



**HAL**  
open science

## Localisation et suivi robustes d'un robot mobile grâce à l'analyse par intervalles

Michel Kieffer, Luc Jaulin, Eric Walter, Dominique Meizel

► **To cite this version:**

Michel Kieffer, Luc Jaulin, Eric Walter, Dominique Meizel. Localisation et suivi robustes d'un robot mobile grâce à l'analyse par intervalles. *Traitement du Signal*, 2000, 17 (3), pp.207-219. hal-00844925

**HAL Id: hal-00844925**

**<https://hal.science/hal-00844925>**

Submitted on 16 Jul 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Localisation et suivi robustes d'un robot mobile

grâce à l'analyse par intervalles

## Robust Robot Localization and Tracking

using Interval Analysis

Michel KIEFFER<sup>1</sup>, Luc JAULIN<sup>1,2</sup>, Éric WALTER<sup>1</sup> et Dominique MEIZEL<sup>3</sup>

<sup>1</sup>Laboratoire des Signaux et Systèmes, CNRS — Supélec — Université Paris-Sud

Plateau de Moulon, 91192 Gif-sur-Yvette, France

{jaulin, kieffer, walter}@lss.supelec.fr

<sup>2</sup>en délégation du Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers

<sup>3</sup>HEUDIASYC, CNRS, Université de Technologie de Compiègne,

B.P. 20529, 60205 Compiègne, France

meizel@hds.univ-compiegne.fr

29 février 2000

**Résumé:** Cet article présente une solution originale au problème de localisation et de suivi d'un robot mobile à partir de données télémétriques fournies par des capteurs à ultrasons. Le problème est traité dans un contexte d'erreurs de mesure et d'incertitudes sur l'état bornées. Grâce à une technique d'estimation de paramètres et d'état utilisant l'analyse par intervalles, les mesures provenant de 24 capteurs à ultrasons sont fusionnées en tenant compte de la présence inévitable de nombreuses données aberrantes. Un *ensemble* contenant la configuration (position et orientation) du robot est fourni à chaque instant. Un exemple simulé réaliste est présenté en détail pour illustrer les propriétés de la technique.

**Abstract:** This paper presents a new solution to the problem of localizing and tracking a robot from distance

measurements provided by on-board ultrasonic sensors. The measurement errors and state perturbations are assumed to be bounded. A parameter and state estimation technique based on interval analysis is employed to compute, as a function of time, a *set* guaranteed to contain all robot configurations (position and orientation) that are consistent with the measurements provided by 24 ultrasonic sensors, given the bounds on the acceptable errors. The unavoidable presence of many outliers is taken into account to provide robust estimates. A realistic simulated example is treated in detail to illustrate the properties of the technique.

**Mots-clefs:** calcul par intervalles, capteurs à ultrasons, données aberrantes, estimation de paramètres, estimation d'état, fusion de données, robotique, robustesse.

**Keywords:** data fusion, interval computation, outliers, parameter estimation, robotics, robustness, state estimation, ultrasonic sensors.

## 1 Introduction

La localisation et le suivi d'un robot mobile tel que celui représenté sur la figure 1 à partir de données télémétriques rentrent dans le cadre de l'estimation de paramètres et d'état de dispositifs à partir de mesures expérimentales. Tout au long de l'étude, nous ferons l'hypothèse que les erreurs de mesure et les incertitudes sur l'état sont bornées, tout en admettant que des données expérimentales puissent ne pas vérifier cette hypothèse.

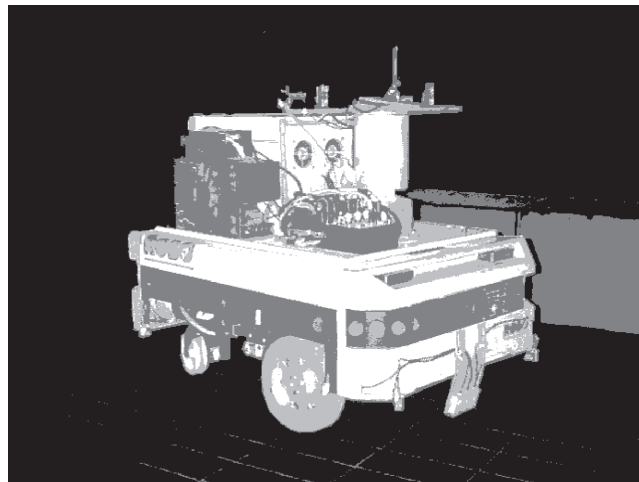


Figure 1: Le robot mobile ROButer

Dans un premier temps, quelques résultats concernant l'estimation paramétrique grâce à l'analyse par intervalles seront rappelés au paragraphe 2. Une attention particulière sera accordée au traitement des mesures aberrantes, c'est-à-

dire ne satisfaisant pas les hypothèses sur les erreurs. En effet, celle-ci sont malheureusement pratiquement inévitables lors de l'utilisation de mesures télémétriques fournies par des capteurs à ultrasons. Un exemple de localisation sera détaillé au paragraphe 3. Il s'agira de déterminer la position et l'orientation du robot dans un environnement de surface au sol plane. Le paragraphe 4 présentera un algorithme ensembliste d'estimation d'état pour des systèmes non-linéaires [12]. Comme pour le filtre de Kalman, une phase de prédiction à partir des informations passées alterne avec une phase de correction pour la prise en compte des nouvelles mesures. Cette étape de correction met en œuvre une technique d'estimation robuste décrite dans [13] et [14]. L'application de cette technique au suivi du robot lors de son déplacement sera ensuite détaillée.

## 2 Estimation paramétrique à erreurs bornées

L'objectif de l'estimation paramétrique est de déterminer le vecteur des paramètres  $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$  d'un modèle du dispositif étudié à partir de mesures expérimentales, le but étant d'obtenir un modèle dont le comportement soit optimal ou acceptable en un sens à définir.

### 2.1 Notations

Soit  $y(k)$  le résultat de la  $k$ -ième mesure. Dans le cas du robot,  $k$  représente le numéro d'un capteur. En supposant que la sortie  $y(k)$  peut être décrite par le modèle  $y_m(k, \mathbf{x})$ , nous cherchons à estimer le vecteur des paramètres  $\mathbf{x} \in \mathcal{X}$  de ce modèle. L'erreur entre les sorties mesurées et prédites est

$$e(k, \mathbf{x}) = y(k) - y_m(k, \mathbf{x}). \quad (1)$$

Cette erreur correspond d'une part à notre aptitude limitée à modéliser les phénomènes physiques intervenant dans le processus à décrire (on parle d'*erreur structurelle*) et d'autre part aux *erreurs (ou bruits) de mesure*, dues aux précisions limitées des appareils de mesure, aux diverses perturbations, etc.

Parmi les nombreuses méthodes d'identification disponibles (voir [28], par exemple), beaucoup procèdent par optimisation d'un critère. Tout argument  $\hat{\mathbf{x}}$  de l'optimum de ce critère correspond à une valeur *optimale* des paramètres  $\mathbf{x}$ , *au sens du critère* retenu, car la valeur de  $\hat{\mathbf{x}}$  peut varier fortement suivant la fonction coût choisie. Outre le fait que  $\hat{\mathbf{x}}$  n'est pas forcément unique, son obtention est en général difficile. De plus, l'évaluation de l'incertitude avec laquelle  $\hat{\mathbf{x}}$  est estimé reste un problème largement ouvert.

Ici, nous allons rechercher l'ensemble des valeurs de  $\mathbf{x}$  *compatibles* avec les données au sens où toutes les erreurs sont

comprises entre les bornes prédéfinies. C'est le contexte d'*erreurs bornées* (voir [19], [22], [23], [28] et [27]). L'estimée de  $\mathbf{x}$  ne sera alors plus ponctuelle, mais constituée d'un ensemble de valeurs.

## 2.2 Estimation à erreurs bornées

Par hypothèse,  $e(k, \mathbf{x})$  appartient à l'intervalle *connu*  $[e(k)] = [\underline{e(k)}, \overline{e(k)}]$ . Une valeur  $\mathbf{x}$  du vecteur des paramètres sera compatible avec la  $k$ -ième mesure si  $y_m(k, \mathbf{x}) \in [y(k)] = [y(k) - \underline{e(k)}, y(k) + \overline{e(k)}]$ . Cette propriété correspond à un *test* sur  $\mathbf{x}$ , qui permet de définir l'ensemble  $\mathcal{S}_k$  des valeurs du vecteur des paramètres compatibles avec la  $k$ -ième mesure comme

$$\mathcal{S}_k = \{ \mathbf{x} \in \mathcal{X} \mid t_k(\mathbf{x}) = \text{vrai} \}, \quad (2)$$

avec

$$t_k(\mathbf{x}) = (y_m(k, \mathbf{x}) \in [y(k)]). \quad (3)$$

L'intersection des  $\mathcal{S}_k$ , pour  $k = 1, \dots, N$  définit l'ensemble  $\mathcal{S}$  des valeurs des paramètres compatibles avec toutes les mesures

$$\mathcal{S} = \bigcap_{k=1}^N \mathcal{S}_k = \{ \mathbf{x} \in \mathcal{X} \mid t(\mathbf{x}) = \text{vrai} \}, \quad (4)$$

avec

$$t(\mathbf{x}) = t_1(\mathbf{x}) \wedge \dots \wedge t_N(\mathbf{x}), \quad (5)$$

où  $\wedge$  représente l'opérateur logique ET (voir la figure 2, pour une situation où seules deux mesures sont disponibles).

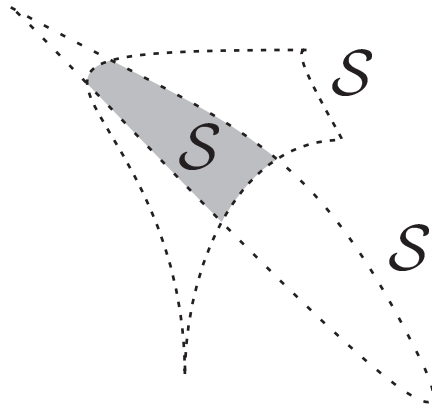


Figure 2: Ensemble des vecteurs de paramètres compatibles avec deux mesures

$\mathcal{S}$  n'est en général pas un singleton. Il peut être vide, ce qui signifie qu'il n'est pas possible de trouver de valeurs de  $\mathbf{x}$  satisfaisant (3) pour tout  $k$ , ou encore être un sous-ensemble de  $\mathbb{R}^p$ , éventuellement non connexe, auquel cas

plusieurs valeurs de paramètres très différentes les unes des autres conduisent à un modèle fournissant des résultats compatibles avec les données.

### 2.3 Traitement des données aberrantes

Un cas particulier où  $\mathcal{S}$  peut être vide correspond à la situation où les hypothèses sur les erreurs de mesure ne sont pas satisfaites pour toutes les valeurs de  $k$ ; certaines mesures sont alors dites *aberrantes*. Celles-ci sont dues par exemple à un dysfonctionnement temporaire du dispositif de mesure. Une parade à ce problème consiste à rechercher l'ensemble des paramètres qui sont compatibles avec le plus grand nombre possible de mesures.

Le test  $t(\mathbf{x})$  dans (4) doit alors être modifié. Définissons l'opérateur Booléen *et- $q$ -relaxé*, noté  $\bigoplus_q$ , tel que

$$\begin{aligned} \text{Soit } (b_1, b_2, \dots, b_n) &\in \{\text{vrai, faux}\}^n, \\ \bigoplus_q (b_1, b_2, \dots, b_n) = \text{vrai} &\iff \text{au moins } n - q \text{ des } b_i \text{ valent vrai.} \end{aligned}$$

Il est alors possible de tolérer jusqu'à  $q$  mesures aberrantes en caractérisant l'ensemble

$$\mathcal{S}_{r,q} = \{\mathbf{x} \in \mathcal{X} \mid t_{\text{robuste}}(\mathbf{x}, q) \text{ est vrai}\}, \quad (6)$$

où

$$\begin{aligned} t_{\text{robuste}}(\mathbf{x}, q) &= \bigoplus_q (t_1(\mathbf{x}), \dots, t_n(\mathbf{x})) \\ &= \bigoplus_{i=1}^n (t_i(\mathbf{x})). \end{aligned} \quad (7)$$

La figure 3 illustre une situation où une seule donnée aberrante est tolérée; l'ensemble solution  $\mathcal{S}_{r,1}$  est constitué ici de deux sous-ensembles disjoints.

Pour déterminer la valeur de  $q$ , une technique possible est de considérer au départ  $q = 0$ , ce qui revient à utiliser (4), et d'incrémenter  $q$  d'une unité tant que l'ensemble des configurations possibles reste vide. Cette technique correspond à l'estimateur OMNE (*Outlier Minimal Number Estimator*, voir [15], [24], et [26]), que nous mettons en œuvre de façon garantie (voir [10], [13] et [14]).

### 2.4 Inversion ensembliste et analyse par intervalles

La description de  $\mathcal{S}$  (ou  $\mathcal{S}_{r,q}$ ) peut être formulée comme un problème d'*inversion ensembliste*, résolu de manière approchée mais garantie grâce à l'analyse par intervalles et l'algorithme SIVIA (*Set Inversion Via Interval Analysis*, voir [9]). Cette technique rappelée brièvement ici permet de caractériser, de manière approchée mais avec une précision

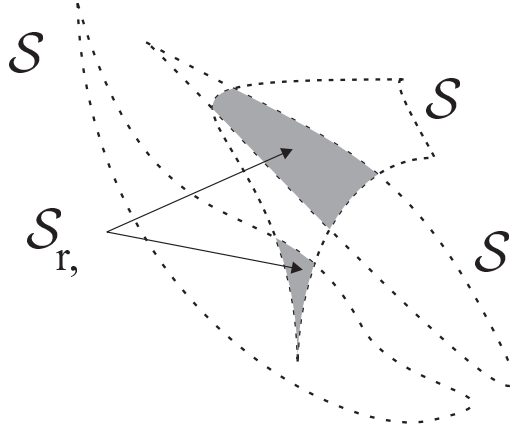


Figure 3: Estimation robuste avec trois données ; une seule donnée aberrante est tolérée

que l'on peut choisir, l'ensemble  $\mathcal{S}$  par un *sous-pavage*, c'est-à-dire par un ensemble constitué de l'union de vecteurs d'intervalles (ou *pavés*) disjoints. Grâce à l'analyse par intervalles, il est possible d'évaluer un test  $t(\cdot)$ , tel que (4), non plus sur des vecteurs de réels, mais sur des pavés (voir l'annexe 1). Pour un pavé  $[\mathbf{x}]$  donné, trois situations seront possibles :

- $t([\mathbf{x}])$  est faux ; il n'existe pas alors de vecteur dans  $[\mathbf{x}]$  qui soit compatible avec les mesures. Le pavé  $[\mathbf{x}]$  a une intersection vide avec  $\mathcal{S}$ .
- $t([\mathbf{x}])$  est vrai ; tous les vecteurs contenus dans  $[\mathbf{x}]$  sont donc acceptables. Le pavé  $[\mathbf{x}]$  est inclus dans  $\mathcal{S}$ .
- $t([\mathbf{x}])$  est indéterminé ; tout ou partie de  $[\mathbf{x}]$  peut éventuellement appartenir à  $\mathcal{S}$ . On peut alors découper le pavé *indéterminé*  $[\mathbf{x}]$  et appliquer le test  $t(\cdot)$  sur les sous-pavés obtenus.

SIVIA réalise un découpage ordonné de l'espace de recherche initial  $[\mathbf{x}_0]$ . Afin d'obtenir une solution en un temps fini, le découpage n'est plus effectué lorsque la taille d'un pavé indéterminé est inférieure à un facteur de précision  $\epsilon$ . L'ensemble solution  $\mathcal{S}$  est enfermé dans une union de pavés disjoints (ou *sous-pavage*)  $\widehat{\mathcal{S}}$  constitué de tous les pavés compatibles avec les mesures et des pavés indéterminés jugés suffisamment petits pour ne plus être découpés. Cette approximation *extérieure* peut être rendue aussi précise que souhaitée en réduisant  $\epsilon$ , et elle est *garantie* car toutes les valeurs du vecteur des paramètres  $\mathbf{x}$  compatibles avec les mesures sont incluses dans  $\widehat{\mathcal{S}}$ .

**Exemple 2.1** Soit  $\mathbf{x} \in \mathbb{R}^2$ , cherchons à caractériser  $\mathcal{S} \subset [-2, 2]^2$  défini par les tests suivants

$$t_1(\mathbf{x}) = \{x_1^2 + x_2^2 - 2 * \text{abs}(x_1) \in [0.7, 1]\},$$

$$t_2(\mathbf{x}) = \{x_1^2 + x_2^2 \in [0.7, 1]\}.$$

Si on applique SIVIA à ce problème avec un paramètre de précision  $\epsilon = 0.1$ , on obtient le sous-pavage représenté en gris foncé sur la figure 4. Les sous-pavages correspondant à  $\mathcal{S}_1$  et  $\mathcal{S}_2$  ont été représentés en gris clair.

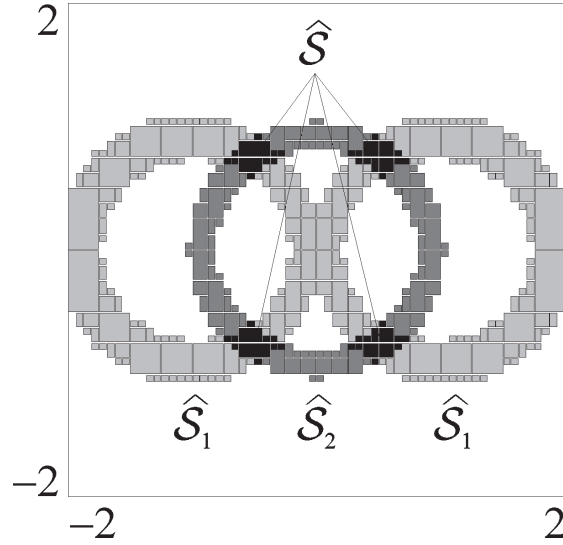


Figure 4: Utilisation de SIVIA pour l'exemple 2.1;  $\hat{\mathcal{S}}$  est en gris foncé

Quelques éléments sur la manière de déterminer la valeur d'un test sur un pavé sont donnés dans l'annexe 1.

### 3 Localisation statique d'un robot

L'objectif est de déterminer la position  $(x_c, y_c)$  et l'orientation  $\theta$  du repère  $\mathcal{R}$  attaché au robot dans le repère  $\mathcal{W}$  associé à son environnement. Le vecteur des paramètres  $\mathbf{x} = (x_c, y_c, \theta)^T$  à estimer est appelé la *configuration* du robot (voir la figure 5).

Les informations disponibles pour la localisation (carte de l'environnement et mesures de distance) seront décrites au paragraphe 3.1. Elles permettront de construire un test, décrit au paragraphe 3.2, que l'on pourra appliquer à une configuration donnée pour décider si elle est compatible avec les mesures. Ensuite, en appliquant les techniques intervalles présentées au paragraphe 2, il sera possible d'évaluer l'ensemble des configurations compatibles avec les informations disponibles, comme illustré au paragraphe 3.3.



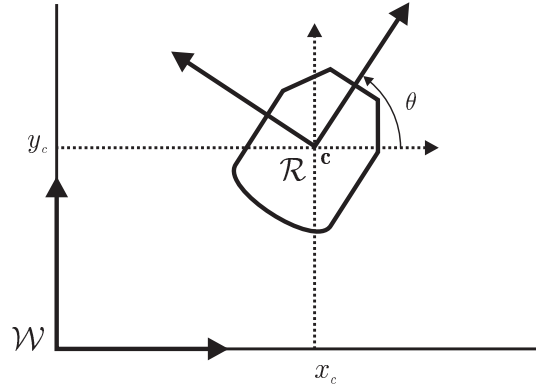


Figure 5: Configuration du robot

### 3.1 Informations disponibles

La carte  $\mathcal{M}$  de l'environnement du robot, constituée de  $n_o$  segments orientés, décrit les obstacles (murs, portes, piliers, *etc.*) entourant le robot

$$\mathcal{M} = \{[\mathbf{a}_j, \mathbf{b}_j] \mid j = 1, \dots, n_o\}. \quad (8)$$

Par convention, lors d'un déplacement de  $\mathbf{a}_j$  vers  $\mathbf{b}_j$ , la partie réfléchissante du segment est sur la gauche (la matière de l'obstacle est à droite). Une telle carte est représentée sur la figure 6 ; les obstacles sont en gris.

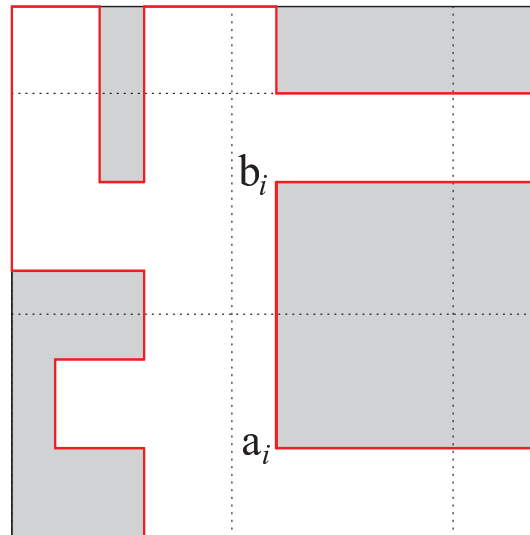


Figure 6: Carte de l'environnement du robot (les distances sont en m.)

Chaque capteur à ultrasons  $i$  émet une onde dans une direction donnée. Cette onde, dont on suppose qu'elle se propage à l'intérieur d'un *cône d'émission* de demi-ouverture  $\gamma_i$ , sera réfléchié par les éléments de l'environnement du

robot. Lorsque l'obstacle présente une surface irrégulière, la réflexion est à la fois spéculaire et diffuse (voir [8]). L'angle d'incidence sur un obstacle détermine si l'onde réfléchie est ou non directement reçue par le capteur. Pour un cône d'émission de demi-ouverture nulle, on peut définir un angle d'*incidence limite*  $\beta_j$  au-delà duquel l'onde réfléchie n'est plus perçue directement. Lorsque  $\gamma_i$  est non nul, l'angle d'incidence moyen du faisceau doit être inférieur à  $\beta_j + \gamma_i$ . La valeur de  $\beta_j$  dépend de l'état de surface de l'obstacle représenté par le  $j^{\text{ème}}$  segment de la carte. Cette valeur sera importante pour des surfaces rugueuses, et très faibles pour des surfaces lisses. Deux situations sont illustrées sur la figure 7.

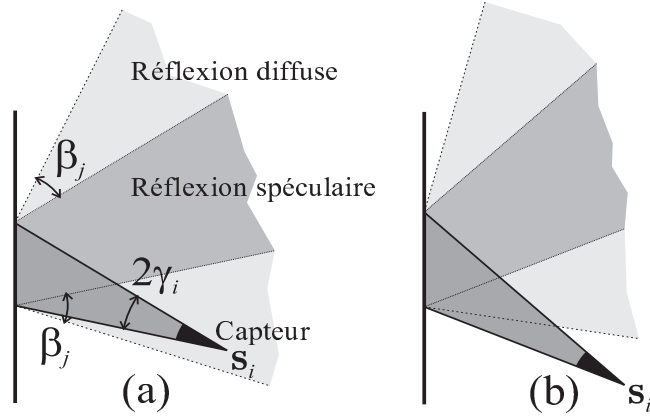


Figure 7: (a) : L'onde réfléchie est directement reçue par le capteur ; (b) : l'onde n'est pas reçue

Dans le cas d'une réception, le temps écoulé depuis l'émission est converti en une distance  $d_i$ , correspondant à la distance à l'obstacle le plus proche dans la direction d'émission du capteur. Les incertitudes de mesure sont supposées bornées, de ce fait, l'intervalle  $[d_i] = [(1 - \alpha_i) d_i, (1 + \alpha_i) d_i]$  devrait contenir la vraie distance à l'obstacle le plus proche dans le cône d'émission. Cependant, l'onde pourra être reçue après plusieurs réflexions, voire ne pas être reçue du tout. Dans ces conditions, l'interprétation de la mesure de distance n'est plus aussi facile, et les bornes sur les incertitudes perdent leur sens. Les mesures seront alors considérées comme aberrantes, par rapport à la compréhension qu'autorise notre modèle. La présence, malheureusement à peu près inévitable, de telles mesures aberrantes rend le problème de localisation particulièrement ardu.

L'ensemble des mesures peut être résumé par un *diagramme d'émission* (voir la figure 8). Chaque segment correspond à une distance  $d_i$  mesurée par un capteur. Un obstacle doit se trouver au moins en partie entre les deux arcs de cercle matérialisant l'incertitude relative de mesure  $\alpha_i$  et la demi-ouverture du cône d'émission  $\gamma_i$ .

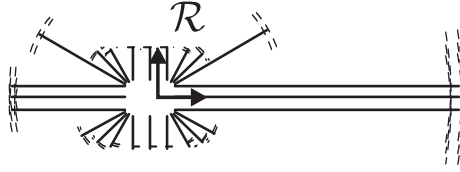


Figure 8: Diagramme d'émission

### 3.2 Test sur les configurations

On suppose disposer d'un *pavé* (ou vecteur d'intervalles) de recherche  $[\mathbf{x}]_0$  dans lequel se trouve la configuration réelle du robot. Ce pavé peut être très grand si nécessaire. L'objectif est de déterminer le sous-ensemble  $\mathcal{S}$  de  $[\mathbf{x}]_0$  correspondant à des configurations compatibles avec la carte et les mesures fournies par les capteurs, c'est-à-dire

$$\mathcal{S} = \{ \mathbf{x} \in [\mathbf{x}]_0 \mid t(\mathbf{x}) = \text{vrai} \}, \quad (9)$$

où  $t(\mathbf{x})$  est un test qui permet de statuer sur la compatibilité d'une configuration.

Pour construire  $t(\mathbf{x})$ , le processus télémétrique est simulé. Il s'agit de déterminer l'ensemble des mesures prédites par le modèle

$$\{ d_i^m(\mathbf{x}), i = 1, \dots, n_s \} \quad (10)$$

lorsque le robot se trouve dans une configuration  $\mathbf{x}$  donnée. Ces mesures simulées permettent de définir un test élémentaire  $t_i(\mathbf{x})$  pour la  $i^{\text{ème}}$  mesure

$$t_i(\mathbf{x}) = \text{vrai} \iff d_i^m(\mathbf{x}) \in [d_i]. \quad (11)$$

Le test  $t(\mathbf{x})$  peut alors être construit en utilisant une combinaison de tests élémentaires  $t_i(\mathbf{x})$  associés aux  $n_s$  capteurs :

$$t(\mathbf{x}) = t_1(\mathbf{x}) \wedge \dots \wedge t_{n_s}(\mathbf{x}). \quad (12)$$

Une première version de  $t_i(\mathbf{x})$  ne prenant pas en compte l'angle d'incidence limite  $\beta$  est donnée dans [11] ou dans [14].

Dans l'annexe 2, un test  $t_i(\mathbf{x})$  ne présentant plus cette limitation est construit pas à pas.

Dans le cas où des données aberrantes sont présentes, la technique présentée au paragraphe 2.3 pourra directement être mise en œuvre, on cherchera alors à caractériser

$$\mathcal{S}_{r,q} = \{ \mathbf{x} \in [\mathbf{x}]_0 \mid t_{\text{robuste}}(\mathbf{x}, q) = \text{vrai} \}, \quad (13)$$

où  $t_{\text{robuste}}(\mathbf{x}, q)$  est donné par (7).

### 3.3 Exemple

Le robot a été placé dans un environnement dont la carte est représentée sur la figure 6. La configuration réelle (mais inconnue) du robot est  $(x_c, y_c, \theta) = (9 \text{ m}, 9 \text{ m}, \pi \text{ rad})$ . Les mesures de distance correspondent au diagramme d'émission de la figure 8. Toutes ces mesures ont été simulées de la façon suivante. Les capteurs sont supposés tous identiques et l'état de surface est le même pour tous les obstacles, avec  $\alpha = 0.02$ ,  $\gamma = 0.1978 \text{ rad}$  et  $\beta = 0.5 \text{ rad}$ . Afin de simuler un des phénomènes qui peuvent donner naissance à des données aberrantes, lorsque l'angle d'incidence de l'onde émise par un capteur est supérieur à  $\gamma + \beta$ , la mesure est remplacée par une distance tirée aléatoirement selon une distribution uniforme entre 0.5 m et 10 m. Le pavé de recherche  $[\mathbf{x}_0]$  est égal à  $[0 \text{ m}, 12 \text{ m}] \times [0 \text{ m}, 12 \text{ m}] \times [0 \text{ rad}, 2\pi \text{ rad}]$ .

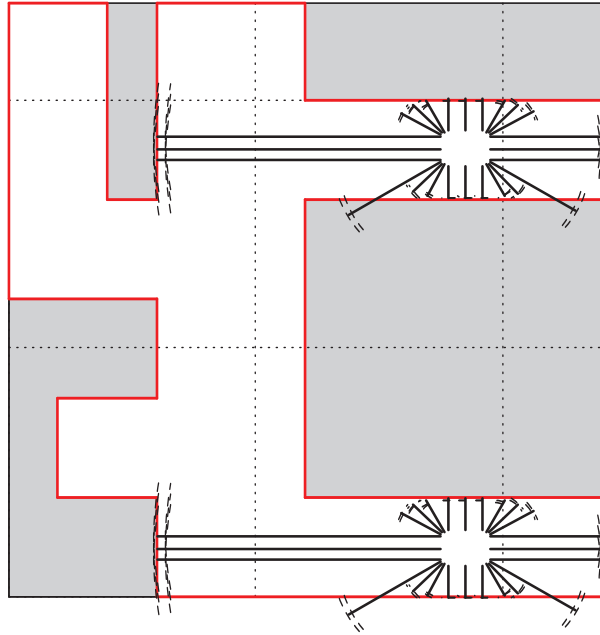


Figure 9: Deux des configurations possibles du robot après localisation statique

L'ensemble  $\widehat{\mathcal{S}}_{r,q}$  reste vide tant que  $q$  est inférieur à quatre. Lorsque  $q = 4$ , l'ensemble solution  $\widehat{\mathcal{S}}_{r,4}$  est constitué de deux sous-ensembles disjoints, qui contiennent de manière garantie la configuration réelle du robot, à condition qu'il n'y ait pas plus de quatre données aberrantes. Cette ambiguïté de la localisation est une conséquence des symétries locales de la pièce et ne doit bien sûr pas être attribuée à un défaut de la méthode qui ne fait que la révéler alors que nombre de techniques de localisation se borneraient à proposer une solution ponctuelle unique.

Lorsque  $q$  augmente, le volume de l'ensemble solution peut également augmenter, mais ce n'est pas nécessairement le cas, comme on le constate ici. Le tableau 1 résume les résultats de l'estimation pour  $q$  allant de 0 à 5, les temps de calcul correspondent à un Pentium-II à 450 MHz.

$q$	volume de $\widehat{\mathcal{S}}_{r,q}$	temps de calcul	temps cumulé
0	0	14 s	14 s
1	0	28 s	42 s
2	0	34 s	76 s
3	0	46 s	122 s
4	0.0086	59 s	181 s
5	0.0086	81 s	262 s

Tableau 1: Résultats de la procédure de localisation statique

## 4 Estimation d'état non-linéaire

Lorsqu'un système évolue dans le temps, il peut être intéressant de déterminer l'évolution de son *état* (qui pour notre robot sera sa configuration). Pour être efficace, cette évaluation doit exploiter, à un instant  $l$  donné, toutes les connaissances acquises jusqu'à  $l$  et de nouvelles mesures lorsqu'elles deviennent disponibles. Nous supposons disposer du modèle non-linéaire à temps discret

$$\begin{cases} \mathbf{x}_{l+1} = \mathbf{f}_l(\mathbf{x}_l, \mathbf{u}_l, \mathbf{v}_l), \\ \mathbf{y}_l = \mathbf{h}_l(\mathbf{x}_l) + \mathbf{w}_l, \end{cases} \quad l = 0, 1, \dots, \quad (14)$$

où  $\mathbf{x}_l \in \mathbb{R}^n$  est le vecteur d'état,  $\mathbf{u}_l \in \mathbb{R}^m$  est le vecteur d'entrée et  $\mathbf{y}_l \in \mathbb{R}^p$  est le vecteur de mesures. Le vecteur  $\mathbf{w}_l \in \mathbb{R}^p$  représente les perturbations sur la mesure et la perturbation d'état  $\mathbf{v}_l$  permet de tenir compte de dynamiques non modélisées, par exemple des frottements pour un système mécanique. La valeur initiale de l'état  $\mathbf{x}_0$  appartient par hypothèse à un compact  $\mathcal{X}_0 \subset \mathbb{R}^n$  donné (par exemple un sous-pavage résultant de l'application de l'algorithme de localisation statique).  $\{\mathbf{v}_l\}$  et  $\{\mathbf{w}_l\}$  sont des suites supposées appartenir respectivement aux suites de pavés  $\{[\mathbf{v}]_l\}$  et  $\{[\mathbf{w}]_l\}$  connues. Enfin,  $\mathbf{f}_l$  et  $\mathbf{h}_l$  sont des algorithmes finis connus.

### 4.1 Algorithme d'estimation d'état

Dans un premier temps, nous présenterons une version idéalisée d'un algorithme ensembliste d'estimation d'état, puis nous donnerons quelques éléments sur la méthode employée pour réaliser une version *approchée* mais *garantie* de cet algorithme.

L'objectif est de caractériser un ensemble  $\mathcal{X}_l$  contenant toutes les valeurs possibles de l'état  $\mathbf{x}_l$  à un instant  $l$  donné. Le raisonnement est par récurrence. Supposons qu'à l'instant  $l$ ,  $\mathcal{X}_l$  soit disponible. A l'instant  $l + 1$ , on définit l'ensemble  $\mathcal{X}_{l+}$  contenant toutes les valeurs de l'état qui sont accessibles à partir d'une valeur quelconque de  $\mathbf{x}_l$  pris dans  $\mathcal{X}_l$ . Cet ensemble est évalué (au moins en théorie) grâce à la première équation de (14). L'ensemble ① de la figure 10 est obtenu. Les ensembles  $\mathcal{X}_l$  et  $\mathcal{X}_{l+}$  peuvent être des compacts absolument quelconques.

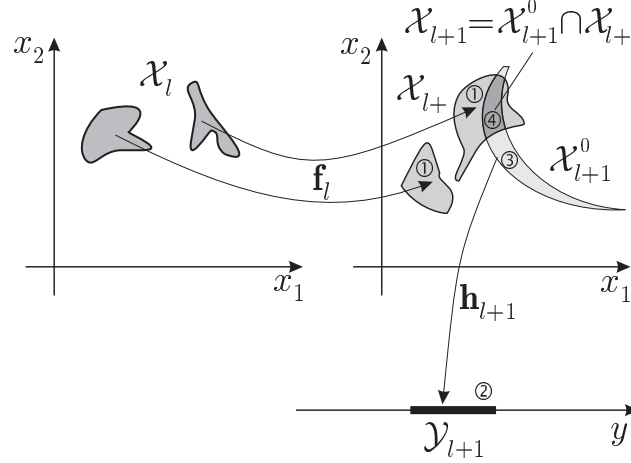


Figure 10: Algorithme idéalisé d'estimation d'état ensembliste

Supposons qu'à l'instant  $l + 1$ , un nouveau vecteur de mesure  $\mathbf{y}_{l+1}$  soit disponible. En considérant une erreur de mesure bornée, nous savons que la sortie ne peut appartenir qu'à l'ensemble  $\mathcal{Y}_{l+1}$  représenté sur la figure 10 - ensemble ②. Dans la seconde équation de (14), la fonction  $\mathbf{h}_{l+1}$  décrit comment le vecteur de mesure est obtenu à partir de l'état (dans le cas du robot, c'est la fonction décrite dans l'annexe 2 qui sera utilisée). De ce fait, l'état ne peut appartenir qu'à  $\mathcal{X}_{l+1}^o = \mathbf{h}_{l+1}^{-1}(\mathcal{Y}_{l+1})$  (voir la figure 10 - ensemble ③). Par conséquent, l'ensemble  $\mathcal{X}_{l+1}$  de toutes les valeurs de l'état  $\mathbf{x}_{l+1}$  compatibles avec les informations disponibles à l'instant  $l + 1$  est l'intersection de  $\mathcal{X}_{l+}$  et de  $\mathcal{X}_{l+1}^o$ , schématisée sur la figure 10 par l'ensemble ④. La récurrence est initialisée par  $\mathcal{X}_0$ , qui rassemble la connaissance sur les valeurs possibles de l'état à l'instant initial. La figure 10 représente ainsi les deux étapes de base d'un algorithme ensembliste et récursif d'estimation d'état. L'étape de *prédiction* détermine l'évolution de l'ensemble contenant l'état et l'étape de *correction* tient compte des nouvelles informations disponibles en éliminant certaines parties de l'ensemble obtenu par prédiction.

En général, il n'est pas possible d'évaluer ces ensembles de manière exacte. Par contre, il est possible de réaliser une version approchée mais garantie de l'algorithme d'estimation d'état grâce à l'analyse par intervalles. Celle-ci permet en effet, grâce à la notion de *fonction d'inclusion*, d'obtenir un pavé contenant l'ensemble des valeurs prises par une

fonction sur un pavé donné. Ainsi, les ensembles  $\mathcal{X}_l$  et  $\mathcal{X}_{l+}$  sont enfermés dans des sous-pavages. La technique qui permet de réaliser l'étape de prédiction de manière approchée est la suivante. Les pavés composant le sous-pavage  $\widehat{\mathcal{X}}_l$  sont découpés, et l'image de chaque pavé résultant est calculée. Ces images sont ensuite rassemblées pour former un sous-pavage approximant  $\mathcal{X}_{l+}$ . L'étape de correction permettant de déterminer  $\widehat{\mathcal{X}}_{l+1}$  se fait ensuite grâce à SIVIA. L'ensemble  $\mathcal{X}_{l+1}^o$  n'est en fait pas calculé, seule son intersection avec  $\widehat{\mathcal{X}}_{l+}$  l'est, ce qui permet d'accélérer de manière spectaculaire la procédure de recherche.

Cet estimateur approché est garanti, au sens où toutes les valeurs de l'état compatibles avec les hypothèses sont contenues dans les ensembles  $\widehat{\mathcal{X}}_l$ . Pour plus de détails sur cette technique, voir [11] et [12].

## 4.2 Prise en compte de données aberrantes

Les mesures aberrantes éventuelles n'interviennent que dans l'étape de correction. Dans le cas du robot, cette étape correspond en fait à une localisation dans un espace de recherche réduit à l'ensemble  $\widehat{\mathcal{X}}_{l+}$ . Le traitement des données aberrantes se fait donc de la même façon que pour la localisation statique. Une des politiques envisageables, qui est celle qui sera suivie ici, consiste dans un premier temps à rechercher l'ensemble  $\widehat{\mathcal{X}}_l$  des états compatibles avec toutes les mesures à l'instant  $l$  ( $q_l = 0$ ), puis à tolérer de plus en plus de données aberrantes en incrémentant  $q_l$  d'une unité tant que  $\widehat{\mathcal{X}}_l$  reste vide.

En pratique, il pourrait s'avérer plus prudent de donner à  $q_l$  une valeur supérieure à la plus petite valeur qui assure que  $\widehat{\mathcal{X}}_l$  soit non-vide. En effet, la procédure n'est garantie que si à chaque étape le nombre réel de données aberrantes est inférieur ou égal au nombre de données aberrantes tolérées  $q_l$ . Cette modification de la politique de choix de  $q_l$  est extrêmement simple à mettre en œuvre, mais cette augmentation de robustesse se fait au détriment de la précision de la localisation, puisqu'un nombre plus important de mesures doit être négligé.

## 4.3 Application au suivi du robot mobile

Cet algorithme d'estimation d'état est appliqué au problème de suivi du robot mobile. L'équation d'état est obtenue par discrétisation exacte des équations cinématiques du robot. Pour simplifier l'exposé, aucun bruit d'état n'a été considéré, mais la prise en compte d'un tel bruit ne pose aucun problème de principe pourvu que des bornes soient disponibles. L'étape de correction correspond à la technique de localisation statique présentée au paragraphe 3. La pièce est toujours décrite par la carte représentée sur la figure 6. L'ensemble  $\widehat{\mathcal{X}}_0$ , correspond au résultat de la localisation statique effectuée au paragraphe 3.3.

Le robot se déplace du couloir représenté sur la partie supérieure droite de la figure 6 vers le couloir central. Toutes les secondes, une nouvelle série de mesures est effectuée afin de corriger l'ensemble obtenu par l'étape de prédiction de l'algorithme d'estimation d'état. L'évolution de la projection des ensembles  $\widehat{\mathcal{X}}_i$  sur le plan  $(x, y)$  est représentée sur la partie droite de la figure 11. Sur la partie gauche de cette figure, le robot est représenté dans une ou deux configurations appartenant à l'ensemble solution trouvé. Jusqu'à l'instant 5, l'ensemble solution est composé de deux sous-ensembles disjoints. Après cet instant, le sous-ensemble inférieur a pu être éliminé. La simulation nécessite un temps de calcul inférieur à 7 s sur un Pentium-II à 450 MHz, ce qui est compatible avec une mise en œuvre en temps réel. Le tableau 2 indique l'évolution du nombre de données aberrantes estimé au cours du processus de localisation. Transitoirement, le nombre minimum de données aberrantes  $q_i$  se révèle très important (la moitié des mesures).

Instant	0	1	2	3	4	5	6	7	8	9
$q$	4	5	4	4	4	3	3	7	12	8

Instant	10	11	12	13	14	15	16	17	18
$q$	8	5	4	6	5	3	4	5	7

Tableau 2: Evolution de  $q_i$ , le nombre minimum de données aberrantes estimé par l'algorithme, en fonction du temps

## 5 Conclusions

Cet article a présenté une technique d'estimation (de paramètres ou d'état) qui se révèle très robuste vis-à-vis de données aberrantes comme celles qui sont inévitables en robotique mobile. Une solution originale est ainsi apportée à certaines limitations dont souffraient les techniques de localisation et de suivi précédemment employées. Elle est garantie, contrairement aux techniques fondées sur le filtrage de Kalman (voir [2], [16] et [17]) ou aux techniques à erreur bornée nécessitant une linéarisation préalable, comme celles décrites dans [1], [4], [18] et [25]. Aucune étape d'association des mesures avec les segments de la carte n'est nécessaire, contrairement à ce qui est fait dans [3] et [5]. Pour le suivi, la gestion d'hypothèses de localisation multiples est naturellement effectuée, contrairement à ce qui est fait dans [6]. Enfin, la procédure de localisation n'étant qu'une étape de l'estimation d'état, aucune procédure séparée de localisation n'est nécessaire contrairement à ce qui se passe dans [16] et [21].

L'un des principaux défauts des techniques ensemblistes utilisant l'analyse par intervalles qui viennent d'être présentées est leur complexité exponentielle en la dimension de  $\mathbf{x}$ . Dans le contexte de la localisation et du suivi du



robot mobile, ce point ne pose cependant pas de problème insurmontable.

Une démonstration du suivi du robot mobile peut être trouvée à l'adresse suivante

<http://www.supelec.fr/lss/fr/personnels/kieffer/robotloc.htm>

**Remerciements:** Les auteurs remercient INTAS pour son soutien financier partiel. (Contrat INTAS-RFBR-97-10782).

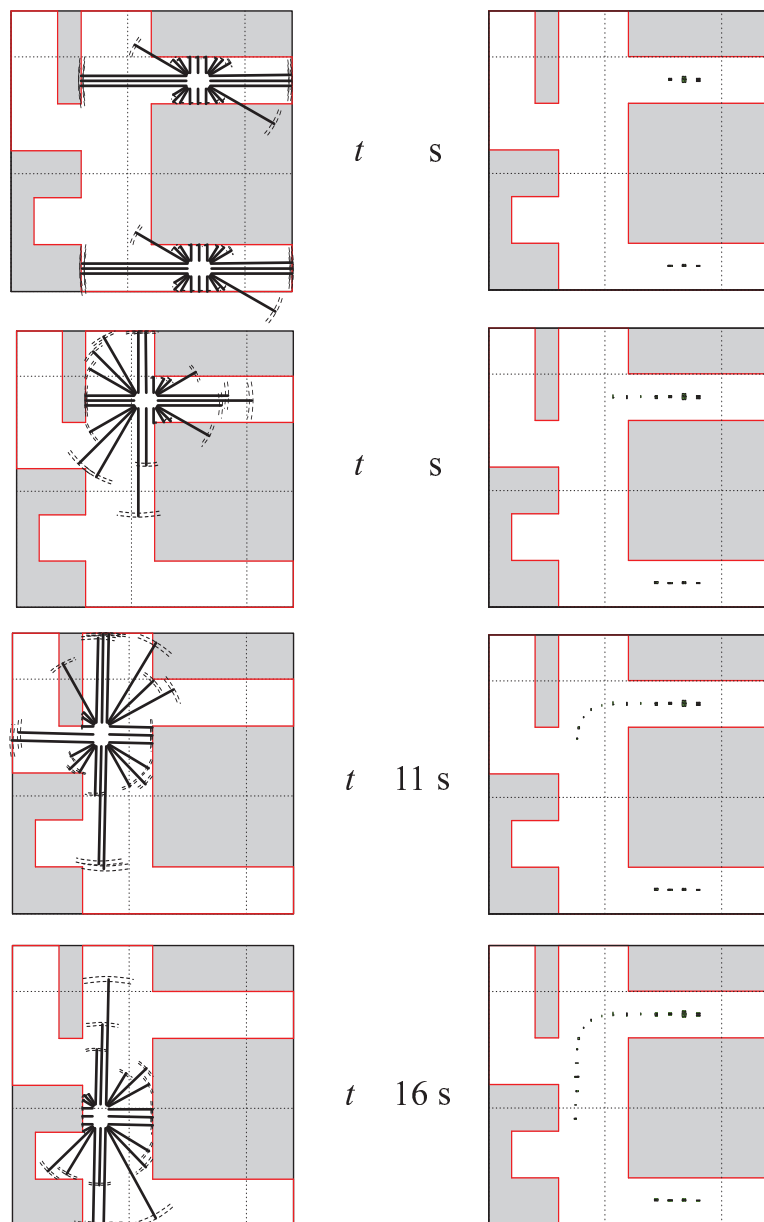


Figure 11: Evolution de deux configurations possibles du robot et projection des ensembles solution sur le plan  $(x, y)$ ; à partir de  $t = 5$  s, l'une des deux configurations est jugée incompatible avec les données

# ANNEXES

## 1 Éléments de calcul sur les intervalles

Un intervalle  $[x] = [\underline{x}, \bar{x}]$  est un ensemble fermé borné de  $\mathbb{R}$  ; il est caractérisé par ses bornes inférieure  $\underline{x}$  et supérieure  $\bar{x}$ . L'ensemble des intervalles réels est noté  $\mathbb{IR}$ . Il est possible de redéfinir les opérations de base sur les intervalles : soit  $\circ \in \{-, +, \cdot, /\}$  et  $([x], [y]) \in \mathbb{IR}^2$ ,

$$[x] \circ [y] = \{x \circ y \mid x \in [x], y \in [y]\}.$$

Caractériser  $[z] = [x] \circ [y]$  est aisé ; par exemple

$$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}],$$

$$[x] / [y] = [\inf(\underline{x}/\underline{y}, \underline{x}/\bar{y}, \bar{x}/\underline{y}, \bar{x}/\bar{y}), \sup(\underline{x}/\underline{y}, \underline{x}/\bar{y}, \bar{x}/\underline{y}, \bar{x}/\bar{y})], \text{ lorsque } 0 \notin [y].$$

On peut également décrire par un intervalle l'ensemble des valeurs prises par une fonction élémentaire monotone sur un intervalle, par exemple

$$\exp([x]) = [\exp(\underline{x}), \exp(\bar{x})].$$

Pour des fonctions élémentaires non monotones (par exemple  $\cos$  ou l'élevation à une puissance paire), l'obtention de l'ensemble des valeurs prises par cette fonction se fait par un algorithme simple (voir [7], [20]). Dans le cas de fonctions quelconques, il reste possible dans de très nombreux cas d'évaluer un *encadrement* des valeurs prises par la fonction grâce aux *fonctions d'inclusion*. Une fonction d'inclusion  $[f]$  d'une fonction scalaire  $f$  définie sur  $\mathcal{D} \subset \mathbb{R}$  est telle que

$$\text{pour tout } [x] \subset \mathcal{D}, f([x]) \subset [f]([x]).$$

Une technique simple pour obtenir la fonction d'inclusion *naturelle* d'une fonction  $f$  consiste à remplacer toutes les occurrences de la variable réelle par la variable intervalle correspondante. Ainsi, si on doit évaluer  $f(x) = x^3 - 0.6 \cos(x \exp(x^2))$  sur  $[-1, 1]$ , la fonction d'inclusion naturelle est

$$\begin{aligned} [f]([-1, 1]) &= [-1, 1]^3 - 0.6 \cos\left([-1, 1] \exp\left([-1, 1]^2\right)\right) \\ &= [-1, 1] - 0.6 \cos\left([-1, 1] \exp([0, 1])\right) \\ &= [-1, 1] - 0.6 \cos([-e, e]) \\ &= [-1, 1] - 0.6 [-1, 1] \\ &= [-1.6, 1.6]. \end{aligned}$$

Or  $f([-1, 1]) = [-0.6, 1.6]$ , on a donc bien  $f([-1, 1]) \subset [f]([-1, 1])$ .

Au paragraphe 2.4, on souhaite évaluer le résultat d'un test sur un intervalle. Dans ce but, il est aisé de construire des *tests d'inclusion* à partir des fonctions d'inclusion. Par exemple, soit le test  $t(x) = (f(x) \in [y])$ , le test d'inclusion correspondant utilisera une fonction d'inclusion de  $f$  :

$$\begin{aligned} [t]([x]) &= \text{faux si } [f]([x]) \cap [y] = \emptyset, \\ [t]([x]) &= \text{vrai si } [f]([x]) \subset [y], \\ [t]([x]) &= \text{ambigu dans les autres cas.} \end{aligned}$$

## 2 Test de localisation

Cette annexe explique comment est construit le test de compatibilité d'une configuration  $\mathbf{x}$  avec les données fournies par un capteur  $\mathbf{s}$ . Dans le repère  $\mathcal{R}$  du robot, les coordonnées de  $\mathbf{s}$  sont notées  $\tilde{\mathbf{s}}$ , et la direction d'émission  $\tilde{\theta}$ . L'onde émise par ce capteur se propage dans un cône noté  $\mathcal{C}(\tilde{\mathbf{s}}, \tilde{\theta}, \gamma)$ , où  $\gamma$  représente la demi-ouverture.

Pour bâtir le test associé à  $\mathbf{s}$ , les informations disponibles dans  $\mathcal{R}$  doivent être traduites dans le repère du laboratoire  $\mathcal{W}$ . Pour une configuration  $\mathbf{x}$  donnée,  $\mathcal{C}$  est décrit de manière équivalente dans  $\mathcal{W}$  (en utilisant des formules de changement de repère) par les coordonnées de son sommet  $\mathbf{s}(\mathbf{x})$  et deux vecteurs unitaires  $\vec{\mathbf{u}}_1(\mathbf{x}, \tilde{\theta}, \gamma)$  et  $\vec{\mathbf{u}}_2(\mathbf{x}, \tilde{\theta}, \gamma)$  correspondant à ses arêtes. Ainsi, on peut aussi noter le cône  $\mathcal{C}(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2)$  (en omettant la dépendance en  $\mathbf{x}, \tilde{\theta}$  et  $\gamma$ ). Par convention,  $\vec{\mathbf{u}}_1$  et  $\vec{\mathbf{u}}_2$  sont choisis de telle manière que  $\vec{\mathbf{u}}_2$  résulte de  $\vec{\mathbf{u}}_1$  par une rotation d'angle  $2\gamma$  dans le sens trigonométrique. Comme  $\gamma$  est physiquement toujours inférieur à  $\pi/2$ , la condition d'appartenance d'un point quelconque  $\mathbf{m} \in \mathbb{R}^2$  au cône d'émission se traduit par

$$\mathbf{m} \in \mathcal{C}(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2) \Leftrightarrow (\det(\vec{\mathbf{u}}_1, \vec{\mathbf{s}}\mathbf{m}) \geq 0) \wedge (\det(\vec{\mathbf{u}}_2, \vec{\mathbf{s}}\mathbf{m}) \leq 0). \quad (15)$$

L'algorithme testant la cohérence d'une configuration avec les données est fondé sur la notion d'*éloignement* entre un capteur et la carte. Nous allons dans un premier temps définir ce concept pour un capteur et un segment  $[\mathbf{a}, \mathbf{b}]$  de la carte pris isolément. On suppose que ce segment représente une partie d'un obstacle dont l'angle limite de réflexion est  $\beta$ .

**Definition 1** *L'éloignement  $r(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \mathbf{a}, \mathbf{b}, \beta)$  entre le cône  $\mathcal{C}(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2)$  et le segment  $[\mathbf{a}, \mathbf{b}]$  est la plus petite distance entre  $\mathbf{s}$  et l'intersection (lorsqu'elle existe) de  $[\mathbf{a}, \mathbf{b}]$  avec le cône  $\mathcal{C}$ , à condition que l'incidence se fasse sous un angle*

moyen inférieur à  $\gamma + \beta$  (condition d'incidence). Il est évalué comme suit

$$r(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \mathbf{a}, \mathbf{b}) = \infty \begin{cases} \text{si la condition d'incidence n'est pas vérifiée,} \\ \text{ou si } [\mathbf{a}, \mathbf{b}] \cap \mathcal{C} = \emptyset, \end{cases} \quad (16)$$

$$= \min_{\mathbf{m} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}} \|\vec{\mathbf{sm}}\| \text{ dans les autres cas.}$$

En pratique, on vérifie dans un premier temps si l'incidence se fait avec un angle inférieur à  $\beta + \gamma$ . Soit le vecteur

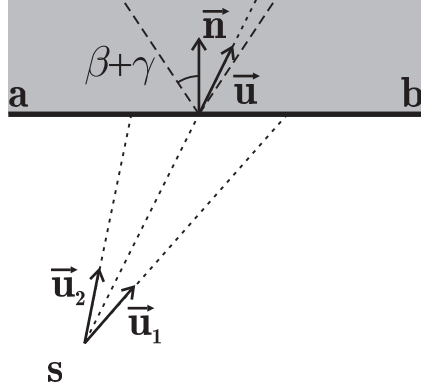


Figure 12: L'angle d'incidence doit être inférieur à  $\gamma + \beta$  pour qu'une onde réfléchie soit perçue

unitaire

$$\vec{\mathbf{u}} = \frac{\vec{\mathbf{u}}_1 + \vec{\mathbf{u}}_2}{\|\vec{\mathbf{u}}_1 + \vec{\mathbf{u}}_2\|}$$

correspondant à la direction d'émission moyenne du capteur et  $\vec{\mathbf{n}}$  un vecteur unitaire perpendiculaire à  $\vec{\mathbf{ab}}$ , et orienté vers le côté non réfléchissant de  $[\mathbf{a}, \mathbf{b}]$  (voir la figure 12). La condition d'incidence se traduit par

$$\langle \vec{\mathbf{u}}, \vec{\mathbf{n}} \rangle \geq \cos(\beta + \gamma), \quad (17)$$

ou encore par

$$\det \left( \vec{\mathbf{u}}, \frac{\vec{\mathbf{ab}}}{\|\vec{\mathbf{ab}}\|} \right) \geq \cos(\beta + \gamma). \quad (18)$$

Lorsque cette condition d'incidence est vérifiée, on cherche le minimum de  $\|\vec{\mathbf{sm}}\|$  lorsque  $\mathbf{m}$  décrit  $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ . Cette minimisation impose l'examen de plusieurs situations. Soit  $\mathbf{h}$  la projection orthogonale de  $\mathbf{s}$  sur la droite  $(\mathbf{a}, \mathbf{b})$ . Si  $\mathbf{h} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ , alors  $r(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \mathbf{a}, \mathbf{b}) = \|\vec{\mathbf{sh}}\|$ . Pour tester si  $\mathbf{h} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ , il faut déterminer dans un premier temps si  $\mathbf{h} \in [\mathbf{a}, \mathbf{b}]$ . Comme  $\mathbf{h}$  est la projection orthogonale de  $\mathbf{s}$  sur  $[\mathbf{a}, \mathbf{b}]$ , on a

$$\mathbf{h} \in [\mathbf{a}, \mathbf{b}] \iff \left( \langle \vec{\mathbf{ab}}, \vec{\mathbf{as}} \rangle \geq 0 \right) \wedge \left( \langle \vec{\mathbf{ba}}, \vec{\mathbf{bs}} \rangle \geq 0 \right). \quad (19)$$

De plus, il faut que  $\mathbf{h} \in \mathcal{C}$ . En utilisant (15), on obtient

$$\mathbf{h} \in \mathcal{C} \iff \left( \det \left( \vec{\mathbf{u}}_1, \vec{\mathbf{s}}\mathbf{h} \right) \geq 0 \right) \wedge \left( \det \left( \vec{\mathbf{u}}_2, \vec{\mathbf{s}}\mathbf{h} \right) \leq 0 \right). \quad (20)$$

Cette condition n'est cependant guère exploitable, car elle nécessite la connaissance des coordonnées de  $\mathbf{h}$ , qui sont relativement complexes à obtenir. Or, si la condition d'incidence (18) est respectée, l'angle entre  $\vec{\mathbf{s}}\mathbf{h}$  et  $\vec{\mathbf{a}}\mathbf{b}$  est de  $\frac{\pi}{2}$  et ainsi, (20) devient après quelques manipulations simples

$$\mathbf{h} \in \mathcal{C} \iff \left( \langle \vec{\mathbf{a}}\mathbf{b}, \vec{\mathbf{u}}_1 \rangle \geq 0 \right) \wedge \left( \langle \vec{\mathbf{a}}\mathbf{b}, \vec{\mathbf{u}}_2 \rangle \leq 0 \right). \quad (21)$$

En combinant (19) et (21), on obtient finalement une condition d'appartenance de  $\mathbf{h}$  à  $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$  indépendante des coordonnées de  $\mathbf{h}$

$$\begin{aligned} \mathbf{h} \in [\mathbf{a}, \mathbf{b}] \cap \mathcal{C} \iff & \left( \langle \vec{\mathbf{a}}\mathbf{b}, \vec{\mathbf{s}}\mathbf{a} \rangle \leq 0 \right) \wedge \left( \langle \vec{\mathbf{a}}\mathbf{b}, \vec{\mathbf{s}}\mathbf{b} \rangle \geq 0 \right) \\ & \wedge \left( \langle \vec{\mathbf{a}}\mathbf{b}, \vec{\mathbf{u}}_1 \rangle \leq 0 \right) \wedge \left( \langle \vec{\mathbf{a}}\mathbf{b}, \vec{\mathbf{u}}_2 \rangle \geq 0 \right). \end{aligned} \quad (22)$$

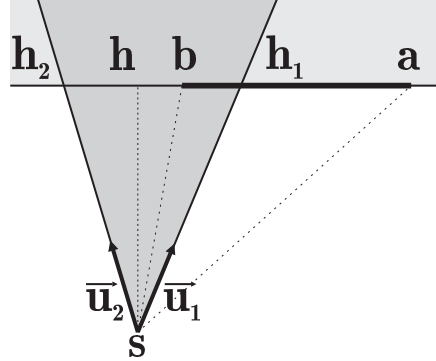


Figure 13: Espaceur entre un capteur et un segment  $[\mathbf{a}, \mathbf{b}]$  isolé

Si  $\mathbf{h} \notin [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ , la distance minimale peut être infinie (lorsque  $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C} = \emptyset$ ), ou être obtenue pour un point  $\mathbf{m}$  correspondant à l'une des extrémités du segment  $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ . Soit  $\mathbf{h}_1$  et  $\mathbf{h}_2$  les intersections de la droite  $(\mathbf{a}, \mathbf{b})$  avec les droites  $(\mathbf{s}, \vec{\mathbf{u}}_1)$  et  $(\mathbf{s}, \vec{\mathbf{u}}_2)$ . L'ensemble des extrémités potentielles de  $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$  est donc  $\mathcal{K} = \{\mathbf{a}, \mathbf{b}, \mathbf{h}_1, \mathbf{h}_2\}$ . De ce fait, si  $\mathbf{h} \notin [\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$ ,  $r(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \mathbf{a}, \mathbf{b}, \beta)$  est soit infini, soit égal à  $\|\vec{\mathbf{s}}\mathbf{m}\|$ , avec  $\mathbf{m}$  appartenant à  $\mathcal{K}$ . Dans le cas de la figure 13,  $r(\mathbf{s}, \vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \mathbf{a}, \mathbf{b}, \beta) = \|\vec{\mathbf{s}}\mathbf{b}\|$ .

Un test pour déterminer si un élément de  $\mathcal{K}$  appartient à l'ensemble  $[\mathbf{a}, \mathbf{b}] \cap \mathcal{C}$  est facile à obtenir à partir de

l'équation (15). Soit  $\mathbf{v} \in \{\mathbf{a}, \mathbf{b}\}$

$$\mathbf{v} \in \mathcal{C} \iff (\det(\vec{\mathbf{u}}_1, \vec{\mathbf{sv}}) \geq 0) \wedge (\det(\vec{\mathbf{u}}_2, \vec{\mathbf{sv}}) \leq 0). \quad (23)$$

Par construction,  $\mathbf{h}_i \in \mathcal{C} \cap \{\mathbf{a}, \mathbf{b}\}$ , il suffit alors de tester si  $\mathbf{h}_i$  appartient à  $\{\mathbf{a}, \mathbf{b}\}$ , ce qui revient tester si  $\mathbf{h}_i \in \mathcal{C} \left( \mathbf{s}, \frac{\vec{\mathbf{sa}}}{\|\vec{\mathbf{sa}}\|}, \frac{\vec{\mathbf{sb}}}{\|\vec{\mathbf{sb}}\|} \right)$ . Ainsi, pour  $i = 1, 2$ ,

$$\begin{aligned} \mathbf{h}_i \in \{\mathbf{a}, \mathbf{b}\} \cap \mathcal{C} \iff & \left( \det \left( \frac{\vec{\mathbf{sa}}}{\|\vec{\mathbf{sa}}\|}, \vec{\mathbf{sh}}_i \right) \geq 0 \right) \\ & \wedge \left( \det \left( \frac{\vec{\mathbf{sb}}}{\|\vec{\mathbf{sb}}\|}, \vec{\mathbf{sh}}_i \right) \leq 0 \right), \end{aligned}$$

or comme  $\frac{\vec{\mathbf{sh}}_i}{\|\vec{\mathbf{sh}}_i\|} = \vec{\mathbf{u}}_i$ , on a

$$\mathbf{h}_i \in \{\mathbf{a}, \mathbf{b}\} \cap \mathcal{C} \iff (\det(\vec{\mathbf{sa}}, \vec{\mathbf{u}}_i) \geq 0) \wedge (\det(\vec{\mathbf{sb}}, \vec{\mathbf{u}}_i) \leq 0). \quad (24)$$

En résumé, lorsque ni  $\mathbf{h}$ , ni aucun élément de  $\mathcal{K}$  n'appartient à  $\{\mathbf{a}, \mathbf{b}\} \cap \mathcal{C}$ , alors il faut conclure que  $\{\mathbf{a}, \mathbf{b}\} \cap \mathcal{C} = \emptyset$ , et l'éloignement entre le cône et le segment est infini.

Dans la situation normale où la carte comprend  $n_o$  segments, il faut prendre en compte le fait qu'un segment puisse ne pas avoir réfléchi l'onde sonore parce qu'il est masqué par un autre segment, plus proche du capteur à ultrasons. Soit  $r_{ij}(\mathbf{x})$  l'éloignement entre le  $j^{\text{ème}}$  segment, considéré comme isolé, et le cône associé au  $i^{\text{ème}}$  capteur lorsque la configuration est  $\mathbf{x}$ ; il est exprimé par

$$r_{ij}(\mathbf{x}) = r \left( \mathbf{s}_i(\mathbf{x}), \mathbf{u}_{1i}(\mathbf{x}, \tilde{\theta}_i, \tilde{\gamma}_i), \mathbf{u}_{2i}(\mathbf{x}, \tilde{\theta}_i, \tilde{\gamma}_i), \mathbf{a}_j, \mathbf{b}_j, \beta_j \right). \quad (25)$$

L'éloignement  $r_i$  entre le capteur  $i$  et tous les segments de la carte est le plus petit éloignement entre le capteur et un segment de la carte se trouvant en partie dans le cône d'émission du capteur. Lorsque la configuration est  $\mathbf{x}$ ,  $r_i(\mathbf{x})$  est donc donné par

$$r_i(\mathbf{x}) = \min_{j=1, \dots, n_o} r_{ij}(\mathbf{x}). \quad (26)$$

## Références

- [1] D. P. Bertsekas et I. B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Trans. on Automatic Control*, 16: 117–128, 1971.
- [2] J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. Dans *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 674–680, Scottsdale, Arizona, 1989.

- [3] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2): 325–332, 1987.
- [4] C. Durieu, B. Polyak, et E. Walter. Trace versus determinant in ellipsoidal outer bounding with application to state estimation. Dans *Proc. 13th IFAC World Congress*, volume I, pages 43–48, San Francisco, 30 juin - 5 juillet 1996.
- [5] W. E. Grimson et T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4): 469–482, 1987.
- [6] E. Halbwachs et D. Meizel. Multiple hypothesis management for mobile vehicle localization. Dans *CD Rom of the European Control Conference*, Louvain, 1997.
- [7] R. Hammer, M. Hocks, U. Kulish, et D. Ratz. *C++ Toolbox for Verified Computing*. Springer Verlag, Berlin, 1995.
- [8] E. Hecht. *Optics*. Addison-Wesley, Reading, second edition, 1987.
- [9] L. Jaulin et E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4): 1053–1064, 1993.
- [10] L. Jaulin, E. Walter, et O. Didrit. Guaranteed robust nonlinear parameter bounding. Dans *Proc. CESA'96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, pages 1156–1161, Lille, 9-12 juillet 1996.
- [11] M. Kieffer. *Estimation ensembliste par analyse par intervalles, application à la localisation d'un véhicule*. Thèse de doctorat, Université Paris-Sud, Orsay, 1999.
- [12] M. Kieffer, L. Jaulin, et E. Walter. Guaranteed recursive nonlinear state estimation using interval analysis. Dans *Proc. 37th IEEE Conference on Decision and Control*, pages 3966–3971, Tampa, 16-18 décembre, 1998.
- [13] M. Kieffer, L. Jaulin, E. Walter, et D. Meizel. Nonlinear identification based on unreliable priors and data, with application to robot localization. Dans A. Garulli, A. Tesi, et A. Vicino, (Eds.), *Robustness in Identification and Control*, pages 190–203, LNCIS 245, Londres, 1999. Springer.
- [14] M. Kieffer, L. Jaulin, E. Walter, et D. Meizel. Robust autonomous robot localization using interval analysis. A paraître dans *Reliable Computing* 6(3), 2000.

- [15] H. Lahanier, E. Walter, et R. Gomeni. OMNE: a new robust membership-set estimator for the parameters of nonlinear models. *J. of Pharmacokinetics and Biopharmaceutics*, 15: 203–219, 1987.
- [16] J. J. Leonard et H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3): 376–382, 1991.
- [17] J. J. Leonard et H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, Boston, 1992.
- [18] D. Maksarov et J. P. Norton. State bounding with ellipsoidal set description of the uncertainty. *Int. J. of Control*, 65(5): 847–866, 1996.
- [19] M. Milanese, J. Norton, H. Piet-Lahanier, et E. Walter (Eds.). *Bounding Approaches to System Identification*. Plenum Press, New York, 1996.
- [20] R. E. Moore. Practical aspects of interval computation. *Appl. Math.*, (13): 52–92, 1968.
- [21] J. Neira, J. Horn, J. D. Tardoz, et G. Schmidt. Multisensor mobile robot localization. Dans *Proc. IEEE International Conference on Robotics and Automation*, pages 673–679, Mineapolis, 1996.
- [22] J. P. Norton (Ed.). Special issue on bounded-error estimation: Issue 1. *Int. J. of Adaptive Control and Signal Processing*, 8(1): 1–118, 1994.
- [23] J. P. Norton (Ed.). Special issue on bounded-error estimation: Issue 2. *Int. J. of Adaptive Control and Signal Processing*, 9(1): 1–132, 1995.
- [24] L. Pronzato et E. Walter. Robustness to outliers of bounded-error estimators and consequences on experiment design. Dans M. Milanese, J. Norton, H. Piet-Lahanier, et E. Walter, (Eds.), *Bounding Approaches to System Identification*, pages 199–212, New York, 1996. Plenum Press.
- [25] F. C. Schweppe. *Uncertain Dynamic Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [26] E. Walter et H. Piet-Lahanier. Estimation of the parameter uncertainty resulting from bounded-error data. *Mathematical Biosciences*, 92: 55–74, 1988.
- [27] E. Walter (Ed.). Special issue on parameter identifications with error bounds. *Mathematics and Computers in Simulation*, 32(5&6): 447–607, 1990.



- [28] E. Walter et L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, Londres, 1997.