

Contexte et objectif

Ce travail vise à localiser trois bateaux coopérant au sein d'une même flotte : un bateau-mère **MS0**, dont la position absolue (GPS) peut être connue, et deux éclaireurs **S1** et **S2** dépourvus de GPS.

Les éclaireurs peuvent ponctuellement mesurer, par acoustique, les distances inter-bateaux :

$$d_{01}, d_{12}, d_{20},$$

chaque mesure étant entachée d'une incertitude modélisée par des intervalles. Par ailleurs, chaque unité a connaissance, à tout instant, de l'enveloppe de la position de ses voisins sous la forme de boîtes englobantes (*bounding boxes*).

Ils disposent en outre de mesures instantanées de cap et de vitesse, elles aussi incertaines. En intégrant ces mesures entre deux instants, on obtient une estimation du déplacement total — avec son incertitude — procédé que l'on appelle le "dead-reckoning".

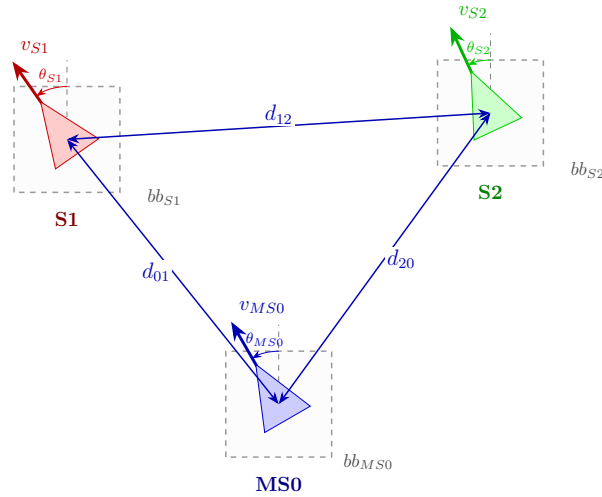


FIGURE 1 – Modélisation du problème

Dans notre modèle, nous ne considérons pas les instants intermédiaires entre deux mesures acoustiques : au moment de la prochaine mesure, on suppose disponible le déplacement total d depuis la dernière communication, où d est un vecteur d'intervalles représentant l'incertitude issue du dead-reckoning. Le cycle d'estimation est alors :

- **prédiction** : propager l'ensemble des positions possibles à l'aide de d (dead-reckoning) ;
- **correction** : réduire cet ensemble lorsque des mesures acoustiques et/ou GPS sont disponibles, en exploitant les distances mesurées ainsi que les boîtes englobantes des voisins, via l'application de contracteurs adaptés.

Prédiction par contracteurs équivalents

Un contracteur réduit un ensemble de positions possibles en appliquant une contrainte (par exemple une mesure de distance ou une boîte GPS). On peut maintenir, tout au long du processus, un contracteur équivalent représentant les positions compatibles avec les contraintes accumulées. En translatant et en intersectant ce contracteur au fil du temps, puis en le pavant (par exemple via SIVIA) lorsqu'on souhaite obtenir explicitement l'ensemble des positions, on récupère l'enveloppe des solutions compatibles — d'où l'appellation de "contracteur équivalent".

Considérons le contracteur équivalent $C_{S1}(t-1) \subset \mathbb{R}^2$ du bateau **S1** à l'instant $t-1$. Le déplacement entre $t-1$ et t est modélisé par

$$d = \begin{pmatrix} dx \\ dy \end{pmatrix},$$

et la fonction arrière (utilisée dans le code comme `f_back`) est

$$f_{\text{back}} : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad f_{\text{back}}(x) = x - d.$$

Dans notre modèle, dx et dy sont des intervalles (par exemple $dx = [\underline{dx}, \overline{dx}]$) : $d = (dx, dy)$ est le vecteur d'intervalles représentant le déplacement relatif. Pour un point réel $x \in \mathbb{R}^2$ et un vecteur d'intervalles d , l'expression $x - d$ s'interprète composante par composante dans l'arithmétique d'intervalles ; par exemple

$$x - d = [x - \overline{dx}, x - \underline{dx}].$$

Ainsi $x - d$ combine translation (par la valeur moyenne du déplacement) et inflation (par l'étendue des intervalles). La préimage s'écrit

$$f_{\text{back}}^{-1}(C) = \{x \in \mathbb{R}^2 \mid x - d \in C\},$$

et ainsi nous avons la prédiction de l'ensemble des positions à l'instant t à partir de l'ensemble à l'instant $t-1$:

$$C_{S1}(t) = f_{\text{back}}^{-1}(C_{S1}(t-1)),$$

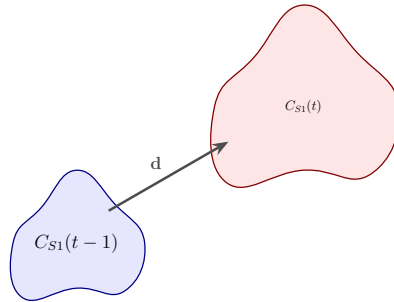


FIGURE 2 – Prédiction : translation et inflation de l'ensemble des positions.

En termes de contracteurs, la prédiction s'effectue par

$$C_{S1}(t) = \text{CtcInverse}(f_{\text{back}}, C_{S1}(t-1)),$$

le contracteur inverse retournant l'ensemble des états à l'instant t dont l'image par f_{back} appartient à $C_{S1}(t-1)$. On pourra appliquer la même logique à C_{MS0} et C_{S2}

Ce calcul de prédiction par contracteur inverse est exact, mais peut être coûteux à maintenir récursivement et on observe dans la pratique qu'avoir une connaissance approximative de l'ensemble des positions à chaque instant est souvent suffisant pour la correction et c'est pourquoi on peut aussi opter pour une approche non récursive : au lieu de chaîner les contracteurs inverses, on peut simplement pavé $C(t-1)$ et utiliser le pavage comme antécédent pour la prédiction à l'instant t .

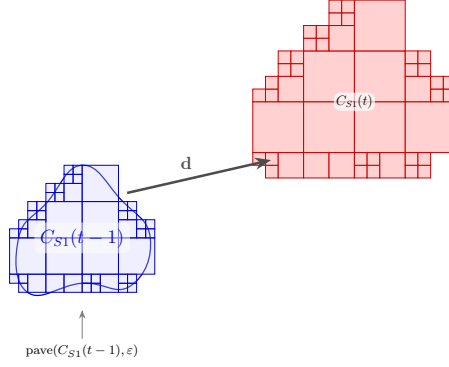


FIGURE 3 – Approche non récursive : pavage de $C_{S1}(t-1)$ et translation vers $C_{S1}(t)$.

Contracteurs de distance

Les données d'entrée pour la phase de correction sont les mesures de distances inter-bateaux (d_{01}, d_{12}, d_{20}) , modélisées par des intervalles pour inclure l'incertitude de mesure. De plus, chaque véhicule a connaissance de l'enveloppe de la position de ses voisins, transmise sous la forme de boîtes englobantes (`neighbor_box` de type `IntervalVector`).

Justification du modèle L'utilisation de la boîte englobante pour représenter la position du voisin est un choix pragmatique lié aux contraintes du milieu sous-marin. Transmettre un `IntervalVector` est très économe en bande passante, contrairement à l'envoi d'un pavage complet (l'ensemble précis des sous-pavés validés) qui, bien qu'offrant une géométrie plus fine et potentiellement une meilleure contraction, induirait une complexité de communication et de calcul prohibitive.

Formulation de la contrainte Soit $x = (x_0, x_1)$ la position du bateau considéré et $b = (b_x, b_y)$ une position appartenant à la boîte du voisin B_{neigh} . La fonction de distance euclidienne est définie par :

$$f_{\text{dist}}(x) = \sqrt{(x_0 - b_x)^2 + (x_1 - b_y)^2}.$$

Une mesure de distance reçue sous la forme d'un intervalle $[d_{\min}, d_{\max}]$ induit la contrainte d'appartenance suivante :

$$f_{\text{dist}}(x) \in [d_{\min}, d_{\max}].$$

En pratique, cette contrainte est résolue à l'aide d'un contracteur inverse, noté `CtcInverse`($f_{\text{dist}}, [d_{\min}, d_{\max}]$), qui réduit le domaine de x en ne conservant que les positions compatibles avec la mesure et la boîte du voisin.

Fusion des mesures Dans le cas où le bateau S_1 dispose de mesures vers deux voisins (par exemple d_{01} et d_{12}), nous construisons un contracteur pour chaque contrainte, notés respectivement $C_{d_{01}}$ et $C_{d_{12}}$. La correction globale s'obtient par l'intersection de ces contraintes avec l'ensemble des positions courantes $C_{S_1}(t)$:

$$C_{S_1}(t) \leftarrow C_{S_1}(t) \cap (C_{d_{01}} \cap C_{d_{12}}).$$

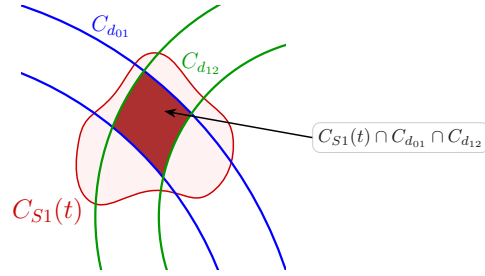


FIGURE 4 – Exemple de contraction par intersection : $C_{S1}(t) \cap C_{d01} \cap C_{d12}$.

Cette opération réduit l'ensemble des possibles en ne gardant que l'intersection entre l'estimation courante et les anneaux de distances définis par les voisins, comme illustré par la Figure 4.

Implémentation de l'algorithme de localisation

Cette section décrit, en pseudocode et en langage accessible, le déroulé de l'algorithme de prédiction/correction utilisé dans `fleet_prediction.py`. Nous utilisons les noms introduits précédemment : le bateau mère $MS0$ et les éclaireurs $S1$, $S2$. Le but est d'indiquer à quel moment se produisent les opérations principales : prédiction (dead-reckoning), construction des contracteurs de distance, intersection des contracteurs et émission des boîtes voisins.

Remarque : les opérations sur contracteurs sont indiquées de façon informelle, l'objectif est la clarté du flux plutôt que la précision. La version de l'algorithme sera légèrement différente car on écrira l'algorithme du point de vue d'un bateau donné pour montrer qu'il marche avec une quantité d'information limitée.

Vue d'ensemble pour $S1$

À chaque cycle d'échange (instant acoustique) : - récupérer les mesures locales (dead-reckoning : cap, vitesse) - prédire l'ensemble compatible au nouvel instant (translation/inflation) ; - collecter les mesures acoustiques disponibles (d_{01} , d_{12} , bb_{MS0} , bb_{S2}) et la boîte GPS si fournie ; - construire les contracteurs de distance pour chaque mesure et les combiner par intersection ; - corriger chaque contracteur équivalent local en l'intersectant avec le contracteur de distances correspondant ;

Pseudocode — version récursive

BoatPredictionRecursive (Vue locale S1)

```

1: Initialisation :  $C_{S1} \leftarrow \text{CtcWrapper}(\text{initial box})$ 

2: for chaque pas acoustique  $t$  do
3:    $\triangleright$  1. Phase de Prédiction (Dead-Reckoning)
4:    $d_{S1} \leftarrow \text{IntegrateAndFilter}(\text{cap}, \text{speed}, S1)$ 
5:   Définir  $f_{\text{back},S1}(x) \leftarrow x - d_{S1}$ 
6:    $C_{S1} \leftarrow \text{CtcInverse}(f_{\text{back},S1}, C_{S1})$ 

7:    $\triangleright$  2. Phase de Communication
8:   Envoyer  $bb_{S1}$  (IntervalVector) aux voisins
9:   Recevoir  $bb_{MS0}, bb_{S2}$ 
10:  Lire les distances acoustiques  $d_{01}, d_{12}$ 

11:   $\triangleright$  3. Phase de Correction (Contraction)
12:  % Construction des contracteurs de distance
13:   $C_{d01} \leftarrow \text{CtcInverse}(f_{\text{dist}}(\cdot, MS0), d_{01})$ 
14:   $C_{d12} \leftarrow \text{CtcInverse}(f_{\text{dist}}(\cdot, S2), d_{12})$ 

15:  % Intersection et mise à jour
16:   $C_{\text{dist},S1} \leftarrow C_{d01} \cap C_{d12}$ 
17:   $C_{S1} \leftarrow C_{S1} \cap C_{\text{dist},S1}$ 

18:  if gps box disponible then
19:     $C_{S1} \leftarrow C_{S1} \cap \text{CtcWrapper}(\text{gps\_box})$ 
20:  end if
21: end for

```

Pseudocode — version non-réursive (pavage)

BoatPredictionNonRecursive (Pavage)

```

1: Initialisation :  $C_{S1} \leftarrow \text{CtcWrapper}(\text{initial box})$ 
2:  $\text{pavage\_S1} \leftarrow \text{SIVIA}(C_{S1}) \triangleright \text{Pavage initial}$ 

3: for chaque pas acoustique  $t$  do
4:    $\triangleright 1. \text{Prédiction via Pavage}$ 
5:    $d_{S1} \leftarrow \text{IntegrateAndFilter}(S1)$ 
6:   Définir  $f_{\text{back},S1}(x) \leftarrow x - d_{S1}$ 
7:    $C_{S1} \leftarrow \text{CtcInverse}(f_{\text{back},S1}, \text{CtcWrapper}(\text{pavage\_S1}))$ 

8:    $\triangleright 2. \text{Communication \& Lecture}$ 
9:   Envoyer  $bb_{S1}$  ; Recevoir  $bb_{MS0}, bb_{S2}$  ; Lire  $d_{01}, d_{12}$ 

10:   $\triangleright 3. \text{Correction}$ 
11:   $C_{d01} \leftarrow \text{CtcInverse}(f_{\text{dist}}(\cdot, MS0), d_{01})$ 
12:   $C_{d12} \leftarrow \text{CtcInverse}(f_{\text{dist}}(\cdot, S2), d_{12})$ 

13:   $C_{\text{dist},S1} \leftarrow C_{d01} \cap C_{d12}$ 
14:   $C_{S1} \leftarrow C_{S1} \cap C_{\text{dist},S1}$ 
15:  if gps box disponible then
16:     $C_{S1} \leftarrow C_{S1} \cap \text{CtcWrapper}(\text{gps\_box})$ 
17:  end if

18:   $\triangleright 4. \text{Re-Pavage (Spécifique non-récuratif)}$ 
19:   $\text{pavage\_S1} \leftarrow \text{SIVIA}(C_{S1}) \triangleright \text{Mise à jour fine de la forme}$ 
20: end for

```