

# Localisation absolue et relative d'un drone sous-marin

Paul-Antoine GRAU

Profil Robotique

CI 2019



**Tuteur entreprise :** Hervé RICHER DE FORGES

**Tuteur école :** Luc JAULIN

**Résumé du stage de fin d'études**

30 août 2019

## Introduction

Kopadia est une jeune entreprise, créée en février 2017. Elle est en train d’arriver sur le marché de l’inspection robotique sous-marine. Pour se faire, Kopadia a, entre autre, besoin :

- de pouvoir suivre un drone depuis la surface
- que le drone puisse se localiser dans l’eau

Ce sont les deux objectifs principaux de mon stage.

Un autre objectif a plus long terme est de développer une solution permettant au drone de se localiser dans des grands fonds pour pouvoir y naviguer. Pour cela, 3 élèves et moi même avons au cours de mes 6 mois de formation à l’ENSTA Bretagne mis en place un suivi d’isobathe.

Il y a 8 personnes qui travaillent pour Kopadia. Thierry en tant que président, Hervé comme Directeur technique, Agnès qui est office manageuse, François qui est responsable du développement informatique, Adrien en tant que responsable développement électronique, Gaëtan comme responsable développement mécanique, Oriane en contrat de professionnalisation en formation à l’ISEN Brest.

Kopadia a un objectif, devenir leader dans le domaine de l’inspection sous-marine. Pour cela, les fondateurs veulent que Kopadia soit innovant et impliqué dans le développement durable.

## Analyse de l’existant

### Suivi d’un AUV

Pour suivre un drone sous-marin depuis la surface, j’ai cherché des solutions de trilatération et j’ai trouvé ce dépôt GitHub [Wie19] qui correspond exactement à ce que je cherchais à faire. L’algorithme utilisé est celui de LEVENBERG-MARQUARDT. Je vais vous le présenter succinctement en m’appuyant sur ces deux articles : [Gav11] et [Ran04].

Cet algorithme a pour but d’améliorer la méthode de descente de gradient et la méthode GAUSS-NEWTON. Pour chacune de ces 3 méthodes, il minimise le critère d’erreur appelé *khi-carré* qui mesure l’adéquation de l’estimation aux données. Pour cela, il reprend les deux méthodes pour trouver la valeur de perturbation qu’on appellera  $h$  du paramètre  $p$  qui réduit *khi-carré*.

La méthode de LEVENBERG-MARQUARDT utilise un mélange des méthodes de descente de gradient et de GAUSS-NEWTON et bénéficie ainsi des avantages de chacune en minimisant leur désavantage :

$$(J^T W J + \lambda I) h_{LM} = J^T W (y - \hat{y}) \quad (1)$$

Avec cette méthode, plus  $\lambda$  est petit plus on se rapproche de la méthode de GAUSS-NEWTON et plus  $\lambda$  est grand et plus on effectue une descente de gradient.

### Localisation d’un AUV

Pour qu’un AUV puisse naviguer sous l’eau, il a besoin de connaître sa position et donc de se localiser. Cette localisation permet aussi de cartographier l’environnement dans lequel le drone évolue. On peut s’autoriser à repositionner en post traitement les données recueillies par le Folaga mais plus le travail sera bon sous l’eau meilleur le résultat sera.

Le "Simultaneous Localization and Mapping" – SLAM – semble être performant dans la localisation. Un autre moyen serait uniquement l'intégration d'un filtre observateur comme un filtre de Kalman pour introduire la cinématique du drone, ce qui peut aussi fonctionner correctement. Cependant, on fait l'hypothèse que les balises ne bougent pas pendant toute la mission. Avec une technique de SLAM on pourrait imaginer des balises qui se déplacent.

Le SLAM est une méthode qui permet, comme son nom l'indique de localiser le robot et de cartographier son environnement. Il existe deux principales méthodes : le "online" SLAM qui estime uniquement la position actuelle de l'AUV et le "full" SLAM qui recalculé l'intégralité de la trajectoire du drone depuis le début de sa mission.[PSSL14]

Après l'analyse de l'existant, j'ai organisé une réunion avec l'équipe pour présenter ce que j'avais trouvé et pour choisir la ou les méthodes de localisation. A ce moment, nous avons donc décidé d'implémenter 2 versions différentes de localisation pour en choisir une selon notre retour d'expérience depuis le terrain, un Kalman et un filtre de KALMAN étendu pour un SLAM.

Ayant effectué mon choix en début de stage, une possible mission en mer Méditerranée est venue changer ce qui était prévu. Cela fait bien entendu partie de la vie d'une jeune entreprise qui cherche de l'argent pour se développer et assurer sa survie. J'ai donc opté pour un développement plus sûr et rapide en choisissant d'implémenter uniquement un filtre de Kalman que je connaissais bien mieux. Cependant, les autres méthodes seront implémentées à la suite de ce stage, au second semestre 2019.

## Suivi de Mission

Dans cette partie, on cherche à savoir depuis la surface où le drone se situe sous l'eau. J'explique donc ici comment j'ai intégré le suivi d'un drone sur l'IHM que possède Kopadia.

## Communication

Pour pouvoir suivre le drone, Kopadia possède 4 balises flottantes – qui ont été développées pendant mon stage – équipées d'une tête acoustique et d'une antenne permettant une communication suivant le protocole LoRa. Les drones possèdent eux aussi une tête acoustique. Pour suivre la mission, nous avons besoin d'un ordinateur sur un bateau. Les horloges des balises et des drones sont synchronisées et une table de parole est alors établie. Une fois en mission, quand vient au tour de l'AUV d'envoyer un *ping*, chaque balise reçoit ce *ping* avec un retard. Un temps de vol est calculé et on remonte à une distance en estimant la vitesse moyenne de l'onde acoustique. Chaque balise envoie, via la communication LoRa, sa position et sa distance au drone. C'est l'IHM qui reçoit la donnée et qui s'occupe d'effectuer la multilatération pour positionner le drone sur une carte.

## Suivi de l'AUV

Mathématiquement, on se retrouve avec ce système d'équations que l'on doit résoudre pour obtenir la position du drone définie par les coordonnées  $x$ ,  $y$  et  $z$  :

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = d_{AUV \rightarrow balise_1}^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = d_{AUV \rightarrow balise_2}^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = d_{AUV \rightarrow balise_3}^2 \\ (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 = d_{AUV \rightarrow balise_4}^2 \end{cases} \quad (2)$$

En utilisant l’algorithme des moindres carrés de LEVENBERG-MARQUARDT, on aboutit à une position d’une précision satisfaisante.

## Localisation d’un AUV

Pour naviguer sous l’eau, un drone a besoin de connaître sa position. Pour réaliser cela nous utilisons 4 balises flottantes équipées chacune d’une tête acoustique pour communiquer avec le drone.

### Principe de localisation

On mesure le temps de vol de l’onde acoustique dans l’eau et on remonte ainsi à la position du drone.

En utilisant un filtre de Kalman, nous alternons entre navigation à l’estime – entre chaque *ping* de balise – et correction à l’aide de la mesure de la distance à une balise dont la position est connue.

On est obligé de linéariser l’équation qui lie la distance du drone à la balise et la position du drone et de la balise pour pouvoir utiliser le filtre de KALMAN.

On obtient donc une équation de la forme :

$$\tilde{Y} = \frac{dg}{dX}(\hat{X})X = Y - g(\hat{X}) + \frac{dg}{dX}(\hat{X})\hat{X} \quad (3)$$

avec  $g(X)$  qui est la fonction de mesure et qui correspond donc à l’équation  $Y = g(X)$ .

## Suivi d’isobathe

Dans ce chapitre, nous allons revenir sur ce que Maria-Luiza, Antony, Louis et moi-même avons développé dans le cadre du projet Guerlédan durant les 6 derniers mois à l’ENSTA Bretagne. Cette stratégie de navigation est très intéressante puisqu’elle ne nécessite que très peu de capteurs et que ses applications sont nombreuses. L’AUV se « localise » relativement à une ligne d’isobathe qu’il tente de positionner et vers lequel il s’oriente. Une fois arrivé en stage, j’ai repris ce mode de navigation et je l’ai intégré à l’architecture de Kopadia. J’ai dû développer une nouvelle application pour que ce mode de navigation soit exécutable depuis l’interface graphique.

En fin de projet à l’école, le suivi d’isobathe était fonctionnel. Je vais détailler ici ce que j’ai essayé d’améliorer pendant ma période de stage ainsi que la modélisation du système. Cette partie synthétise une partie de l’article [Jau18] et ajoute quelques modifications pour adapter l’article aux Folaga, en ayant pour but d’améliorer ce qui a été fait à l’école pour le projet Guerlédan.

## Modélisation du système

On définit les équations d'état du drone par

$$\dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ u_1 \\ u_2 \end{pmatrix} \quad (4)$$

avec  $x$ ,  $y$  et  $z$  la localisation du drone dans l'espace et  $\phi$  son cap.  $u_1$  et  $u_2$  sont les commandes en profondeur et en cap du drone.

On introduit aussi les variables qui vont nous être utiles pour la régulation

$$\begin{cases} y_1 &= z - h(x, y) \\ y_2 &= \text{angle}(\nabla h(x, y)) - \phi \\ y_3 &= -z \end{cases} \quad (5)$$

où  $h$  correspond à la hauteur d'eau à la position  $(x, y)$ .

## La commande finale

La commande finale du robot est la somme d'une commande qui oriente le drone vers l'isobathe souhaitée  $-h_0$  et l'orientation de l'isobathe au dessus de laquelle il se trouve. On obtient alors

$$u = \tanh(y_3 + y_1 + h_0) + \text{sawtooth}(y_2 \pm \frac{\pi}{2}) \quad (6)$$

## Améliorations

Etant donné que l'on utilise pas la boussole de l'article [Jau18], je l'ai intégrée dans le filtre de KALMAN qui estime la pente du fond et qui permet donc de calculer  $y_2$ . On obtient donc l'équation d'évolution suivante :

$$p(k+1) = \begin{pmatrix} 1 & \phi_k - \phi_{k-1} & 0 \\ -\phi_k + \phi_{k-1} & 1 & 0 \\ dt & 0 & 1 \end{pmatrix} p(k) + \alpha(k) \quad (7)$$

J'ai aussi rendu possible le paramétrage de la loi de commande, ce qui devrait nous permettre de nous adapter plus facilement à un fond très pentu ou au contraire plus plat. Voici l'équation de la commande correspondante :

$$u = K_1 \times \tanh(y_3 + y_1 + h_0) + K_2 \times \frac{\text{sawtooth}(y_2 \pm \frac{\pi}{2})}{\pi} \quad (8)$$

# Conclusion

Lors de ce stage, j'ai pu me confronter une fois de plus à la vie d'une jeune entreprise. Je connaissais déjà Kopadia, mais l'organisation entre mon stage d'assistant ingénieur et mon contrat de professionnalisation était bien différente. Beaucoup de changements ont été effectués pendant ma période de formation à l'école et il m'a été bien plus difficile de m'intégrer dans cette nouvelle architecture que ce à quoi je m'attendais. Peu d'applications que j'avais développées l'année dernière étaient restées, mais les briques technologiques que j'avais utilisées étaient, elles, restées.

Pour la partie technique de mon stage, j'ai développé : le suivi d'une mission en surface d'un drone, la localisation du drone pour qu'il puisse dans le futur naviguer en position et référencer les données qu'il aura récolté lors de sa mission ; quelques améliorations du suivi d'isobathe pour utiliser pleinement des capteurs présents sur le Folaga.

Jusqu'à maintenant, je n'avais pas conscience de la rapidité à laquelle 6 mois en entreprise passent vite. Je pensais avoir plus de temps pour implémenter et tester toutes les solutions que l'on avait retenues au début du stage. Finalement, je n'ai eu que des résultats de simulation. Cependant, les résultats en lac et/ou mer sont prévus pour courant septembre et la validation pour octobre. J'ai pu assister à un certain nombre de réunions de mises en place de processus de gestion d'entreprise, développement, ... répondant aux critères de la norme ISO 9001, ce qui je pense est une chance pour un élève-ingénieur en stage.

Fin juin, j'ai pu participer à une journée du groupe de travail robotique marine et sous-marine grâce à M.JAULIN, et d'y présenter le suivi d'isobathe. Cela m'a permis de m'entraîner une fois de plus à présenter et qui plus est, devant à une grande majorité de chercheurs dans le domaine de la robotique sous-marine.

J'ai la chance de continuer l'aventure avec Kopadia, une jeune entreprise innovante et dynamique qui possède une équipe avec laquelle je me suis tout de suite bien entendu, par son envie d'être innovante dans tous les domaines de vie d'une entreprise. Mon rôle au sein de Kopadia pour le prochain semestre 2019 est de continuer la localisation puis d'implémenter des stratégies de navigation à partir de la position.

# Bibliographie

- [Gav11] Henri Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, pages 1–15, 2011.
- [Jau18] Luc Jaulin. Isobath following using an altimeter as a unique exteroceptive sensor. In *International Robotic Sailing Conference 2018 (IRSC 2018)*, Southampton, United Kingdom, Aug. 2018.
- [PSSL14] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. Auv navigation and localization : A review. *IEEE Journal of Oceanic Engineering*, 39(1) :131–149, 2014.
- [Ran04] Ananth Ranganathan. The Levenberg-Marquardt algorithm. *Tutorial on LM algorithm*, 11(1) :101–110, 2004.
- [Wie19] Scott Wiedemann. Solves a formulation of n-D space trilateration problem using a nonlinear least squares optimizer : lemmingapex/trilateration, July 2019.