

SAUC'ISSE, un robot sous-marin autonome

Fabrice Le Bars, Jan Sliwka, Luc Jaulin

DTN (Département des Technologies Nouvelles)
ENSIETA (Ecole Nationale Supérieure des Ingénieurs des Etudes et Techniques d'Armement)
2 rue François Verny, 29806 Brest Cedex 9 - France

lebarsfa@ensieta.fr, sliwkaja@ensieta.fr, jaulinlu@ensieta.fr

Résumé – Depuis 2007, l'ENSIETA (Ecole Nationale Supérieure des Ingénieurs des Etudes et Techniques d'Armement) construit des robots sous-marins autonomes. Ces robots participent tous les ans au concours SAUC-E (Student Autonomous Underwater Challenge - Europe) et permettent de tester et valider des algorithmes de régulation, détection, localisation, estimation d'état et de paramètres ou encore SLAM (Simultaneous Localization And Mapping) utilisant le calcul par intervalles. Le robot que nous allons présenter est SAUC'ISSE, notre premier AUV (Autonomous Underwater Vehicle). Un deuxième robot aux caractéristiques similaires, nommé SARDINE, a aussi été construit pour étudier la faisabilité d'une collaboration entre robots sous-marins.

Mots clés – robotique, sous-marin, AUV, localisation, intervalles, SAUC-E.

1. Introduction

Le robot sous-marin autonome SAUC'ISSE a été construit à l'origine par des étudiants pour participer au concours SAUC-E 2007. Les buts de cette compétition sont de développer la robotique sous-marine autonome en permettant à des équipes multidisciplinaires d'étudiants, ingénieurs et industriels de travailler ensemble. La compétition se passe en général dans une piscine (ou une marina) où les robots sous-marins doivent effectuer des missions autonomes de déplacement, détection, identification et localisation telles que passer à travers des cadres, suivre une boule rouge, suivre un coin du bord de la piscine, rentrer dans une boîte ou encore faire surface au-dessus d'une balise acoustique... Ces missions, déjà difficiles sur terre sont souvent encore plus complexes dans l'eau, notamment à cause de problèmes de visibilité, étanchéité ou communication. D'abord, nous détaillerons l'architecture physique (mécanique et électronique) du robot. Ensuite, nous expliquerons les algorithmes qui y sont embarqués, notamment pour la localisation et le contrôle du robot.

2. Description de l'AUV SAUC'ISSE

2.1 Aspect extérieur du sous-marin

La partie principale de SAUC'ISSE est un tube en aluminium de 18 cm de diamètre et 70 cm de long. Ces dimensions sont un bon compromis pour à la fois faciliter le transport du robot et avoir suffisamment d'espace pour contenir tous les composants internes nécessaires. L'aluminium a été choisi pour son amagnétisme (ceci est important si on veut utiliser une boussole magnétique) et sa résistance à la corrosion. Le tube est rendu étanche en plaçant 2 tapes (sorte de couvercles) munies d'un joint torique et maintenues par des vis de part et d'autre du tube. Des vis d'extraction sont utilisées pour faciliter l'ouverture du tube. Pour communiquer avec les capteurs et actionneurs externes du sous-marins, des connecteurs étanches (de marque Switchcraft et Bulgin) sont placés sur la tape arrière. La tape avant a été laissée sans connecteurs pour faciliter son ouverture et ainsi pouvoir changer rapidement les batteries du

sous-marin. Une structure spéciale fixée au tube a aussi été faite pour servir de support aux propulseurs.

Les angles de roulis et tangage ne sont pas contrôlés et restent stables grâce à une quille lestée qui sert de support pour le sonar et le propulseur vertical. Ainsi, le centre de volume du sous-marin est au-dessus du centre de gravité. De plus, des petits lests troués (en plomb) peuvent être ajoutés sur des tiges aux quatre coins du sous-marin pour atteindre la limite de flottaison et affiner l'équilibrage. De cette manière, le sous-marin n'a besoin que d'une force très faible pour descendre en profondeur et lorsque le propulseur vertical est arrêté, le sous-marin remonte tout seul à la surface.

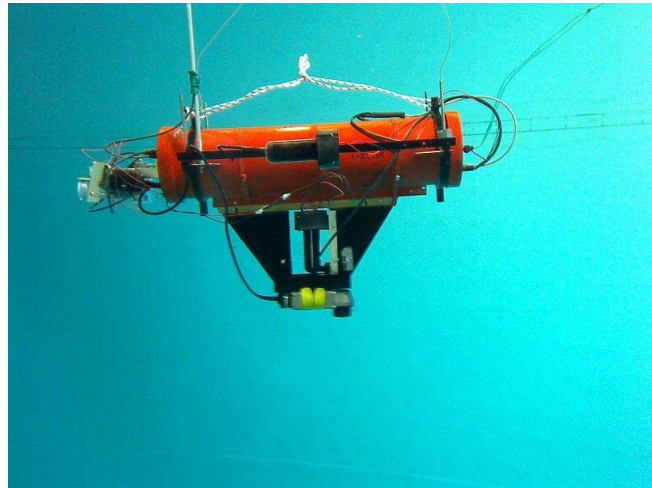


Figure 1 : SAUC'ISSE dans l'eau

2.2 Organisation intérieure

Des rails collés dans le tube en aluminium permettent de tirer une plaque en Plexiglas qui sert de support à l'ensemble des éléments internes de SAUC'ISSE. Un tiroir attaché sous cette plaque permet d'accéder aux batteries sans avoir à déplacer les autres éléments.

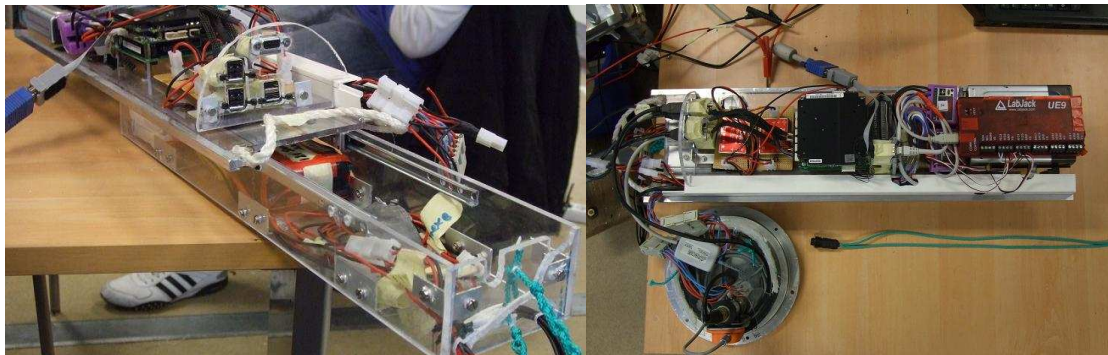


Figure 2 : Tiroir et éléments internes de SAUC'ISSE

2.3 Electronique

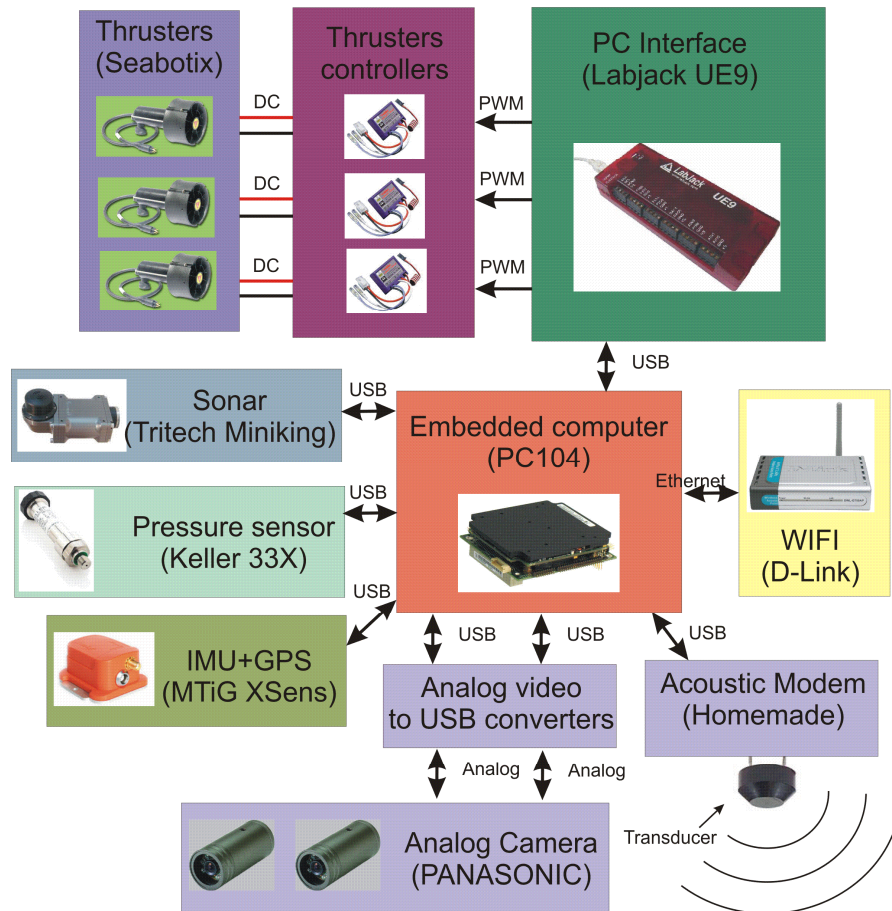


Figure 3 : Electronique de SAUC'ISSE

3 propulseurs (Seabotix BTD150) sont utilisés pour faire bouger le robot :

- 1 propulseur vertical sert à régler la profondeur du sous-marin.
- 2 propulseurs horizontaux permettent de contrôler sa vitesse et sa direction.

Pour contrôler ces propulseurs, nous utilisons des variateurs (Robbe Rokraft) qui font varier la puissance envoyée aux moteurs selon la forme du signal PWM (Pulse Width Modulation) qu'ils reçoivent. Pour générer ces signaux à partir de programmes exécutés sur l'ordinateur, nous utilisons un module d'interface (Labjack UE9) connecté au PC embarqué par USB.

L'ordinateur embarqué dans SAUC'ISSE est un PC/104 (Eurotech CPU-1482) équipé d'un processeur Pentium M à 1.4 GHz et 512 MB de RAM. Le système d'exploitation et les programmes sont stockés sur un disque dur 2.5 pouces de 320 Go. 8 ports USB, 1 Ethernet, 2 RS232 et 1 VGA nous permettent de connecter facilement n'importe quel périphérique d'ordinateur classique (écran, souris, clavier...). Ce PC peut être alimenté directement à partir de batteries 24 V grâce à son étage d'alimentation, qui fournit en plus des tensions régulées de +3.3, +5, +12 et -12 V.

2 webcams étanches analogiques (une orientée vers l'avant et une vers le bas) sont connectées à l'ordinateur embarqué via des convertisseurs audio-vidéo vers USB.

Pour obtenir la profondeur du sous-marin, nous utilisons un capteur de pression (Keller PAA33X) connecté au PC avec un convertisseur RS485 – USB. Ce capteur est fixé sur la tige arrière du sous-marin.

Une centrale inertielle (MTi Xsens) prêtée par le GESMA (Groupe d'Etudes Sous-Marines de l'Atlantique) nous permet d'obtenir l'orientation du robot. Elle est équipée d'un filtre qui fusionne les données magnétiques et inertielles. Elle se connecte au PC embarqué via un convertisseur RS232 – USB.

Le principal capteur servant à obtenir la position du sous-marin dans la piscine est le sonar (Tritech MiniKing). Celui-ci est aussi connecté à l'ordinateur via un convertisseur RS232 – USB.

Un point d'accès Wifi (D-Link DWL G700AP) ainsi qu'une antenne externe de 1 m nous permettent de communiquer avec le robot lorsqu'il est proche de la surface de l'eau. S'il doit aller plus profondément, l'antenne est mise sur un flotteur et un déport d'antenne de 5 m la relie au sous-marin. La Connexion au Bureau à Distance (commande mstsc) de Windows (le PC/104 embarqué fonctionne sous Windows XP) permet à n'importe quel ordinateur portable ou même téléphone mobile équipé de Windows de lancer des programmes sur le PC embarqué (ceci peut aussi être fait avec VNC sous Linux par exemple).

L'alimentation de SAUC'ISSE est divisée en 2 parties :

- Les propulseurs sont alimentés par une batterie de 12 V.
- Le PC/104, le point d'accès Wifi (via le 5 V fourni par l'étage d'alimentation du PC/104), les caméras analogiques (via le 12 V fourni par le même étage d'alimentation) et le sonar sont alimentés par une batterie de 24 V.

Tous les autres périphériques (capteur de pression, centrale inertielle, Labjack) sont alimentés par les ports USB de l'ordinateur embarqué.

3. Localisation et contrôle

3.1 Principe

Des algorithmes de régulation en profondeur, cap, vitesse et distance classiques associés à d'autres algorithmes de détection, localisation et contrôle plus innovants forment l'intelligence du robot.

L'une des principales difficultés pour le robot est de pouvoir se localiser par rapport aux bords de la piscine où il évolue. La plupart des solutions existantes pour résoudre le problème de la localisation sont basées sur des techniques probabilistes (filtre de Kalman, estimation Bayésienne, filtres particulaires, voir [1]). Dans notre cas, un observateur d'état utilisant le calcul par intervalles [2] et prenant en compte des données aberrantes venant du sonar rotatif du sous-marin lui permet d'estimer sa position. De plus, une méthode originale supplémentaire mettant en jeu un contracteur sur l'image (voir [3] pour plus d'informations sur les contracteurs et [4] pour le contracteur sur l'image) est actuellement en cours d'élaboration. Un algorithme de SLAM utilisant la propagation de contraintes pourrait aussi être testé sur le sous-marin (voir [5] et [6] pour son application offline sur des données des sous-marins Redermor et Daurade du GESMA).

Le robot peut :

- Etre téléopéré de différentes façons en utilisant les algorithmes de régulation et contrôle voulus : pas de régulation, régulation en profondeur et régulation en cap et vitesse dans le plan...
- Effectuer des enchaînements de missions prédéfinies sous forme de scripts (fichiers texte avec quelques mots-clés prenant des paramètres) : successions et combinaisons

entre régulations en profondeur, cap, vitesse ou distance dont les consignes changent au cours du temps ou appel à des fonctions particulières telles que l'activation du suivi de waypoints, mur ou boule rouge...

Les équations d'état du robot sous-marin peuvent être décrites de cette façon :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = u_2 - u_1 \\ \dot{v} = u_1 + u_2 - v \end{cases} \quad (1)$$

où x, y sont les coordonnées du robot, θ son orientation et v sa vitesse. Les entrées u_1 et u_2 sont les accélérations fournies par les propulseurs droite et gauche. Ce modèle normalisé correspond à un robot sous-marin à une profondeur constante (la régulation en profondeur du robot est considérée comme déjà résolue et ne sera pas considérée ici) sans roulis ni tangage. De cette manière, notre sous-marin peut être considéré comme un robot évoluant en 2 dimensions.

Pour prendre en compte la présence de données aberrantes, nous considérons qu'une pose du robot est valide si elle est consistante avec toutes les mesures faites aux temps $k \dots k-l$ sauf q d'entre elles (voir q-intersection dans [7] et [8]).

3.2 Observateur utilisé pour la localisation

Le système peut être discrétisé comme suit :

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) \quad (2)$$

où δ est le pas de temps, $\mathbf{x} = (x, y, \theta, v)$ est le vecteur d'état et

$$\mathbf{f}_k \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x_1 + \delta \cdot x_4 \cdot \cos x_3 \\ x_2 + \delta \cdot x_4 \cdot \sin x_3 \\ x_3 + \delta \cdot u_2(k) - \delta \cdot u_1(k) \\ x_4 + \delta \cdot u_1(k) + \delta \cdot u_2(k) - \delta \cdot x_4 \end{pmatrix} \quad (3)$$

On suppose que le robot se déplace dans une piscine de forme connue. Son sonar permet de mesurer la distance horizontale entre le robot et le bord de la piscine selon la direction pointée par le sonar (cette direction change au cours du temps car c'est un sonar rotatif). Si la piscine est composée de bords verticaux, l'équation d'observation du système est

$$d = g_k(x) \quad (4)$$

où g_k est une fonction d'observation donnée par un algorithme simple de calcul de distance à des segments.

Le principe de notre observateur est de calculer l'ensemble des états possibles $\mathbb{X}(k)$. Le centre $\hat{\mathbf{x}}(k)$ de $\mathbb{X}(k)$ est retourné par l'observateur en tant qu'estimation du vecteur d'état

actuel du robot. Ce sera cette estimation qui sera utilisée par le contrôleur pour calculer le vecteur de contrôle $\mathbf{u}(k)$.

3.3 Contrôleur utilisé pour le suivi de waypoints

Le principe de ce contrôleur est décrit dans la figure suivante :

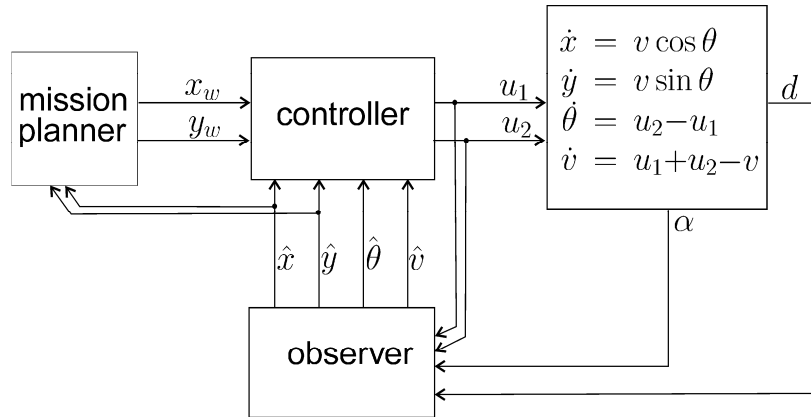


Figure 4 : Principe du contrôle du robot sous-marin

D'abord, le planificateur de mission envoie au contrôleur un waypoint (x_w, y_w) à atteindre. Lorsque le waypoint courant est atteint avec une précision prédéfinie (i.e. $(\hat{x} - x_w)^2 + (\hat{y} - y_w)^2 \leq \varepsilon$), le planificateur passe au waypoint suivant. Le contrôleur choisi est donné par l'expression suivante :

$$\mathbf{u} = \begin{pmatrix} 1 - \omega \\ 1 + \omega \end{pmatrix}, \text{ où } \omega = \text{sign} \left(\det \begin{pmatrix} \cos \hat{\theta} & x_w - \hat{x} \\ \sin \hat{\theta} & y_w - \hat{y} \end{pmatrix} \right) \quad (5)$$

Le principal avantage de ce contrôleur est sa simplicité. La direction que le robot doit suivre est donnée par le vecteur $\mathbf{e} = (x_w - \hat{x}, y_w - \hat{y})^T$. L'orientation estimée du robot est donnée par le vecteur $\mathbf{v} = (\cos \hat{\theta}, \sin \hat{\theta})^T$. Si \mathbf{v} est à la droite de \mathbf{e} (i.e. $\det(\mathbf{v}, \mathbf{e}) < 0$), le robot tourne à droite ($\omega = 1$), autrement il tourne à gauche ($\omega = -1$).

4. Conclusion

Dans cet article, nous avons décrit un AUV utilisant à la fois des techniques classiques de l'automatique et des algorithmes plus innovants. Cette plateforme nous permet à la fois d'initier des étudiants à la robotique sous-marine et de tester en conditions réelles les algorithmes développés dans notre laboratoire.

5. Bibliographie

- [1] Thrun S., Burgard W., Fox D. (2005), *Probabilistic Robotics*, MIT Press, Cambridge, M.A., United Kingdom.
- [2] Jaulin L., Kieffer M., Didrit O., Walter E. (2001), *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Edition Springer-Verlag, London, United Kingdom.

- [3] Chabert G., Jaulin L., “*QUIMPER, A Language for Quick Interval Modelling and Programming in a Bounded-Error Context*”, Artificial Intelligence, 173:1079-1100, 2009.
- [4] Sliwka J., Le Bars F., Reynet O., Jaulin L., “*Using interval methods in the context of robust localization of underwater robots*”, submitted to NAFIPS, El Paso, 2011.
- [5] Jaulin L., “*A Nonlinear Set-membership Approach for the Localization and Map Building of an Underwater Robot using Interval Constraint Propagation*”, IEEE Transaction on Robotics, Vol 45, 25(1):88-98, 2009.
- [6] Le Bars F., Bertholom A., Sliwka J., Jaulin L., “*Interval SLAM for underwater robots; a new experiment*”, NOLCOS, Bologna, 2010.
- [7] Jaulin L., “*Robust set membership state estimation; Application to Underwater Robotics*”, Automatica, Vol 45, Issue 1, pp. 202-206, 2009.
- [8] Sliwka J., Le Bars F., Jaulin L., “*Calcul ensembliste pour la localisation et la cartographie robustes*”, JD-JN-MACS, Angers, 2009.