd'hui ce projet est sur le point d'aboutir (plus d'informations sur [www.simon-rohou.fr](http://www.simon-rohou.fr)) ; malgré les innombrables heures passées sur cette réalisation et les nombreux échecs accumulés, je n'imaginai pas en apprendre autant sur le plan technique, scientifique et humain. Cette passion m'a donc naturellement poussé à rechercher un stage dans ce milieu. J'ai d'abord contacté le LIRMM ; si un stage dans ce laboratoire n'était pas possible cette année, on m'a en revanche conseillé de contacter l'ENSTA de Bretagne, une école qui pourrait me proposer un sujet de stage particulièrement intéressant.

## II | Le stage de spécialité : un tremplin vers ma carrière

Ce stage s'inscrit parfaitement dans la transition entre mes études et mon entrée dans le monde professionnel. Il me permet de confirmer mon intérêt pour cette discipline et de garnir mon carnet d'adresses dans ce milieu. Ce fût donc une excellente occasion de rencontrer les chercheurs de l'ENSTA, dans laquelle j'ai réalisé ce stage de spécialité.

### 1. L'ENSTA Bretagne et son pôle STIC

#### Une école tournée vers la mer.

L'École Nationale Supérieure de Techniques Avancées de Bretagne [2] est une école d'ingénieurs civils et militaires située à Brest (29). L'école forme ses étudiants pendant trois ans dans les domaines de l'électronique, de l'informatique, de la mécanique ou de la pyrotechnique. Elle accueille des promotions de 180 étudiants dont 85% de civils.

*L'esprit « Grand Large » !*

De par sa localisation et son histoire, l'école propose des formations liées aux domaines maritimes comme la robotique sous-marine, l'hydrographie, l'océanographie, l'architecture navale et l'ingénierie offshore.



FIGURE 3 – Les locaux de l'ENSTA

#### Une des références françaises en robotique sous-marine.

C'est de cette manière que des amis du LIRMM m'ont décrit l'ENSTA pendant que je recherchais un stage. Et cela s'est vérifié pendant mon séjour à Brest. Comme pour l'INSA, l'ENSTA abrite des laboratoires de recherche dans les domaines de formation qu'elle propose. Pour la recherche en robotique sous-marine, l'école met à disposition des chercheurs des moyens matériels et financiers qui rendent possible la mise en application des travaux du laboratoire. L'école possède notamment plusieurs capteurs tels que des sonars et des caméras ainsi que des robots terrestres et sous-marins. Enfin, une piscine<sup>5</sup> couverte<sup>6</sup> creusée dans les locaux de l'école facilite les expérimentations des étudiants et chercheurs roboticiens et sonaristes.

L'école accueille aussi des colloques de scientifiques et organise des concours de robotique tels que la WRSC<sup>7</sup>, une compétition de voiliers autonomes. Les nouvelles méthodes de calculs par intervalles (outil numérique permettant la résolution d'une grande classe de problèmes non linéaires), encouragées par Luc JAULIN, font aussi l'objet de nombreux projets au sein de l'école.

5. une surface de 12m<sup>2</sup> pour une profondeur de 3m

6. *couverte* : il est vrai qu'à Brest, c'est indispensable

7. World Robotic Sailing Championship

## 2. Sujet du stage

Le sujet de mon stage est le suivant :

« *Préparation du concours de robotique sous-marine SAUC-E* »

Le concours SAUC-E est un concours se déroulant en Italie et rassemblant plusieurs équipes de robotique sous-marine pour la réalisation d'une succession d'épreuves en mer. Un jury évalue les tâches réalisées.

Lors de mon arrivée à l'ENSTA, on m'a présenté un ROV conçu par la société CISCREA. Pour le faire participer au concours, il était nécessaire de l'équiper d'un système embarqué pour le rendre autonome (pour qu'il devienne AUV). Il s'agit de le doter de capteurs et d'unités de calculs. Mon objectif était donc de rendre ce ROV autonome en 5 semaines, avant notre départ en Italie.

## 3. Planning - déroulement des activités

D'une durée de 9 semaines, ce stage a eu lieu pendant les mois de juin et de juillet. Voici une brève description de son déroulement :

- **Semaine 1** : découverte de l'ENSTA, des robots, de MOOS et des règles du concours SAUC-E
- **Semaine 2** : conception de l'architecture du projet, premières implémentations MOOS
- **Semaine 3** : développement des applications MOOS : utilisation des capteurs de l'AUV
- **Semaine 4** : aménagement du système embarqué - réception des mexicains à la gare
- **Semaine 5** : derniers préparatifs - rassemblement du matériel - départ de Brest le vendredi
- **Semaine 6** : **Italie**, concours SAUC-E à La Spezia
- **Semaine 7** : rangement du matériel - rassemblement des données collectées en Italie
- **Semaine 8** : nouveaux essais sur les algorithmes de suivi de murs et de localisation
- **Semaine 9** : préparation d'une mission autonome autour d'une île

J'ai pu bénéficier d'une grande autonomie pendant ces 9 semaines : j'avais le robot à ma disposition en permanence et mes travaux m'ont permis d'avoir une vue globale et précise du fonctionnement du projet. Mohamed Saad IBN SEDDIK, doctorant dans le laboratoire, m'a d'abord présenté le concours et le ROV pendant les premiers jours de ce stage. Après cela, j'ai travaillé en autonomie puis en collaboration avec Vincent DREVELLE, post-doctorant, qui m'aidait dans la réalisation des interfaces logicielles des capteurs.

Des essais quotidiens en piscine ont eu lieu dès le premier jour. Une bonne prise en main du robot est essentielle pour le concours.

Une description complète des activités qui ont précédé le challenge est disponible dans les annexes de ce rapport.

Deuxième partie  
Préparation du robot

### III | Mise en place du hardware et des interfaces logicielles

#### 1. La base du CISCREA

CISCREA est une entreprise française de conception/réalisation de systèmes robotiques fonctionnant en milieux hostiles. La société travaille donc sur la fabrication de robots sous-marins.

L'ENSTA souhaitait développer davantage ses travaux de recherche en robotique sous-marine ; elle a ainsi fait appel à la société CISCREA pour un projet de réalisation de quatre ROV identiques et répondant aux spécifications fixées par le laboratoire. De cette demande, quatre sous-marins nommés *Jack* (et souvent surnommés CISCREA) sont nés et appartiennent aujourd'hui au laboratoire.



FIGURE 4 – Un ROV *Jack* avant les travaux pour le rendre autonome

Ces sous-marins sont mécaniquement conçus pour descendre à 50m de profondeur. Ils disposent d'une batterie amovible, extérieure au compartiment étanche dans lequel se loge l'électronique capable de contrôler les six propulseurs. Chaque modèle est aussi doté d'une caméra orientable, d'une centrale inertielle basique, ainsi que de quelques capteurs de température interne, de détection de fuite, ou de pression pour déterminer la profondeur d'évolution du robot. La communication avec le robot se fait par liaison filaire à travers le protocole Modbus.

L'ENSTA a donc souhaité transformer ces ROV pour les rendre autonomes. C'est ici que je suis intervenu dans le cadre de mon stage. L'un des robots conçus par CISCREA, nommé *Jack*, est renommé *Vici*<sup>8</sup> après avoir été rendu autonome.

8. l'idée sera d'intégrer *Vici* dans une meute de trois AUV pour de la robotique de collaboration : *Veni Vidi Vici*

## 2. La propulsion vectorielle : intérêts et implémentation

Le CISCREA ne dispose pas d'un système de propulsion classique, où l'on retrouve des propulseurs agissant de manière indépendante pour aller dans une direction donnée. Ici, le robot dispose de 6 propulseurs :

- 2 dans un plan vertical, non dissociables ;
- 4 dans un plan horizontal, indépendants.



FIGURE 5 – L'un des 6 propulseurs du CISCREA, en action

La combinaison de ces propulseurs permet au robot d'évoluer selon 4 degrés de liberté :

- translation selon  $x$  (avancer/reculer) ;
- translation selon  $y$  (bâbord/tribord) ;
- translation selon  $z$  (immersion/émersion) ;
- rotation selon  $z$  : le *yaw*.

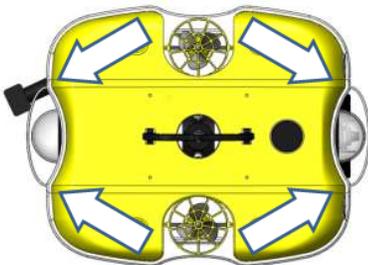


FIGURE 6 – Vue de dessus

Ces combinaisons peuvent être exécutées simultanément ou indépendamment, cela dépend de la commande envoyée à l'ensemble de ces propulseurs. Cette commande est déterminée à l'aide d'un système matriciel [3].

Deux degrés manquent à la liste (rotations selon  $x$  et  $y$ ) mais ils ne sont pas indispensables pour la manœuvre du robot. La disposition des quatre propulseurs horizontaux permet de faire évoluer le sous-marin selon deux degrés de translation et un degré de rotation.

Prenons  $C_{aaw}$ , un vecteur comportant les consignes à appliquer au robot. Dans un soucis de généralisation, on lui donne une dimension de 6 afin de représenter les 6 degrés de liberté possibles.

$$C_{auv} = (Vx, Vy, Vz, Rx = 0, Ry = 0, Rz)^T \quad (1)$$

Considérons maintenant le vecteur  $F_{prop}$ , représentant les 6 propulseurs du sous-marin. Nous cherchons à obtenir les puissances à appliquer aux propulseurs ( $F_{prop}$ ) à partir des consignes choisies ( $C_{auv}$ ) et connaissant la matrice de configuration du robot :  $E$ .

Cette matrice est déterminée en fonction de la configuration des propulseurs sur le robot.

$$E = \begin{pmatrix} -\cos(\theta) & \cos(\theta) & \cos(\theta) & -\cos(\theta) & 0 & 0 \\ -\sin(\theta) & -\sin(\theta) & \sin(\theta) & \sin(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \cos(\theta) & -\cos(\theta) & \cos(\theta) & -\cos(\theta) & 0 & 0 \end{pmatrix} \quad (2)$$

Avec  $\theta$ , l'angle d'orientation des propulseurs. L'équation générale est la suivante :

$$C_{auv} = E \times F_{prop} \quad \text{et donc :} \quad F_{prop} = E^{-1} \times C_{auv} \quad (3)$$

$E^{-1}$  pourrait être calculée une fois pour toute sur un logiciel de calcul numérique mais j'ai préféré implémenter directement  $E$  dans le programme embarqué sur le CISCREA : cela permet de revenir facilement sur les valeurs de cette matrice de configuration (ce qui, en mission, est assez intéressant). Son inversion est donc réalisée à chaque lancement du programme, par OPENCV.

Les combinaisons de mouvements sont désormais possibles, comme par exemple une double translation mixée avec une rotation :

$$C_{auv} = (5 \ 0 \ 8 \ 0 \ 0 \ 4)^T \quad (4)$$

Cela aura pour effet d'actionner les six propulseurs du robot.

### 3. Régulation en cap et en profondeur

C'est l'une des étapes primordiales dans cette adaptation robotique. Le robot doit pouvoir maintenir :

- son cap (connu grâce à sa boussole ou à la centrale inertielle) ;
- sa profondeur (connue grâce au capteur de pression).

Un régulateur PID (proportionnel intégral dérivé) a été mis en place pour effectuer ces corrections. Le middleware utilisé sur ce robot (et dont il est question dans la partie *Architecture logicielle* de ce rapport) intègre un régulateur PID qu'il nous suffit de paramétrer. Les paramètres ont fait l'objet de nombreuses séances de manipulation en piscine, puis en mer.

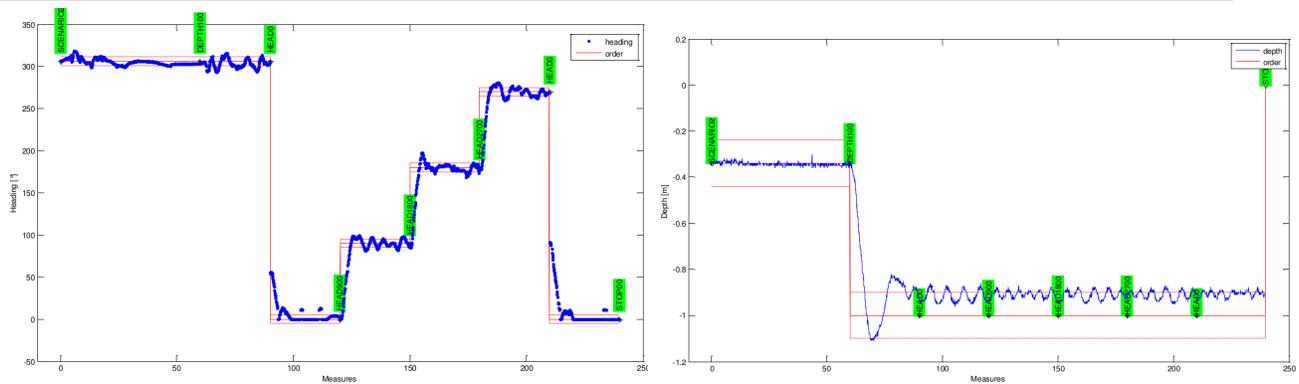


FIGURE 7 – Visualisation des régulations en cap (à gauche) et en profondeur (à droite)

#### 4. Contenu de la boîte en aluminium

Le CISCREA est donc un ROV qu'il va falloir *aüvéiser*<sup>9</sup>. Pour ce faire, le robot doit disposer d'une nouvelle unité de calculs ainsi que de nouveaux capteurs (dont certains sont à placer en milieu sec). Il a été décidé de rajouter une boîte étanche sous la base déjà existante du CISCREA.

J'ai eu la chance de débiter mon stage alors que la boîte était toujours en cours d'usinage. Je devais me charger de son aménagement interne et de son intégration dans le sous-marin : l'occasion d'avoir une vue globale et précise sur l'ensemble du robot. La boîte est *taillée dans la masse* pour répondre au cahier des charges fixé par l'ENSTA :

- disposer d'un volume suffisamment important pour y héberger le matériel nécessaire ;
- être d'une résistance élevée pour tenir les fortes pressions (elle doit pouvoir descendre à 50m) ;
- pouvoir être mise en dépression pour les essais d'étanchéité et le renouvellement d'air manuel.

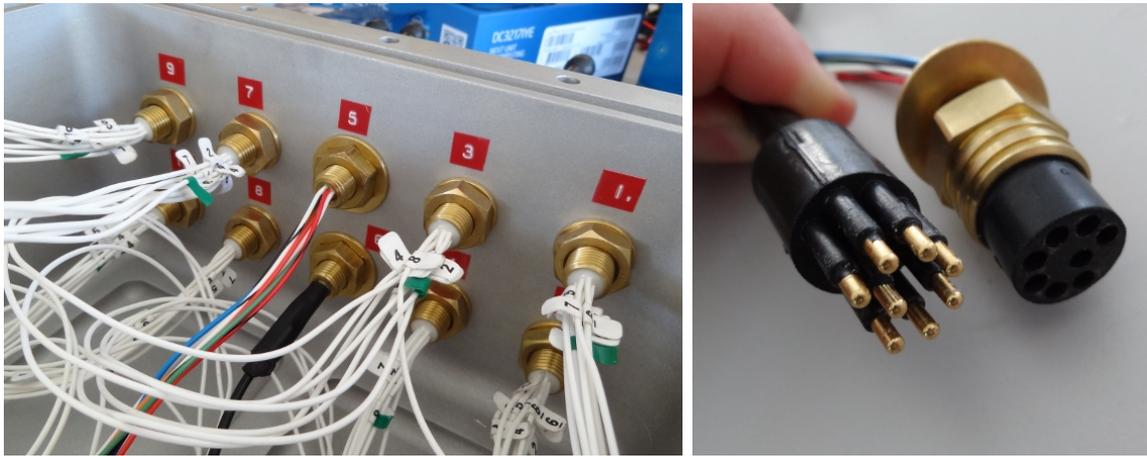
Le matériau retenu est un alliage d'aluminium adapté pour une exposition à l'humidité et au sel. La boîte est arrivée à la fin de ma première semaine de stage :



FIGURE 8 – La boîte lors de sa réception - tests d'étanchéité et de flottabilité

Dix sorties électriques de huit pins sont prévues au travers de la boîte. Ces connexions relient le contenu de la boîte aux capteurs et autres périphériques de l'AUV. Fabriqués au Mexique, il s'agit de connecteurs conçus pour tenir des profondeurs dépassant les 50 mètres. Ils sont appelés *Seacon* et *Subconn*.

9. néologisme : le fait de rendre un sous-marin autonome

FIGURE 9 – Vue sur les connecteurs *Seacon* et *Subconn*

Le contenu de la boîte est le suivant :

- le NUC d'Intel ;
- une batterie 19V ;
- un régulateur 19 - 12V ;
- une centrale inertielle RAZOR ;
- un quadruple convertisseur serial - USB ;
- un driver pour hydrophones ;
- deux drivers pour caméras ;
- un hub USB.

### La centrale inertielle

La centrale inertielle<sup>10</sup> renseigne l'AUV :

- sur son angle de rotation selon  $X$ ,  $Y$  ou  $Z$  :  
c'est la partie *magnétomètre* ;
- sur son accélération selon  $X$ ,  $Y$  ou  $Z$  :  
c'est la partie *accéléromètre*.

Elle est indispensable pour la navigation du robot ; elle est aussi extrêmement sensible et très perturbable par les champs magnétiques avoisinants<sup>11</sup>. Une fois installée dans l'AUV, il est nécessaire de la calibrer pour gommer les perturbations liées au robot et à sa structure. Une interface prévue par le constructeur facilite cette calibration.

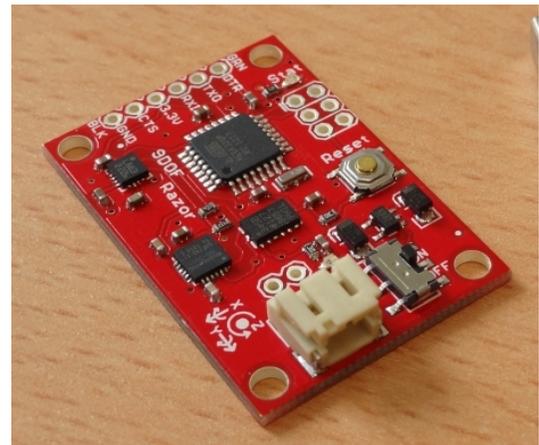


FIGURE 10 – La centrale inertielle

### Le Next Unit of Computing d'Intel

Le « NUC » d'Intel représente le cœur logiciel du robot. L'unité abrite une carte mère initialement destinée à un fonctionnement dans des Ultrabooks. Un processeur *Core i3* y prend place avec un double-cœur hyper-threadé, cadencé à 1,8 GHz. On trouve parmi ses sorties une prise HDMI, un port Gigabit Ethernet et trois ports USB 2.0.

Pour les besoins de l'AUV, le NUC sera aussi doté d'une antenne Wifi.

10. communément appelée IMU : Inertial Measurement Unit

11. anecdote : une équipe écossaise s'est rendue au concours cette année ; l'AUV a suivi dans l'avion et la centrale a, semble-t-il, été perturbée pendant le transport. Malgré le niveau de leur construction, ils n'ont malheureusement pas pu participer aux épreuves car l'AUV ne parvenait pas à se contrôler. Ceci prouve la grande sensibilité de certains capteurs et leur impact important dans l'ensemble du système.

## Emplacements

Les composants sont placés en fonction de leurs émissions thermiques et des flux de ventilation du NUC. J'ai aussi pris soin de placer la centrale inertielle le plus loin de toutes interférences et perturbations magnétiques.

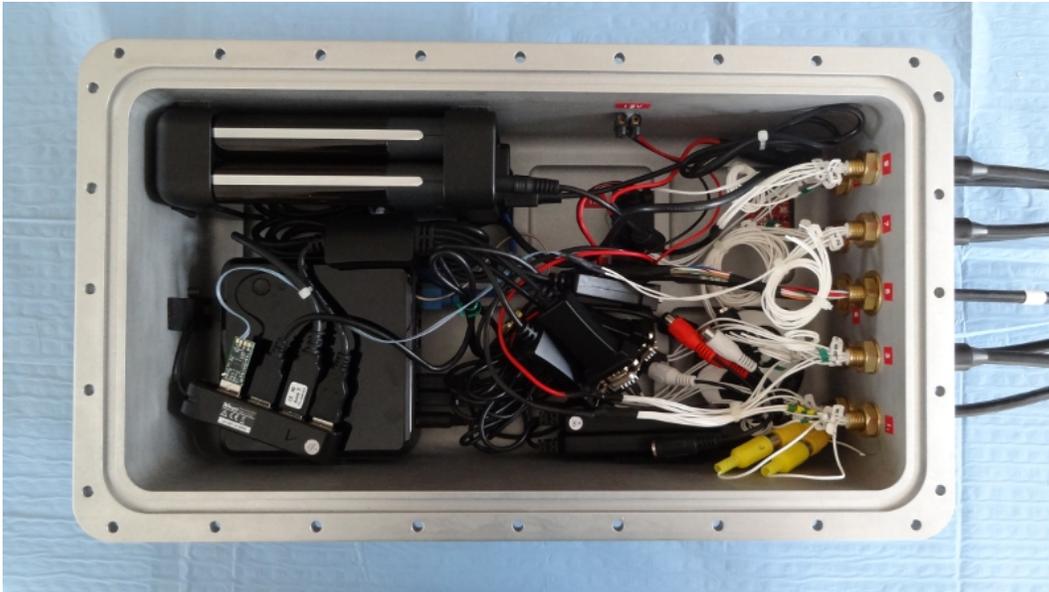


FIGURE 11 – Disposition finale des composants ; c'est *embarqué* !

Les capteurs externes du sous-marin sont ensuite placés autour de sa coque.

## 5. Un sonar pour cartographier

Les données sonar sont difficiles à analyser : une présentation en polaires, une forte présence de bruits, un jeu de données non simultanées ; autant de paramètres qu'il faut pouvoir traiter. C'est ce qui en fait une discipline toute particulière dans le monde de la recherche sous-marine. L'image qui suit est une présentation des données sonar récupérées lors d'un scan de la piscine de l'ENSTA pendant des essais du matériel. Si la piscine peut être assimilée à un environnement parfait pour de nombreux travaux (analyse d'images nettes en eau limpide, déplacement dans un milieu calme, absence de nuisances sonores), c'est loin d'être le cas lors de l'utilisation du sonar : les parois du bassin sont si nettes qu'elles réfléchissent les ondes sonar, provoquant un écho important dans les données.

Ceci se remarque particulièrement sur la visualisation polaire ci-après, où la surface rectangulaire du bassin se retrouve plusieurs fois. Le point central noir, rappelant la position du sonar, nous indique quel ensemble de données étudier. Lors des essais, le sonar se trouvait dans un coin de la piscine et il devait détecter un sous-marin évoluant près de la surface ; les deux images ont été capturées au même instant.

Le sonar est positionné sur le robot et offre une vision de son environnement en coupe horizontale.

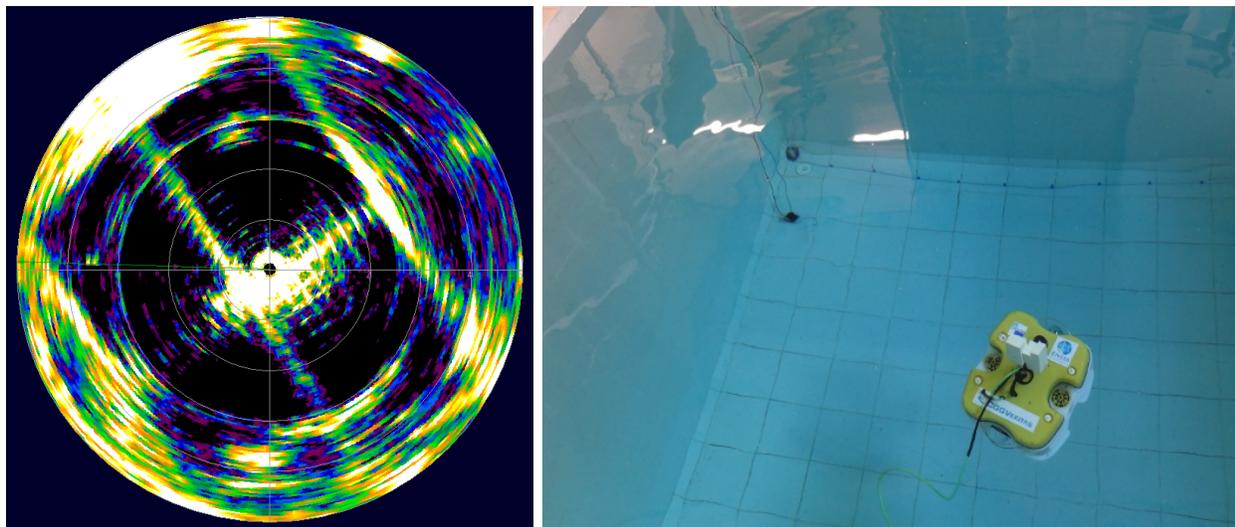


FIGURE 12 – Analyse de la piscine de l'ENSTA par sonar

## 6. Un échosondeur pour donner une échelle

Un échosondeur est un simple détecteur d'obstacle dans une dimension. Il émet un signal sonore et calcule son délai de réponse. La précision des données récupérées est plutôt bonne mais j'ai malgré tout appliqué un filtre médian sur ces résultats. Cela permet de ne pas prendre en compte certaines anomalies dues au passage d'un poisson ou à la réception d'un mauvais écho.

Cet échosondeur, pointé vers le bas, est utilisé dans deux cas : il renseigne sur la distance qui sépare le robot du fond (ce qui se révèle utile pour éviter les collisions avec des roches en mer) et donne une échelle aux images prises par l'une des caméras, elle aussi tournée vers le fond.

## 7. Un modem pour communiquer avec un autre AUV



FIGURE 13 – Un modem Tritech

La communication entre deux AUV dans leur milieu d'évolution est une chose très intéressante. Elle introduit rapidement la notion de meute et de travail en collaboration : un domaine où de nombreuses innovations vont prochainement voir le jour. Si la communication entre plusieurs robots est intéressante, elle est aussi particulièrement complexe surtout lorsqu'elle est sous-marine. Le fort taux d'absorption de l'eau ne permet pas de transmettre beaucoup d'information. Les meilleurs résultats sont obtenus par transmission acoustique sous-marine. Le débit de communication est de l'ordre de deux octets par seconde, ce qui est suffisant pour communiquer une position ou une information simple comme l'état d'une bouée clignotante.

Des modems Tritech sont utilisés pour cette transmission. L'information à transmettre est envoyée sur leur port série, leur utilisation est relativement simple.

Un de ces modems a été installé sur *Vici*. Les messages sont envoyés par blocs de quatre octets. Cette transmission acoustique est limitée dans l'espace et peut facilement s'altérer. Il est donc nécessaire de mettre en place un mécanisme logique de détection et de correction d'erreurs, ce qui réduit le débit à chaque envoi de messages.

Plusieurs solutions ont été envisagées :

- le contrôle de parité ;
- le LRC<sup>12</sup> : contrôle de parité croisé ;
- le CRC<sup>13</sup> : contrôle de redondance cyclique.

Mais par manque de temps, une solution rapide et temporaire a été mise en place. Sur les quatre octets du message, deux sont utilisés pour communiquer l'information et le reste est utilisé pour vérifier son exactitude ; le message est simplement envoyé en double pendant la transmission. Il s'agit d'un code de détection et non de correction d'erreur. Si les deux paires d'octets ne sont pas identiques, l'AUV récepteur ne considère pas le message et attend que l'émetteur renvoie son information. Sinon, il confirme la bonne réception du message en diffusant deux paires d'octets particuliers. L'émetteur envoie son message tant qu'il n'a pas reçu la confirmation du récepteur.

Cette méthode de communication a été rapidement implémentée et a fait ses preuves dans la piscine de l'ENSTA. Elle est perfectible mais suffisamment fonctionnelle pour le concours : les AUV devront communiquer des messages relativement légers. Enfin, un protocole de communication a été mis en place pour l'interprétation des octets reçus.

## 8. Deux caméras pour percevoir l'environnement

Deux caméras étanches ont été installées, l'une orientée vers le fond et l'autre vers tribord :



FIGURE 14 – Les deux caméras étanches ajoutées au CISCREA

---

12. Longitudinal Redundancy Check

13. Cyclic Redundancy Check

Plusieurs heures ont été passées sur la configuration des drivers et API vidéos (les `v4l2` et `em28xx`). Il était impossible de faire fonctionner les deux caméras simultanément. Il nous a fallu recompiler les drivers officiels et utiliser certaines bibliothèques `OPENCV` pour parvenir à utiliser ces deux capteurs. Nous récupérons les images en `320x240`, un format suffisant pour le traitement d'images que nous souhaitons appliquer.

## Vue d'ensemble

Les capteurs sont disposés autour de la boîte en aluminium et restent amovibles en cas de besoin. La géométrie du robot les protège contre les chocs. Enfin, des zones de lestage sont prévues sur le châssis du robot. Le sonar est visible sur le capot jaune ; il balaye la zone horizontale se trouvant au dessus du sous-marin.



FIGURE 15 – *Vici*, physiquement prêt pour le concours

## IV | L'architecture logicielle

La partie logicielle s'implémente sur le NUC, embarqué dans la boîte en aluminium. Un système Ubuntu y est installé, dans sa dernière version LTS 12.04. Les programmes sont majoritairement développés en C++ car c'est sur ce langage qu'est basé le middleware MOOS. Aussi, l'analyse d'images et certains calculs matriciels sont réalisés à l'aide d'OPENCV, en C++.

Pour la gestion des versions, il a été décidé d'utiliser le service web Github. L'occasion d'organiser les travaux effectués pendant l'été et de fournir un accès facile et partagé aux sources de l'AUV.

### 1. L'architecture MOOS : un middleware du MIT

#### Présentation générale [4]

MOOS est un middleware développé par le MIT<sup>14</sup> et utilisé en robotique marine et sous-marine comme base de communication entre les différentes applications embarquées sur les robots. MOOS<sup>15</sup>, « a Mission Oriented Operating Suite », est un équivalent de ROS, « Robot Operating System », très utilisé en robotique terrestre.



Un des membres de l'équipe a proposé de mettre en place une architecture MOOS sur le CISCREA. J'ai saisi l'occasion de découvrir ce middleware mais ne connaissant pas ROS, je ne suis pas en mesure de comparer ces deux outils. J'ai en revanche une idée précise du fonctionnement de MOOS et je me rends compte aujourd'hui qu'un middleware de ce type, s'il est fiable et bien maîtrisé, est un réel atout dans les applications robotiques. Voici une brève description de son fonctionnement<sup>16</sup>.

MOOS permet de diviser le programme du robot en une suite de processus pouvant être démarrés simultanément ou séquentiellement au lancement d'une mission. Les applications communiquent les unes avec les autres au travers de la MOOSDB : une base de données centrale et dans laquelle il est uniquement possible de voir l'état courant des variables qu'elle contient. Pour le CISCREA, on y stocke des positions géographiques, des vitesses ou des caps de navigations, des images en binaire, des températures de processeurs, etc.

Nous avons développé notre architecture MOOS en C++<sup>17</sup>, un langage commun à MOOS et à OPENCV. Chaque application est donc un programme C++ basé sur l'instanciation d'une classe qui hérite de la classe `CMOOSApp` :

---

14. Massachusetts Institute of Technology, USA

15. proche de *moose* (trad : élan), rappelant aux chercheurs américains les manipulations de drones sous-marins en hiver dans la *Charles River*

16. pour plus d'informations, voir le site officiel ; si la documentation de MOOS est pauvre, les tutoriaux officiels d'apprentissage sont en revanche très riches

17. une implémentation en Python aurait aussi été possible

```

1  class ApplicationPerso : public CMOOSApp
2  {
3      public:
4          ApplicationPerso();
5          ~ApplicationPerso();
6
7      protected:
8          bool OnNewMail(MOOSMSG_LIST &NewMail);
9          bool Iterate();
10         bool OnConnectToServer();
11         bool OnStartup();
12         void RegisterVariables();
13     }

```

Les méthodes définies ci-dessus sont à surcharger et permettent d'établir le comportement de notre application.

- `OnStartup()` : méthode appelée au lancement de l'application ; c'est ici que l'on récupère les paramètres pour la configuration du programme ;
- `RegisterVariables()` : inscription de l'application à l'évolution de certaines variables de la MOOSDB ; dès qu'une variable sera mise à jour, la fonction `OnNewMail` sera exécutée ;
- `OnConnectToServer()` : méthode appelée dès que le contact avec la MOOSDB est effectué ;
- `OnNewMail(MOOSMSG_LIST &NewMail)` : méthode appelée lorsqu'une variable de la MOOSDB est mise à jour ; c'est ici que l'on récupère la nouvelle valeur de la variable modifiée ;
- `Iterate()` : méthode appelée automatiquement périodiquement. Si par exemple l'application est une interface avec l'un des capteurs du robot, c'est dans cette méthode que la récupération de données et leur publication sur la MOOSDB aura lieu.

L'équipe du MIT a développé plusieurs programmes pour faciliter l'utilisation de ce middleware.

En voici cinq principaux :

- `pAntler` : un programme permettant de lancer un ensemble de MOOSAPP ayant des paramètres bien définis ;
- `uXMS` : un programme affichant les variables de la MOOSDB dans une console ;
- `uPokeDB` : un programme pour changer manuellement la valeur d'une variable de la MOOSDB ;
- `pLogger` et `uPlayback` : un programme enregistrant tous les changements de valeurs sur les variables et un autre permettant de *rejouer* une mission en faisant une simulation de ces changements.

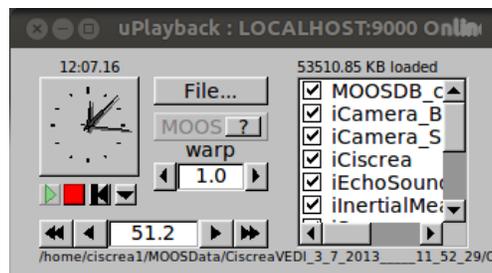


FIGURE 16 – Le programme permettant de rejouer des missions

Les rejeux de missions ont été particulièrement utiles lors du concours : on loggait nos essais en mer dans la journée et on rejouait les missions le soir pour améliorer nos programmes et voir, *en temps réel*, leur exécution.

### Application sur le CISCREA

Après quelques jours de prise en main, il était temps d'en venir au concret en définissant les applications de notre projet. On peut définir 3 types d'applications :

- les [i]nterfaces avec les capteurs et les autres éléments du hardware ;
- les interfaces [u]tilisateurs, pour de la visualisation de données ;
- les [p]rocessus, véritable logique métier au sein du projet.

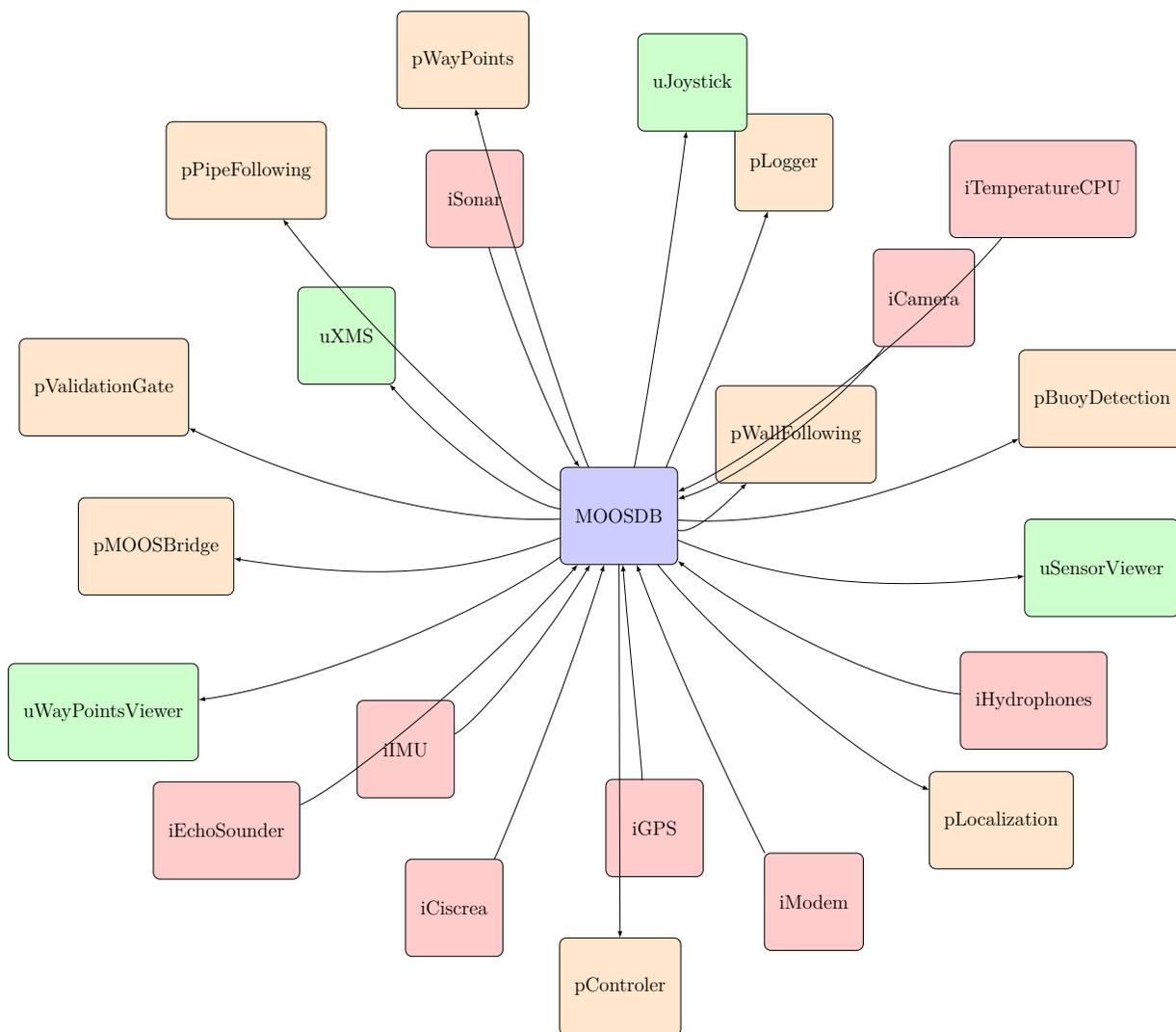


FIGURE 17 – L'architecture logicielle du projet avec la base de données en position centrale

Les applications pWallFollowing, pBuoyDetection, pValidationGate, pPipeDetection et pPingLocator décrivent le comportement du robot pendant les épreuves du concours SAUC-E. Elles sont détaillées dans la troisième partie de ce rapport.

## En bref

Une architecture de ce genre présente l'avantage de pouvoir :

- **séparer** les applications et donc faciliter la répartition du travail au sein d'une équipe ;
- **paralléliser** facilement les processus et tirer profit des  $n$  cœurs du processeur utilisé ;
- **répartir** les applications sur différentes machines car elles sont toutes liées à un serveur ;
- **rejouer** des missions car il est facile de logger tous les changements sur la MOOSDB ;
- **réutiliser** les applications déjà implémentées par les chercheurs du MIT et qui sont généralement tournées vers des problématiques maritimes.

Si je n'y vois que des avantages, d'autres ne sont toutefois pas du même avis. Nous étions la seule équipe pendant le concours à utiliser ce middleware qui est largement délaissé au profit de son voisin ROS, considéré comme plus réputé et mieux documenté.

## Tests unitaires

J'ai pris la décision de mettre en place de nombreux tests unitaires sur le projet. Cela s'est révélé utile pour déterminer rapidement la bonne exécution des programmes. Je pouvais ainsi voir d'un coup d'œil si un capteur ne répondait plus.

Mes collègues se sont rendus compte que l'approche rigoureuse qu'ont les informaticiens sur des travaux logiciels manquait chez les roboticiens. L'implémentation de tests unitaires et un formalisme de conception sont deux points qui devraient être appliqués lors des travaux sur les robots.

## 2. La communication par Modbus

Le protocole de communication Modbus [5] était à l'origine présent dans l'architecture logicielle des ROV *Jack*. C'est par cette communication que les ordres étaient envoyés au robot. Il a donc fallu utiliser à nouveau ce protocole pour la communication entre l'ordinateur NUC (partie autonome) et l'interface de contrôle du robot (partie ROV).

Ce protocole est non propriétaire et relève du niveau 7 du modèle OSI. Il est en particulier utilisé dans certaines applications robotiques et automatiques. Le Modbus utilisé sur le CISCREA est encapsulé dans une trame Ethernet. La communication se base sur la norme RS-485 : une norme électrique assurant une communication sur de longues distances. Ce standard était donc particulièrement adapté lorsque le robot était utilisé en tant que ROV car le câble qui le reliait à l'ordinateur de contrôle pouvait mesurer jusqu'à 100m.

Ce long câble n'étant plus de rigueur depuis que le robot est devenu AUV, je regrette que nous soyons encore contraints aujourd'hui d'utiliser ce protocole sur le sous-marin. Nous avons eu quelques problèmes de latence et certains ordres n'étaient pas correctement interprétés ce qui, pour un système fonctionnant en temps réel, est contraignant. Toutefois, remplacer ce mode de communication par un autre impose de modifier le boîtier ROV de *Vici* : une opération loin d'être triviale.

**Troisième partie**  
**Le concours SAUC'E**

## V | Règles et organisation

### 1. Le Student AUV Challenge Europe

Le concours SAUC-E [6] est un challenge rassemblant chaque année plusieurs écoles autour d'un bassin dans le but de faire évoluer, de manière autonome, des AUV préparés à effectuer une suite de missions définies par les organisateurs du concours. Créé en 2006 par le DSTL<sup>18</sup>, ce challenge a pour vocation de développer l'état de l'art autour des AUV et d'encourager les étudiants à s'engager dans ce nouveau domaine de recherche.

Chaque année, les missions à accomplir sont de difficulté croissante. Le concours rassemble des robots très différents les uns des autres, ce qui prouve que la créativité est un aspect important dans ce secteur récent et particulièrement technique.

Cette année et pour la troisième fois, le concours se déroule au NURC<sup>19</sup>, le centre de recherche de l'OTAN axé sur les activités sous-marines. Le centre se trouve sur la cote méditerranéenne, à La Spezia en Italie : à quelques 1550km des locaux de l'ENSTA. Le voyage se fera par la route avec une voiture et un camion de location dans lequel cinq sous-marins et beaucoup de matériel seront embarqués.

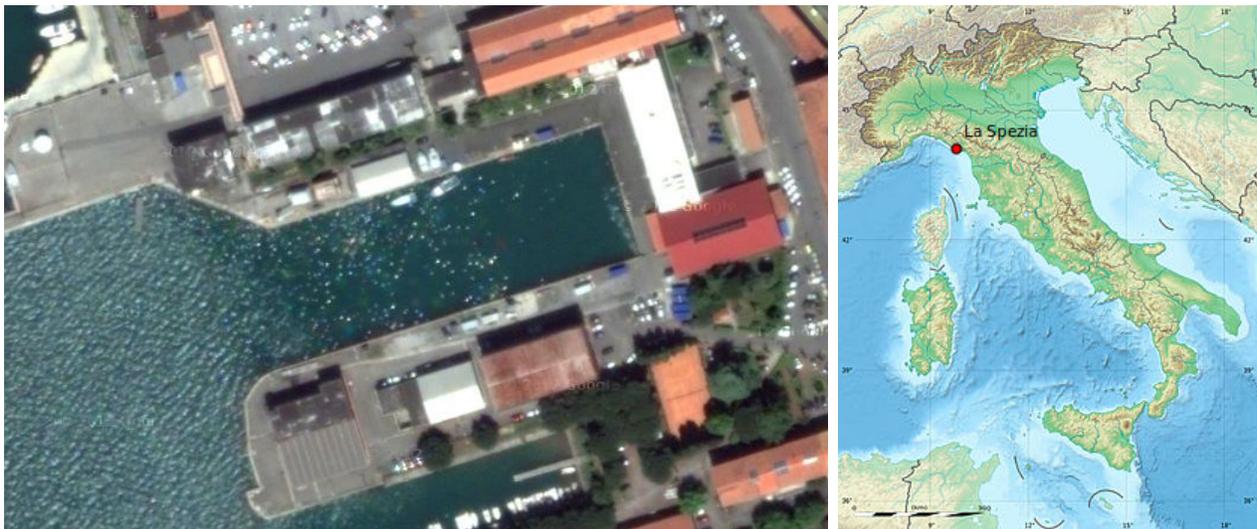


FIGURE 18 – Image Google et localisation du bassin du centre de recherche de l'OTAN

Le bassin dans lequel se déroule la compétition se situe en zone militaire et fait 50 mètres de large, 120 mètres de long et a une profondeur de 5 à 6 mètres. C'est une zone ouverte sur la Méditerranée : l'eau y est donc trouble et salée. Si l'AUV était prêt pour une mission dans la piscine d'eau douce limpide de l'ENSTA, il ne l'est plus pour ce nouvel environnement.

18. le DSTL est un centre d'excellence scientifique regroupant 3500 chercheurs en Angleterre

19. NATO Undersea Research Centre

## 2. Les règles du concours

### Le système de notation

Le challenge base sa notation sur un ensemble de tâches à réaliser en mer ou en préparation avant le début des missions en Italie. Parmi ces tâches préparatoires, on compte la réalisation d'une vidéo présentant le projet ainsi qu'un *Journal Paper* rappelant les grandes lignes de fonctionnement de l'AUV. D'autres points étaient donnés selon la qualité de la réalisation, le degré d'innovation du projet, le niveau de sécurité du robot (comportement dans l'eau en cas d'avarie) et une présentation orale devant un jury. À ces points s'ajoutaient les évaluations des missions en mer, détaillées dans les prochaines pages de ce rapport.



FIGURE 19 – Le bassin du NURC, tourné vers la Mer Méditerranée

### La décision des juges

Les missions étaient difficiles et seules trois équipes se sont qualifiées à la fin du concours. Parmi ces trois équipes, nous avons été celle à avoir réalisé le plus d'actions dans les missions qui suivent. Nous avons donc les meilleurs résultats dans les tâches à effectuer en mer. Toutefois, la faible proportion d'équipes qualifiées a incité les juges à revoir le système de notation en mettant davantage l'accent sur les travaux préparatoires. Cela nous a désavantagé et nous avons été surpris de cette décision qui pénalise les équipes les plus aptes à réaliser de réelles missions autonomes.



FIGURE 20 – Le camion et la moitié de l'atelier robotique de l'ENSTA

## VI | Les épreuves

### 1. Les qualifications : passage à travers une gate

Le passage à travers une gate est systématiquement le passage obligé avant de pouvoir réaliser une des épreuves suivantes. Pendant cette mission, il est donc favorable que le sous-marin ait un comportement fiable et robuste afin de pouvoir passer la gate autant de fois que nécessaire. La gate se trouve au centre du bassin. Elle est constituée de deux bouées flottantes entre deux eaux et reliées au fond du bassin par deux câbles verticaux ; l'objectif est de passer entre ces deux câbles. Les bouées se voient à peine depuis la surface et elles sont difficilement détectables à l'aide du sonar. Il en sera de même pour les autres épreuves du concours : l'objectif de ces missions étant de confronter les équipes à la réalité d'un environnement marin.



FIGURE 21 – *Vici* (trace jaune, en haut) s'apprêtant à franchir la gate (points oranges, à droite)

Nous avons envisagé plusieurs stratégies :

- **par sonar** : détecter, dans les données sonars, deux anomalies qui auraient un éloignement cohérent. Cette technique est rapidement abandonnée car les données sonar sont bruitées et la marque des bouées est minime comparée à l'étendue du bassin ;
- **par vision** : les bouées sont oranges et les essais de seuillage sur des teintes de cette couleur se sont révélés positifs. Toutefois, le robot doit suffisamment s'approcher du passage afin de voir les bouées. Cette stratégie pourrait, au mieux, servir à corriger sa position une fois qu'il s'apprête à franchir la gate ;
- **par *wall following*** : on le sait, la gate se trouve au centre du bassin et orientée vers la mer. L'idée serait donc de se maintenir à une certaine distance de l'un des murs du bassin tout en avançant vers sa sortie. La problématique de la détection de mur se pose ; celle-ci peut être faite en analysant les données sonar du bassin et en effectuant une régression linéaire sur la position des principaux obstacles reconnus.
- **par *dead reckoning***<sup>20</sup> : il s'agit de programmer le robot pour le faire aller dans une direction pendant un certain temps. Si cette méthode est la plus simple à mettre en place, c'est aussi la plus incertaine : elle ne permet pas de correction de trajectoire du robot en cas de vague, d'affaiblissement d'un propulseur ou de perturbations liées à l'approche du zodiac du jury.

20. méthode de positionnement par navigation à l'estime

FIGURE 22 – Bouée vue par *Vici*

La solution du *wall following* a été retenue pour sa robustesse. Un algorithme a été développé pour cette mission mais nous n'avons pas eu suffisamment de temps pour l'éprouver dans le bassin avant les qualifications. Nous avons dû envisager une solution de secours, plus rapide à mettre en place : celle du *dead reckoning*. Le sous-marin devait donc partir du startpoint en avançant tout droit jusqu'au centre du bassin, puis il devait effectuer une rotation de  $90^\circ$  vers la mer et avancer suffisamment pour franchir la gate. La rotation est rigoureuse puisque basée sur la régulation en cap implémentée sur les données de la centrale inertielle.

Après différents réglages, le robot franchissait la porte en nous qualifiant dans un grand « *hourra !* ».

## 2. Suivi de mur et détection d'anomalie

Cette épreuve consiste à inspecter un mur immergé afin d'y trouver une anomalie, symbolisée par une bouée orange clignotante. La mission se fait donc en deux temps : longer le mur à une distance constante en enregistrant les données vidéo et analyser ces images pour la détection de la bouée.

### Le cas du wall following

Le suivi de mur se base sur l'étude des données du sonar. Un seuillage est effectué sur les données récupérées (nuances sonores). Cela nous permet de déterminer où se trouve l'obstacle sur chaque scanline traitée.

Une fois paramétré de manière empirique, ce seuillage nous permet de pointer les obstacles majeurs sur un espace 2D. Il est ensuite nécessaire de déterminer un modèle (ligne, coin, rectangle) correspondant à la répartition de ces points. Les essais montrent que la recherche d'une ligne *collant au mieux* avec les obstacles permet au robot de suivre les murs même lorsque ceux-ci sont perpendiculaires.

Le robot ajuste son cap en fonction de l'orientation de la ligne calculée. Lorsque qu'il n'y a pas d'angle dans les données sonar, la ligne représente au mieux la réalité du bassin et l'AUV longe le mur. Lorsqu'un coin apparaît, la ligne calculée est de moins en moins calquée sur le mur que le robot est en train de suivre : elle effectue progressivement un virage vers le mur perpendiculaire suivant. Le robot

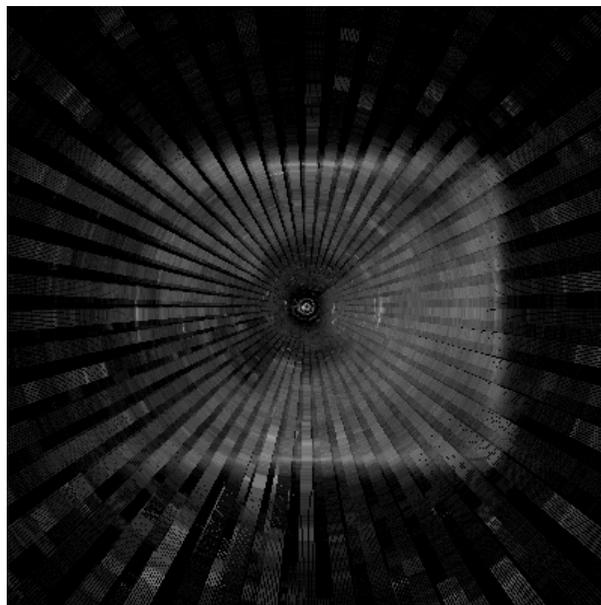


FIGURE 23 – Visualisation par sonar du bassin du NURC

se régulant directement par rapport à cette régression, il va passer d'un mur à l'autre en faisant automatiquement le bon virage.

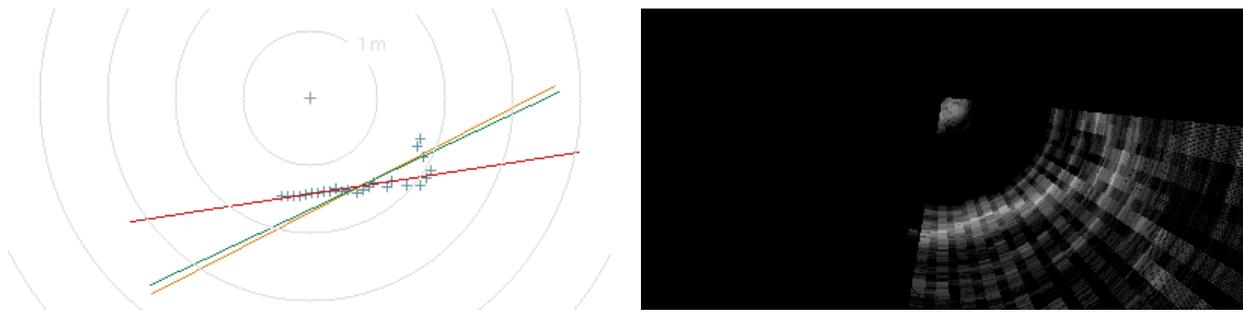


FIGURE 24 – Régressions linéaires sur les obstacles détectés

L'algorithme calcule la ligne en minimisant  $\sum p(r_i)$  où  $r_i$  est la distance entre le  $i^{\text{ième}}$  point et la ligne et où  $p(r)$  est une fonction synthétisant cette distance.

Trois méthodes  $p$  ont été expérimentées [7] :

- $p(r) = r$  : la méthode la plus légère
- $p(r) = r^2/2$  : une simple et rapide méthode de moindres carrés
- $p(r) = 2 \cdot \left( \sqrt{1 + \frac{r^2}{2}} - 1 \right)$  : la méthode finalement retenue car particulièrement robuste face aux valeurs aberrantes.

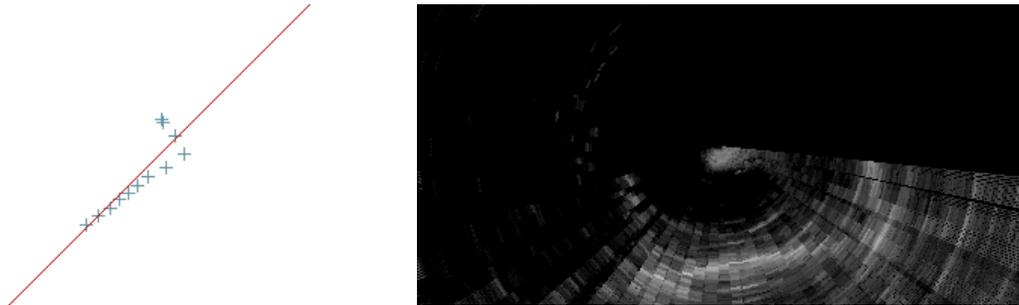


FIGURE 25 – Régulation progressive à l'approche d'un coin du bassin

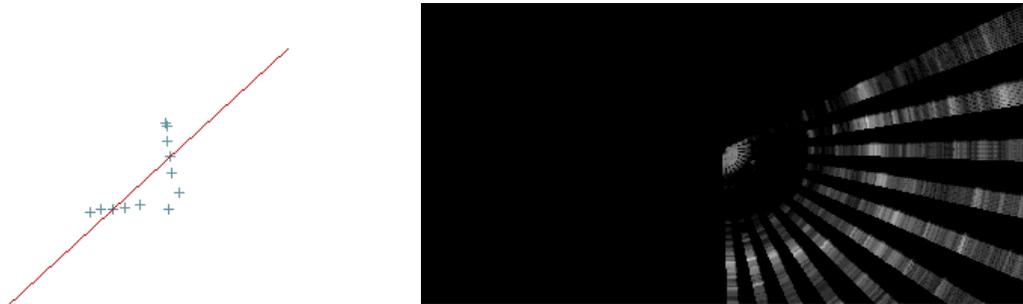


FIGURE 26 – La régression provoque un virage pour passer au mur suivant

L'implémentation de la régression est effectuée avec OPENCV et sa méthode `fitLine` :

```
void fitLine(const Mat& points, Vec4f& line, int distType, double param, double reps, double aeps)
```

## Détection d'une bouée par vision

Le robot longe le mur et s'arrête lorsqu'il détecte la bouée clignotante. Il doit alors envoyer la position de la bouée par communication modem à un autre AUV qui devra se rendre sur place et déterminer la fréquence de clignotement de l'anomalie.



FIGURE 27 – L'AUV localisant la bouée qui symbolise l'anomalie

La bouée est jaune et c'est sa couleur qui va être recherchée dans les images fournies par l'une des caméras. L'étude des images ne se fait pas sur chacune des composantes RGB mais sur une décomposition en valeurs HSV<sup>21</sup>.



FIGURE 28 – Bouée vue par *Vici* en Italie (à gauche) et à l'ENSTA (à droite)

La teinte jaune est recherchée et elle s'exprime selon trois composantes dans l'espace RGB alors qu'elle ne prend qu'une valeur dans l'espace HSV. Enfin, un seuillage est effectué sur la teinte retenue ; l'ensemble de points récupérés correspond à la visualisation de la bouée dans l'image. Ce sont les points rouges que l'on voit sur la figure ci-dessus. Le robot considère qu'il a vu la bouée lorsqu'une certaine proportion de points dans l'image est atteinte. La position de la bouée dans l'image est aussi calculée en faisant la médiane des coordonnées des points du seuillage. La position obtenue permet au robot de se réguler pour garder la bouée dans le champ de la caméra.

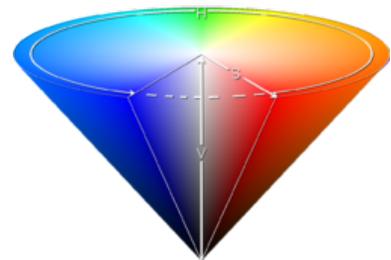


FIGURE 29 – Répartition des couleurs dans l'espace HSV [8]

21. HSV : *Hue (teinte), Saturation, Value*

### 3. Inspection et suivi d'une structure immergée

Une autre épreuve consiste à repérer et à suivre une structure immergée. Il s'agit typiquement de tuyaux de couleur jaune assemblés au fond du bassin. Les années précédentes, les tubes étaient disposés les uns à la suite des autres selon un chemin courbé ; le robot devait prendre des images de cette structure en la suivant d'une extrémité à l'autre.

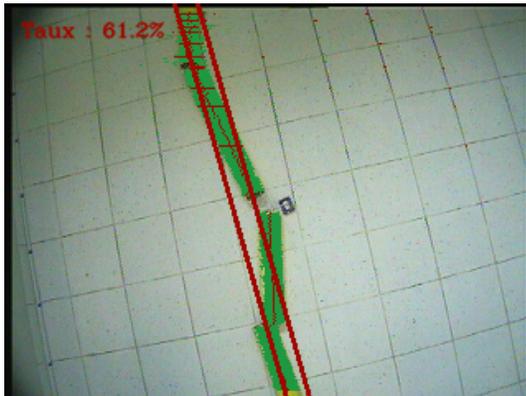


FIGURE 30 – Détection de l'orientation du tube vu en piscine

#### Le suivi de tuyaux

Ce suivi passe d'abord par la détection des tubes et ici, la technique employée est la même que celle utilisée pour la reconnaissance de la bouée. Un seuillage est donc effectué sur les couleurs des pixels de l'image. Puis, chaque ligne horizontale de l'image (scanline) est étudiée : la position médiane des pixels seuillés est calculée. Ensuite, une régression linéaire est réalisée sur l'ensemble des médianes de chaque scanline. Cela permet de déterminer l'orientation du tube par rapport à la caméra.

Cette technique présente toutefois ses limites lorsque le tube est visible horizontalement dans l'image car plus la droite est horizontale, plus son coefficient directeur sera infini. Il faudrait donc améliorer cette reconnaissance en utilisant un modèle de droites exprimé en coordonnées polaires et non en coordonnées cartésiennes.

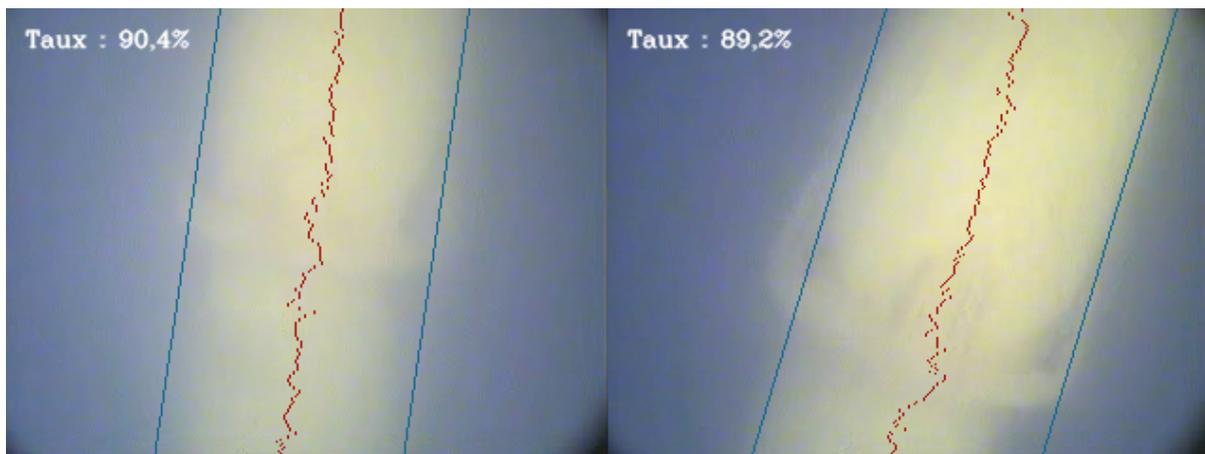


FIGURE 31 – Détection de l'orientation et de la largeur du tube vu en mer

Sur l'image ci-dessus, la visibilité est bien moindre : les essais ne sont plus faits en piscine mais dans le bassin du NURC. Les points rouges déterminent les médianes des points seuillés sur chaque scanline. Les droites bleues déterminent l'orientation et la largeur du tube. La largeur est évaluée dynamiquement après un changement de repère dans le sens du tube.



FIGURE 32 – La structure immergée, vue pour la première fois

### Le cas de la structure de l'édition de 2013

Ce type de suivi de tuyaux ne faisait pas l'objet du concours cette année. La structure à repérer était en effet bien plus complexe, non linéaire. La photo ci-contre a été prise par *Vici* pendant une mission d'exploration dans le bassin du NURC. La structure est constituée de tubes empilés les uns sur les autres.

Par manque de temps et à cause d'une édition difficile cette année, cette structure n'a pas été étudiée davantage.

## 4. À la recherche d'une boîte noire

Une autre épreuve du concours consistait à localiser une boîte noire comme celles que l'on trouve dans les avions. Cette boîte, déposée au fond du bassin, émet un son caractéristique que les AUV doivent pouvoir détecter. C'est pour cette raison qu'il était prévu d'installer deux hydrophones ainsi que le matériel audio nécessaire sur *Vici*. Si plusieurs algorithmes ont été développés pour diriger le robot vers la source sonore, ils n'ont pas pu être testés lors du concours. La boîte noire n'a même pas été plongée dans le bassin car aucune équipe n'avait le temps de s'en préoccuper.

La localisation par hydrophones est particulière ; elle impose un positionnement précis et suffisamment éloigné des deux capteurs. Une des méthodes retenues était la localisation par différence de phases entre les deux signaux considérés. Je ne détaille pas cette technique dans ce rapport car n'ayant pas travaillé sur cette partie, je ne suis pas en mesure de l'expliquer.

## 5. Cartographie et localisation

L'ultime épreuve du concours consistait à tracer, à la fin d'une succession de missions, une carte globale du bassin. Les positions de la structure immergée, de la bouée sur le mur, de la gate et de la boîte noire y seraient indiquées. Pour réaliser un tel rendu, il est impératif de parfaitement se localiser dans le bassin. Un programme de localisation a donc été implémenté à partir de l'étude des données sonar. La difficulté majeure est qu'un jeu de données à 360° n'est pas obtenu instantanément car le sonar met un certain temps pour effectuer une rotation complète. Pendant cette acquisition, le robot peut être soumis à une variation de son environnement (déplacement d'un zodiac) ou à un mouvement (rotation, translation du robot lui-même) qui brouillerait l'ensemble des données.

Cette problématique n'est pas triviale, elle fait l'objet de nombreux travaux de recherche notamment sous le nom de Simultaneous Localization And Mapping (SLAM) : lorsqu'un robot autonome, évoluant dans un environnement inconnu, cherche à dessiner une carte de son environnement tout en utilisant cette carte pour calculer sa position. Des travaux de recherche sur cette thématique

ont été entrepris à l'ENSTA en particulier par Luc JAULIN, Fabrice LE BARS et Jan SLIWKA : trois chercheurs que j'ai pu rencontrer pendant mon stage. Ils proposent une nouvelle approche au problème en utilisant des méthodes de calculs par intervalles.

L'arithmétique des intervalles est une méthode de calculs particulière qui permet de garantir l'égalité dans chaque relation mathématique. Elle est donc intéressante dans des applications robotiques où l'imprécision des capteurs demande une certaine flexibilité dans l'utilisation de leurs données. Cette méthode est particulièrement étudiée à l'ENSTA, lieu de colloques scientifiques mondiaux dans ce domaine.

## 6. Conclusion sur les résultats obtenus

On regrettera évidemment de ne pas avoir pu mettre en œuvre tous les programmes implémentés avant le concours. Le manque de temps est une réalité dans ce genre de challenge. De nombreux problèmes techniques sont apparus et il a bien fallu y faire face. Mais si tout n'a pas pu être fait en Italie, nous avons toutefois pu enregistrer des données en grande quantité lors des mises à l'eau de *Vici* : de quoi rejouer de nombreuses missions virtuelles à l'aide de l'interface de MOOS.

Le gestionnaire de versions Github n'a plus été utilisé en Italie. Ceci est dû au manque de connexion Internet entre chaque déplacement du robot et des ordinateurs. Il devenait plus simple de partager ses sources par support USB. Si le concours était à refaire, il faudrait replacer Github au cœur du projet en toutes situations : nous aurions pu en avoir besoin pendant cette semaine à La Spezia, et le manque de versionnage pendant cette période aurait pu être critique. Aussi, Github peut fonctionner hors-ligne en s'installant sur une machine devenant alors *Git server*. C'est la solution qu'il aurait fallu appliquer.

L'AUV *Vici* a parfaitement fonctionné pendant les épreuves. On retiendra juste la perte d'une caméra (infiltration d'eau par capillarité à travers son câble) ainsi que des problèmes d'algues dans ses propulseurs. La propulsion vectorielle du robot était un réel atout pendant les manœuvres dans le bassin.

Enfin, l'équipe s'est bien entendue pendant ces quelques jours intenses. Il y a toujours quelques pics de stress et de pression qu'il faut pouvoir surmonter sans perdre en productivité. Pour le reste, nous étions tous autant motivés à travailler sur ces épreuves. Avant le concours, il nous arrivait de rester jusqu'à minuit à l'ENSTA pour les préparatifs du robot. En Italie, les AUV retournaient au camping le soir et de nouveaux programmes étaient implémentés avant leur mise en application, le lendemain, dans les eaux du NURC.

Nous sommes donc rentrés en Bretagne déçus de ne pas avoir remporté le prix espéré après nos résultats dans le bassin, mais néanmoins satisfaits des travaux effectués en Italie au début de ce mois de juillet.

**Quatrième partie**  
**Mission autonome**

## VII | Faire le tour d'une île de manière autonome

Lorsque Luc JAULIN a proposé de profiter du beau temps pour faire quelques manipulations avec les AUV, j'ai évoqué l'idée de faire le tour d'une île de manière autonome. D'après les chercheurs, c'est une tâche qui n'avait pas encore été faite dans ce domaine ; de quoi motiver une équipe quelques jours avant les vacances d'été.

Le CISCREA pourrait faire le tour d'une île de deux manières :

- en utilisant les données sonar pour se maintenir à une certaine distance de la côte ;
- en utilisant un GPS pour faire un suivi de waypoints (en traçant le parcours auparavant).

La première solution est intéressante puisqu'elle ne nécessite pas de connaître la géométrie des lieux. L'AUV a pour consigne d'avancer en se maintenant à une certaine distance d'une série d'obstacles : c'est un algorithme de suivi de mur (wall following).

Toutefois, l'implémentation de ce type d'algorithme nécessite de disposer au préalable d'un jeu de données pour pouvoir fixer certains paramètres de manière empirique.

Il nous manquait ce jeu de données<sup>22</sup> et c'est pourquoi nous avons choisi de faire le tour de l'île en suivant des points GPS. La mission que nous planifions avait donc deux buts :

- faire un premier tour d'une île en autonomie ;
- récupérer un jeu de données pour expérimenter un algorithme de suivi de côte.

### 1. L'île des Morts, dans la rade de Brest

Nous avons donc cherché une île ayant un contour suffisamment découpé pour avoir des données exploitables ; une île de taille raisonnable avec peu de plages, proche de Brest, éloignée des courants et sans trop de fréquentations. Deux îles répondent à la demande : l'Île Trébéron et l'Île des Morts<sup>23</sup>. La première se trouvant en partie dans la zone protégée de l'Île Longue<sup>24</sup>, on s'intéressera à la seconde. Si le mouillage autour de l'île est possible, il est en revanche interdit d'y poser le pied. Ceci ne devrait pas gêner la mission.



FIGURE 33 – Localisation

22. le CISCREA a toujours évolué à côté de parois artificielles telles que des digues, des bords de piscines, des quais de ports - loin de la complexité que représente une plage inclinée ou une côte rocheuse

23. le nom de l'île fait référence au cimetière qui s'y trouvait au XVIII<sup>ème</sup> siècle

24. l'Île Longue est la base sous-marine nationale de France, elle abrite les sous-marins nucléaires lanceurs d'engins

## 2. L'algorithme de suivi de waypoints GPS

L'algorithme de suivi de points est issu des travaux de Luc JAULIN et de Fabrice LE BARS [9] sur le projet VAIMOS<sup>25</sup>. Basé sur un calcul de champs de vecteurs, il est conçu pour être rapidement implémentable et fiable : il nécessite peu de paramétrages, ce qui permet de gagner du temps pendant les essais.

L'idée n'est pas d'atteindre directement le prochain point du parcours, mais de se rapprocher au plus tôt de la ligne à suivre (celle reliant le point précédent au point suivant). En effectuant un *suivi de lignes*, le robot est davantage fidèle au parcours désiré. Les écarts éventuels sont plus rapidement corrigés que dans un suivi de points trivial.

L'algorithme est défini de la façon suivante :

### Fonction : suivi de lignes

#### Entrées :

- $m$  : position actuelle du robot
- $a$  : le précédent waypoint
- $b$  : la destination à atteindre
- $r$  : le rayon de courbure lors du rapprochement de la ligne
- $\gamma_\infty$  : l'angle à suivre lorsque l'éloignement est considéré comme infini

#### Sortie :

- $\theta$  : l'angle que doit prendre le robot

#### Étapes :

1.  $e = \det \left( \frac{b-a}{\|b-a\|}, m-a \right)$
2.  $\phi = \text{atan2}(b-a)$
3.  $\theta = \phi - \frac{2 \cdot \gamma_\infty}{\pi} \cdot \text{atan} \left( \frac{e}{r} \right)$

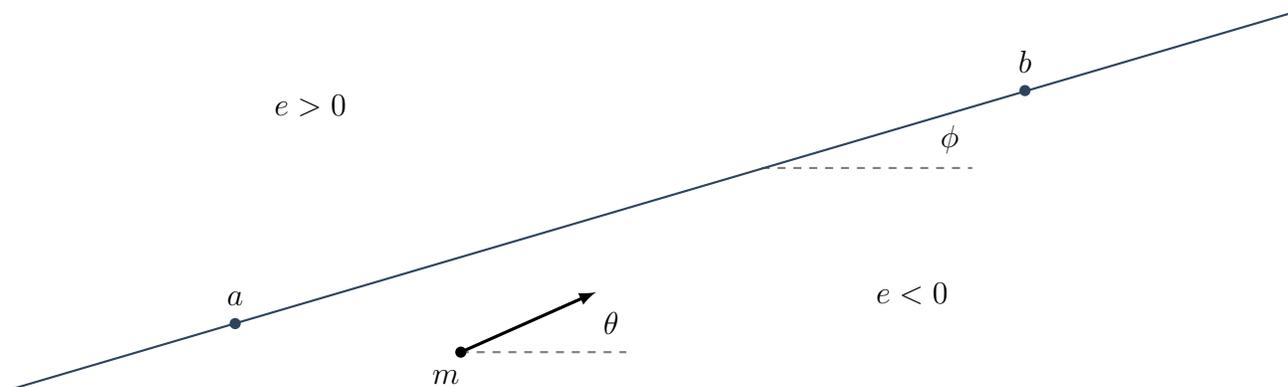
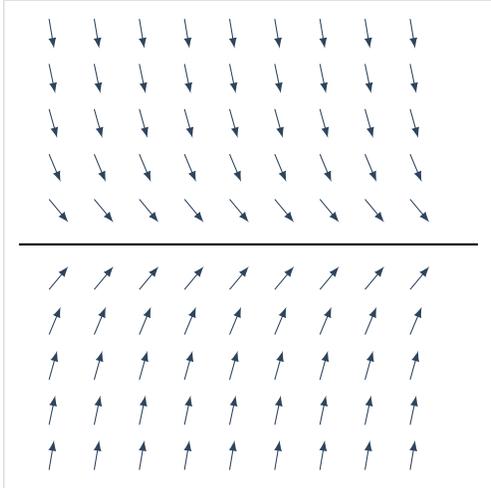


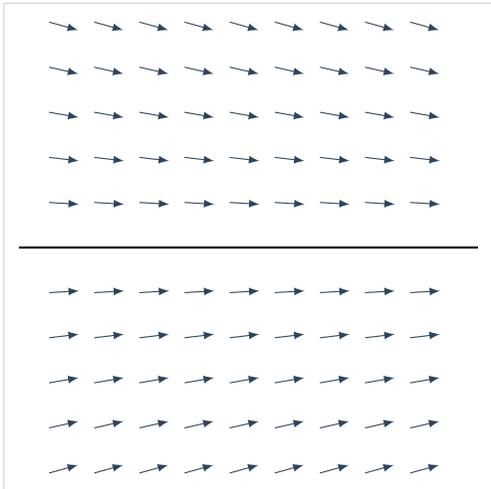
FIGURE 34 – Représentation d'une correction de trajectoire

25. VAIMOS est un voilier autonome issu d'une collaboration entre l'ENSTA et l'IFREMER

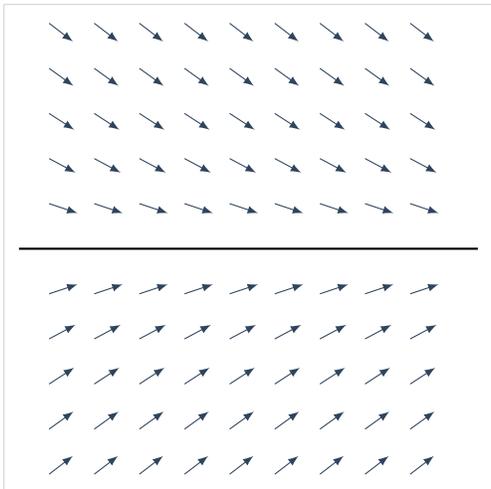
**Description des étapes**



a)  $\gamma_\infty = \pi/2, r = 0.5$



e)  $\gamma_\infty = \pi/4, r = 5$



c)  $\gamma_\infty = \pi/4, r = 0.8$

1. On calcule la distance algébrique entre le robot et la ligne<sup>26</sup>. Si  $e > 0$ , le robot se trouve à gauche de la ligne; si  $e < 0$ , il se trouve à sa droite. Le calcul du déterminant est le suivant :

$$\det(a, b) = a_x b_y - a_y b_x$$

2. On détermine l'angle  $\phi$  de la ligne à suivre. Ici,  $\text{atan2}(u) = \text{atan2}(u_x, u_y)$ . Le résultat correspond à l'angle du vecteur par rapport à l'Est.

3. On calcule enfin l'angle que doit prendre le robot pour se rapprocher de la ligne. Lorsque  $e = \pm\infty$ , on a :

$$\theta = \phi - \frac{2 \cdot \gamma_\infty}{\pi} \cdot \left( \pm \frac{\pi}{2} \right) = \phi \pm \gamma_\infty$$

C'est à dire que le robot cherchera à atteindre la ligne avec un angle de  $\gamma_\infty$  par rapport à celle-ci.

Pour l'expérience, les paramètres suivants ont été fixés :

$$\gamma_\infty = \pi/4 \quad r = 5 \text{ (en mètres)}$$

Ils correspondent à la figure ci-contre (schéma b). Les paramètres du schéma a traduisent un comportement différent du robot, où celui-ci cherche à atteindre la ligne plus rapidement.

**Déterminer qu'un waypoint a été franchi**

Un waypoint est considéré comme dépassé lorsque  $|\widehat{ABM}| > \frac{\pi}{2}$ . Cela revient à calculer le produit scalaire entre les vecteurs  $\overrightarrow{BA}$  et  $\overrightarrow{BM}$  et à vérifier que celui-ci est négatif.

L'implémentation de ces algorithmes est présentée dans les annexes.

FIGURE 35 – Champs de vecteurs

26. en pratique, la Terre n'étant pas plate, il convient de respecter une distance raisonnable entre deux waypoints (moins de 100km). Ce détail ne concerne pas la mission dont il est question.

### 3. L'expérience du 24 juillet 2013

Ce mercredi, le CISCREA a effectué un tour complet de manière autonome autour de l'Île des Morts. Le sous-marin devait donc longer l'île en suivant des points GPS définis. Le GPS se trouvait dans la bouée Wifi de l'AUV, flottant en surface et reliée par Ethernet au robot. Le GPS est lui-même relié en Ethernet au routeur se trouvant dans la bouée ; il peut ainsi capter les signaux satellites et communiquer ces informations au sous-marin.



FIGURE 36 – l'AUV faisant le tour de l'île, on ne voit que sa bouée et le plongeur

La communication avec le sous-marin était essentielle, depuis la vedette où nous nous trouvions. Une antenne Wifi *Ubiquiti* a été sommairement installée sur le toit du bateau afin de pouvoir capter la bouée dans les meilleures conditions. Nous pouvions ainsi récupérer la position GPS du robot, la trace de son parcours sur une carte numérique, ainsi que des informations importantes sur son autonomie électrique, sa profondeur et sa proximité avec le fond. En revanche, aucune directive de navigation ne lui a été donnée ; il était autonome.

Après quelques essais peu concluants le mercredi, une dernière mission a finalement été lancée à 19h au nord-ouest de l'île. La mission a duré 1h02 et le sous-marin a parcouru 1,2km. Il n'a fait surface qu'après avoir fait son tour complet.

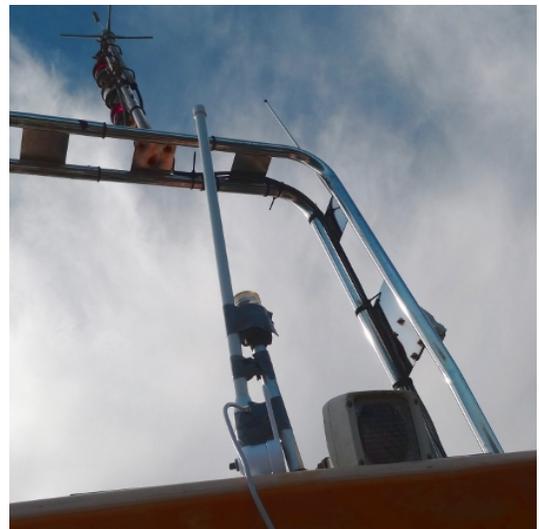


FIGURE 37 – L'antenne Wifi

Des données sonar et bathymétriques ont été logées pendant la mission. Les images des deux caméras (dont une orientée vers le fond) ont également été enregistrées. Les données sonar récupérées vont permettre de tester un algorithme robuste de suivi de murs par rebonds, basé sur des données sonar pouvant être peu précises (une idée de Luc). Le CISCREA sera donc peut-être amené à refaire le tour de cette île, cette fois-ci sans GPS.

Le parcours réel du robot est visible sur la vue satellite ci-dessous. On remarque :

- en bleu, les waypoints fixés à l'ENSTA ;
- en jaune, la trace du robot.

Le parcours s'est effectué dans le sens direct avec un départ concluant entre les points 10 et 11. La mission s'est effectuée sans encombre<sup>27 28</sup>.



FIGURE 38 – Trace GPS de l'AUV

Nous n'avons pas eu besoin de changer les paramètres de l'algorithme en cours de mission. Il a donc montré sa robustesse ; limiter le nombre de paramètres d'un code s'avère particulièrement utile en mission, lorsqu'il y a déjà de nombreux problèmes techniques à résoudre.

27. un bref moment d'inattention et l'AUV nous surprenait en passant sous la vedette, entre les points 15 et 16

28. une petite frayeur entre les points 20 et 21 : l'AUV ayant frôlé (sans toucher) le bord de l'île

# Conclusion

Ce stage prend fin après 9 semaines chargées en préparatifs, en participation à un challenge et en missions scientifiques. J'ai beaucoup appris de ces expériences riches en problématiques scientifiques et en travail de groupe. Je regrette juste que la phase de préparation du concours ait été si courte (5 semaines), mais ceci n'est dû qu'à la proximité des dates du challenge avec celles de la fin de mon année en ASI4. Les quelques nuits courtes à l'approche du concours témoignent sans doute de notre motivation à donner le maximum pour réussir, au mieux, les épreuves du concours. La récompense n'a peut-être pas été officielle mais nous avons eu de meilleurs résultats que les autres équipes, pendant les épreuves dans le bassin ; c'est ce que je retiendrai de cette semaine à La Spezia. Les choses ne sont pas toujours comme on les imagine, et on pouvait ici regretter la décision assez scolaire des juges qui ont préféré mettre l'accent sur de la documentation et de la recherche théorique plutôt que sur les véritables résultats obtenus en mer.

De nombreuses évolutions sont possibles sur l'AUV *Vici*. Si les capteurs et leur interface logicielle sont stables et peuvent être réutilisés pour de nombreuses missions, il conviendrait d'approfondir les algorithmes traitant ces données pour les tâches autonomes dont il était question pendant ce stage. Aussi, nous n'avons pas eu le temps d'installer les deux hydrophones initialement prévus. Ces capteurs permettraient au sous-marin d'augmenter significativement ses capacités de détection de l'environnement. Enfin, on pourrait développer son système de communication pour pouvoir intégrer cet AUV dans une meute de sous-marins : un domaine nouveau dans le monde de la robotique sous-marine, qui ne me laisse pas indifférent.

Ce stage mélange donc passion, innovation, initiation à la recherche et challenge. Les problématiques sur lesquelles j'ai été amené à travailler étaient centrées sur des points de vision, d'informatique répartie, de réseau, de traitement de signaux, d'analyse numérique, de capteurs, de systèmes d'exploitation, d'automatique et bien évidemment de programmation avancée : autant de matières enseignées en ASI et auxquelles il serait difficile d'oublier les thèmes de la mécanique et de l'électronique : des domaines qu'il faut connaître pour ce type de projet.

La robotique constitue donc un carrefour des technologies où il est difficile de s'ennuyer tant les problématiques sont nombreuses, variées et intéressantes. J'ajouterais aussi qu'elle présente l'avantage de pouvoir travailler sur des systèmes réduits où il est toujours possible de garder une excellente vue d'ensemble du projet ; et à ce titre j'ai particulièrement apprécié de pouvoir travailler sur un robot de A à Z, c'est à dire de son état de ROV jusqu'à son *aïvéesation* complète. Je n'ai manqué aucune partie du projet et je remercie mes collègues de m'avoir fait confiance en me donnant ces responsabilités.

# Table des figures

1	L'AUV <i>Vici</i> sur lequel j'ai travaillé pendant ce stage . . . . .	7
2	Le <i>Collins</i> , une réalisation personnelle de type ROV . . . . .	8
3	Les locaux de l'ENSTA . . . . .	9
4	Un ROV <i>Jack</i> avant les travaux pour le rendre autonome . . . . .	12
5	L'un des 6 propulseurs du CISCREA, en action . . . . .	13
6	Vue de dessus . . . . .	13
7	Visualisation des régulations en cap (à gauche) et en profondeur (à droite) . . . . .	15
8	La boîte lors de sa réception - tests d'étanchéité et de flottabilité . . . . .	15
9	Vue sur les connecteurs <i>Seacon</i> et <i>Subconn</i> . . . . .	16
10	La centrale inertielle . . . . .	16
11	Disposition finale des composants; c'est <i>embarqué!</i> . . . . .	17
12	Analyse de la piscine de l'ENSTA par sonar . . . . .	18
13	Un modem Tritech . . . . .	18
14	Les deux caméras étanches ajoutées au CISCREA . . . . .	19
15	<i>Vici</i> , physiquement prêt pour le concours . . . . .	20
16	Le programme permettant de rejouer des missions . . . . .	22
17	L'architecture logicielle du projet avec la base de données en position centrale . . . . .	23
18	Image Google et localisation du bassin du centre de recherche de l'OTAN . . . . .	26
19	Le bassin du NURC, tourné vers la Mer Méditerranée . . . . .	27
20	Le camion et la moitié de l'atelier robotique de l'ENSTA . . . . .	27
21	<i>Vici</i> (trace jaune, en haut) s'apprêtant à franchir la <i>gate</i> (points oranges, à droite) . . . . .	28
22	Bouée vue par <i>Vici</i> . . . . .	29
23	Visualisation par sonar du bassin du NURC . . . . .	29
24	Régressions linéaires sur les obstacles détectés . . . . .	30
25	Régulation progressive à l'approche d'un coin du bassin . . . . .	30
26	La régression provoque un virage pour passer au mur suivant . . . . .	30
27	L'AUV localisant la bouée qui symbolise l'anomalie . . . . .	31
28	Bouée vue par <i>Vici</i> en Italie (à gauche) et à l'ENSTA (à droite) . . . . .	31
29	Répartition des couleurs dans l'espace HSV [8] . . . . .	31
30	Détection de l'orientation du tube vu en piscine . . . . .	32
31	Détection de l'orientation et de la largeur du tube vu en mer . . . . .	32
32	La structure immergée, vue pour la première fois . . . . .	33
33	Localisation . . . . .	36
34	Représentation d'une correction de trajectoire . . . . .	37
35	Champs de vecteurs . . . . .	38
36	L'AUV faisant le tour de l'île, on ne voit que sa bouée et le plongeur . . . . .	39
37	L'antenne Wifi . . . . .	39
38	Trace GPS de l'AUV . . . . .	40
39	Organisation générale du projet - réflexions sur tableau blanc . . . . .	47
40	Le cadre de travail à proximité du bassin . . . . .	50
41	Explication du projet devant le jury anglais . . . . .	50
42	Au moment du passage de la <i>gate</i> . . . . .	51
43	L'équipe dont je faisais partie, juste avant une nouvelle mise à l'eau . . . . .	51
44	Les deux Mexicains qui m'ont accompagné pendant ce stage . . . . .	52
45	La robotique, c'est aussi le plaisir du terrain! . . . . .	52
46	Quelques instants avant une mission de suivi de digue . . . . .	53
47	Quand il s'agit de ramener un robot et un plongeur sur un kayak . . . . .	53
48	La vedette de l'ENSTA utilisée pour les missions autour de l'île . . . . .	54
49	Le plongeur qui n'était jamais loin de l'AUV pendant les missions en pleine mer . . . . .	54

# Références

- [1] D. LEBLANC. *SAUC-E, le concours Européen de robots sous-marins autonomes de l'OTAN*. 2011. URL : [http://shyrobotics.com/sauc-e-le-concours-europeen-de-robots-sous-marins-autonomes-sur-une-base-de-lotan-en-italie\\_20110524.html](http://shyrobotics.com/sauc-e-le-concours-europeen-de-robots-sous-marins-autonomes-sur-une-base-de-lotan-en-italie_20110524.html).
- [2] *Site officiel de l'ENSTA Bretagne*. 2013. URL : <http://www.ensta-bretagne.fr/>.
- [3] Mohamed Saad Ibn SEDDIK. *Rapport de stage sur le contrôle d'AUV*. 2012.
- [4] *Plateforme officielle de moos-ivp*. 2013. URL : <http://oceanai.mit.edu/moos-ivp/pmwiki/pmwiki.php?n=Site.Software>.
- [5] WIKIPÉDIA. *Page Wikipédia - Modbus*. 2013. URL : <http://fr.wikipedia.org/wiki/Modbus>.
- [6] *Site officiel du concours SAUC-E*. 2013. URL : <http://sauc-europe.org/about.html>.
- [7] *Structural Analysis and Shape Descriptors*. 2013. URL : [http://opencv.willowgarage.com/documentation/cpp/structural\\_analysis\\_and\\_shape\\_descriptors.html](http://opencv.willowgarage.com/documentation/cpp/structural_analysis_and_shape_descriptors.html).
- [8] *Page Wikipédia - Teinte Saturation Valeur*. 2013. URL : [http://fr.wikipedia.org/wiki/Teinte\\_Saturation\\_Valeur](http://fr.wikipedia.org/wiki/Teinte_Saturation_Valeur).
- [9] Luc JAULIN et Fabrice Le BARS. *A simple controller for line following of sailboats*. [http://www.ensta-bretagne.fr/jaulin/paper\\_jaulin\\_irsc12.pdf](http://www.ensta-bretagne.fr/jaulin/paper_jaulin_irsc12.pdf). 2013.

# Annexes

Annexe A : implémentation de l'algorithme de suivi de lignes .....	44
Annexe B : journal de bord tenu avant le départ pour l'Italie .....	46
Annexe C : photos complémentaires du stage .....	50

## Annexe A : implémentation de l'algorithme de suivi de lignes

```
1  /**
2   * \fn
3   * \brief Methode appelee automatiquement periodiquement
4   * Implementation du comportement de l'application
5   */
6
7  bool WayPoints::Iterate()
8  {
9      if(!m_waypoints.empty())
10     {
11         // Mise a jour des waypoints :
12
13         while(!m_waypoints.empty() && waypointDepasse())
14         {
15             m_waypoint_precedent = m_waypoints.front();
16             cout << "Waypoint depasse" << endl;
17             m_waypoints.pop_front();
18         }
19
20         if(m_waypoints.empty()) // Fin du parcours
21         {
22             m_Comms.Notify("VVV_Z_DESIRED", 0.);
23             m_Comms.Notify("VVV_VX_DESIRED", 0.);
24             cout << "Fin de la mission" << endl << endl;
25         }
26
27         else
28         {
29             // Suivi de ligne :
30
31             // 1.
32             Vecteur vAB = make_vecteur(m_waypoint_precedent, m_waypoints.front());
33             Vecteur vMB = make_vecteur(m_position, m_waypoints.front());
34             Vecteur vAM = make_vecteur(m_waypoint_precedent, m_position);
35             normaliserVecteur(vAB);
36             double e = determinant(vAB, vAM);
37             // si e > 0, robot a gauche de la ligne
38             // si e < 0, robot a droite de la ligne
39         }
40     }
41 }
```

```
40     // 2.
41     double phi = atan2(vAB.y, vAB.x);
42
43     // 3.
44     double r = 5.; // Precision GPS, pour rejoindre la ligne
45     double gamma_inf = M_PI/4.;
46     double theta = phi - (2.*gamma_inf/M_PI) * atan(e/r);
47
48     m_Comms.Notify("VVV_HEADING_DESIRED", 180. - MOOSRad2Deg(theta));
49     // theta = 0      --> direction Est
50     // theta = PI/2   --> direction Nord
51
52     cout << endl << m_waypoints.front().identifiant << endl;
53     cout << "Distance de la ligne : \t" << e << endl;
54     cout << "Distance du waypoint : \t" << sqrt(vMB.x*vMB.x + vMB.y*vMB.y) << endl <<
55         endl;
56     }
57 }
58 return(true);
59 }
60
61 /**
62  * \fn
63  * \brief Methode testant le depassement d'un waypoint
64  */
65
66 bool WayPoints::waypointDepasse()
67 {
68     Vecteur vBA = make_vecteur(m_waypoints.front(), m_waypoint_precedent);
69     normaliserVecteur(vBA);
70     Vecteur vBM = make_vecteur(m_waypoints.front(), m_position);
71     normaliserVecteur(vBM);
72
73     return produitScalaire(vBA, vBM) < 0;
74 }
```

## Annexe B : journal de bord tenu avant le départ pour l'Italie

### Mardi 28/05/13

Découverte de l'école, du laboratoire, des installations et des 4 ROV CISCREA. Début de prise en main du programme réalisé par Mohamed Saad IBN SEDDIK (projet nommé *Enstauw*) et qui permet de communiquer par Modbus avec le ROV.

### Mercredi 29/05/13

Découverte de MOOS, MOOS-IVP : des outils développés par des chercheurs du MIT et d'Oxford (Paul Newman). But : mettre en place une architecture pour tout le projet sur un CISCREA. Cette architecture permet de combiner tous les capteurs en les faisant communiquer à travers une mémoire partagée (MOOSDB). L'idée est de simplifier les calculs et la gestion du sous-marin en séparant toutes les fonctionnalités dans des *applications* communiquant via cette MOOSDB. L'avantage est de pouvoir greffer sur MOOS une application adaptée à la navigation en milieux marins (développée par Paul Newman) ce qui permet de jouer le rôle de contrôleur sur les propulseurs du CISCREA. Cette application est MOOS-IVP. Ce mercredi donc : lecture approfondie de MOOS, début de prise en main par suivi de tutoriels du MIT. Une partie de l'après-midi est passée avec Saad à essayer de faire fonctionner un exemple de MOOS (`xRelay`) sur ma machine : le problème viendrait d'Ubuntu 13.04. Il va falloir installer la version LTS 12.04 sur mon ordinateur.

### Jeudi 30/05/13

Après de nombreux déboires pour installer la 12.04 sur mon laptop (perte de pilotes graphiques), j'ai de nouveau une machine fonctionnelle. Nouveau travail sur la commande de l'AUV : pouvoir le faire évoluer selon quatre DOG<sup>29</sup> simultanément :  $V_x$ ,  $V_y$ ,  $V_z$  et  $R_z$ . Le matin : mise en place d'un code C++ pour permettre  $V_x$  et  $V_y$  en simultané (par règles de trigonométrie). L'après-midi : nouveau départ sur un système plus complexe mais plus général : sur base de matrices d'Euler. Étude d'un extrait du rapport de stage de Saad, implémentation C++ du système (résolution de systèmes de matrices linéaires, inversion de matrices (pseudo-inversions) et calculs de déterminants). Recherche de composants sur Conrad pour l'équipement du CISCREA. Des équations de déplacement ont été trouvées, il reste à les tester en bassin.

### Vendredi 31/05/13

Gros débriefing en fin de journée sur le concours et sur l'architecture de la boîte en aluminium. Celle-ci est plongée au fond du bassin pendant le week-end. Imagination d'un début de stratégie pour le concours.

### Week-end 01-02/06/13

Suite de la documentation (lecture) sur MOOS, plus précisément MOOS-IVP.

### Lundi 03/06/13

Récupération de la boîte en aluminium : aucune trace de fuite. Début de rassemblement des composants. Imagination d'une stratégie sur les AUV : *Veni Vidi Vici*, trois AUV en collaboration.

---

29. *Degree Of Freedom : degré de liberté*

**Mardi 04/06/13**

But de la journée : confirmer une idée de stratégie. Essais de sonars en bassin : expérimentation d'un sonar fixé sur un ROV au fond du bassin et tentant de localiser un autre ROV (*Vici*, chercheur). L'interprétation des données semble difficile. L'AUV est peu visible lorsqu'il est en surface. Attention à l'orientation du sonar. Essais avec le gros pipeline au fond de la piscine, il est invisible lorsqu'il est couché (dépend de l'inclinaison du sonar). Essais de communication entre les modems. Ceux-ci fonctionnent mais les messages transmis présentent quelques erreurs. Nécessité d'implémenter un code détecteur (et correcteur) d'erreurs. Les messages transmis font 4 octets (32 bits). Un protocole de communication doit être mis en place pour le concours.

**Mercredi 05/06/13**

Schémas globaux de l'architecture (voir ci-dessous). Récupérations des composants.

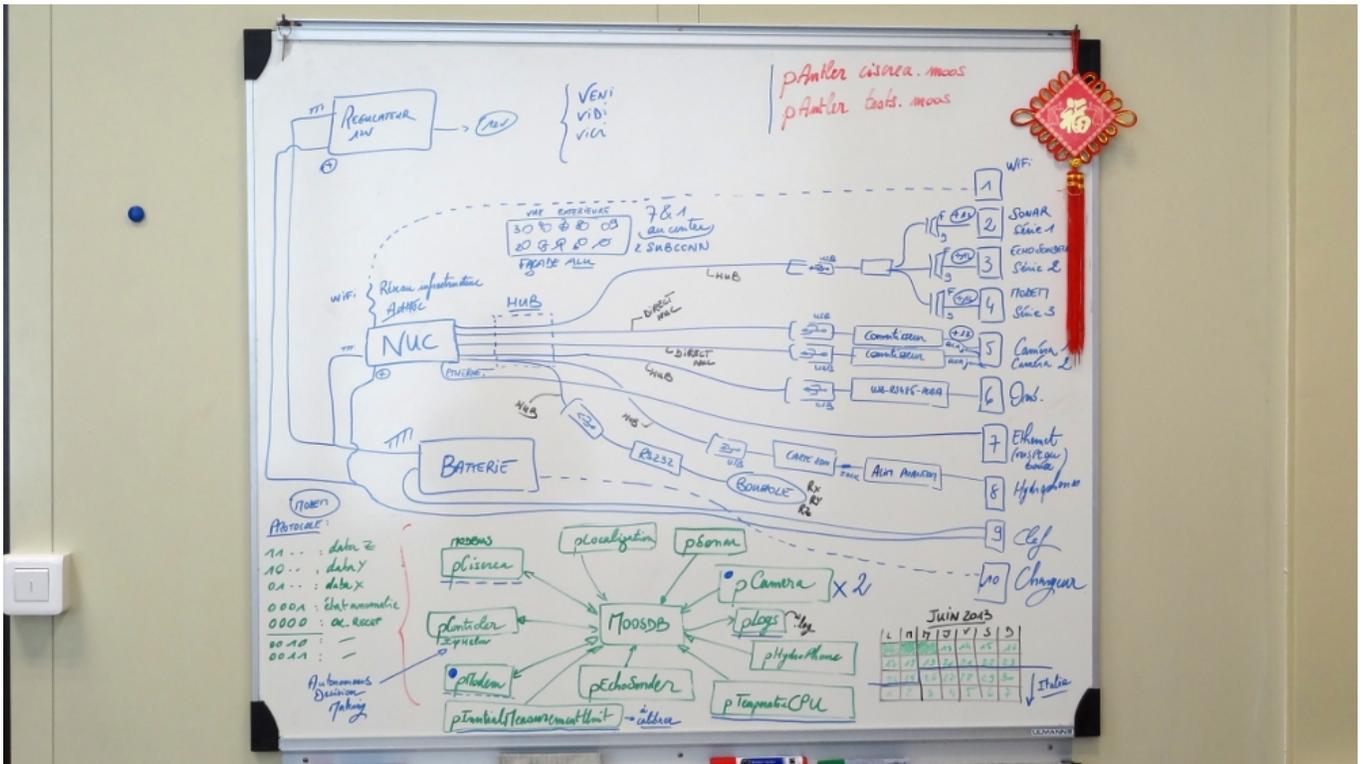


FIGURE 39 – Organisation générale du projet - réflexions sur tableau blanc

**Jeudi 06/06/13**

Essais de navigation en piscine : problèmes sur le pilotage de l'AUV : perturbations liées au câble ? Ou problèmes sur Modbus ? Nécessité de mettre en place des tests unitaires pour, au moins, valider la logique métier ne dépendant pas du hardware...

**Vendredi 07/06/13**

Implémentation de la base des tests. Début de l'implémentation des logs. Nouvelle implémentation des matrices pour le modèle physique (consignes).

**Lundi 10/06/13**

Suite du travail sur les logs. Première utilisation de Github. Soudure de la centrale inertielle, début

de son implémentation. Utilisation d'une librairie allemande (qui intègre un filtre de Kalman). Reste la calibration à faire. Implémentation de `pTemperatureCPU` (début et fin).

### **Mardi 11/06/13**

Suite du travail sur la centrale inertielle. Début de l'implémentation du protocole de communication : message codé sur 32 bits. Recherche de pilotes/drivers pour la caméra. Aucun résultat concluant pour le moment...

### **Mercredi 12/06/13**

Suite du travail sur le protocole de communication. Manque toujours la partie "correction de données" Une bonne partie de la journée a été passée sur la configuration des drivers des deux caméras (*v4l2, em28xx*). Irvin PROBST nous a aidé le matin pour faire fonctionner une caméra, mais le problème reste entier lorsqu'il s'agit de faire fonctionner deux caméras, ensemble... Problème de bande passante sur une port USB ? Difficulté du pilote à gérer deux périphériques ? Faut-il réduire la taille des images pour pouvoir utiliser les deux caméras ? Faut-il recompiler les drivers ? Tout a été testé mais pour le moment, aucun résultat... En parallèle, travail sur les communications par ports série : application `pModem`. Utilisation d'une librairie pour communiquer via ce port.

### **Jeudi 13/03/13**

Suite du travail sur les ports série. Finalisation de `pModem` avec redéfinition du protocole de communication.

### **Vendredi 14/03/13**

Réalisation de `pEchoSonder`. Rassemblement du matériel pour le week-end. Recherche sur les drivers des caméras. Essais Wifi en extérieur et intérieur (en piscine) avec la boîte en aluminium et un routeur. Récupération des images et vidéos de la précédente édition du concours.

### **Week-end 15-16/06/13**

Aménagement de la boîte en aluminium : soudures, collages, organisation des éléments. Réflexion sur l'aménagement des périphériques : un module à l'avant contenant tous les éléments (et amovible) ? À voir...

### **Lundi 17/06/13**

Essais sur `pSonar`. Difficulté de prise en main (protocole différent des autres sonars utilisés dans le labo). Reprise de code... intégrations des interfaces déjà implémentées. Travaux laissés en standbye.

### **Mardi 18/06/13**

Modélisations Solidworks. Présentation virtuelle des éléments devant la boîte en alu : capteurs caméras, echosondeur, etc. Développement d'un programme de reconnaissance d'orientation d'un pipeline : régression linéaire, seuillage, extraction de teintes de couleur.

### **Mercredi 19/06/13**

Complétion de quelques soudures (masse des caméras dans les fiches RCA). Installation de l'interface des hydrophones (soudures des prises jack).

### **Jeudi 20/06/13**

Réception des Mexicains à la gare de Brest. Accueil. Amélioration du programme de suivi de pipe-

line : détection des jonctions entre les éléments, implémentation de médianes au lieu de moyennes. Importantes corrections sur Modbus (délais entre les lectures/écritures de chaque registres).

### **Vendredi 21/06/13**

Comme je suis le plus apte à parler espagnol, je me charge de l'inscription des Mexicains à l'administration de l'ENSTA. Tests d'équilibrage (surprise : la boîte en aluminium flotte!), préparation avant les travaux du week-end, réflexion sur les positions des capteurs. Rédaction du rapport « *JournalPaper* » à envoyer au jury du concours. Manque la vidéo de présentation de l'équipe... Configuration réseau du NUC (doit se connecter automatiquement sur le réseau nommé **vedi**).

### **Week-end 22-23/06/13**

Installation des capteurs/câbles sur le châssis du CISCREA. Fixation du châssis à la boîte alu.

### **Lundi 24/06/13**

Problèmes de connexion au CISCREA par Wifi/Ethernet. Équilibrage. Installation de pipelines au fond du bassin.

### **Mardi 25/06/13 → Jeudi 27/06/13**

Ultimes préparatifs avant le concours. Amélioration des algorithmes des missions et de la régulation en cap et en profondeur sur *Vici*. Récupération du camion de location *Hertz*.

### **Vendredi 28/06/13**

Déménagement de la moitié de l'atelier robotique du laboratoire dans le camion, embarquement de l'AUV *Vici*, de deux autres ROV CISCREA et des deux AUV de Fabrice LE BARS. **Départ pour l'Italie vers 19h**. Les 1500km se feront d'une traite.

## Annexe C : photos complémentaires du stage

Quoi de mieux que quelques photos pour se donner une idée de ce stage. C'est l'objet de cette section.

### À La Spezia dans le centre de recherche de l'OTAN, du 29 juin au 6 juillet

Les organisateurs avaient disposé de nombreuses tentes sous lesquelles les équipes pouvaient travailler. Les outils indispensables étaient éparpillés dans de véritables ateliers de fortune, permettant les ultimes réparations et l'approfondissement des programmes déployés sur les robots.



FIGURE 40 – Le cadre de travail à proximité du bassin



FIGURE 41 – Explication du projet devant le jury anglais

Les qualifications et les épreuves qui suivirent se faisaient sous l'œil du jury, chargé d'évaluer l'évolution des AUV. L'un des juges prenait place dans un zodiac avec l'équipe en cours d'évaluation. À cause de sa forme ou de son déplacement, le zodiac perturbait parfois les robots en cours de mission.



FIGURE 42 – Au moment du passage de la *gate*



FIGURE 43 – L'équipe dont je faisais partie, juste avant une nouvelle mise à l'eau

Deux piscines d'eau salée avaient été installées à côté du grand bassin. Elles devaient permettre aux équipes de préparer l'équilibrage et le lestage des robots avant les réelles missions dans le bassin. Il y a en effet des différences de poids à compenser par rapport aux piscines d'eau douce dans lesquelles on avait l'habitude de les faire évoluer. Mais la chaleur entraînait l'évaporation des piscines et leur salinité changeait. Il n'était pas rare de devoir parfaire l'équilibrage à bord du zodiac. L'une de ces piscines est visible ci-dessous :



FIGURE 44 – Les deux Mexicains qui m'ont accompagné pendant ce stage



FIGURE 45 – La robotique, c'est aussi le plaisir du terrain !

**De retour à Brest mêm'**

Dès notre retour en Bretagne, il était question de mettre à nouveau les AUV dans l'eau. Nous prévoyons dans un premier temps de faire un suivi de mur à plus grande échelle qu'à La Spezia. Nous cherchions donc une structure longue et régulière, nous avons trouvé une digue près de l'ancienne base sous-marine allemande de Brest.



FIGURE 46 – Quelques instants avant une mission de suivi de digue



FIGURE 47 – Quand il s'agit de ramener un robot et un plongeur sur un kayak

Et lorsqu'il s'agit d'aller plus loin, le kayak n'est plus suffisant. Pour traverser la rade, nous avons utilisé un bateau de l'ENSTA.



FIGURE 48 – La vedette de l'ENSTA utilisée pour les missions autour de l'île

Enfin, un plongeur accompagnait le robot pendant les missions brestoises afin de s'assurer qu'il ne heurte pas des hauts fonds. La difficulté est de ne pas apparaître dans les données sonar enregistrées par l'AUV.



FIGURE 49 – Le plongeur qui n'était jamais loin de l'AUV pendant les missions en pleine mer

Placée au cœur de problématiques économiques, militaires, écologiques et archéologiques, la robotique sous-marine est un domaine en plein essor ; elle permet à l'Homme d'atteindre progressivement un environnement qui a toujours été inaccessible et qui, pourtant, abrite d'importantes richesses et lieux stratégiques. Les difficultés de communication dans cet environnement hostile ont poussé les chercheurs à concevoir des robots sous-marins complètement autonomes.

Ce rapport décrit un stage de spécialité réalisé en fin de quatrième année en Architecture des Systèmes d'Information. Il s'est en partie déroulé dans le laboratoire de l'ENSTA de Bretagne, avec une équipe travaillant sur des projets de robotique sous-marine. Le sujet de ce stage consistait à équiper un robot pour le rendre autonome en vue de participer au concours européen SAUC-E : un concours rassemblant plusieurs écoles autour de missions réalistes à effectuer en mer.

Placed at the heart of economic, military, ecological and archaeological issues, underwater robotics is an area rapidly expanding ; it allows Man to gradually achieve an environment that has been inaccessible and yet is home to significant wealth and strategic locations. Communication difficulties in this hostile environment have prompted researchers to develop underwater robots completely independent.

This report describes a technical internship made late in the fourth year in IT formation. It was partly conducted in the laboratory of ENSTA of Britain, with a team working on projects of underwater robotics. The topic of this period was to equip a robot in order to make it autonomous for the European competition SAUC-E : a challenge bringing together several schools around realistic missions carried out in sea.

