

# Calcul ensembliste pour la localisation et la cartographie robustes

Jan SLIWKA, Fabrice LE BARS, Luc JAULIN  
Laboratoire DTN  
ENSIETA, 2 rue François VERNY 29200 Brest  
{sliwkaja,lebarsfa,jaulinlu}@ensieta.fr  
www.ensieta.fr/jaulin/

## Abstract

Dans cet article seront présentées deux applications de la théorie ensembliste, notamment l'analyse par intervalles, dans les problèmes de localisation et cartographie en robotique. La première application porte sur une localisation d'un robot par rapport à un mur et l'autre sur la reconstruction d'un environnement paramétré en utilisant de la télémétrie.

**Mots Clés** : Localisation, données aberrantes, robustesse, versatilité, parallélisme, SLAM

## 1 Introduction

La localisation du robot est une brique essentielle pour que ce dernier puisse mouvoir d'une manière indépendante (sans intervention humaine). Le pouvoir de se localiser permet aussi de faire de la cartographie. Le robot se déplace souvent dans un environnement imparfait donc ses capteurs ne donnent pas toujours des données correctes. Ceci implique que les algorithmes de localisation et de cartographie doivent être robustes. Nous utiliserons la théorie ensembliste pour résoudre ces problèmes comme alternative aux approches probabilistes en robotique [3]. En effet, en présence de données aberrantes et de non linéarités, l'utilisation d'une approche probabiliste, par exemple un filtre de Kalman, n'est pas satisfaisante. Nous essaierons d'expliquer la localisation et cartographie robustes, à travers deux exemples. Le premier exemple est celui de la localisation d'un robot par rapport à un mur infini - voir figure 1- en utilisant

de la télémétrie. En effet, nous voulons connaître son orientation et sa distance par rapport au mur. Le deuxième exemple est celui d'une reconstruction d'un environnement paramétré en utilisant des données télémétriques provenant d'un robot se trouvant dans un tel environnement. La figure 3 représente un robot qui se trouve dans une pièce ayant la forme d'un quadrilatère convexe dont on déterminera les paramètres.

## 2 L'approche ensembliste

### 2.1 Estimation d'ensembles

Dans un contexte d'une erreur bornée [4], un problème d'estimation de paramètres peut être généralement amené à un problème de caractérisation d'un ensemble défini par

$$\mathbb{P} = \bigcap_{i \in \{1, \dots, m\}} \underbrace{\{\mathbf{p} \in \mathbb{R}^{n_p}, \exists \mathbf{y} \in [\mathbf{y}](i), \mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}\}}_{\mathbb{P}_i} \quad (1)$$

$\mathbf{p}$  étant le vecteur de paramètres,  $[\mathbf{y}](i) \subset \mathbb{R}^{n_y}$  étant le  $i$ ème pavé de mesures et  $\mathbf{f} : \mathbb{R}^{n_p} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_f}$  étant la fonction du modèle. Les ensembles  $\mathbb{P}_i$  sont les ensembles de tous les vecteurs de paramètres qui sont consistants avec le  $i$ ème vecteur de mesures. Le problème cité précédemment peut être qualifié de *implicite* car il utilise une équation implicite  $\mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}$ . Dans le cas particulier où  $y$  peut être isoler, *i.e.*,  $\mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0} \Leftrightarrow \mathbf{g}(\mathbf{p}) = \mathbf{y}$ , on trouve

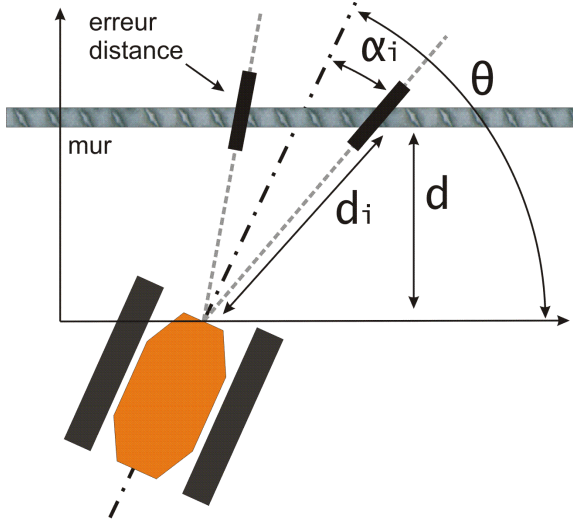


Figure 1: Localisation du robot avec deux mesures télémétriques

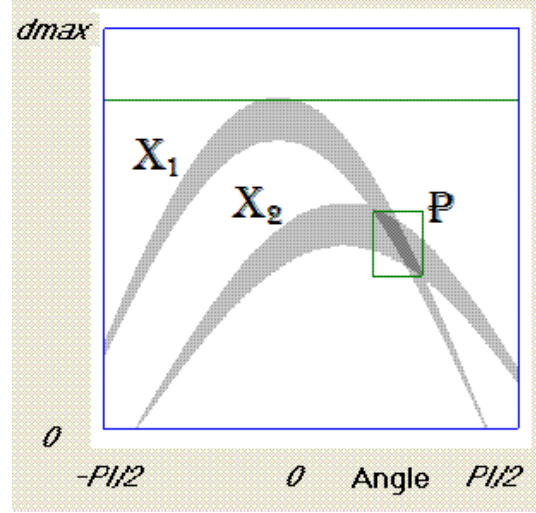


Figure 2: Les différents ensembles solutions représentés dans l'espace de configuration

$$\mathbb{P} = \bigcap_{i \in \{1, \dots, m\}} \underbrace{\{\mathbf{p} \in \mathbb{R}^{n_p}, \mathbf{g}(\mathbf{p}) \in [\mathbf{y}](i)\}}_{\mathbb{P}_i} \quad (2)$$

$$= \bigcap_{i \in \{1, \dots, m\}} \mathbf{g}^{-1}([\mathbf{y}](i)), \quad (3)$$

et le problème devient alors un problème d'inversion ensembliste.

## 2.2 Localisation comme problème d'inversion ensembliste

Dans la figure 1 on veut localiser le robot avec des données télémétriques. En effet, on veut connaître son orientation  $\theta$  et la distance  $d$  par rapport au mur. Une donnée télémétrique  $i$  est composée d'une distance au premier obstacle (au mur)  $\tilde{d}_i$  avec une erreur  $\Delta\tilde{d}_i$  et (ii) l'angle correspondant  $\alpha_i$  du faisceau du télémètre par rapport au robot. On a

$$d = \tilde{d}_i \cos(\theta + \alpha_i)$$

On note par  $[d_i]$  l'intervalle de mesure de  $\tilde{d}_i$ . Si  $\tilde{d}_i$  est la distance mesurée avec une erreur  $\Delta\tilde{d}_i$ . On a  $[d_i] = [\tilde{d}_i - \Delta\tilde{d}_i, \tilde{d}_i + \Delta\tilde{d}_i]$ .

A priori, on suppose que l'ensemble des positions possibles du robot  $(d, \theta)$  est  $\mathbb{D} = [0, D_0] \times [-\pi/2, \pi/2]$ . L'ensemble des positions du robot consistant avec les deux mesures devient

$$\mathbb{P} = \mathbb{X}_1 \cap \mathbb{X}_2 \quad (4)$$

$$\text{où } \mathbb{X}_i \stackrel{\text{def}}{=} \left\{ (d, \theta) \in \mathbb{D} \mid \exists \tilde{d} \in [d_i], d = \tilde{d} \cos(\theta + \alpha_i) \right\}$$

$\mathbb{P}$  et les  $\mathbb{X}_i$  sont représentés sur la figure 2. Cette figure est générée grâce à un programme de simulation écrit en langage C++.

**Remarque:** Dans notre exemple, pour chaque mesure,

$$f(\mathbf{p}, \mathbf{y}) = y_1 \cos(p_2 + y_2) - p_1 \quad (5)$$

Où  $\mathbf{p} = (p_1, p_2)^T$  représente le vecteur de paramètres  $p_1, p_2$  égaux à  $d$  et  $\theta$  respectivement et  $\mathbf{y} = (y_1, y_2)^T$  représente les mesures. A chaque coup de mesure,  $(y_1, y_2)$  représentent la distance au premier obstacle et l'angle du faisceau de mesure respectivement.

Dans le cas réel, les données ne sont pas toutes sûres à cause, par exemple, d'obstacles imprévus entre le télémètre et le mur. Les données erronées ou

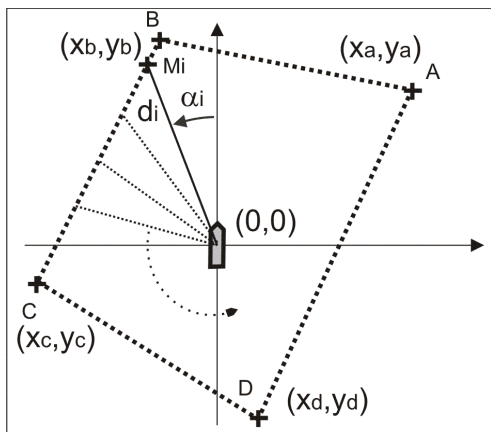


Figure 3: reconstruction de l'environnement, une pièce quadrilatère convexe

encore dites aberrantes devraient être prises en considération. (i) Il faudra d'abord augmenter le nombre de mesures. (ii) Il faudra ensuite changer un peu la méthode. La section 4 expliquera comment on peut rendre la méthode robuste par rapport aux données aberrantes.

### 3 Reconstruction d'un environnement paramétré comme problème d'inversion ensembliste

Soit un robot se trouvant dans une pièce de forme quadrilatère convexe (figure 3). Nous voulons reconstruire l'environnement du robot en utilisant des données télémétriques. Une telle reconstruction peut rentrer dans le cadre du SLAM (Simultaneous Localisation And Mapping) car on se localise dans l'environnement et à la fois on détermine ses caractéristiques. Cependant, il est à noter que le SLAM classique utilise des amers ponctuels alors que dans notre approche nous utilisons la spécificité d'un environnement (*ex*: le fait que la pièce soit un quadrilatère convexe).

Pour faciliter l'explication de la méthode, nous al-

lons supposer que le robot est immobile. Nous allons aussi reconstruire l'environnement dans le repère lié au robot. La pièce sera donc définie par les coordonnées  $(x_i, y_i)$  de ses quatre coins que nous allons déterminer. (Nota: La pièce pourrait être caractérisée différemment en utilisant les longueurs des murs et les angles entre ces derniers). Nous allons utiliser les données télémétriques comme entrée de l'algorithme. Une donnée télémétrique  $i$  est composée d'une distance au premier obstacle (au mur)  $\tilde{d}_i$  avec une erreur  $\Delta\tilde{d}_i$  et (ii) l'angle correspondant  $\alpha_i$  du faisceau du télémètre par rapport au robot. On note par  $[d_i]$  l'intervalle de mesure de  $d_i$ . Si  $\tilde{d}_i$  est la distance mesurée avec une erreur  $\Delta\tilde{d}_i$ . On a  $[d_i] = [\tilde{d}_i - \Delta\tilde{d}_i, \tilde{d}_i + \Delta\tilde{d}_i]$ . Comme dans l'exemple précédent, nous allons utiliser des relations géométriques dans notre raisonnement.

Soit  $M_i(m_{xi}, m_{yi})$  défini par

$$\begin{cases} m_{xi} = -d_i \sin(\alpha_i) \\ m_{yi} = d_i \cos(\alpha_i). \end{cases}$$

On utilisera la contrainte (voir la section sur l'algorithme d'inversion ensembliste) que  $M_i$  doit appartenir à un des murs. Il en résulte 4 groupes d'équations/inéquations par mesure

Par exemple, pour le segment  $[BC]$  nous trouvons les équations suivantes

$$\begin{cases} \min(xb, xc) < m_{xi} < \max(xb, xc) \\ \min(yb, yc) < m_{yi} < \max(yb, yc) \\ \det \left( \begin{pmatrix} xc - xb \\ yc - yb \end{pmatrix}, \begin{pmatrix} m_{xi} - xb \\ m_{yi} - yb \end{pmatrix} \right) = 0. \end{cases}$$

Il en est de même pour tous les autres segments.

Les relations sur les segments  $[AB], [BC], [CD], [DA]$  définissent une contrainte entre l'entrée (donnée du capteur) et les sorties (domaines des coordonnées des coins de la pièce) qui permettra d'affiner (ou autrement dit contracter) les domaines de ces derniers. La contraction de la sortie pour chaque segment se fera d'une manière indépendante. Ensuite, puisqu'on sait que le point  $M_i$  doit appartenir à un segment (mais on ne sait pas lequel), les domaines finaux des sorties seront l'union des domaines respectifs contractés par les contraintes sur chaque segment. On aura ainsi défini

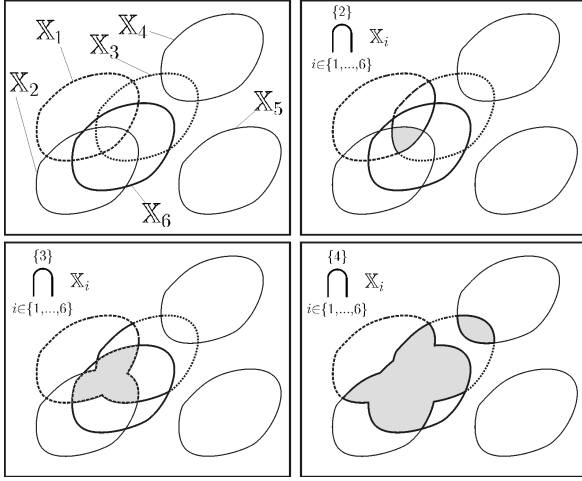


Figure 4: Définition de la  $q$ -intersection

un contracteur global pour chaque mesure. Ensuite, on applique l'algorithme RSIVIA expliqué dans la section 5.

## 4 Un algorithme d'inversion robuste

Soit  $m$  ensembles  $\mathbb{X}_1, \dots, \mathbb{X}_m$  de  $\mathbb{R}^n$ . L'intersection  $q$ -relaxée notée par  $\bigcap_{i \in \{1, \dots, m\}}^{\{q\}} \mathbb{X}_i$  est l'ensemble de tous les  $\mathbf{x} \in \mathbb{R}^n$  qui appartiennent à tous les  $\mathbb{X}_i$ , excepté  $q$  au plus. La figure 4 montre ce concept pour  $m = 6$  et  $q = 2, 3, 4$ . Dans cet exemple nous avons:

$$\bigcap_{i \in \{1, \dots, 6\}}^{\{0\}} \mathbb{X}_i = \bigcap_{i \in \{1, \dots, 6\}}^{\{1\}} \mathbb{X}_i = \emptyset, \quad \bigcap_{i \in \{1, \dots, 6\}}^{\{5\}} \mathbb{X}_i = \bigcup_{i \in \{1, \dots, 6\}} \mathbb{X}_i \text{ and } \bigcap_{i \in \{1, \dots, 6\}}^{\{6\}} \mathbb{X}_i = \mathbb{R}^2. \quad (6)$$

De la même manière, dans le contexte de l'estimation d'ensembles, nous pouvons définir l'ensemble  $q$  relaxé des solutions comme

$$\mathbb{P}^{\{q\}} \stackrel{\text{def}}{=} \bigcap_{i \in \{1, \dots, m\}}^{\{q\}} \{\mathbf{p} \in \mathbb{R}^{n_p}, \exists \mathbf{y} \in [\mathbf{y}](i), \mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}\}. \quad (7)$$

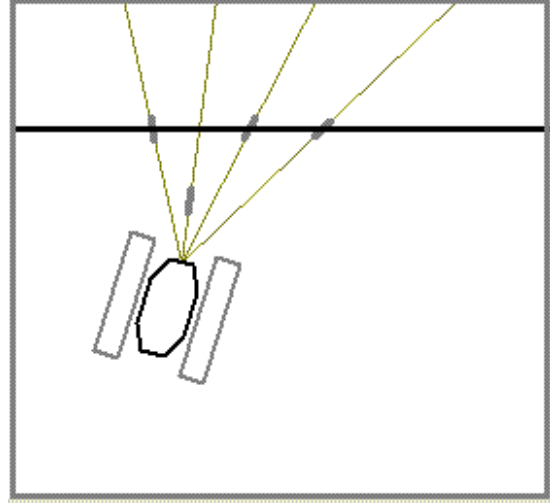


Figure 5: Localisation du robot avec quatre mesures télémétriques

$\mathbb{P}^{\{q\}}$  peut être facilement caractérisé en utilisant l'analyse par intervalle

**Exemple:** Dans la section 2.2, nous avons localisé le robot avec deux mesures. Il est clair qu'une erreur sur l'une des mesures se répercutera sur la solution finale. On augmente donc le nombre de mesures. Pour être plus clair, utilisons 4 mesures dont une est aberrantes. Voir figure 5.

Dans ce cas, si on garde la même définition de  $\mathbb{P}$  comme dans l'équation 1, on obtien l'ensemble vide. Ceci est logique puisque nous sommes consistant qu'avec trois mesures parmi quatre. Nous allons modifier un peu la méthode.

En utilisant le concept de la  $q$ -intersection et l'équation 4 devient

$$\mathbb{P}^{\{1\}} \stackrel{\text{def}}{=} \bigcap_{i \in \{1, \dots, 4\}}^{\{1\}} \mathbb{X}_i. \quad (8)$$

Où les  $\mathbb{X}_i$ , représentés sur la figure 6, sont définis comme dans le cas de deux mesures par l'équation ci dessous

$$\mathbb{X}_i \stackrel{\text{def}}{=} \left\{ (d, \theta) \in \mathbb{D} \mid \frac{d}{\cos(\theta + \alpha_i)} \in [d_i] \right\}$$

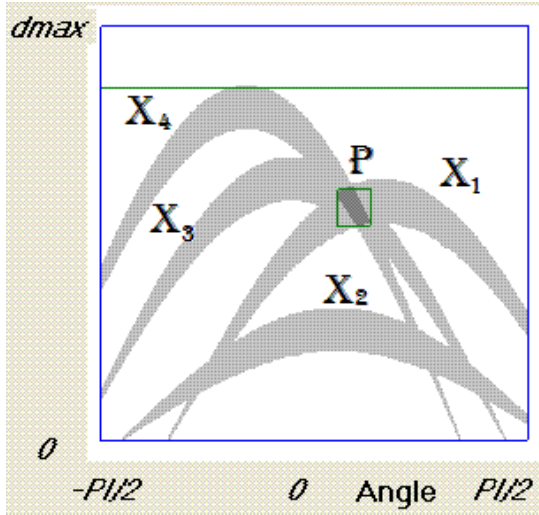


Figure 6: Les différents ensembles solutions représentés dans l'espace de configuration.

où  $[d_i]$  représentent les intervalles de mesures de la distance.

## 5 Algorithme d'inversion ensembliste

Après avoir posé les problèmes, nous proposons une méthode de résolution basée sur la propagation de contraintes. Nous ferons d'abord une petite introduction sur l'analyse par intervalle et les contractions, ensuite, nous expliquerons l'algorithme de résolution RSIVIA ([1]).

### 5.1 Analyse par intervalle

Dans un contexte ensembliste, une variable aléatoire  $x$  de  $\mathbb{R}$  - *exemple* : mesure de capteur, grandeur physique...- est souvent représentée par un intervalle  $[x]$  qui contient le support de sa fonction de densité de probabilité. Cette représentation est évidemment plus pauvre, mais elle représente plusieurs avantages. (i) Puisque un intervalle non singleton est consistant avec une infinité de fonctions de densité de probabilité,

une représentation intervalle est bien adaptée pour la représentation de variables aléatoires avec des fonctions qui ont caractéristiques probabilistes imprécises, d'où la versatilité de l'approche. (ii) Contrairement au calcul probabiliste, une arithmétique peut être développée pour les intervalles ce qui permet de traiter les incertitudes avec une manière sûre et simple même lors de fortes non-linéarités. (iii) Quand des variables aléatoires sont reliées par des contraintes (*i.e.* équations ou inéquations) un processus de propagation de contraintes permet d'avoir des algorithmes polynomiaux efficaces pour calculer des intervalles précis qui contiennent d'une manière garantie les valeurs plausibles pour ces variables aléatoires. En effet, il existe des bibliothèques performantes toutes faites qui résolvent ce genre de problèmes.

### 5.2 Arithmétique des intervalles

Un intervalle est un sous-ensemble fermé et connexe de  $\mathbb{R}$ . Soit deux intervalles  $[x]$  et  $[y]$  et des opérateurs  $\diamond \in \{+, -, \cdot, /\}$ . On définit  $[x] \diamond [y]$  comme le plus petit intervalle qui contient toutes les valeurs possibles pour  $x \diamond y$ , avec  $x \in [x]$  et  $y \in [y]$  (Voir [2]).

$$[x] \diamond [y] = [\inf(\{x \diamond y | x \in [x], y \in [y]\}), \sup(\{x \diamond y | x \in [x], y \in [y]\})]$$

Par exemple

$$\begin{aligned} [-1, 3] + [2, 5] &= [1, 8], \\ [-1, 3] \cdot [2, 5] &= [-5, 15], \\ [-1, 3] / [2, 5] &= [-\frac{1}{2}, \frac{3}{2}]. \end{aligned} \quad (9)$$

pour des fonctions élémentaires  $f$  comme  $\sin, \cos, \dots$  on définit  $f([x])$  comme le plus petit intervalle contenant toutes les valeurs possibles pour  $f(x)$  avec  $x \in [x]$ .

### 5.3 Contractions

Considérons une contrainte  $\mathcal{C}$  (*i.e.*, une équation ou une inéquation), certaines variables  $x_1, x_2, \dots$  impliqués dans  $\mathcal{C}$  qui ont à priori des domaines  $[x_i]$ . L'arithmétique intervalle permet de contracter les domaines  $[x_i]$  sans enlever aucune valeur plausible de  $x_i$ . Par exemple si on considère l'équation  $x_3 = x_1 + x_2$  où les domaines de  $x_1, x_2, x_3$  sont donnés par  $[x_1] =$

$[-\infty, 5], [x_2] = [-\infty, 4]$  and  $[x_3] = [6, \infty]$ . Ces domaines peuvent être contractés à  $[\bar{x}_1] = [2, 5], [\bar{x}_2] = [1, 4]$  et  $[\bar{x}_3] = [6, 9]$ . Le calcul intervalle donnant ce résultat est le suivant

$$\begin{aligned} x_3 = x_1 + x_2 &\Rightarrow z \in [6, \infty] \cap ([-\infty, 5] + [-\infty, 4]) \\ &= [6, \infty] \cap [-\infty, 9] = [6, 9]. \\ x_1 = x_3 - x_2 &\Rightarrow x \in [-\infty, 5] \cap ([6, 9] - [-\infty, 4]) \\ &= [-\infty, 5] \cap [2, \infty] = [2, 5]. \\ x_2 = x_3 - x_1 &\Rightarrow y \in [-\infty, 4] \cap ([6, 9] - [2, 5]) \\ &= [-\infty, 4] \cap [1, \infty] = [1, 4]. \end{aligned}$$

En général, on peut contracter les domaines d'un ensemble de variables liés avec plusieurs contraintes qui sont aussi beaucoup plus complexes (même non linéaires). En effet, il est possible de décomposer une contrainte complexe en un enchainement de contraintes simples et cela est appelé propagation de contraintes ([1]). Un opérateur de contraction est appelé *contracteur*.

## 5.4 Algorithme RSIVIA

L'algorithme qui résout le problème d'inversion ensembliste utilise le principe illustré dans la figure 7:

$\mathbb{X}_i$  est l'ensemble des paramètres (solutions) compatibles avec la  $i$ ème mesure représentée par le vecteur  $y_i$ . La sous-figure (a) représente les ensembles  $\mathbb{X}_i$  avec l'ensemble solution  $\mathbb{X}$  (zone hachurée), qui représente l'intersection  $q$ -relaxée que l'on veut approcher par un pavé. Pour chaque  $i$ , on commence par approcher les ensembles  $[\mathbf{x}] \cap \mathbb{X}_i$  par les pavés  $[\mathbf{x}](i)$  représentés dans la sous-figure (b) par des rectangles en tirets. Ceci est possible en utilisant la relation entre les inconnues et la  $i$ ème mesure  $f^i(x) = y_i$  comme contracteur. Dans la sous-figure (c), les deux pavés gris représentent l'intersection  $q$ -relaxée des pavés  $[\mathbf{x}](i)$ . Ayant fini cette première étape, on traite maintenant la zone hachurée. Dans la sous-figure (d), on se trouve dans la même situation que dans la sous-figure (a). Le pavé courant englobe toujours  $\mathbb{X}$  mais il est plus petit maintenant.

Ce procédé peut être répété une troisième fois comme illustré dans les sous-figures (e) et (f). On convergera donc vers un pavé fixe -sous figure (g)- et

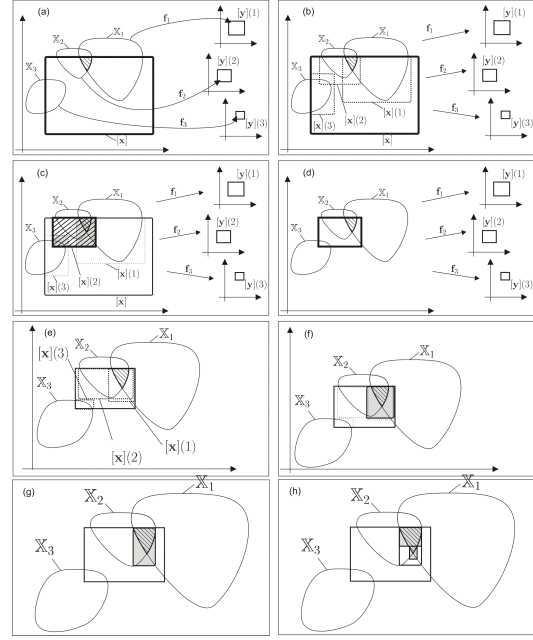


Figure 7: Déroulement de l'algorithme RSIVIA

cela même si la dimension  $n$  de  $x$  est grande. La précision de cette l'approximation peut être contrôlée en faisant plusieurs bisections du pavé courant en sous-pavés et en itérant la procédure de contraction sur tout sous-pavé -la sous-figure (h) montre l'état du pavage après une bissection-. L'algorithme est nommé RSIVIA (*Relaxed Set Inverter Via Interval Analysis*) ([1]).

## 5.5 L'erreur

Si les hypothèses sur le nombre maximal de données aberrantes est respecté, la solution est garantie. L'erreur de la solution calculée converge vers la solution théorique quand le nombre de bisections dans l'algorithme tend vers l'infini. Si des erreurs apparaissent, ce n'est plus le problème de la méthode de résolution mais un problème de modélisation et/ou de conception.(faisabilité). Par exemple, si le problème est symétrique (ex: le cas ou le quadrilatère convexe serait un rectangle.), nous avons l'apparition de

plusieurs solutions dans le cas de la cartographie. Il faudrait par exemple ajouter un capteur pour rompre cette symétrie.

## 5.6 Conclusion

Dans cet article, nous avons vu deux applications de la théorie ensemblistes. Un exemple portait sur la localisation qui donne une certaine autonomie au robot. D'une autre part, l'exemple de reconstruction prouve que l'approche ensembliste pourrait être utilisée dans le SLAM. On a aussi vu quelques propriétés de l'approche ensembliste. Cette dernière a plusieurs avantages comme la robustesse, la versatilité et l'existence de bibliothèques toutes faites pour les programmer.

## References

- [1] L. JAULIN, M. KIEFFER, O. DIDRIT, AND E. WALTER. "Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics". Springer-Verlag, London (2001).
- [2] R. E. MOORE. "Methods and Applications of Interval Analysis". SIAM, Philadelphia, PA (1979).
- [3] S. THRUN, W. BUGARD, AND D. FOX. "Probabilistic Robotics". MIT Press, Cambridge, M.A. (2005).
- [4] E. WALTER AND L. PRONZATO. "Identification of Parametric Models from Experimental Data". Springer-Verlag, London, UK (1997).