

Navigation sous-marine par cycles stables

Amine CHAABOUNI

June 28, 2021

Contents

1	Introduction	2
1.1	Contexte	2
1.2	Présentation de l'AUV	3
1.3	Problématique	4
2	Outils mathématiques utilisés	5
2.1	Modélisation d'un robot	5
2.1.1	Systèmes dynamiques hybrides	6
2.2	Calculs par intervalles	9
2.2.1	Introduction à l'arithmétique des intervalles	9
2.2.2	Vecteurs et matrices d'intervalles	9
2.2.3	Fonctions d'intervalles	10
2.2.4	La forme centrée	11
2.3	Cycles stables	12
2.3.1	Cycle limite	12
2.3.2	Systèmes dynamiques continus et surfaces de Poincaré	13
2.3.3	Étude des cycles stables grâce aux surfaces de Poincaré	14
2.3.4	Stabilité d'un point fixe	14
2.3.5	Cycle limite hybride	15
3	Le tour de l'île	16
4	Preuve de stabilité	19
4.1	Contracteur de stabilité	19
4.2	Exemple de preuve pour un système hybride	20
4.2.1	Formalisation de l'exemple	21
4.2.2	Présentation des calculs formels	22
4.2.3	Implémentation de l'exemple	23
4.3	Stabilité du tour de l'île	23
5	Résultats pratiques	27
6	Vers un tour plus réaliste	29
6.1	Carte d'intervalles	29
6.2	Travail futur	31
A	Équations mécaniques	34
A.1	Équations cinématiques	34
A.2	Équations dynamiques	35
A.2.1	Accélération	36
A.2.2	Rotation	36
A.3	Contrôle	37

1 Introduction

1.1 Contexte

La navigation sous-marine, malgré les nombreux travaux réalisés dans ce domaine, présente encore plusieurs mystères et difficultés. En effet, le monde sous-marin n'a pas été autant exploré que son correspondant terrestre.

La navigation sous-marine est confrontée à une difficulté de localisation à cause de l'impossibilité d'utiliser des capteurs type GPS. D'autre part, les équations physiques sont plus complexes car il faut prendre en compte la pression, la corrosion, la température, les forces exercées par l'eau, etc. Par ailleurs, la vision et l'observation du comportement d'un sous-marin n'étant pas souvent possible, il est nécessaire de prendre toutes les précautions avant de lancer un test en mer, sous risque de perdre le sous-marin.

Dans le cadre de ce stage, j'ai intégré l'équipe du ROBEX à l'ENSTA Bretagne dans le but de faire naviguer un robot sous-marin et effectuer le tour d'une île: l'île des Morts, figure 1. Sous la tutelle de Luc JAULIN et en contact constant avec Morgan LOUEDEC, Nathan FOURNIOL, Robin SANCHEZ, Guillaume SAMAIN et toute l'équipe technique du laboratoire, j'ai manipulé le robot *Riptide*, un véhicule sous-marin autonome (VSA ou AUV en anglais).

Dans un premier temps, il était nécessaire d'étudier le comportement de l'AUV et d'identifier ses paramètres. Ensuite, des tests ont été réalisés dans des environnements plus larges afin de valider certains contrôleurs. Enfin, une phase d'essais préalables a eu lieu pour valider des comportements nécessaires au tour de l'île.



Figure 1: Localisation de l'île des morts (google maps)

1.2 Présentation de l'AUV

On considère le *Riptide* représenté par la figure 3.

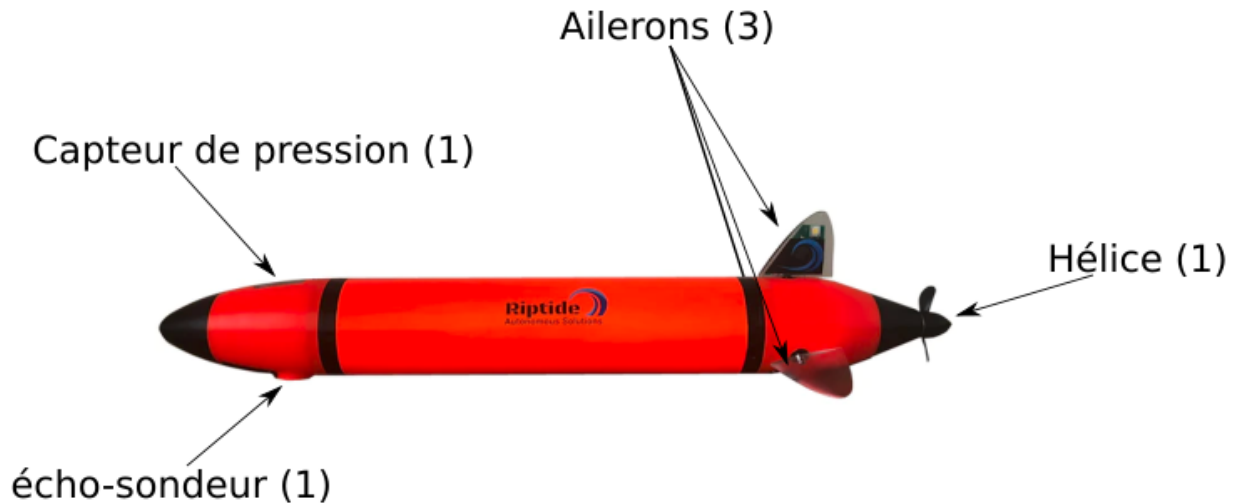


Figure 2: Le Riptide

Le *Riptide* est contrôlé grâce à 4 actionneurs: u_0 qui correspond à la puissance de propulsion demandée, u_1 , u_2 et u_3 qui commandent les angles des ailerons. Dans notre modèle simplifié, on suppose qu'il n'y a pas de glissement latéral et que le *Riptide* n'a qu'une seule translation. De ce fait, la vitesse \mathbf{v}_r du *Riptide* est suivant l'axe local \mathbf{x}_r du *Riptide*. Les équations mécaniques de l'AUV sont présentées à l'annexe A.

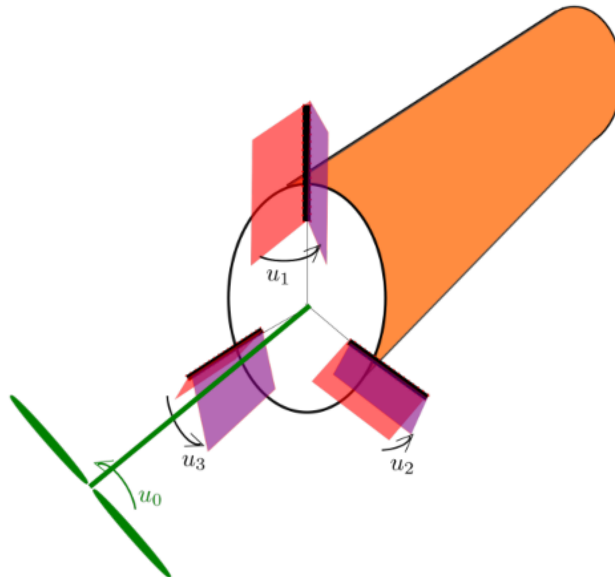


Figure 3: Modélisation du Riptide

1.3 Problématique

Dans le cadre de ce stage, on cherche à contrôler le véhicule sous-marin autonome (AUV) afin d'exécuter le tour de l'Île des Morts. La particularité de cette mission est l'incapacité d'utiliser un capteur de localisation type GPS. En effet, les ondes électromagnétiques ne se propagent pas suffisamment bien dans l'eau pour permettre l'utilisation du GPS par des AUV. Il est à noter néanmoins que des recherches portent sur ce sujet et visent à développer une version sous-marine du GPS, [4], [12]. Autre que l'utilisation du GPS, différentes méthodes de localisations ont été développées et utilisées, [13].

On ne cherche pas ici à reproduire une méthode de localisation développée ou à en créer une nouvelle. On a pour but de vérifier la possibilité d'une navigation "à l'aveugle", dans notre cas, sans connaissance de position. Afin de se séparer du composant de localisation, il est nécessaire d'avoir des garanties de succès. En effet, il est tout à fait possible que notre objectif ne soit pas réalisable à cause d'un mauvais contrôle, d'un défaut de capteurs ou d'une cause externe.

Afin de réaliser notre objectif, on a décidé d'utiliser l'écho-sondeur dont le Riptide est équipé pour exécuter des cycles stables par rebond sur des lignes de niveaux, appelées *isobathes*. Ainsi, on a modélisé notre robot comme étant un système dynamique hybride.

Dans les parties suivantes, on va présenter les outils mathématiques qu'on utilisera, l'approche retenue pour exécuter le tour de l'île, les méthodes mises en place afin de prouver la réalisabilité de cette approche et enfin, les résultats pratiques que l'on a obtenus jusqu'à présent.

2 Outils mathématiques utilisés

Avant de prouver la stabilité d'une trajectoire, on va présenter des notions utiles à notre preuve. [1] présente la majorité de ces concepts. On essaiera donc de se restreindre aux détails nous concernant et de guider vers d'autres références lorsque c'est pertinent.

2.1 Modélisation d'un robot

Souvent, on choisit de modéliser les robots mobiles par un système dynamique, c'est-à-dire, un ensemble d'équations mathématiques qui décrit l'évolution du système dans le temps, [10].

Définition 2.1. Un système dynamique est une fonction $\phi : T \times S \rightarrow S$ telle que:

- $T \in \{\mathbf{R}, \mathbf{R}^+, \mathbf{Z}, \mathbf{N}\}$. On appelle $t \in T$ le *paramètre d'évolution*. On parle de système dynamique continu lorsque $T = \mathbf{R}$ ou \mathbf{R}^+ et de système dynamique discret lorsque $T = \mathbf{Z}$ ou \mathbf{N}
- S est l'espace des *état*.
- $\phi(0, \cdot)$ est la fonction identité.
- $\phi(t, \phi(s, x)) = \phi(t + s, x)$ pour tout $t, s \in T$ et pour tout $x \in S$.

Remarque 2.1. Lorsque $x = x_0$ constante, on appelle $\phi(t, x_0) = \phi_{x_0}(t)$ le *flux* du système passant par x_0 .

D'un autre côté, en utilisant les théories de la mécanique classique, une modélisation du robot aboutit à une équation différentielle ordinaire (ODE en anglais), de la forme suivante:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \quad (2.1)$$

où $\mathbf{f} \in \mathcal{C}^k(\mathbf{R}^n)$ appelé *champ vectoriel*, $t \in \mathbf{R}$ représente le temps et $\mathbf{x}(t) \in \mathbf{R}^n$ représente l'état du robot à l'instant t .

Cette équation admet une unique solution $\phi(t, \mathbf{x}_0)$ qui passe par le point \mathbf{x}_0 à l'instant $t = 0$. Cette solution correspond au *flux* du système dynamique continu associé, initialisé à \mathbf{x}_0 et $t = 0$.

Ainsi, le problème qui nous intéresse dans la suite consiste à trouver une solution numérique à cette équation différentielle en partant d'une valeur initiale.

Définition 2.2. Résoudre un problème de valeur initiale consiste à trouver la trajectoire du système décrit par:

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t)) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{cases}$$

sur l'intervalle de temps $[t_0, t_f]$.

Afin d'exécuter le tour de l'île, on demandera à notre robot d'effectuer une mission consistant à changer de direction à des moments précis. Ne pouvant pas exprimer ces moments au cours du temps, on a décidé de les associer au isobathes (les lignes de niveaux).

Ainsi, entre deux changements de directions, on modélisera le *Riptide* par un système dynamique continu exprimant son évolution au cours du temps. D'autre part, chaque rencontre avec une isobathe constitue un événement majeur modifiant la dynamique du robot et la commande reçue. On modélisera alors le changement de commande par un système dynamique discret. Ce mélange des deux modèles est appelé un système dynamique *hybride*. La partie suivante rentrera plus en détail dans ce cas particulier de systèmes dynamiques.

2.1.1 Systèmes dynamiques hybrides

On peut trouver dans la littérature différentes définitions des systèmes hybrides, que ce soit un sextuple [5], septuple [2] ou octuple [15]. On prendra ici le choix de présenter la définition de [1].

Définition 2.3. Un système hybride est un sextuple $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$ où:

- $\mathcal{Q} = \{q_0, \dots, q_m\}$ est l'ensemble fini des états discrets de \mathcal{H} .
- $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ est l'ensemble des *transitions* entre les états discrets.
- $\mathcal{D} = \{D_{q_i}, q_i \in \mathcal{Q}\}$ est l'ensemble des domaines de \mathcal{H} .
- $\mathcal{F} = \{\mathbf{f}_{q_i}, q_i \in \mathcal{Q}\}$ est l'ensemble des *champs vectoriels*. On associe à chaque \mathbf{f}_{q_i} le système dynamique $\phi_{q_i}(t, \mathbf{x})$.
- $\mathcal{G} = \{G_e, e \in \mathcal{E}\}$ l'ensemble des *gardes*. Pour $e = (q_i, q_j)$, $G_e \subset D_{q_j}$.
- $\mathcal{R} = \{R_e, e \in \mathcal{E}\}$ l'ensemble des *ré-initialisations*, où, pour $e = (q_i, q_j)$, $R_e : G_e \rightarrow D_{q_j}$.

Comme on peut le remarquer, un système hybride trouve son nom dans les variables discrètes $q_i \in \mathcal{Q}$ et la variable continue $\mathbf{x} \in \mathbf{R}^n$. Une exécution du système consiste à évoluer à partir d'un état initial (q_i, \mathbf{x}) , où $\mathbf{x} \in D_{q_i}$ est régi par la relation $\dot{\mathbf{x}} = \mathbf{f}_{q_i}(\mathbf{x})$. L'évolution continue jusqu'à atteindre une condition G_e , $e = (q_i, q_j)$ pour $j \neq i$. Alors, la valeur de \mathbf{x} est réinitialisée par la fonction R_e associée et la variable discrète q_i devient q_j .

Montrons quelques exemples afin d'illustrer l'exécution d'un système hybride.

Exemple 2.1. Commençons d'abord par présenter un exemple classique des systèmes hybrides: le thermostat [15], [8].

Le plus courant, et clair, pour représenter un système hybride est de le mettre sous-forme de machine à état, Figure 4.

Le fonctionnement du thermostat est simple: supposons qu'initialement le thermostat soit à l'arrêt. Lorsque x (la température) est inférieure à 17, le thermostat change d'état et se met en marche. La température augmente alors tant qu'elle est inférieure à 23. En dépassant 21, le thermostat se remet à l'arrêt et la température diminue.

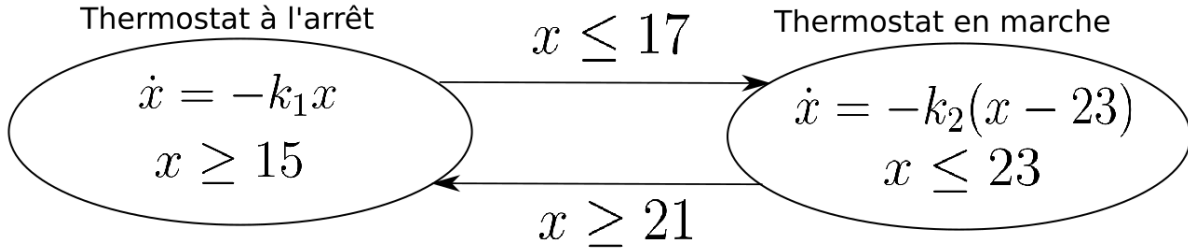


Figure 4: Machine à état d'un thermostat autonome

Exprimons maintenant ce fonctionnement de manière plus rigoureuse, en utilisant les notations de la définition 2.3.

Posons les ensembles suivants:

- $\mathcal{Q} = \{0, 1\}$ où 0 représente l'état d'arrêt et 1 représente l'état de marche.
- $\mathcal{E} = \{(0, 1), (1, 0)\}$ l'ensemble des deux transitions possibles.
- $\mathcal{D} = \{D_0, D_1\}$ où $D_0 = [15, +\infty]$ et $D_1 = [-\infty, 23]$.
- $\mathcal{F} = \{\mathbf{f}_0, \mathbf{f}_1\}$ avec $\mathbf{f}_0(x) = -k_1 x$ et $\mathbf{f}_1(x) = -k_2(x - 23)$.
- $\mathcal{G} = \{G_{(0,1)}, G_{(1,0)}\}$ avec $G_{(0,1)} = \{x \leq 17\}$ et $G_{(1,0)} = \{x \geq 21\}$
- $\mathcal{R} = \{R_{(0,1)}, R_{(1,0)}\}$ avec $R_e = id$ pour $e \in \mathcal{E}$.

Dans chaque état (0 ou 1), le thermostat est représenté par un système dynamique continu avec un champ vectoriel \mathbf{f}_{q_i} . Chaque transition est réalisée après validation des conditions G_e et le vecteur d'état x appartient au domaine D_{q_i} .

Le système hybride que nous venons de définir et son illustration en machine à état sont équivalents. On priorisera la machine à état dans ce qui suit afin de préserver la clarté de nos propos.

Exemple 2.2. Présentons à présent un exemple plus avancé qui sera important dans la validation du tour de l'île.

Prenons comme modèle le char de Dubin, un véhicule régi par le système d'équations suivant:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = u \end{cases} \quad (2.2)$$

On désire contrôler ce véhicule afin qu'il fasse un motif de L. Pour ceci, on contrôlera le robot à la fois à l'aide d'une horloge interne et de sa position. On lui demandera dans un

premier temps d'avancer pendant une certaine durée, de tourner à gauche et avancer jusqu'à une certaine position, de faire demi tour et revenir pendant une certaine durée et enfin de tourner à droite jusqu'à atteindre une certaine position. Au lieu d'exprimer les différents ensembles $(\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$, on se contentera d'une machine à états, figure 5.

On note \mathbf{X} le vecteur (x, y, θ, τ) , où τ l'horloge interne du véhicule. On désigne par θ_d l'angle désiré.

On pose alors

$$\mathbf{f}(\mathbf{X}, \theta_d) = \begin{cases} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= u(\theta, \theta_d) \\ \dot{\tau} &= 1 \end{cases}$$

Enfin, on prend deux fonctions g et d tel que, $g(x, y) = 0$ signifie que le véhicule a atteint la position désirée après avoir tourné à gauche et $d(x, y) = 0$ signifie que le véhicule a atteint la position désirée après avoir tourné à droite.

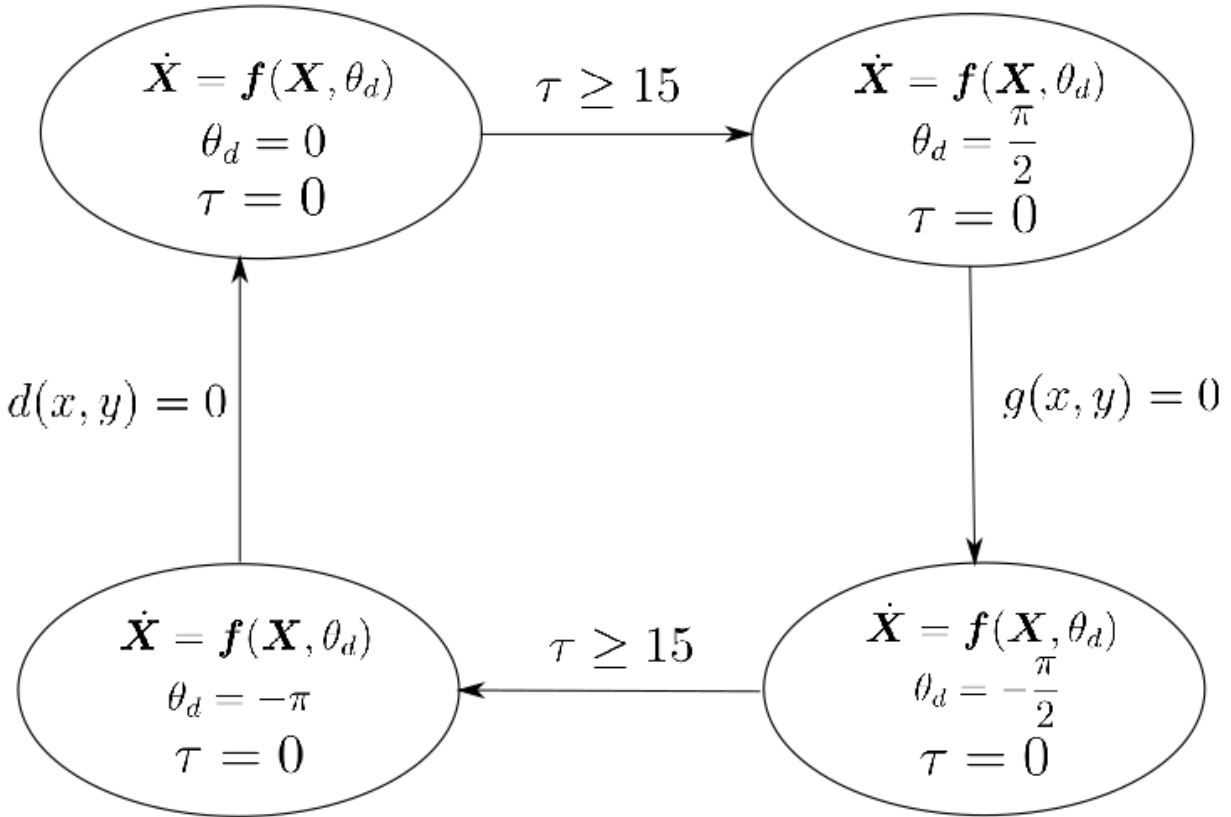


Figure 5: Machine à état du motif L

Remarque 2.2. La différence entre ce système hybride et celui du thermostat est la présence de la ré-initialisation $\tau = 0$ à chaque changement d'état discret.

2.2 Calculs par intervalles

Dans la robotique, il est commun de travailler dans des milieux avec beaucoup d'incertitudes. Ces dernières concernent la position initiale du robot, en particulier les robots sous-marins, les paramètres du robot comme son centre de masse ou son centre de carène, les informations mesurées par les capteurs, les arrondis des nombres flottants par les machines, etc. Il est donc impératif, lorsqu'on cherche à proposer une preuve de garantie, de prendre toutes ces incertitudes en compte. De ce fait, les valeurs utilisées dans nos calculs doivent faire apparaître les moultes erreurs envisageables. On n'utilisera donc plus des valeurs scalaires mais des intervalles. Cette partie présentera brièvement les bases des calculs par intervalles. Pour une explication plus détaillée, se référer à [7].

2.2.1 Introduction à l'arithmétique des intervalles

Un *intervalle* est un ensemble connexe et fermé compris entre deux valeurs, a et b , où $a \leq b$. On le note $I = [a, b]$. On note \mathbf{IR} l'ensemble des intervalles de \mathbf{R} .

Une autre notation de l'intervalle peut être utilisée, et sera préférée dans le reste du rapport: $[x] = [x^-, x^+]$. Ainsi, x^- est la borne inférieure de l'intervalle $[x]$ et x^+ est la borne supérieure. On étend les opérations arithmétiques sur \mathbf{R} à l'ensemble \mathbf{IR} en posant:

$$[x] * [y] = [\{x * y | x \in [x], y \in [y]\}]$$

avec $*$ $\in \{+, -, \times, \div\}$. À noter que pour la division, différents cas se présentent lorsque 0 est inclus dans un des intervalles.

On définit aussi l'intersection et l'union des intervalles par:

$$[x] \cap [y] = \{z \in \mathbf{R} | z \in [x], z \in [y]\} \quad (2.3)$$

$$[x] \sqcup [y] = \{z \in \mathbf{R} | z > \min(x^-, y^-), z < \max(x^+, y^+)\} \quad (2.4)$$

Remarque 2.3. La définition de \sqcup au lieu de l'utilisation de \cup (l'union naturelle des sous-ensemble sur \mathbf{R}) est expliquée par le fait que l'union de deux intervalles n'est pas nécessairement un intervalle.

Enfin, on définit les fonctions suivantes sur les intervalles:

$$\begin{aligned} lb([x]) &= x^- \\ ub([x]) &= x^+ \\ mid([x]) &= \frac{x^- + x^+}{2} \\ width([x]) &= x^+ - x^- \\ |[x]| &= \max(|x^-|, |x^+|) \end{aligned}$$

2.2.2 Vecteurs et matrices d'intervalles

À partir de la définition d'un intervalles $[x]$, on peut définir un vecteur d'intervalle $[x]$ comme étant un produit cartésien d'intervalles. Par exemple,

$$\begin{pmatrix} [-1, 1] \\ [4, 5] \\ [0, 0.2] \end{pmatrix}$$

est un vecteur d'intervalles de dimension 3. On peut aussi appeler ces vecteurs d'intervalles des boîtes. En effet, le vecteur défini précédemment peut représenter une boîte de longueur 2, de largeur 1 et de hauteur 0.2 dans l'espace (x, y, z) .

On définit ensuite une matrice d'intervalles comme étant une concaténation de vecteurs d'intervalles. Par exemple:

$$\begin{pmatrix} [-1, 1] & [-2, -1] & [1, 3] \\ [4, 5] & [0, 2] & [6, 7] \\ [0, 0.2] & [1, 2] & [-4, -3] \end{pmatrix}$$

On étend alors la définition de lb , ub et mid aux vecteurs et matrices d'intervalles. Par exemple:

$$lb \begin{pmatrix} [-1, 1] \\ [4, 5] \\ [0, 0.2] \end{pmatrix} = \begin{pmatrix} -1 \\ 4 \\ 0 \end{pmatrix}$$

$$mid \begin{pmatrix} [-1, 1] & [-2, -1] & [1, 3] \\ [4, 5] & [0, 2] & [6, 7] \\ [0, 0.2] & [1, 2] & [-4, -3] \end{pmatrix} = \begin{pmatrix} 0 & -1.5 & 2 \\ 4.5 & 1 & 6.5 \\ 0.1 & 0.5 & -3.5 \end{pmatrix}$$

Enfin, on définit la norme d'un vecteur d'intervalle comme étant $\|[\mathbf{x}]\| = \max_i |[x_i]|$

Remarque 2.4. Dans ce qui suit, en absence d'ambiguïté, on désignera par vecteurs et matrices les vecteurs et matrices d'intervalles.

2.2.3 Fonctions d'intervalles

Tous comme les vecteurs et les matrices, on étend la définition des fonctions aux intervalles. Ainsi, une image d'un intervalle par une fonction est l'ensemble des images de tous les points de cet intervalle par la même fonction:

$$\mathbf{f}([\mathbf{x}]) = \{\mathbf{f}(\mathbf{x}) | \mathbf{x} \in [\mathbf{x}]\}$$

On gardera la même notation f pour la fonction qu'elle soit une fonction d'intervalle ou une fonction au sens classique du terme.

On définit alors une fonction ensemble $[\mathbf{f}] : \mathbf{IR} \rightarrow \mathbf{IR}$ telle que :

$$\forall [\mathbf{x}] \in \mathbf{IR}^n, \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}])$$

On appelle une telle fonction une *fonction d'inclusion*.

Remarque 2.5. La fonction d'inclusion n'est pas unique. Néanmoins, on peut définir une fonction minimale, qu'on notera $[f]([x])$.

Comme $f([x]) \subset [f]([x])$, on peut, dans certains cas, avoir une approximation très large au bout de quelques pas. Un exemple classique est celui du carré en rotation, [9], qu'on

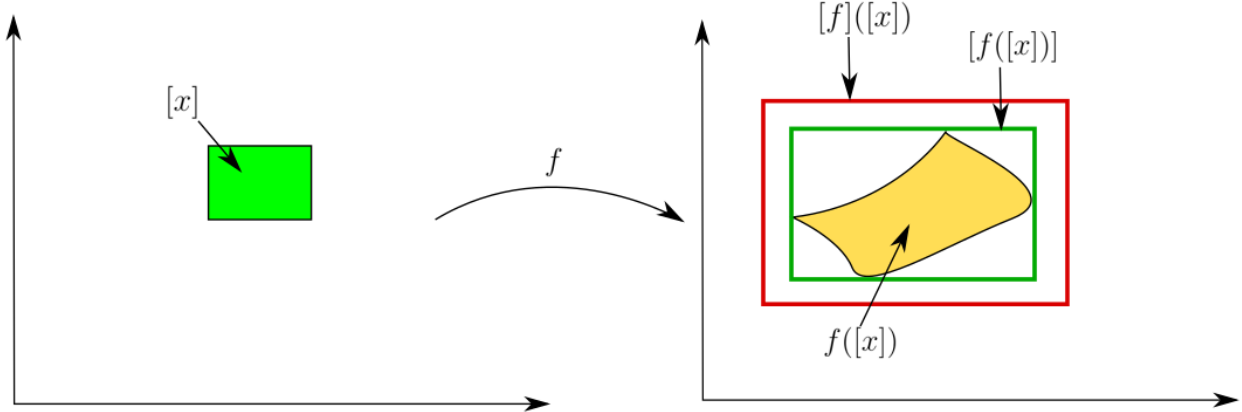


Figure 6: Illustration des fonctions d'inclusion

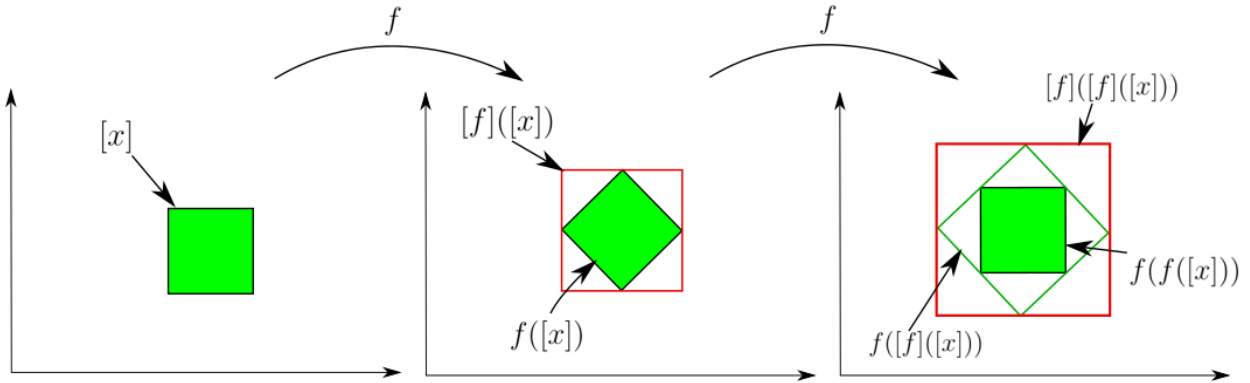


Figure 7: Illustration du wrapping effect

illustre dans la figure 7. On appelle cette sur-approximation le *wrapping effect*.

Il est important de minimiser le *wrapping effect* afin de pouvoir avoir des résultats utilisables et précis. On va donc introduire dans la partie suivante la fonction d'inclusion qu'on utilisera dans nos algorithmes dans le but de s'approcher le plus possible de la fonction d'inclusion minimale et de réduire le *wrapping effect*.

2.2.4 La forme centrée

Afin d'approximer au mieux l'image d'un intervalle, Moore a proposé une définition de fonction d'inclusion dans [11] : *la forme centrée*. [1] propose la définition suivante:

Définition 2.4. Soit la fonction $\mathbf{f} : \mathbf{R}^n \rightarrow \mathbf{R}^n$. Notons \mathbf{J}_f sa matrice jacobienne. Soit $[\mathbf{x}] \in \mathbf{IR}^n$. On définit la forme centrée de \mathbf{f} , notée $[\mathbf{f}_c]$, par:

$$[\mathbf{f}_c]([\mathbf{x}]) = \mathbf{f}(\mathbf{m}) + [\mathbf{J}_f]([\mathbf{x}]) \cdot ([\mathbf{x}] - \mathbf{m})$$

avec $\mathbf{m} \in [\mathbf{x}]$ (par exemple, $\mathbf{m} = \text{mid}([\mathbf{x}])$)

L'avantage de la forme centrée sera discuté plus loin dans la partie 4. Néanmoins, il est à noté que son désavantage majeur est qu'elle ne donne des résultats précis que pour une boîte initiale $[\mathbf{x}_0]$ petite. Mais cela reste suffisant pour notre application.

2.3 Cycles stables

Afin de faire le tour de l'île, deux idées majeures ont été discutées. N'ayant pas beaucoup d'informations sur l'environnement, le robot est plus susceptible de réussir sa mission en se basant sur les isobathes alentours. Ainsi, la première idée consistait à faire un suivi d'isobathes. Cette idée a été vite abandonnée pour deux raisons principales:

1. Dans le cas d'une bathymétrie à faibles variations, le sous-marin peut se retrouver dans une plaine et le suivi d'isobathe l'emmènera loin de l'île. Il pourrait même se retrouver à faire le tour d'une île voisine.
2. Cette technique n'est pas reproductible. En effet, on ne sait pas précisément où sera notre drone et quelle cap il aura à tout instant.

La deuxième raison nous a mené à considérer une approche plus robuste: effectuer des cycles stables.

2.3.1 Cycle limite

La notion de cycle limite est très importante dans moultes domaines. Sa définition repose sur plusieurs concepts qui ne rentrent pas nécessairement dans le sujet de ce rapport. Ainsi, on présentera ici une définition non rigoureuse, mais suffisante pour notre utilisation. Le lecteur pourra se référer à [17], [20] pour des définitions plus rigoureuses.

Définition 2.5. Un cycle limite est une courbe fermée dans l'espace telle que, lorsque le temps tend vers $\pm\infty$, il existe au moins une trajectoire voisine convergant vers cette courbe.

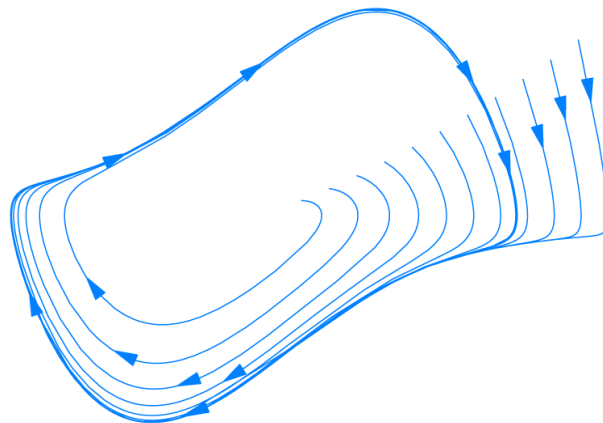


Figure 8: Cycle limite de l'oscillateur de Van Der Pol (image extraite de wikipedia)

Un cycle limite peut être instable ou stable. Un cycle limite instable est un cycle dont les trajectoires divergent au cours du temps. En revanche, un cycle stable est un cycle vers lequel les trajectoires convergent au cours du temps.

L'exemple de l'oscillateur de Van Der Pol, figure 8, présente un cycle limite stable, en gras, et des trajectoires qui convergent vers cette courbe.

Ainsi, afin d'assurer l'exécution du tour de l'île, on va se concentrer sur la recherche d'un cycle limite stable. Il est important à noter que le robot n'est pas obligé de suivre cette trajectoire dès le premier ou deuxième tour. Il nous suffit de savoir qu'au bout de n -tours, le sous-marin convergera vers une trajectoire stable.

2.3.2 Systèmes dynamiques continus et surfaces de Poincaré

Afin d'étudier et de prouver la stabilité d'un cycle limite, une approche consiste à utiliser l'arithmétique des intervalles, présentée dans la partie 2.2, et les surfaces de Poincaré, qu'on présentera dans cette partie. Cette approche a notamment été utilisée par W. Tucker pour prouver que le système de Lorenz est chaotique, [19].

Commençons par donner la définition des surfaces de Poincaré avant d'expliquer la notion et comment on compte s'en servir.

Définition 2.6. Soit un système dynamique continu avec une fonction de *flux* $\phi : \mathbf{R} \times \mathbf{R}^n \rightarrow \mathbf{R}^n$, et soit une trajectoire fermée δ de ce système. On définit une surface \mathcal{S} qui intersecte la trajectoire δ en un point \mathbf{x}_0 . On définit alors la fonction de Poincaré (aussi appelée *Poincaré map*) Π associée à δ par:

$$\begin{aligned} \Pi : \mathcal{S} &\rightarrow \mathcal{S} \\ \mathbf{x} &\mapsto \phi(\tau_{\mathcal{S}}(\mathbf{x}), \mathbf{x}) \end{aligned}$$

avec $\tau_{\mathcal{S}}$ la fonction qui à $\mathbf{x} \in \mathcal{S}$ renvoie la plus petite valeur $t > 0$ tel quel $\phi(t, \mathbf{x}) \in \mathcal{S}$.

On définit souvent la surface \mathcal{S} par une fonction $\sigma : \mathbf{R}^n \rightarrow \mathbf{R}$ tel que $\mathcal{S} = \{\mathbf{x} \in \mathbf{R}^n | \sigma(\mathbf{x}) = 0\}$.

Ainsi, la surface de Poincaré intersecte une trajectoire en différents points $\{x_0, x_1, \dots, x_k, \dots\}$ et permet de suivre l'évolution de cette trajectoire. Une illustration est présentée figure 9. C'est une méthode souvent utilisée pour étudier la stabilité des satellites en orbite. À partir de cette suite de point, on peut étudier la stabilité de l'orbite.

Comme on peut le remarquer, cette définition concerne des trajectoires en orbite et ne permet l'utilisation qu'une seule surface de Poincaré. On étend alors la définition afin d'utiliser plusieurs surfaces consécutives de la manière suivante:

Définition 2.7. Soit un système dynamique continu avec une fonction de *flow* $\phi : \mathbf{R} \times \mathbf{R}^n \rightarrow \mathbf{R}^n$, et soit une trajectoire fermée δ de ce système. On définit une suite de surfaces $(\mathcal{S}_k)_{k \in \mathbf{N}}$ qui intersectent la trajectoire δ en suite de points $(\mathbf{x}_k)_{k \in \mathbf{N}}$. On définit alors la fonction locale de Poincaré (aussi appelée *local Poincaré map*) Π_k associée à δ par:

$$\begin{aligned} \Pi_k : \mathcal{S}_k &\rightarrow \mathcal{S}_{k+1} \\ \mathbf{x}_k &\mapsto \phi(\tau_{\mathcal{S}_{k+1}}(\mathbf{x}_k), \mathbf{x}_k) \end{aligned}$$

avec $\tau_{\mathcal{S}_{k+1}}$ la fonction qui à $\mathbf{x}_k \in \mathcal{S}_k$ renvoie la plus petite valeur t tel quel $\phi(t, \mathbf{x}_k) \in \mathcal{S}_{k+1}$.

On peut maintenant utiliser les surfaces de Poincaré pour étudier des trajectoires de systèmes dynamiques non continus.

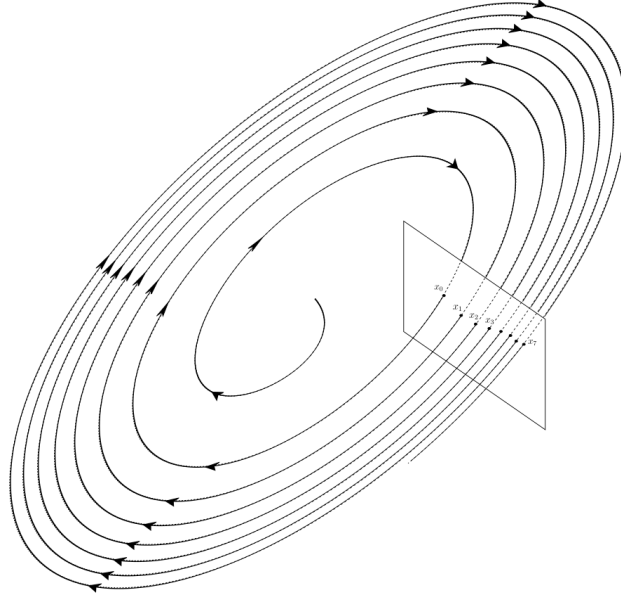


Figure 9: Exemple d'orbite et de surface de Poincaré

2.3.3 Étude des cycles stables grâce aux surfaces de Poincaré

L'utilisation des surfaces de Poincaré nous permet de passer du problème de stabilité d'un cycle limite lié à un système dynamique continu, au problème de stabilité d'un point d'équilibre de la fonction de Poincaré. En effet, [18] propose le théorème suivant:

Théorème 2.1. *Soit un système dynamique continu décrit par l'équation 2.1, $\mathbf{f} \in \mathcal{C}^1(\mathbf{R}^n)$. Soit δ un cycle limite. Soit Π la fonction de Poincaré associée à une surface de Poincaré \mathcal{S} intersectant δ en un point $\bar{\mathbf{x}}$. Alors, δ est (asymptotiquement) stable si et seulement si $\bar{\mathbf{x}}$ est un point d'équilibre de Π (asymptotiquement) stable.*

On est donc amené à étudier la stabilité des systèmes dynamiques discrets, qui s'avérera plus facile à appréhender.

2.3.4 Stabilité d'un point fixe

Proposition 2.1. *Soit un système défini par $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ et soit $\mathbf{J}_{\mathbf{f}}(\bar{\mathbf{x}})$ la matrice jacobienne de \mathbf{f} au point fixe $\bar{\mathbf{x}}$. Alors:*

- *Si les valeurs propres $(\lambda_i)_{i \in \mathbf{N}}$ de $\mathbf{J}_{\mathbf{f}}(\bar{\mathbf{x}})$ satisfont pour tout $i \in \mathbf{N}$, $|\lambda_i| < 1$, alors $\bar{\mathbf{x}}$ est asymptotiquement stable.*
- *S'il existe $j \in \mathbf{N}$ tel que $\lambda_j > 1$, alors $\bar{\mathbf{x}}$ est instable.*
- *Sinon, on ne peut conclure sur la stabilité du point $\bar{\mathbf{x}}$.*

À partir de la proposition précédente, on peut prouver la stabilité d'une suite périodique en étudiant la stabilité d'un point de la suite.

Ayant tous les outils en main, il ne reste qu'à les assembler pour prouver la stabilité d'un système hybride.

2.3.5 Cycle limite hybride

Afin d'étudier la stabilité d'un système hybride, il faut se ramener à un problème de trajectoire périodique, comme le cas dans les système continu avec les orbites, ou dans les systèmes discrets avec les suites périodiques.

Rappelons qu'un système dynamique hybride est décrit par deux variables d'états: $(q(t), \mathbf{x}(t))$, les deux dépendantes du temps. Ainsi, il faut que ces deux variables décrivent un cycle périodique. En notant T la période du système hybride, on obtient, pour tout temps t , $(q(t+T), \mathbf{x}(t+T)) = (q(t), \mathbf{x}(t))$.

Si on suppose qu'un cycle limite existe, et donc que la périodicité est présente, il suffit d'étudier la stabilité de tel cycle en utilisant les surfaces de Poincaré.

Les surfaces utilisées ne seront pas choisies de façon aléatoire, mais seront définies par le système hybride. Considérons un système hybride $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$. Pour tout $e_{ij} = (q_i, q_j) \in \mathcal{E}$, on définit la surface de Poincaré locale $S_{e_{ij}}$ comme étant la surface qui délimite les deux ensembles D_i et D_j .

Ainsi, on définit autant de surfaces locales que de transitions possibles. Ensuite, on définit la fonction de Poincaré locale (*local Poincaré map*):

$$\begin{aligned} \Pi_{q_i} : D_{q_i} &\rightarrow G_{e_{ij}} \\ \mathbf{x} &\mapsto \phi(\tau_{G_{e_{ij}}}(\mathbf{x}), \mathbf{x}) \end{aligned}$$

Si on suppose que notre système hybride possède N états discrets s'exécutant dans l'ordre $(q_0, q_1, \dots, q_{N-1})$, alors on n'étudie plus la variable continue \mathbf{x} du système, mais une suite de de points $(\mathbf{x}_k)_{k \in \mathbf{N}}$ correspondants aux intersections de la trajectoires avec les différentes surfaces de Poincaré locales.

$$\begin{aligned} \mathbf{x}_k &= \Pi_{q_{k-1}} \circ \Pi_{q_{k-2}} \circ \dots \circ \Pi_{q_0}(\mathbf{x}_0), \forall k < N \\ \mathbf{x}_N &= \Pi_{q_{N-1}} \circ \Pi_{q_{N-2}} \circ \dots \circ \Pi_{q_0}(\mathbf{x}_0) = \Pi(\mathbf{x}_0) \\ \mathbf{x}_{k+N} &= \Pi_{q_{k-1}} \circ \Pi_{q_{k-2}} \circ \dots \circ \Pi_{q_0} \circ \Pi(\mathbf{x}_0), \forall k > 1 \end{aligned}$$

où $\Pi = \Pi_{q_{N-1}} \circ \Pi_{q_{N-2}} \circ \dots \circ \Pi_{q_0}$ est la fonction de Poincaré hybride (*hybrid Poincaré map*) associée au système \mathcal{H} .

Avec ces différentes définitions, on s'est ramené à étudier la stabilité d'un point fixe d'une fonction de Poincaré. En utilisant le théorème 2.1 et la proposition 2.1, on a en notre possession tous les outils pour étudier la stabilité d'un système hybride.

3 Le tour de l'île

Expliquons maintenant plus en détail l'approche retenue pour exécuter le tour de l'île, et faisons le lien avec tout ce qui a été introduit précédemment.

Comme mentionné au début de cette partie, on cherche à exécuter des cycles stables avec notre robot. On va donc expliquer comment notre problème de tour de l'île se ramène à une étude de stabilité.

Notre robot est modélisé par un système hybride. Son évolution mécanique au cours du temps, les phénomènes physiques qui agissent sur lui et les forces exercées par ses ailerons et son hélice peuvent être modélisés par un système dynamique continu. Les informations mesurées par ses capteurs et les différentes étapes de sa missions peuvent être modélisées par un système dynamique discret.

Pour exécuter le tour de l'île, on compte effectuer des rebonds sur isobathes, c'est-à-dire, qu'on contrôlera le robot grâce à son écho-sondeur, et que certaines valeurs mesurées amèneront le sous-marin à changer de mission. Plus précisément, ces valeurs induiront un changement de direction. Ainsi, on définira autant d'états discrets q_i que de rebonds nécessaires. Les conditions de transition seront donc des valeurs seuils de mesure par l'écho-sondeur.

Supposons qu'on connaisse la bathymétrie autour de l'île, et que l'on soit capable d'exprimer analytiquement celle-ci. Ainsi, on possède une fonction $h(x, y)$ qui, à chaque point du plan, nous donne la valeur mesurée par l'écho-sondeur. Pour notre preuve, on prendra comme vecteur d'état : $\mathbf{X} = (x, y, \theta, \tau)$ avec θ le cap du sous-marin et τ l'horloge interne du robot. On ne cherche pas à estimer la position (x, y) de notre robot, mais ces valeurs permettront d'obtenir les mesures $h(x, y)$. On gardera aussi le modèle du char de Dubin, introduit dans l'exemple 2.2.

On est, ainsi, passé d'une problématique abstraite à un modèle dynamique hybride visant à exécuter des cycles stables, dont la stabilité sera prouvée avec des surfaces de Poincaré.

Enfin, dans le but de trouver un cycle stable, on visera à minimiser la largeur de l'intervalle après chaque rebond. Un exemple est présenté dans [6], illustré figure 10a

On peut voir ce trajet comme un sous-marin qui exécute des rebonds sur 4 isobathes formant un rectangle. Dans ce cas, le drone "attaque" chaque isobathe avec un angle $\alpha = 72.2$ et en repart avec un angle de "sortie" $\beta = 17.8$. On remarque bien une différence entre le premier rebond sur l'isobathe supérieure et le deuxième (respectivement les deux segments $[GH]$ et $[KL]$) mais que cette différence diminue au niveau de l'isobathe droite et encore plus au niveau de l'isobathe inférieure. En effet, les angles d'"attaque" et de "sortie" ont été bien choisis afin de converger rapidement vers un cycle, celui-ci s'avérant être stable. On peut donc imaginer des angles différents où $\alpha < \beta$ et on obtiendra une trajectoire qui divergera au cours du temps, comme illustré à la figure 10b.

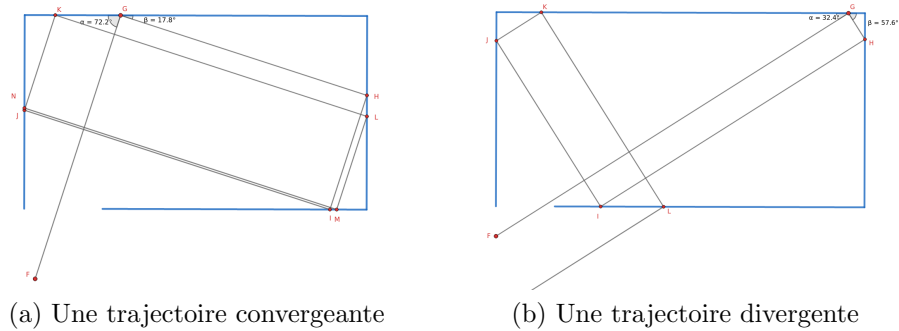


Figure 10: trajectoires et cycle limite

En se basant sur ce principe, on voudrait évaluer la faisabilité du tour de l'île. Considérons la figure 11 ci-dessous où la fonction de bathymétrie est une somme de quatre gaussiennes aléatoires.

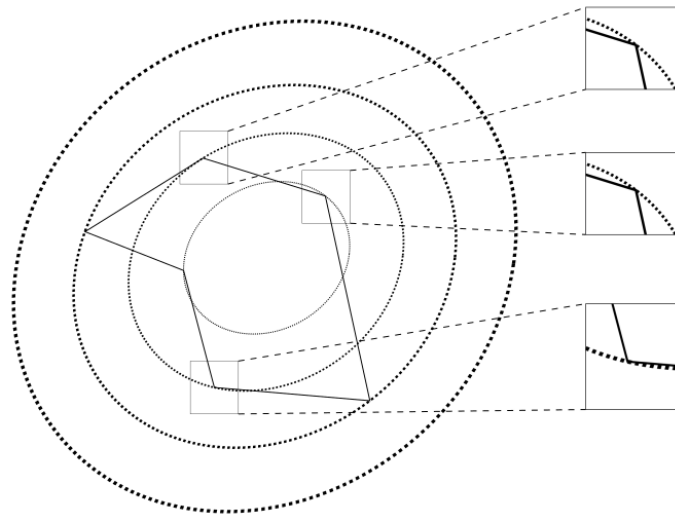


Figure 11: Exemple de rebonds sur isobathes

Ainsi, chaque isobathe a une forme d'ellipse. La première approche retenue fut de faire des rebonds sur trois niveaux distincts, pour un total de six rebonds. N'arrivant pas à trouver une manière de retourner du niveau le plus profond (le plus excentré) vers le niveau le moins profond (le moins excentré) sans augmenter l'erreur, on ne minimise les angles que sur quatre rebonds. On décide aussi de parcourir cette trajectoire en partant du niveau le plus profond.

Afin de tester le bon choix des rebonds, on implémente cette trajectoire sur *Python* où on utilise le modèle du char de Dubin pour notre simulation. La figure 12 illustre l'exécution de l'algorithme.

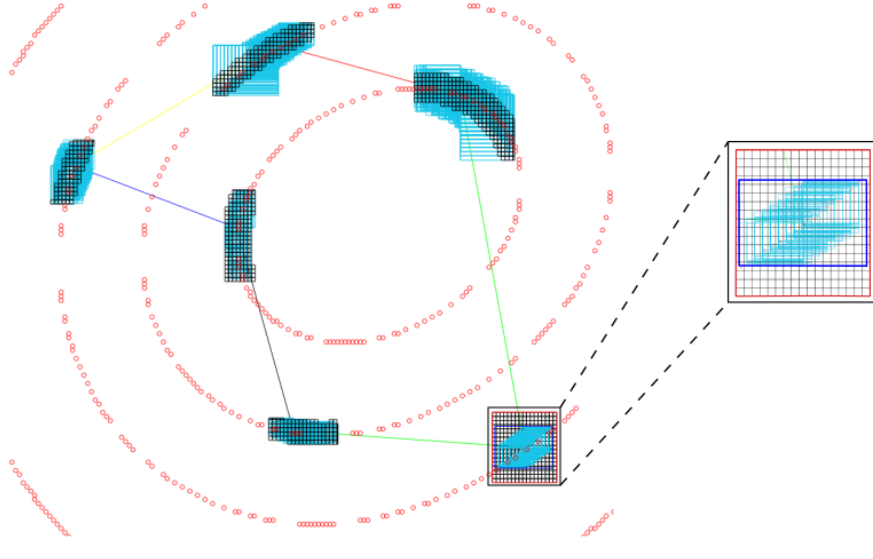


Figure 12: Trajectoire avec rebonds sur isobathes

On considère donc notre trajectoire. On se place au sommet en bas à droite de la figure, et on observe le comportement de tous les points présents dans une boîte de départ (boîte rouge). Pour cela, on se repose sur les calculs par intervalles et sur le principe de pavage. On découpe notre boîte en pavés, on regarde l'image de chaque pavé sur une arête, et on identifie l'union de toutes les images comme celle de la boîte initiale. Ensuite, on redécoupe cette image en pavé. Néanmoins, on ne considère que ceux sur l'isobathe. Le résultat final est l'image de la boîte initiale après avoir exécuté la trajectoire (boîte bleue). On remarque que la boîte bleue est incluse dans la boîte rouge, ce qui nous donne un premier résultat : *l'invariance positive*.

Définition 3.1. [14] Soit $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ un système dynamique, \mathbf{x}_0 l'état initial et $\mathbf{x}(t)$ une trajectoire partant de \mathbf{x}_0 . Soit un ensemble $\mathcal{O} \triangleq \{\mathbf{x} \in \mathbf{R}^n | \phi(\mathbf{x}) = 0\}$ avec ϕ une fonction à valeurs réelles. On dit que \mathcal{O} est un ensemble positivement invariant lorsque:

$$\mathbf{x}_0 \in \mathcal{O} \implies \forall t \geq 0, \mathbf{x}(t) \in \mathcal{O}$$

Ainsi, lorsqu'une trajectoire rentre dans \mathcal{O} , elle y reste pour toujours.

On est ainsi assuré qu'un départ de la boîte rouge nous permet de revenir à celle-ci. Néanmoins, on ne possède aucune information concernant la stabilité du cycle.

Dans la partie suivante, on prouvera la stabilité de ce cas particulier.

4 Preuve de stabilité

La méthode utilisée pour prouver la stabilité de notre cycle est celle présentée dans [1]. Celle-ci repose sur la bonne propriété suivante:

Proposition 4.1. *Soit un système discret défini par $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ tel que $\mathbf{f}(\bar{\mathbf{x}}) = \bar{\mathbf{x}}$. Soit $[\mathbf{x}_0]$ un intervalle englobant la position initiale du système. On a:*

$\exists \alpha < 1, \exists q \in \mathbf{N}^*, [f_c^q](\mathbf{x}_0) \subset \alpha[\mathbf{x}_0] \Rightarrow$ *le système dynamique est asymptotiquement stable.*

Après avoir introduit quelques dernières notions, on utilisera cette propriété sur la fonction de Poincaré hybride Π de notre système pour prouver la stabilité de certains cycles et du tour de l'île.

4.1 Contracteur de stabilité

Commençons par introduire l'outil mathématique permettant de garantir la stabilité asymptotique d'un système.

Définition 4.1. Soit $[\mathbf{x}_0] \ni \mathbf{0}$ une boîte de \mathbf{R}^n . Soit $[\mathbf{a}] \ni \mathbf{0}$ et $[\mathbf{b}] \ni \mathbf{0}$ deux sous-boîte de $[\mathbf{x}_0]$. On appelle *contracteur de stabilité* de ratio $\alpha < 1$ une application $\Psi : \mathbf{IR}^n \rightarrow \mathbf{IR}^n$ satisfaisant les propriétés suivantes:

1. $[\mathbf{a}] \subset [\mathbf{b}] \Rightarrow \Psi([\mathbf{a}]) \subset \Psi([\mathbf{b}])$
2. $\Psi([\mathbf{a}]) \subset [\mathbf{a}]$
3. $\Psi(\mathbf{0}) = \mathbf{0}$
4. $\Psi([\mathbf{a}]) \subset \alpha \cdot [\mathbf{a}] \Rightarrow \forall k \geq 1, \Psi^k([\mathbf{a}]) \subset \alpha^k \cdot [\mathbf{a}]$

Avant d'avancer plus loin dans les notions, on peut remarquer que le point 4 de la définition nous permet de proposer la propriété suivante:

Proposition 4.2. *Si $[\mathbf{x}_0] \ni \mathbf{0}$ est une boîte de \mathbf{R}^n et si Ψ est un contracteur de stabilité de ratio $\alpha < 1$, alors $\lim_{k \rightarrow \infty} \Psi^k([\mathbf{a}]) = \mathbf{0}$*

Cette proposition nous permet de prouver le résultat important suivant:

Proposition 4.3. *Soit un système discret défini par $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ tel que $\mathbf{f}(\mathbf{0}) = \mathbf{0}$. Soit $[\mathbf{x}_0]$ un intervalle englobant la position initiale du système. Soit Ψ est un contracteur de stabilité de ratio $\alpha < 1$ tel que $\Psi([\mathbf{x}_0]) \subset \alpha \cdot [\mathbf{x}_0]$. Si pour tout $k \in \mathbf{N}$, $\mathbf{f}^k([\mathbf{x}_0]) \subseteq \Psi^k([\mathbf{x}_0])$, alors le système est asymptotiquement stable.*

Avant de prouver cette affirmation, montrons comment elle permet de prouver la proposition 4.1.

Dans la proposition 4.3, il nous faut définir trois objets et les lier par une inclusion pour obtenir la stabilité asymptotique. Prenons \mathbf{f} et $[\mathbf{x}_0]$ convenablement. (À noter qu'ici, on prendra $\bar{\mathbf{x}} = \mathbf{0}$.) On sait, par définition de la forme centrée, que $\mathbf{f}([\mathbf{x}_0]) \subseteq [\mathbf{f}_c](\mathbf{x}_0)$.

Donc, il existe $q \geq 1$ tel que $\mathbf{f}([\mathbf{x}_0]) \subseteq [\mathbf{f}_c^q]([\mathbf{x}_0])$, et alors, pour tout $k \in \mathbf{N}$, $\mathbf{f}^k([\mathbf{x}_0]) \subseteq [\mathbf{f}_c^{kq}]([\mathbf{x}_0])$. Ainsi, la relation d'inclusion est déjà présente. Prouver la stabilité asymptotique du système revient alors à montrer qu'il existe $q \geq 1$ tel que $[\mathbf{f}_c^q]$ soit un contracteur de stabilité.

Preuve. Soit Ψ un contracteur de stabilité de ratio $\alpha < 1$ tel que $\Psi([\mathbf{x}_0]) \subset \alpha \cdot [\mathbf{x}_0]$. Supposons que pour tout $k \in \mathbf{N}$, $\mathbf{f}^k([\mathbf{x}_0]) \subseteq \Psi^k([\mathbf{x}_0])$. Par la proposition 4.2, on a la suite d'implications suivante:

$$\begin{aligned} \Psi([\mathbf{x}_0]) \subset \alpha \cdot [\mathbf{x}_0] &\Rightarrow \lim_{k \rightarrow \infty} \Psi^k([\mathbf{x}_0]) = \mathbf{0} \\ &\Rightarrow \lim_{k \rightarrow \infty} \mathbf{f}^k([\mathbf{x}_0]) = \mathbf{0} \\ &\Rightarrow \forall \mathbf{x}_0 \in [\mathbf{x}_0] \lim_{k \rightarrow \infty} \mathbf{f}^k(\mathbf{x}_0) = \mathbf{0} \\ &\Rightarrow \forall \mathbf{x}_0 \in [\mathbf{x}_0] \lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{0} \end{aligned}$$

qui correspond à la définition de la stabilité asymptotique. \square

On a ainsi montré la proposition 4.1 pour $\bar{\mathbf{x}} = \mathbf{0}$. Afin de la prouver pour tout $\bar{\mathbf{x}}$, il suffit d'adapter la définition d'un contracteur de stabilité comme suit:

Définition 4.2. Soit $[\mathbf{x}_0] \ni \bar{\mathbf{x}}$ une boîte de \mathbf{R}^n . Soit $[\mathbf{a}] \ni \bar{\mathbf{x}}$ et $[\mathbf{b}] \ni \bar{\mathbf{x}}$ deux sous-boîte de $[\mathbf{x}_0]$. On appelle *contracteur de stabilité* de ratio $\alpha < 1$ une application $\Psi : \mathbf{IR}^n \rightarrow \mathbf{IR}^n$ satisfaisant les propriétés suivante:

1. $[\mathbf{a}] \subset [\mathbf{b}] \Rightarrow \Psi([\mathbf{a}]) \subset \Psi([\mathbf{b}])$
2. $\Psi([\mathbf{a}]) \subset [\mathbf{a}]$
3. $\Psi(\bar{\mathbf{x}}) = \bar{\mathbf{x}}$
4. $\Psi([\mathbf{a}]) \subset \alpha \cdot ([\mathbf{a}] - \bar{\mathbf{x}}) + \bar{\mathbf{x}} \Rightarrow \forall k \geq 1, \Psi^k([\mathbf{a}]) \subset \alpha^k \cdot ([\mathbf{a}] - \bar{\mathbf{x}}) + \bar{\mathbf{x}}$

4.2 Exemple de preuve pour un système hybride

Reprenons l'exemple 2.2, et rappelons le champ vectoriel du système dynamique hybride.

$$\mathbf{f}(\mathbf{X}, \theta_d) = \begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = u(\theta, \theta_d) \end{cases}$$

où $\mathbf{X} = (x, y, \theta)$. Dans cet exemple, définissons la fonction de contrôle u comme étant la fonction d'erreur:

$$u : \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}$$

$$(\theta, \theta_d) \mapsto \theta_d - \theta$$

4.2.1 Formalisation de l'exemple

Afin d'approcher cet exemple du tour de l'île, on ne contrôlera pas le système en position, mais en profondeur mesurée. On utilisera donc les fonctions g et d comme des expressions analytiques d'isobathes qui définiront deux surfaces de Poincaré.

L'exemple ayant pour objectif d'être reproduit en mer, prenons pour g et d des fonctions s'approchant un peu de la réalité. Ainsi, prenons des fonctions paraboliques comme isobathes:

$$g(x, y) = x - 0.1 * y^2 - 20$$

$$d(x, y) = x - 0.1 * y^2 + 2$$

Afin de contrôler le sous-marin grâce à son horloge interne, on va modifier le système légèrement de la manière suivante:

$$\mathbf{X} = (x, y, \theta, \tau)$$

$$\mathbf{f}(\mathbf{X}, \theta_d) = \begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \theta_d - \theta \\ \dot{\tau} = 1 \end{cases}$$

Ainsi, on peut définir 4 surfaces de Poincaré locales, associées aux 4 surfaces délimitantes $G_e \in \mathcal{G}$:

$$G_{(0,1)} = \{\mathbf{X} \in \mathbf{R}^4 | \tau \geq 15\}$$

$$G_{(1,2)} = \{\mathbf{X} \in \mathbf{R}^4 | x - 0.1 \cdot y^2 - 20 = 0\}$$

$$G_{(2,3)} = \{\mathbf{X} \in \mathbf{R}^4 | \tau \geq 15\}$$

$$G_{(3,0)} = \{\mathbf{X} \in \mathbf{R}^4 | x - 0.1 \cdot y^2 + 2 = 0\}$$

et définissons les fonctions de Poincaré locale:

$$\Pi_0 : \mathbf{R}^4 \rightarrow G_{(0,1)}$$

$$\mathbf{x} \mapsto \phi_0(\tau_{G_{(0,1)}}(\mathbf{x}), \mathbf{x})$$

$$\Pi_1 : \mathbf{R}^4 \rightarrow G_{(1,2)}$$

$$\mathbf{x} \mapsto \phi_1(\tau_{G_{(1,2)}}(\mathbf{x}), \mathbf{x})$$

$$\Pi_2 : \mathbf{R}^4 \rightarrow G_{(2,3)}$$

$$\mathbf{x} \mapsto \phi_2(\tau_{G_{(2,3)}}(\mathbf{x}), \mathbf{x})$$

$$\Pi_3 : \mathbf{R}^4 \rightarrow G_{(3,0)}$$

$$\mathbf{x} \mapsto \phi_3(\tau_{G_{(3,0)}}(\mathbf{x}), \mathbf{x})$$

Enfin, définissons la fonction de Poincaré hybride définie par:

$$\Pi : \mathbf{R}^4 \rightarrow G_{(3,0)}$$

$$\mathbf{x} \mapsto \Pi_3 \circ \Pi_2 \circ \Pi_1 \circ \Pi_0(\mathbf{x})$$

Celle-ci permet alors d'obtenir la suite de points $\mathbf{x}_{k+1} = \Pi(\mathbf{x}_k)$

4.2.2 Présentation des calculs formels

On souhaite appliquer la proposition 4.1 et le théorème 2.1 afin de prouver la stabilité du système.

Pour calculer l'image de $[\mathbf{x}_0]$ par Π , posons:

$$[\mathbf{x}_a] = [\Pi_0]([\mathbf{x}_0])$$

$$[\mathbf{x}_b] = [\Pi_1]([\mathbf{x}_a])$$

$$[\mathbf{x}_c] = [\Pi_2]([\mathbf{x}_b])$$

$$[\mathbf{x}_1] = [\Pi_3]([\mathbf{x}_c])$$

Ainsi, $[\mathbf{x}_1] = [\Pi]([\mathbf{x}_0])$

De même, afin de calculer la matrice jacobienne de Π en $[\mathbf{x}_0]$, posons:

$$[M_a] = \left[\frac{\partial \Pi_0}{\partial \mathbf{x}} \right] ([\mathbf{x}_0])$$

$$[M_b] = \left[\frac{\partial \Pi_1}{\partial \mathbf{x}} \right] ([\mathbf{x}_a])$$

$$[M_c] = \left[\frac{\partial \Pi_2}{\partial \mathbf{x}} \right] ([\mathbf{x}_b])$$

$$[M_1] = \left[\frac{\partial \Pi_3}{\partial \mathbf{x}} \right] ([\mathbf{x}_c])$$

et on obtient :

$$\left[\frac{\partial \Pi}{\partial \mathbf{x}} \right] ([\mathbf{x}_0]) = [M_1] \cdot [M_c] \cdot [M_b] \cdot [M_a]$$

La définition 2.4 nous donne alors:

$$[\Pi_c]([\mathbf{x}_0]) = \left[\frac{\partial \Pi}{\partial \mathbf{x}} \right] ([\mathbf{x}_0]) \cdot ([\mathbf{x}_0] - [\bar{\mathbf{x}}]) + [\Pi]([\bar{\mathbf{x}}])$$

4.2.3 Implémentation de l'exemple

Afin de réaliser ces calculs analytiquement difficiles, voire même impossibles, on se tourne vers la librairie *CAPD* dans C++, [21].

Les résultats des figures 13 ont été réalisées avec:

$$\epsilon = 0.05$$

$$v = 1$$

$$[\mathbf{X}_0] = \begin{pmatrix} [-4.39 - \epsilon, -4.39 + \epsilon] \\ [-4.89 - \epsilon, -4.89 + \epsilon] \\ [-\pi - \epsilon, -\pi + \epsilon] \end{pmatrix}$$

Après avoir déterminé un point fixe de notre système, on considère une boîte $[\mathbf{x}_0]$ contenant ce point. On détermine ensuite l'image de cette boîte à travers les différentes surfaces de Poincaré. Enfin, on calcule la forme centrée de la *hybrid Poincaré map* associée au système. Comme $[\Pi_c]([\mathbf{x}_0]) \subset [\mathbf{x}_0]$, alors, par la proposition 4.1, ce système est stable.

Ne connaissant pas parfaitement la bathymétrie et ne voulant pas risquer de perdre de Riptide au large, on a réfléchi à un cycle légèrement différent où le sous-marin devra faire des rebonds successifs sur la même isobathe. Ainsi, $g(x, y) = d(x, y)$ et on adapte les différents θ_d afin de convenir à notre cycle. La simulation via *CAPD* nous donne les résultats figures 14 et 15, prouvant la stabilité du système.

4.3 Stabilité du tour de l'île

Reprenons la méthode discutée dans la partie 3 et illustrée à travers les figures 11 et 12. On désire maintenant appliquer les nouveaux outils à notre disposition afin d'apporter une

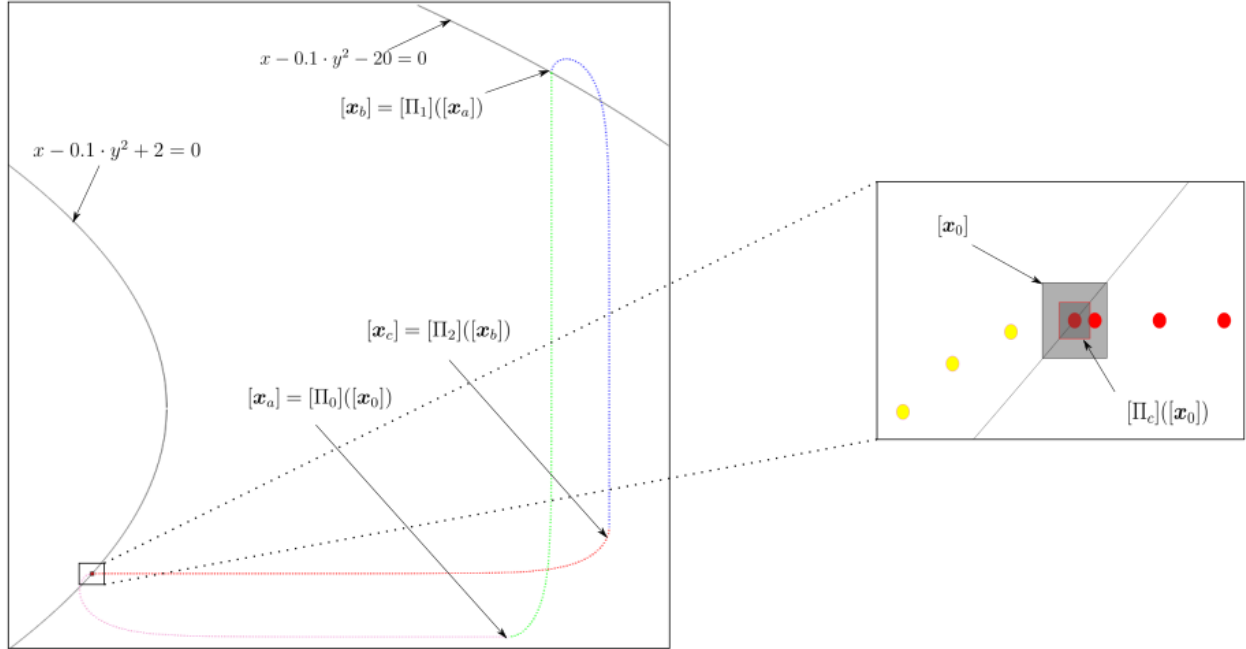


Figure 13: Preuve de stabilité du système hybride

preuve plus rigoureuse à la stabilité du système présenté. On utilise donc l'outil CAPD en définissant 6 surfaces de Poincaré pour les 6 rebonds présents.

Remarque 4.1. En réalité, on ne définit que 3 surfaces, une pour chaque isobathe utilisée, car on a choisi un exemple analytiquement simple à exprimer. Néanmoins, si, pour une isobathe, la courbe qui l'approxime au niveau d'un rebond ne correspond pas à celle qui l'approxime au niveau du second rebond, il est nécessaire de définir deux surfaces distinctes.

On reprend ainsi notre bathymétrie définie par la somme de quatre gaussiennes. On donne ci-dessous les gaussiennes générées:

Paramètre	k_i	σ_i	x_i	y_i
valeur	1.72360383	4.73574262	-6.98657222	-3.2560395
	1.25013936	4.75095462	1.14222231	-0.37564894
	1.82006509	6.05487422	-4.51795131	-4.16665923
	1.22974667	4.58370601	-0.72522199	-0.30465499

Table 1: Paramètres des gaussiennes

Les résultats des figures 16 ont été réalisées avec:

$$\epsilon = 0.005$$

$$[\mathbf{X}_0] = \begin{pmatrix} [1.513 - \epsilon, 1.513 + \epsilon] \\ [-7.245 - \epsilon, -7.245 + \epsilon] \\ [-0.065 - \epsilon, -0.065 + \epsilon] \end{pmatrix}$$

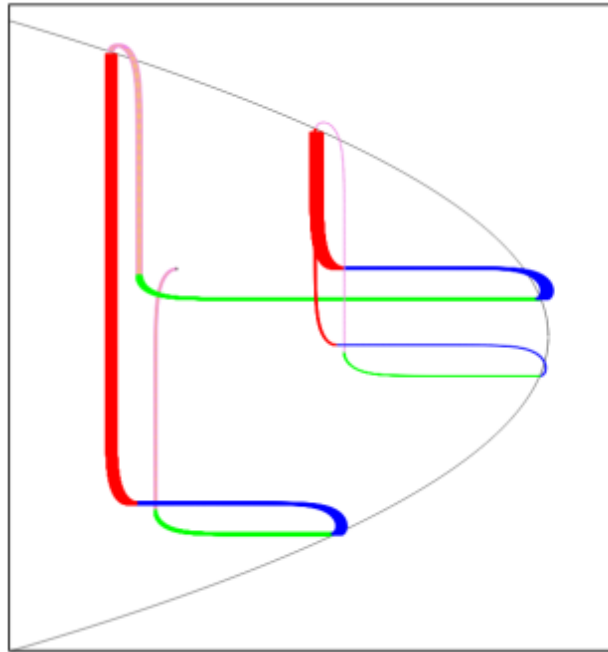


Figure 14: Convergence vers un cycle limite

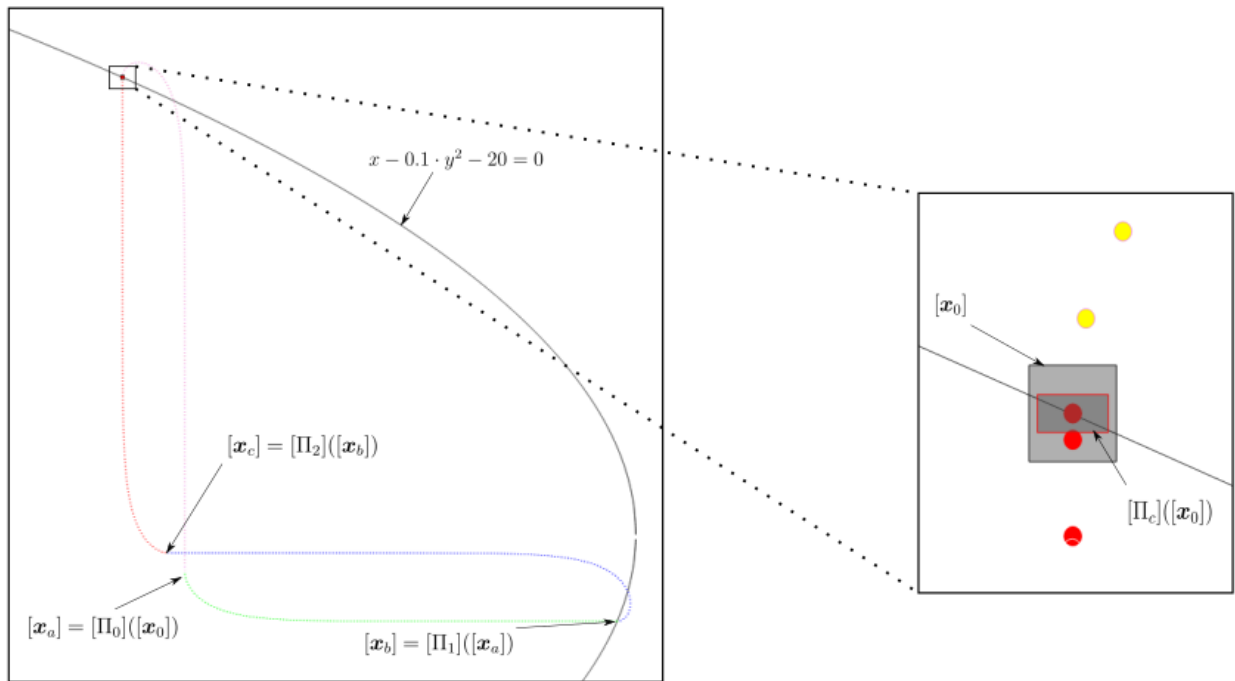


Figure 15: Preuve de la stabilité du cycle limite

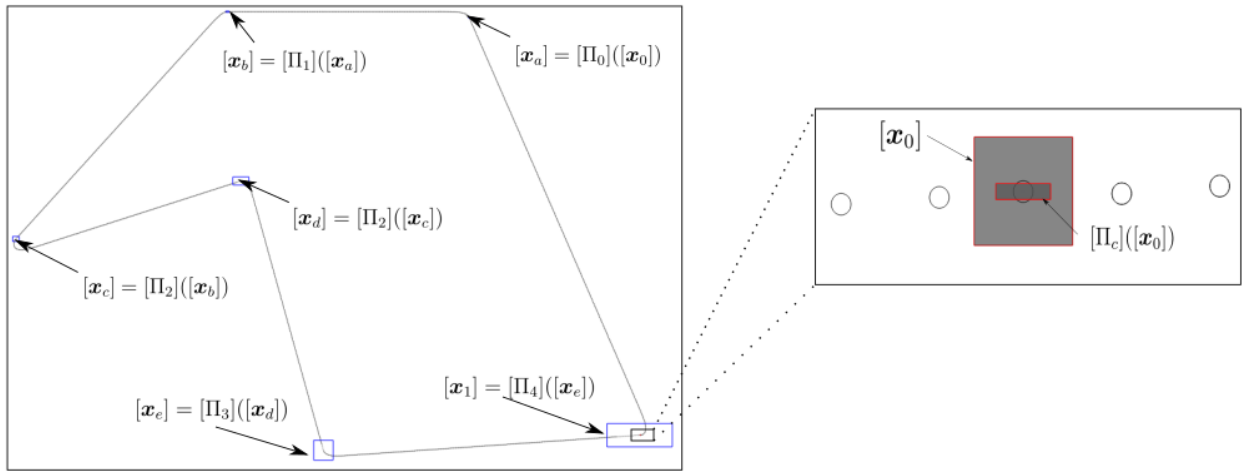


Figure 16: Preuve de stabilité du tour de l'île

5 Résultats pratiques

Notre objectif est la réalisation du tour de l'île des Morts grâce à des rebonds sur isobathes en convergeant vers un cycle stable. Malgré la réussite de la théorie, on ne peut envoyer le *Riptide* en mer sans effectuer quelques tests préliminaires.

Reprenons notre exemple du cycle L et posons nous dans le cas $g(x, y) = d(x, y)$. On désire donc effectuer des tests afin d'obtenir les résultats figures 14 et 15. Les tests ont été réalisés au niveau du port de Auberlac'h. À deux reprises, le *Riptide* a exécuté une dizaine de cycle.



Figure 17: Localisation du port de l'Auberlac'h

Malgré la bonne exécution des rebonds, le nombre de cycle effectué n'a pas permis la convergence vers une trajectoire périodique. La figure 18 reproduit les résultats du test en montrant les points importants du cycle. Cette reproduction a été réalisé grâce à des vidéos suivant le *Riptide* et en se basant sur des éléments fixes de l'environnement comme des boués. En vert, le premier rebond exécuté, le gris le dernier rebond exécuté. Les cases jaunes représentent les rebonds lorsque $g(x, y) = 0$, les cases rouges représentent les rebonds lorsque $d(x, y) = 0$. Les cases blanches représentent les rebonds lorsque l'horloge interne atteint une certaine valeur. (ici, 4s après le rebonds rouge, 8s après le rebond jaune). On remarque que les rebonds ne se produisent pas exactement au même niveau et qu'il est assez difficile de déterminer l'isobathe utilisée à partir de ces résultats.

Plusieurs causes possibles peuvent expliquer cet échec:

1. Le fond marin est relativement plat. Il est donc assez difficile de définir une ligne de niveau. Vu que le *Riptide* prend des mesures discrètes avec son écho-sondeur, il est probable qu'il rate la profondeur désirée et qu'il fasse le rebond plus loin. De plus, il est très probable que la profondeur visée ne soit pas unique, c'est à dire que le fond n'est pas strictement monotone.
2. Les tests étaient réalisés sur une longue durée. Entre le début et la fin de chaque mission, une différence de profondeur de 50cm à 1m peut se produire (dépendant du

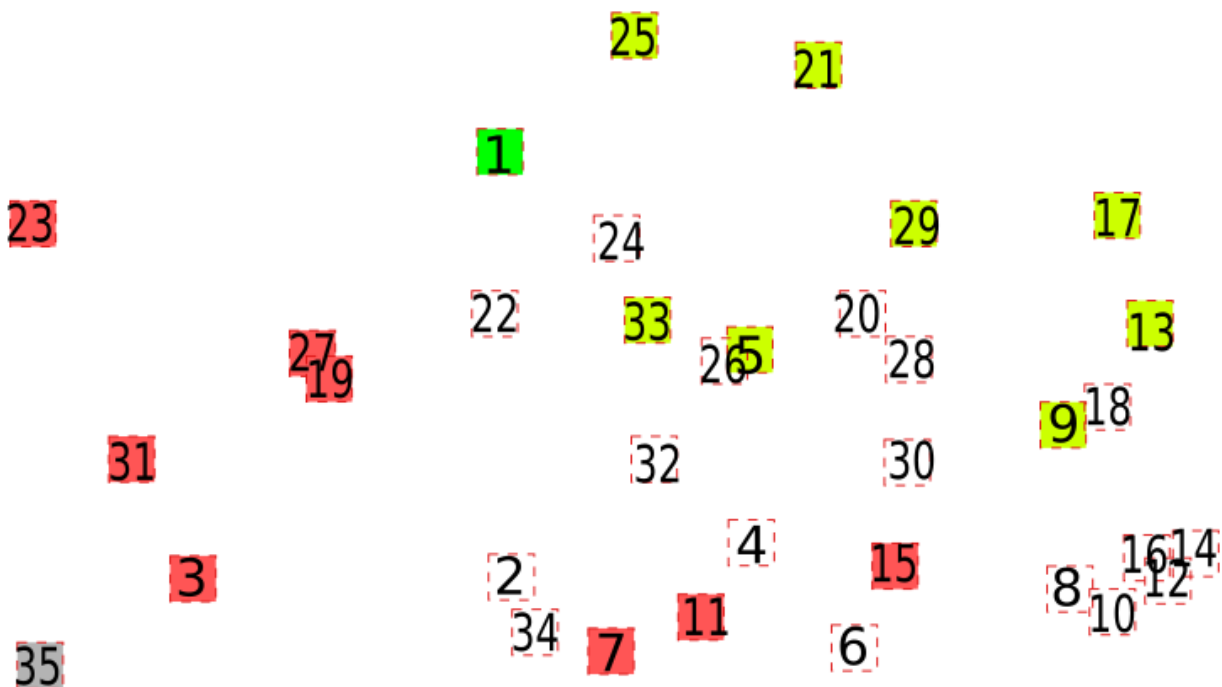


Figure 18: Points clés des cycles

coefficient de la marée le jour du test). Ce changement fausse considérablement la mission (vu que l'isobathe n'est pas la même).

3. Une phase de remontée à la surface est présente dans la mission. La distance parcourue en surface n'est pas la même car elle varie selon la courant d'eau. Cette variation déstabilise le cycle.

Néanmoins, certains résultats restent satisfaisants et nous rapprochent de notre objectif:

- Les tests de rebonds ont déjà été effectués dans un autre environnement sans réussite (au niveau de la Penfeld). Une hypothèse expliquant cet échec est l'incompatibilité du fond avec l'écho-sondeur. En effet, les expériences ont été réalisées sur un fond vaseux. Cette trajectoire, bien que non concluante, nous a permis d'observer le bon fonctionnement de l'écho-sondeur. Même si on ne connaît pas la bathymétrie, on peut observer que les valeurs mesurées par le *Riptide* correspondent à ce qu'on peut voir réellement.
- Les grandes distances parcourues par le *Riptide* en mer nous ont permis d'obtenir des données importantes concernant les contrôleurs implémentés et de valider les gains choisis.

6 Vers un tour plus réaliste

Dans les parties précédentes, on a réussi à prouver le tour d'une île dans un cas particulier: une bathymétrie exprimée par la somme de quatre gaussiennes. Ainsi, on approxime chaque isobathe par une ellipse. Le cas global n'est pas intéressant, donc, ce qu'on réalise réellement est une approximation locale centrée autour de la zone de rebond.

Le défaut de cette preuve est sa non-généralisation. En effet, il est rare qu'on puisse exprimer une bathymétrie de cette façon.

On peut alors se dire qu'on a englobé l'isobathe en question dans un tube dont la forme est une ellipse. Une question se pose alors: Si on réussit le rebond sur chaque borne du tube, est-on certain de le réussir sur l'isobathe réelle?

Considérons par exemple la situation suivante, figure 19.

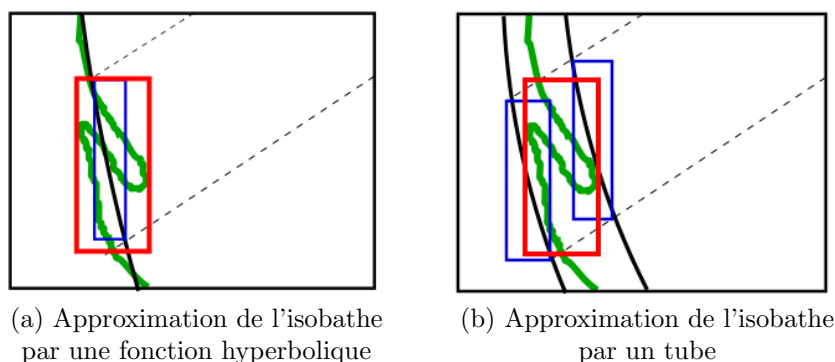


Figure 19: Isobathe et approximations

En rouge, l'intervalle d'incertitude réel, en bleu, les intervalles d'incertitude obtenus par approximation de l'isobathe. On remarque que la boîte rouge est plus large que les boîtes bleues. Ainsi, il y a différentes possibilités qu'on ne prend pas en compte.

Une première approche possible consiste à considérer le cas de l'approximation par tube et de prendre comme intervalle image l'union des deux boîtes bleues. Néanmoins, comme mentionné à la fin de la partie 2.2.4, la forme centrée perd son intérêt pour des intervalles larges. Par ailleurs, cette approche nécessite l'obtention de la bathymétrie réelle. Différentes méthodes existent, adaptées à certaines profondeurs et avec des résolutions distinctes, [3].

La deuxième approche, celle qu'on a retenue et qu'on essaie de développer, est la définition d'une nouvelle carte bathymétrique. Dans ce qui suit, on va présenter le travail réalisé, les idées développées, ainsi que les éléments qui restent à explorer.

6.1 Carte d'intervalles

Dans le but de travailler avec une bathymétrie correcte, et considérant l'incapacité à produire des mesures continues avec une précision de l'ordre du millimètre (dans le domaine qui nous

intéresse), on se pose la question de travailler avec des cartes d'intervalles. Ce qu'on désire posséder est donc une carte divisée en pavés qui ne se recouvrent pas. Chaque pavé contiendra des intervalles de valeurs utiles. Pour une carte bathymétrique, on peut demander que chaque pavé contienne un intervalle des lignes de niveau, ainsi que des intervalles des gradients.

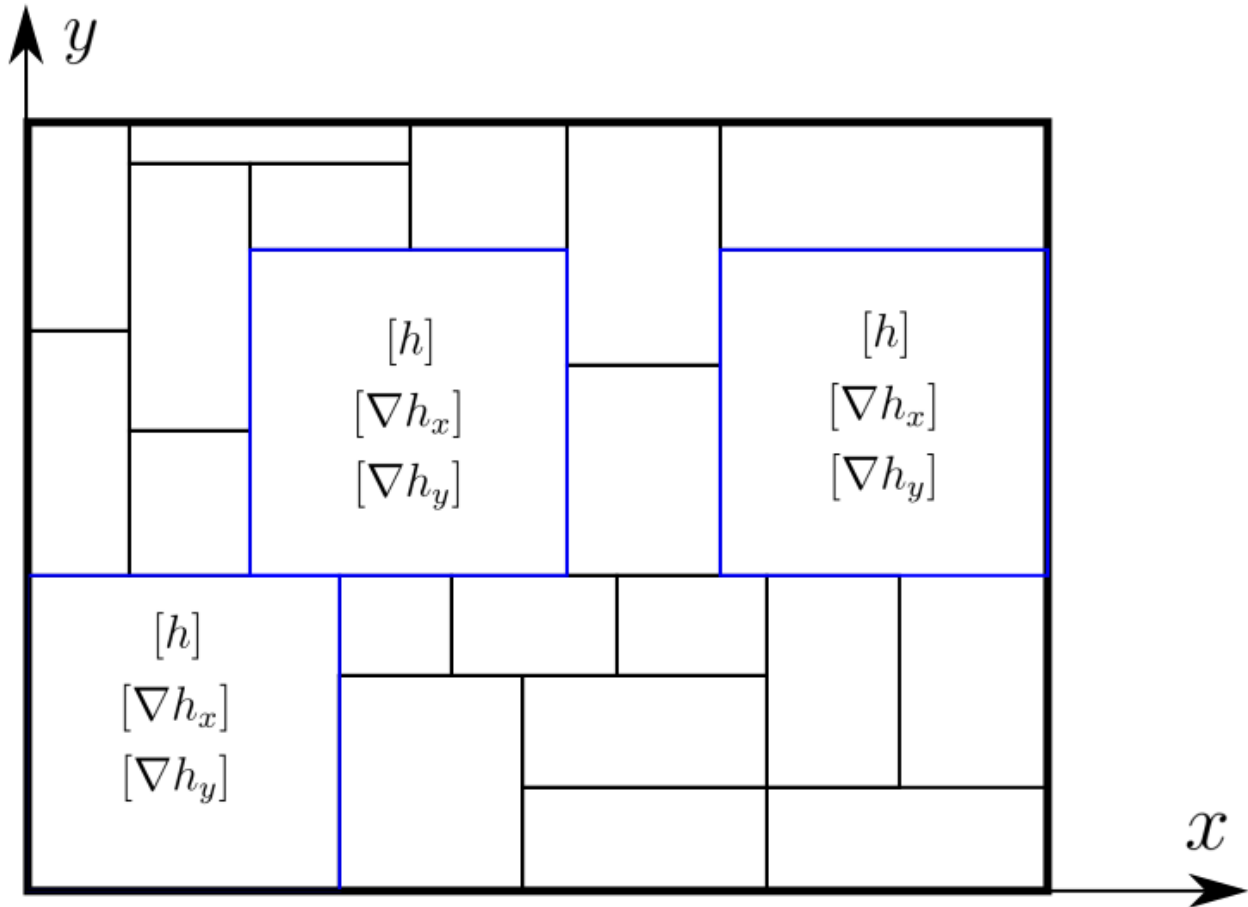


Figure 20: Exemple de carte d'intervalles

Ces données sont suffisantes pour continuer à travailler avec les surfaces de Poincaré et l'outil CAPD. Montrons, dans un cas très simple les résultats qu'on obtient avec une telle carte.

Exemple 6.1. Prenons le cas d'un véhicule qui se déplace sur un seul axe, par exemple l'axe x . Considérons la surface de Poincaré $x = 5$. Ajoutons des incertitudes à cette surface en pavant notre espace 2D sous la forme d'une carte d'intervalles, figure 21.

Dans ce cas, on connaît les dérivées de la surface de Poincaré selon x et selon y , ce qui rend l'exemple plus simple à traiter.

Dépendant de la précision de la carte (*width*) et de l'incertitude sur la position initiale du véhicule, on peut obtenir différents résultats. La figure 22 présente ces résultats.

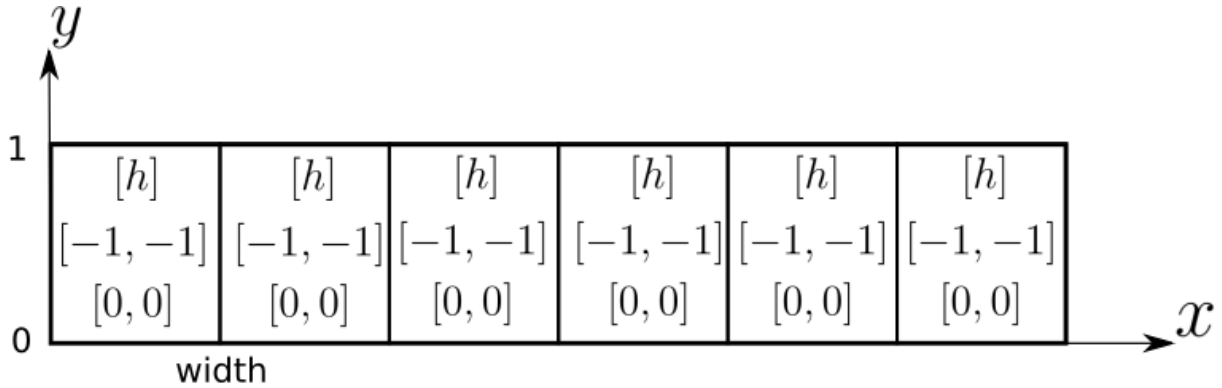


Figure 21: Exemple de pavage

On remarque d'un côté que la précision de la bathymétrie joue beaucoup dans nos résultats. Dans le cas où l'incertitude sur la position est de 10^{-3} , c'est à dire que $[x_0] = [4.499, 4.501]$, et une précision bathymétrique de 10^{-1} on obtient une image $[x] = [4.899, 5.101]$. Ainsi, l'incertitude est multiplier par 200. Dans le cas où la précision est de 10^{-3} , on obtient une image $[x] = [4.998, 5.00201]$. L'incertitude a juste doublé. D'un autre côté, l'incertitude sur la position initiale influe aussi sur les résultats. Pour une incertitude 10^{-1} , augmenter la précision bathymétrique dans ce cas n'a pas aboutit a des résultats satisfaisants.

On en conclut que cette méthode fonctionne, et est plus efficace pour des petites incertitudes de positions.

6.2 Travail futur

La carte d'intervalles a le potentiel de résoudre notre problème d'absence de bathymétrie réelle. Néanmoins, je n'ai pas encore eu le temps de le développer suffisamment pour prouver le tour de l'île des Morts. Ainsi, durant le temps qu'il me reste, j'envisage de:

- générer des cartes d'intervalles bathymétriques aléatoires. Ceci a pour but de regarder des exemples différents et d'étudier la stabilité du tour de l'île avec une carte contenant les données désirées. Pour cela, je compte utiliser les fonctions gaussiennes pour le relief général et rajouter un bruit de Perlin afin de rendre le fond marin plus naturel.
- trouver une méthode d'obtenir des cartes d'intervalles à partir des cartes bathymétriques existantes afin de prouver le tour de l'île des Morts sans avoir à attendre une nouvelle carte,
- exécuter le tour de l'île.

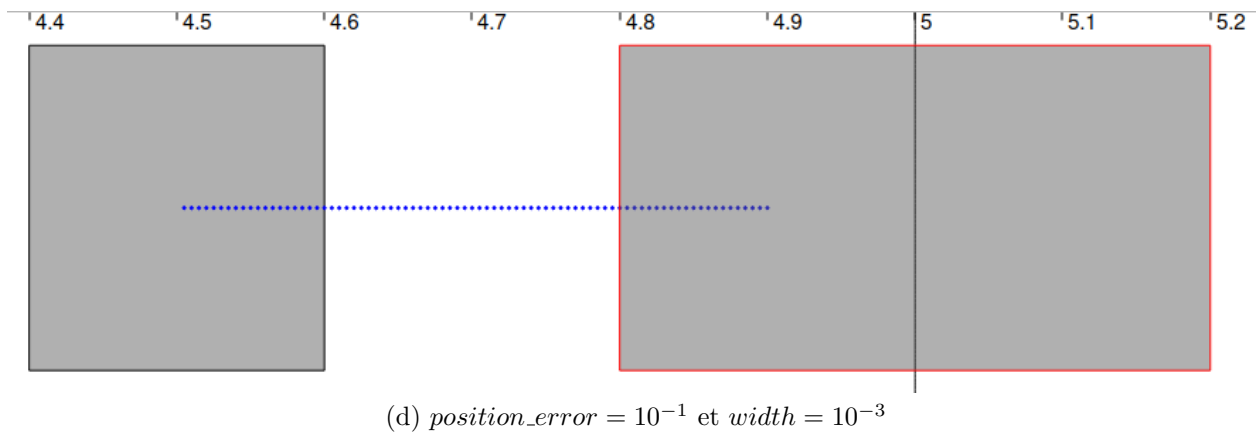
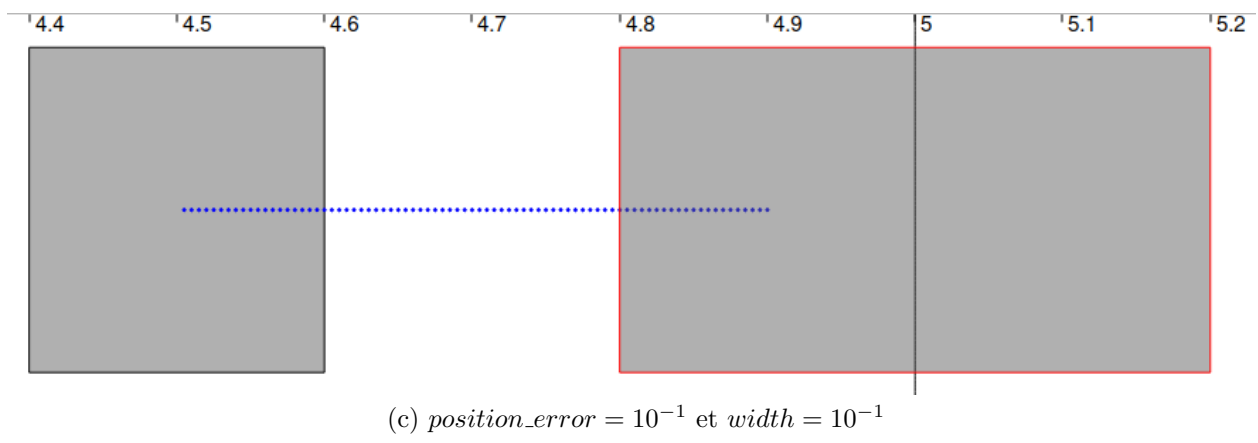
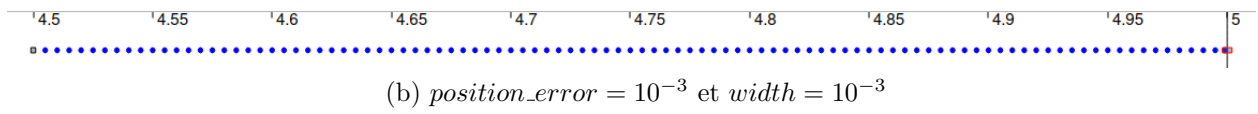
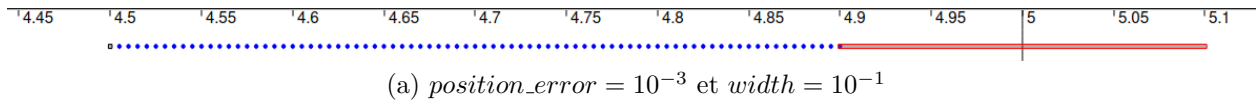


Figure 22: Résultats avec différentes précisions de bathymétrie

References

- [1] A. Bourgois. “Safe and collaborative autonomous underwater docking”. In: (2021), p. 182.
- [2] M. S. Branicky, V. S. Borkar, and S. K. Mitter. “A unified frame-work for hybrid control: model and optimal control theory”. In: (1998), p. 15.
- [3] H. M. Dierssen and A. E. Theberge Jr. “Bathymetry: Assessing Methods”. In: (2014), p. 8.
- [4] R. Ghaffarivardavagh et al. “Underwater Backscatter Localization: Toward a Battery-Free Underwater GPS”. In: (2020), p. 7.
- [5] W. P. M. H. Heemels, D. Lehmann J. Lunze, and B. De Schutter. “Introduction to hybrid systems”. In: (2009), p. 28.
- [6] L. Jaulin. *Underwater navigation with stable cycles*. Mar. 2020.
- [7] L. Jaulin et al. “Computing capture tubes”. In: (2014), p. 16.
- [8] K. H. Johansson, J. Lygeros, and S. Sastry. “Modeling of hybrid systems”. In: (), p. 25.
- [9] U. Kulisch, R. Lohner, and A. Facius. *On the Ubiquity of the Wrapping Effect in the Computation of Error Bounds*. 2001.
- [10] M. Brin and G. Stuck. *Introduction To Dynamical Systems*. cambridge univercity press, 2002.
- [11] R. E. Moore. “Methods and Applications of Interval Analysis”. In: (1979).
- [12] *Navigateur GPS pour plongeur*. URL: <http://subteq.io/fr/#1475701022410-bbfec03a-a502>.
- [13] L. Paull et al. “AUV Navigation and Localization - A Review”. In: (2013), p. 25.
- [14] *Positively invariant set*. URL: https://en.wikipedia.org/wiki/Positively_invariant_set.
- [15] S. Putot. *Lecture notes in Modeling, Simulation and Reachability Analysis of Hybrid Systems*. Nov. 2020.
- [16] J. Solà, J. Deray, and D. Atchuthane. “A micro Lie theory for state estimation in robotics”. In: (2020), p. 17.
- [17] Xiaojuan Sun and Jinzhi Lei. “Limit Cycle”. In: *Encyclopedia of Systems Biology*. Ed. by Werner Dubitzky et al. New York, NY: Springer New York, 2013, pp. 1126–1127. ISBN: 978-1-4419-9863-7.
- [18] G. Teschl. *Ordinary differential equations and dynamical systems*. American Mathematical Soc., 2012.
- [19] W. Tucker. “The Lorenz attractor exists”. In: (1999).
- [20] *TWO DIMENSIONAL FLOWS - Lecture 5: Limit Cycles and Bifurcations*.
- [21] D. Wilczak. *Computer-assisted proofs in dynamics Part I: Topological methods*.

A Équations mécaniques

On modélise le Riptide par sa position $\mathbf{p}_r(t)$, sa vitesse $\mathbf{v}_r(t)$ et sa matrice de rotation (son orientation) $\mathbf{R}_r(t)$. On pose donc $\mathbf{X}(t) = (\mathbf{p}_r(t), \mathbf{R}_r(t), \mathbf{v}_r(t))$.

Remarque A.1. On rajoute l'indice \square_r aux vecteurs et matrices liés au Riptide. Ceci permettra, dans la suite, de différencier les paramètres du Riptides des autres paramètres utilisés. Néanmoins, cet indice sera omit occasionnellement en absence d'ambiguïté.

Le contrôle du Riptide revient donc au problème de contrôle suivant:

$$\dot{\mathbf{X}}(t) = f(t, \mathbf{X}(t), u(t))$$

où u décrit le contrôle et f décrit la dynamique du système. Dans ce qui suit, on s'abstiendra de marquer la variable t quand aucune confusion n'est présente. Ainsi, le problème se réécrit plus clairement:

$$\dot{\mathbf{X}} = f(\mathbf{X}, u) \tag{A.1}$$

Dans la suite de cette partie, on détaillera les différentes équations qui régissent le Riptide avant d'expliquer le contrôle mis au point.

Ces équations feront jouer différents paramètres physiques qu'on exprimera dans différents repères, afin de garder une visualisation simple et clair du problème. Ainsi, il est important d'identifier les repères utilisés.

On notera avec un indice \square^{world} les vecteurs exprimés dans le domaine monde, avec un indice \square^d les vecteurs exprimés dans le domaine désiré (plus précisément, le repère qu'on désire avoir). On omettra l'indice lorsqu'on exprime les vecteurs dans le domaine du Riptide, par soucis de lisibilité.

Remarque A.2. En ce qui concerne les matrices de rotations, elles seront toutes exprimées dans le repère du monde. Ainsi, on omettra l'indice du repère par soucis de lisibilité.

A.1 Équations cinématiques

Afin de déterminer la fonction f , il faut d'abord exprimer les dérivées des différents paramètres du Riptide.

On a, de façon direct:

$$\dot{\mathbf{p}}_r = \mathbf{v}_r^{world}$$

qu'on peut réécrire plus facilement:

$$\dot{\mathbf{p}}_r = \mathbf{R}_r(\phi, \theta, \psi) \cdot \mathbf{v}_r \tag{A.2}$$

On utilisant l'hypothèse d'absence de glissement latéral, Equation A.2 se réécrit:

$$\dot{\mathbf{p}}_r = \mathbf{R}_r(\phi, \theta, \psi) \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot v_r \tag{A.3}$$

De même,

$$\dot{\mathbf{v}}_r = \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r$$

ou, en écriture scalaire:

$$\dot{v}_r = a_{rx} \quad (\text{A.4})$$

En ce qui concerne la matrice de rotation \mathbf{R}_r , en se référant à la théorie de Lie [16], on obtient la succession d'équation suivante:

$$\begin{aligned} \mathbf{R}^T \mathbf{R} &= \mathbf{I} \\ \dot{\mathbf{R}}^T \mathbf{R} + \mathbf{R}^T \dot{\mathbf{R}} &= \mathbf{0} \\ \mathbf{R}^T \dot{\mathbf{R}} &= -\dot{\mathbf{R}}^T \mathbf{R} \\ \mathbf{R}^T \dot{\mathbf{R}} &= -(\mathbf{R}^T \dot{\mathbf{R}})^T \end{aligned}$$

On a donc que la matrice $\mathbf{R}^T \dot{\mathbf{R}}$ est anti-symétrique et la relation:

$$\dot{\mathbf{R}} = \mathbf{R} \cdot (\boldsymbol{\omega}_r^{world} \wedge) \quad (\text{A.5})$$

où

$$(\boldsymbol{\omega}_r \wedge) = \begin{pmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{pmatrix}$$

Remarque A.3. $\boldsymbol{\omega}_r^{world}$ représente le vecteur de rotation du robot. On peut réécrire ce vecteur sous la forme $\alpha \mathbf{n}$ où α est un angle et \mathbf{n} un vecteur unitaire.

Remarque A.4. lorsque $\boldsymbol{\omega}_r$ est constante, l'intégration de l'Equation A.5 se met sous la forme:

$$\mathbf{R}(t + dt) = \mathbf{R}(t) \cdot \expw(dt \cdot \boldsymbol{\omega}_r) \quad (\text{A.6})$$

La fonction \expw utilisée est celle décrite dans [16]. On peut la réécrire de la manière suivante:

$$\expw(\boldsymbol{\omega}) = \exp((\boldsymbol{\omega} \wedge))$$

À ce stade, on peut réécrire l'Equation A.1 de la manière suivante, afin d'obtenir l'équation cinématique du Riptide:

$$\begin{cases} \dot{\mathbf{p}}_r = \mathbf{R}_r(\phi, \theta, \psi) \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot v_r \\ \dot{\mathbf{R}}_r = \mathbf{R} \mathbf{R}_r \cdot (\boldsymbol{\omega}_r \wedge) \\ \dot{v}_r = a_{rx} \end{cases}$$

A.2 Équations dynamiques

À partir de l'équation cinématique, on va approfondir le modèle afin de faire sortir quelques paramètres physiques.

A.2.1 Accélération

Selon la deuxième loi de Newton, l'accélération du Riptide est produite suite à une force exercée sur ce dernier. Ainsi,

$$f_x = m_{total}a_{rx} = (m_r + m_{added})a_{rx}$$

où m_{added} correspond à la masse d'eau ajoutée. On a aussi:

$$m_{added} = \underbrace{k_a}_1 \underbrace{\rho_{water}V}_{m_r} = m_r$$

Ainsi,

$$f_x = 2m_r a_{rx}$$

Par ailleurs, la force f_x exercée est composée principalement de la poussée qu'exerce les hélices du Riptide et du frottement visqueux résultant du déplacement.

$$f_x = k_p \omega_{propeller}^2 - \frac{1}{2} c_x S_x \rho_{water} v$$

La première composante de la force représente la vitesse de rotation des hélices du Riptide. Ce terme est bien évidemment lié à la puissance du propulseur, c'est-à-dire, u_0 . De plus, $\omega_{propeller} \propto u_0$ ¹

Ainsi, on obtient l'équation suivante:

$$a_{rx} = p_1 u_0^2 - p_2 v_r^2 \quad (\text{A.7})$$

A.2.2 Rotation

La vitesse de rotation ω_r du Riptide est liée à la fois à la vitesse de celui-ci et à l'orientation de ces ailerons. Cette relation est exprimée ci-dessous, dans le repère du Riptide:

$$\mathbf{w}_r = v \underbrace{\begin{pmatrix} p_3 & 0 & 0 \\ 0 & p_4 & 0 \\ 0 & 0 & p_4 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & \sin(\frac{2\pi}{3}) & -\sin(\frac{2\pi}{3}) \\ -1 & \cos(\frac{2\pi}{3}) & \cos(\frac{2\pi}{3}) \end{pmatrix}}_{B(p_3, p_4)} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (\text{A.8})$$

où, rappelons-le, u_1 est l'angle pris par l'aileron du dessus, u_2 l'angle pris par l'aileron de droite et u_3 l'angle pris par l'aileron de gauche.

Ainsi, on obtient un système d'équation faisant jouer, entre autres, nos différents paramètres de contrôle u_0 , u_1 , u_2 et u_3 .

¹graphique "RPM at 10-20 V" - <https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/>

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R}(\phi, \theta, \psi) \cdot \begin{pmatrix} v \\ 0 \\ 0 \end{pmatrix} \\ \dot{\mathbf{R}}_r &= \mathbf{R}_r \cdot \left(\left(v \cdot \mathbf{B}(p_3, p_4) \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \right) \wedge \right) \\ \dot{v}_r &= p_1 u_0^2 - p_2 v^2 \end{cases} \quad (\text{A.9})$$

A.3 Contrôle

À partir du système d'équation précédent, on extrait nos paramètres de contrôle afin d'obtenir:

$$\begin{aligned} u_0 &= \sqrt{\frac{\overline{a_{rx}} + p_2 v^2}{p_1}} \\ \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} &= \frac{1}{v} \mathbf{B}^{-1}(p_3, p_4) \overline{\boldsymbol{\omega}}_r \end{aligned} \quad (\text{A.10})$$

où $\overline{a_{rx}}$ et $\overline{\boldsymbol{\omega}}_r$ sont respectivement l'accélération désirée et la vitesse de rotation désirée.

Il reste donc à déterminer la vitesse de rotation $\overline{\boldsymbol{\omega}}_r$ désirée à partir de l'orientation désirée. Notons \mathbf{R}_d la matrice de rotation désirée, exprimée dans le repère monde. La matrice erreur de rotation peut se mettre sous la forme:

$$\mathbf{E} = \mathbf{R}_d \mathbf{R}^T$$

qui se réécrit, on se reposant sur la théorie des groupe de Lie,

$$\mathbf{e}^{world} = \log w(\mathbf{R}_d \mathbf{R}^T)$$

On prend alors, comme erreur de rotation:

$$\mathbf{e} = \mathbf{R}^T \cdot \log w(\mathbf{R}_d \mathbf{R}^T) \quad (\text{A.11})$$

Sachant que $\frac{d}{dt} \log(R) = R^T \cdot \dot{R}$ [16], et en dérivant \mathbf{e}^{world} , on obtient:

$$\begin{aligned} \dot{\mathbf{e}}^{world} &= \wedge^{-1} \left(\mathbf{R} \mathbf{R}_d^T \cdot \left(\dot{\mathbf{R}}_d \mathbf{R}^T + \mathbf{R}_d \dot{\mathbf{R}}^T \right) \right) \\ &= \wedge^{-1} \left(\mathbf{R} \mathbf{R}_d^T \dot{\mathbf{R}}_d \mathbf{R}^T + \mathbf{R} \dot{\mathbf{R}}^T \right) \\ &= \wedge^{-1} \left(\mathbf{R} (\boldsymbol{\omega}_d^{world} \wedge) \mathbf{R}^T + \mathbf{R} \dot{\mathbf{R}}^T \right) \\ &= \wedge^{-1} \left((\mathbf{R} \boldsymbol{\omega}_d^{world} \wedge) + (\mathbf{R} (\mathbf{R}^T \dot{\mathbf{R}}) \mathbf{R})^T \right) \\ &= (\mathbf{R} \boldsymbol{\omega}_d^{world}) + \wedge^{-1} \left((\mathbf{R} (\boldsymbol{\omega}_r^{world} \wedge) \mathbf{R})^T \right) \\ &= \mathbf{R} \boldsymbol{\omega}_d^{world} - \mathbf{R} \boldsymbol{\omega}_r^{world} \end{aligned}$$

puis, en multipliant à gauche par \mathbf{R}^T , on obtient:

$$\dot{\mathbf{e}} = \boldsymbol{\omega}_d^{world} - \boldsymbol{\omega}_r^{world} \quad (\text{A.12})$$

Dans notre application, on va se restreindre à la première dérivée de l'erreur et utiliser un contrôleur proportionnelle tel quel:

$$\dot{e} + \alpha_0 e = 0 \quad (\text{A.13})$$

et on obtient:

$$\boldsymbol{\omega}_r^{world} = \boldsymbol{\omega}_d^{world} + \alpha_0 \mathbf{R}^T \cdot \text{logw}(\mathbf{R}_d \mathbf{R}^T) \quad (\text{A.14})$$

et on pose enfin: $\overline{\boldsymbol{\omega}}_r = \boldsymbol{\omega}_r^{world}$.

Afin d'assurer la stabilité du système, il faut que les racines du système A.13 soient négatives donc que α_0 soit positif.

Remarque A.5. Dans notre application, ω_d est nul. On effectue, on va contrôler le Riptide en lui fournissant une matrice de rotation désirée constante. Ainsi, $\dot{\mathbf{R}}_d = \mathbf{0}$.