

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد- تلمسان

Université Aboubakr Belkaïd- Tlemcen

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du diplôme de **MASTER II**

En : Génie électrique

Spécialité : Automatique

Par : KHALDI née LAOUEDJ Ferial

INVERSION ENSEMBLISTE POUR L'ETALONNAGE D'UN ROBOT

Soutenu publiquement, le 10/11/2020, devant le jury composé de :

Dr. Amel CHOUKCHOU BRAHAM

Maitre de conférences, Université de Tlemcen

Présidente

Dr. Choukri BENSALAH

Maitre de conférences, Université de Tlemcen

Examineur

Dr. Salima BORSALI née KAHOUADJI

Maitre de conférences, Université de Tlemcen

Encadreur

Je tiens à dédier ce modeste travail avant tout :

A ce que j'ai de plus chers au monde, ma mère,

Mon époux,

Mes enfants,

Ilyesse, Riheme,

Nadine et Yazane

Mes sœurs et me frères

Remerciements

Le travail présenté dans ce mémoire a été dirigé sous la direction de Madame Salima BORSALI née KAHOUADJI que je remercie pour son encadrement, son aide, ses directives et ses précieux conseils.

Je remercie les membres du jury Madame Amel CHOUKCHOU BRAHAM et Monsieur Choukri BENSALAH qui m'ont fait l'honneur de participer à l'examen de ce travail.

Je remercie toute personne qui m'a apporté son aide, ne serait-ce que minime.

Résumé

Dans cette étude, une méthode inversion ensembliste utilisant L'algorithme SIVIA sur les intervalles est proposée pour l'étalonnage géométrique d'un robot à 3 degrés de libertés. Cette méthode permet d'encadrer chaque paramètre à estimer à l'intérieur d'un intervalle dont la largeur dépend de l'incertitude sur les mesures collectées, des erreurs de modélisation et du pessimisme inhérent aux méthodes ensemblistes. Le problème de l'étalonnage géométrique est présenté et formalisé comme un problème de satisfaction de contraintes. La plus petite boîte encadrant l'ensemble des solutions n'étant pas forcément obtenue, un ajout de contraintes redondantes basé sur le principe de projections de contraintes est proposé afin d'améliorer les résultats obtenus par l'inversion ensembliste.

Mots clés

L'étalonnage – robotique – inversion ensembliste – arithmétique intervalle

Abstract

In this study, a set inversion method using the SIVIA algorithm on intervals is proposed for the geometric calibration of a robot with 3 degrees of freedom. This method makes it possible to frame each parameter to be estimated within an interval whose width depends on the uncertainty of the collected measurements, on modeling errors and on the pessimism inherent in set methods. The problem of geometric calibration is presented and formalized as a problem of constraint satisfaction. Since the smallest box surrounding the set of solutions is not necessarily obtained, an addition of redundant constraints based on the principle of constraint projections is proposed in order to improve the results obtained by the set inversion.

Keywords

Calibration - robotics - set inversion - interval arithmetic

المخلص

في هذه الدراسة، تم اقتراح طريقة انعكاس مجموعة باستخدام خوارزمية SIVIA على فترات زمنية للمعايرة الهندسية للروبوت مع 3 درجات من الحرية. تتيح هذه الطريقة تأطير كل معلمة ليتم تقديرها خلال فترة زمنية يعتمد عرضها على عدم اليقين في القياسات التي تم جمعها، وعلى أخطاء النمذجة وعلى التشاؤم المتأصل في الأساليب المحددة. يتم تقديم مشكلة المعايرة الهندسية وإضفاء الطابع الرسمي عليها كمشكلة رضا القيد. نظرًا لأنه لا يتم بالضرورة الحصول على أصغر مربع يحيط بمجموعة الحلول، يُقترح إضافة قيود زائدة على أساس مبدأ إسقاطات القيد من أجل تحسين النتائج التي تم الحصول عليها من خلال انعكاس المجموعة.

الكلمات الدالة

المعايرة - الروبوتات - ضبط الانعكاس - حساب الفترات

Table des matières

Table des figures	9
Introduction générale	11
1 L'arithmétique du l'intervalle	14
1-1 Introduction	13
1-2 Notions sur L'arithmétique des intervalles	13
1.2.1 Définition	13
1.2.2 Propriétés des intervalles.....	13
1.2.3 Opérations élémentaires	14
1.2.4 Propriétés algébriques (perdues)	15
1.3 Les pavés	16
1.3.1 Définition	16
1.3.2 Propriétés des pavés	17
1.4 Définition d'un sous pavage	
1.4.1 Fonction d'inclusion	18
1.4.2 Test d'inclusion	18
1.5 Pessimisme.....	19
1.6 Conclusion.....	20
2 Méthodes ensemblistes	22
2.1 Introduction	22
2.2 Opérations sur les ensembles	22
2.2.1 Ensembles et sous-ensembles	22
2.2.2 propriétés.....	23
2.3 Intersection, réunion	24
2.3.1 Intersection.....	24
2.3.2 Réunion.....	24

2.4	Applications	25
2.4.1	Premières définitions	25
2.4.2	Composition d'applications	27
2.5	Image directe et image réciproque	27
2.5.1	Définition	27
2.5.2	Bijections	29
2.6	Algorithme ensembliste	30
2.6.1	Inversion ensembliste.....	31
2.6.2	Contraction.....	32
2.6.2.1	Problème de satisfaction de contraintes	32
2.6.2.2	Notion de consistance.....	33
2.6.2.3	Projection de contraintes	33
2.6.2.4	Sous-résolveurs	34
2.7	Identifications dans le contexte à erreurs bornées.....	36
2.8	Conclusion.....	37
3	La modélisation des robots manipulateurs	39
3.1	Introduction.....	39
3.2	Définitions.....	40
3.3	Description de la géométrie des robots :.....	40
3.3.1	Le système mécanique articulé (S.M.A.)	41
3.3.2	Architecture des robots	43
3.4	Degrés de liberté d'un solide dans l'espace :.....	44
3.4.1	Définition	44
3.4.2	Degrés de libertés	45
3.5	Matrices de transformation homogène	47
3.5.1	Coordonnées homogènes.....	47
3.5.2	Transformation homogène.....	47
3.5.3	Transformation de vecteurs.....	48
3.5.4	Matrice de translation pure homogène.....	49
3.5.5	Matrice de rotation homogène.....	49

3.5.6 Le torseur cinématique.....	50
3.6 Modèle géométrique des robots.....	50
3.6.1 Convention de Denavit -Hartenberg modifiée.....	50
3.6.2 Présentation des différents modèles.....	52
a. Modèle géométrique direct des robots (MGD).....	52
b. Modèle géométrique inverse des robots (MGI).....	53
3.7 Description géométrique du robot d'application.....	55
a. Placement des repères	55
b. Les paramètres géométriques du robot.....	56
3.8 Conclusion.....	56
4 Application	58
4.1 Le principe d'application.....	58
4.2 les résultats de la simulation.....	58
4.3 Modification des 3 axes en mouvement.....	61
4.4 Conclusion	66
Conclusion générale	69
ANNEXE	70
Bibliographies	74

Table des figures

Figure 1.1 Composant de vecteurs intervalles.....	1
Figure 1.2 Effet d'enveloppement conséquent à la rotation d'un pave.....	21
Figure 2.1 Illustration de l'intersection.....	25
Figure 2.2 Illustration de la réunion.....	25
Figure 2.3 Image d'un élément.....	26
Figure 2.4 Antécédents d'un élément.....	27
Figure 2.5 Antécédents par la fonction carrée.....	27
Figure 2.6 Diagramme sagittal d'une composée.....	27
Figure 2.7 Image directe.....	29
Figure 2.8 Image réciproque.....	29
Figure 2.9 Représentation d'une application bijective.....	30
Figure 2.10 Sivia.....	32
Figure 2.11 Contraction d'un ensemble.....	34
Figure 2.12 – Modèle d'identification.....	37
Figure 3.1 Constituants d'un robot.....	40
Figure 3.2 Articulation rotoïde.....	42
Figure 3.3 Articulation prismatique.....	42
Figure 3.4 Structure ouverte simple (a), Structure arborescente(b).....	44
Figure 3.5. a d.d.l. dans l'espace.....	45
Figure 3.5.b Angle d'Euler.....	46
Figure 3.6 -changements de repères.....	47
Figure 3.7 Transformation de point et de vecteur entre 2 repères consécutifs.....	48
Figure 3.8 Robot a structure ouverte simple.....	51
Figure 3.9 Convention et paramètres de Denavit -Hartenberg modifiée.....	51
Figure 3.10 robot plant 3 axes.....	55
Figure 3.11 placement des repères sur le robot.....	55
Figure 4.1 Pavés solutions relatives aux paramètres r_1 , r_2 et r_3 obtenu par Sivia.....	59
Figure 4.2 Pavés solutions relatives aux paramètres r_1 , r_2 et r_3 obtenu par Sivia.....	59
Figure 4.3 Pavés solutions relatives aux paramètres r_1 , r_2 et r_3 obtenu par Sivia.....	60

Figure 4.4 Pavés solutions relatives aux paramètres d'intervalle p obtenu par Sivia.....	61
Figure 4.5 Pavés solutions relatives aux paramètres d'intervalle p après amélioration obtenu par Sivia.....	62
Figure 4.6 Pavés solutions relatives aux paramètres d'intervalle p obtenu par SiviaP.....	63
Figure 4.7 Pavés solutions relatives aux paramètres d'intervalle p obtenu par SiviaP.....	64
Figure 4.8 Pavés solutions relatives aux paramètres p_1, p_2, p_3 testé par un solveur Proj2D.....	65
Figure 4.9 Pavés solutions relatives aux paramètres p_1, p_2, p_3 testé par un solveur Proj2D.....	66

Introduction générale

L'utilisation des robots manipulateurs pour des différentes opérations est devenue un enjeu de développement des robots ces dernières années. L'objectif est de proposer une alternative aux machines-outils. Toutefois, les robots développés présentent des précisions de suivi de trajectoire uniquement de l'ordre du dixième de millimètre, même si leur rigidité a été améliorée par rapport aux robots utilisés pour des opérations moins exigeantes. En effet, les comportements géométriques, statiques et dynamiques de ces robots ne permettent pas de réaliser des usinages avec une grande précision. Le comportement géométrique observé sans sollicitation de la structure par une tâche particulière montre déjà une erreur de suivi de trajectoire de l'ordre du dixième de millimètre. Ce rapport se s'intéresse à l'étude de la précision géométrique de robots manipulateur, et plus particulièrement à la proposition d'une méthodologie de définition du modèle géométrique et d'identification des paramètres géométriques associés avec l'inversion ensembliste.

Les modèles géométriques des robots reposent généralement sur un ensemble d'équations mathématiques permettant de définir la pose de l'effecteur final en fonction des valeurs des variables articulaires et de paramètres géométriques. Les méthodes de modélisation les plus répandues reposent généralement sur le formalisme de Denavit-Hartenberg (DH) [1]. Ce formalisme est enrichi par divers travaux scientifiques afin de s'adapter plus particulièrement à certains types de structures. Toutefois, la complexité des modèles développés ne garantit pas une meilleure précision géométrique. En effet, l'ajout de paramètres géométriques rend le modèle plus sensible aux erreurs d'identification. Il est nécessaire de limiter le nombre de paramètres géométriques aux paramètres influents et identifiables. Les méthodes ensemblistes sont bien adaptées pour l'étalonnage des robots manipulateurs en raison de l'intervalle d'erreur important introduit. Dans ce travail, l'association d'une méthode ensembliste, particulièrement l'inversion ensembliste, à l'arithmétique des intervalles nous permet d'obtenir un intervalle pour les paramètres géométriques qu'on cherche à identifier. Ces intervalles nous assurent un résultat fiable et garanti et contenant à coup sur la valeur réelle.

La suite de ce document est organisée de la manière suivante :

1. Le chapitre 1 est consacré à l'arithmétique des intervalles ainsi que les différentes opérations élémentaires étendues aux intervalles. Certaines propriétés algébriques sont perdues. La définition des pavés, des sous pavés et des fonctions d'inclusions est à la base pour la résolution des systèmes d'équations linéaires et non linéaires par les méthodes ensemblistes [3] [4].
2. Le chapitre 2, nous allons aborder les fondements de base de l'approche ensembliste [5]. Motivée par sa simplicité dans la manipulation et dans les calculs, la représentation par des intervalles (ou vecteurs

intervalles) est choisie pour modéliser les incertitudes influant sur un système donné [8]. Nous présentons particulièrement l'inversion ensembliste utilisée pour l'identification des paramètres du robot plan [9]. Les contracteurs seront introduits, ils seront associés à l'inversion ensembliste afin de diminuer le pessimisme, notion qui a été définie dans ce chapitre 1 [6] [7] [10].

3.- Le troisième chapitre est consacré à la description du robot plan sur lequel nous testons notre méthode ensembliste. Après un rappel des différentes structures en robotique, nous définissons le modèle géométrique direct de ce robot en fonction de ses caractéristiques structurales et obtenu par l'application de la convention de Denavit-Hartenberg[11] [12] .

4.- Le dernier chapitre sera consacré à l'application de l'inversion ensembliste à l'étalonnage d'un robot manipulateur plan trois axes. Nous proposons ensuite une amélioration de la méthode d'identification du robot en rajoutant un contracteur basé sur le principe de projection de contrainte.

Chapitre I

L'arithmétique par l'intervalle

1 L'arithmétique du l'intervalle

1-1 Introduction :

La majorité des méthodes ensemblistes développées pour l'estimation sont basées sur un outil de calcul qui est l'arithmétique des intervalles. Initialement, cette arithmétique est développée pour quantifier les erreurs numériques dues à la représentation finie des nombres réels sur un calculateur [3] et [4]. L'analyse par intervalles est devenue un outil de résolution numérique globale et garanti des problèmes d'estimation.

Durant ces dernières décennies, différentes méthodes basées sur l'analyse par intervalles ont été développées dans le but d'étudier et de quantifier les incertitudes (numériques ou physiques) influant sur les données manipulées.

1.2 Notions sur L'arithmétique des intervalles

1.2.1 Définition

Un intervalle $[X]$ est un ensemble connexe, borné, fermé de borne inférieure \underline{x} de borne supérieure \bar{x} qui sont des nombres réels, l'intervalle $[x]$ est représenté de manière suivante

$$[X] = [\underline{x}, \bar{x}] := \{\forall x \in \mathbb{R}, \underline{x} \leq x \leq \bar{x}\} \quad (1 - 1)$$

- L'ensemble de tous les intervalles \mathbb{R} est noté \mathbb{IR} , soit $[x] \in \mathbb{IR}$.
- Un intervalle $[x]$ est dit dégénéré si les bornes supérieure et inférieure sont identiques.

Exemple :

$[1,4]$, $\{1\}$, $]-\infty,3]$, \mathbb{R} , \emptyset sont des intervalles, alors que $]1,3[$, $[5,1]$, $[2,3] \cup [5,6]$ ne sont pas des intervalles car ils ne vérifient pas la définition donnée ci-dessus.

1.2.2 Propriétés des intervalles

Tout nombre réel x sera confondu avec l'intervalle $[\underline{x}, \bar{x}]$ correspondant. Les intervalles seront désormais repérés par des caractères majuscules et les caractères minuscules désigneront des nombres réels ou entiers.

$$\{X = [1,3] \text{ ou } = \left[\frac{\pi}{2}, \frac{3\pi}{2}\right]\} \quad (1 - 2)$$

- a) *Borne inférieure de $[X]$* : $\inf([X]) = \underline{x}$
- b) *Borne supérieure de $[X]$* : $\sup([X]) = \bar{x}$

c) *Milieu d'intervalle de* $[X]$: fonction définie de \mathbb{I} dans \mathbb{IR}

$$\begin{cases} mid[X] = \frac{\bar{x} + \underline{x}}{2}, \quad \forall X \in \mathbb{IR} \\ \text{par extention aux forme vectorielle} \\ mid([X]) = (mid([x_1]), \dots, \dots, \dots, mid([x_n]))^T X \in \mathbb{IR}^n \end{cases} \quad (1 - 3)$$

d) *Largeur d'intervalle* $([X])$: Fonction définie de \mathbb{I} dans \mathbb{IR} ou dans le cas vectoriel de \mathbb{I}^n dans \mathbb{IR}^n

$$\begin{cases} w([X]) = \bar{x} - \underline{x}, \quad \forall X \in \mathbb{IR} \\ \text{par extention aux forme vectorielle} \\ w([X]) = (w([x_1]), \dots, \dots, \dots, w([x_n]))^T X \in \mathbb{IR}^n \end{cases} \quad (1 - 4)$$

e) *Rayon de l'intervalle* $([X])$: $rad([X]) = \frac{\bar{x} - \underline{x}}{2}, \forall x \in \mathbb{IR}$ (1 - 5)

1.2.3 Opérations élémentaires

Les opérations mathématiques élémentaires sont étendues aux intervalles, le résultat d'une opération entre deux intervalles obtenus en travaillant seulement sur leurs bornes.

Soit $[x], [y] \in \mathbb{IR}$, et une opération binaire $\odot \in \{+, -, *, /\}$

On définit : $[x] \odot [y] = \{ x \odot y \mid x \in [X] \mid y \in [Y] \}$

Définition : le résultat d'une opération \odot entre deux intervalles $\begin{cases} [X] = [\underline{x}, \bar{x}] \\ [Y] = [\underline{y}, \bar{y}] \end{cases}$ (1 - 6)

- $[X] + [Y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$ (1 - 7)
- $[X] - [Y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$ (1 - 8)
- $[X] * [Y] = [\min(\underline{x} * \underline{y}, \underline{x} * \bar{y}, \bar{x} * \underline{y}, \bar{x} * \bar{y}), \max(\underline{x} * \underline{y}, \underline{x} * \bar{y}, \bar{x} * \underline{y}, \bar{x} * \bar{y})]$ (1 - 9)
- $[X]^2 = \begin{cases} [\min(\underline{x}^2, \bar{x}^2), \max(\text{idem})] \text{ si } 0 \notin [X] \\ [0, \max(\underline{x}^2, \bar{x}^2)] \text{ si } 0 \in [X] \end{cases}$ (1 - 10)

$$\bullet \left[\frac{1}{\underline{x}} \right] = \left[\min\left(\frac{1}{\underline{x}}, \frac{1}{\bar{x}}\right), \max(\text{idem}) \text{ si } 0 \notin [X] \right] \quad (1-11)$$

$$\bullet \left[\frac{X}{Y} \right] = [\underline{x}, \bar{x}] * \frac{1}{[\underline{y}, \bar{y}]} \text{ si } 0 \notin [Y] \quad (1-12)$$

$$\bullet e^{[X]} = [e^{\underline{x}}, e^{\bar{x}}] \quad (1-13)$$

Exemple :

Expression polynomiale :

$$\begin{cases} fx = x^3 - 2x^2 + x - 3, \\ x = [-5, 2] \end{cases} \quad (1-14)$$

$$\begin{aligned} fx &= [-5, 2]^3 - 2[-5, 2]^2 + [-5, 2] - 3 \\ &= [-125, 8] - 2[0, 25] + [-5, 2] - 3 \\ &= [-125, 8] - [0, 50] + [-5, 2] - 3 \\ &= [-183, 7] \end{aligned}$$

Démonstration :

$$\begin{aligned} &[-5, 2]^3 \neq [-5, 2]^2 * [-5, 2]^1 \\ &[-5, 2]^3 = [-5, 2]^n \quad \text{tel que } n = 3 \end{aligned}$$

En effet :

$$X^n = \begin{cases} [1, 1] \text{ si } n = 0 \\ \left\{ \begin{array}{l} [(\underline{x})^n, (\bar{x})^n] \text{ si } \underline{x} \geq 0 \text{ ou } n \text{ impair} \\ [(\bar{x})^n, (\underline{x})^n] \text{ si } \bar{x} \leq 0 \text{ et } n \text{ pair} \end{array} \right. \\ [0, \max((\underline{x})^n, (\bar{x})^n)] \text{ si } 0 \in [X] \text{ et } n \text{ pair} \end{cases}$$

1.2.4 Propriétés algébriques (perdues)

- La soustraction n'est pas la réciproque de l'addition

$$\begin{aligned} \text{si } x = [2, 3] \text{ et } y = [-3, 5] &\Rightarrow \begin{cases} z = (x + y) \\ = [-1, 8] \end{cases} \\ &\begin{cases} (z - y) = [-1, 8] - [-3, 5] = [-6, 11] \\ x = [2, 3] \in (z - y) = [-6, 11] \end{cases} \end{aligned}$$

Ou encore :

$$(x - x) = [2, 3] - [2, 3] = [-1, 1] \neq 0$$

- La division n'est pas la réciproque de la multiplication

$$\text{si } x = [2,3] \Rightarrow \frac{x}{x} = \frac{[2,3]}{[2,3]} = \left[\frac{2}{3}, \frac{3}{2}\right] \ni 1$$

➤ La multiplication n'est pas égale à l'élevation carrée.

$$\text{si } x = [-3,2] \begin{cases} [x] * [x] = [-6,9] \\ [x]^2 = [0,9] \end{cases}$$

➤ La multiplication est sous-distributive par rapport à l'addition

$$\text{si } \begin{cases} x = [-2,3], y = [1,4], z = [-2,1] \\ x * (y + z) \neq (x * y) + (x * z) \end{cases}$$

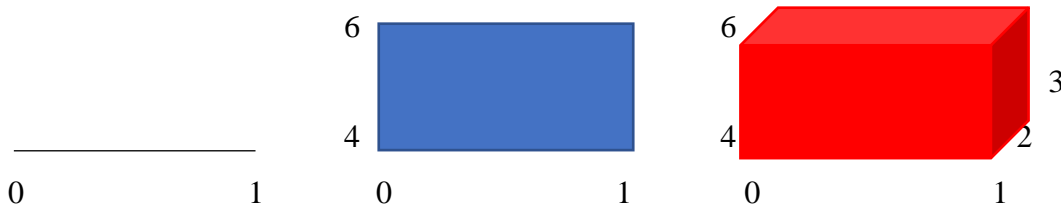
1.3 Les pavés

1.3.1 Définition

Soit $\mathbb{I}\mathbb{R}^n$ l'ensemble des intervalles de \mathbb{R}^n un pavé $[x]$ est un vecteur intervalle de dimension(n) représenté de la manière suivante :

$$([X]) = ([\underline{x}_1, \bar{x}_1] \dots \dots \dots [\underline{x}_n, \bar{x}_n]) \quad (1 - 14)$$

Les opérateurs élémentaires définis précédemment pour les intervalles sont également redéfinis pour les vecteurs d'intervalles :



$$x = [0,1]x = \begin{pmatrix} [0 & , & 1] \\ [4 & , & 6] \end{pmatrix} x = \begin{pmatrix} [0 & , & 1] \\ [4 & , & 6] \\ [2 & , & 3] \end{pmatrix}$$

FIGURE 1-1 : Composant de vecteurs intervalles

1.4.1 Fonction d'inclusion

Considérons deux ensembles ainsi que deux récipients x & y , contenant les deux ensemble x & y respectivement, soit un sous ensemble x_1 de x si $x_1 \in X$, on peut étendre une fonction $F : X \Rightarrow Y$ Aux récipients tel que :

$$[F](X_1) = \{F(x) | x \in X_1\}$$

Et encore $[F](X_1) \supset F(X_1)$ (1 -16)

Ce qui introduit la notion de fonction d'inclusion, considérons une fonction \mathcal{F} de $\mathbb{R}^n \Rightarrow \mathbb{R}^m$, la fonction intervalle $[F]$ de $\mathbb{I}\mathbb{R}^n \Rightarrow \mathbb{I}\mathbb{R}^m$ est une fonction d'inclusion

$$F \text{ si } \begin{cases} \forall [x] \in \mathbb{R}^n, F[x] \subseteq [F]([x]) \\ \text{ou} \\ [F]([x]) \supset (\{F(x) | x \in [x]\}) \end{cases} \quad (1 - 17)$$

Propriétés

- La convergence : Une fonction d'inclusion $[F]$ d'une fonction F est convergente si pour une suite de pavé $[x]_k$

$$\lim_{i \rightarrow \infty} w([x]_k) = 0 \Rightarrow \lim_{i \rightarrow \infty} w([F][x]_k) = 0 \quad (1 - 18)$$

- La monotonie : Une fonction d'inclusion $[F]$ d'une fonction F est monotone si : $[x] \subset [y] \Rightarrow [F]([x]) \in [F]([y])$ (1--19)

1.4.2 Test d'inclusion :

L'une des techniques utilisées pour réduire l'effet du pessimisme défini précédemment est la représentation par des fonctions d'inclusion dont l'efficacité est explicitée en s'appuyant sur des exemples illustrant cet effet :

En définit l'ensemble des intervalles booléens $IB \in \mathbb{I}\mathbb{R}$, telle que :

$$IB = \{\phi, [0], [1], \dots \dots \dots, [0,1]\}$$

Ou l'élément $\begin{cases} [\Phi]: \text{pour impossible} & [\mathbf{0}]: \text{pour faux} \\ [1]: \text{pour vrai} & [\mathbf{0}, \mathbf{1}]: \text{pour indetermine} \end{cases} \quad (1 - 20)$

On définit une fonction test $t: \mathbb{R}^n \rightarrow B$ avec $B = \{[0], [1]\}$

Comme pour les réels, la notion de fonction $[t]: \mathbb{IR}^n \rightarrow \text{IB}$ est appelée test d'inclusion pour :
 t si $\forall [x] \in \mathbb{R}^n$

$$\begin{cases} [t]([x]) = 1 \Rightarrow \forall x \in [x]: t(x) = 1 \\ [t]([x]) = 0 \Rightarrow \forall x \in [x]: t(x) = 0 \end{cases} \quad (1 - 21)$$

Exemple : Soit le test $t: \mathbb{R}^2 \rightarrow \{0,1\}$

$$\begin{cases} (\mathbf{x}_1, \mathbf{x}_2)^t \rightarrow (x_1 + x_2 \leq 10) \\ \text{ce qui veut dire :} \\ \mathbf{t}(\mathbf{x}) = \begin{cases} 1 : \text{si}(x_1 + x_2) \leq 10 \\ 0 : \text{si}(x_1 + x_2) > 10 \end{cases} \end{cases}$$

La fonction d'inclusion t est définie par :

$$[t](x) = \begin{cases} [1] : & \text{si}(x_1 + x_2) \leq 10, \\ [0] : & \text{si}(x_1 + x_2) > 10, \\ [0,1] : & \text{ailleurs.} \end{cases}$$

On peut réécrire aussi : $[t](x) \Leftrightarrow ([x_1], [x_2]) \leq 10$ (1 - 22)

1.5 Pessimisme

Lorsque le résultat des opérations entre intervalles n'est pas minimal, on parle alors de pessimisme, le pessimisme est une conséquence de deux phénomènes, le premier et le phénomène d'enveloppement, il provoque une surestimation du pavé résultant, l'exemple suivant permet de mieux comprendre cet effet :

Exemple :

Soit le pavé $[x] = [x_1] \times [x_2]$, si l'on fait subir à ce pavé des rotations successives à l'aide de la matrice A définie par :

$$\Rightarrow A = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \quad \text{avec } \theta = \frac{\pi}{4} \quad (1 - 23)$$

La représentation de ce pavé image est un rectangle qui l'enveloppe, Après plusieurs rotations, ce rectangle augmente, ce qui provoque une surestimation du résultat finale.

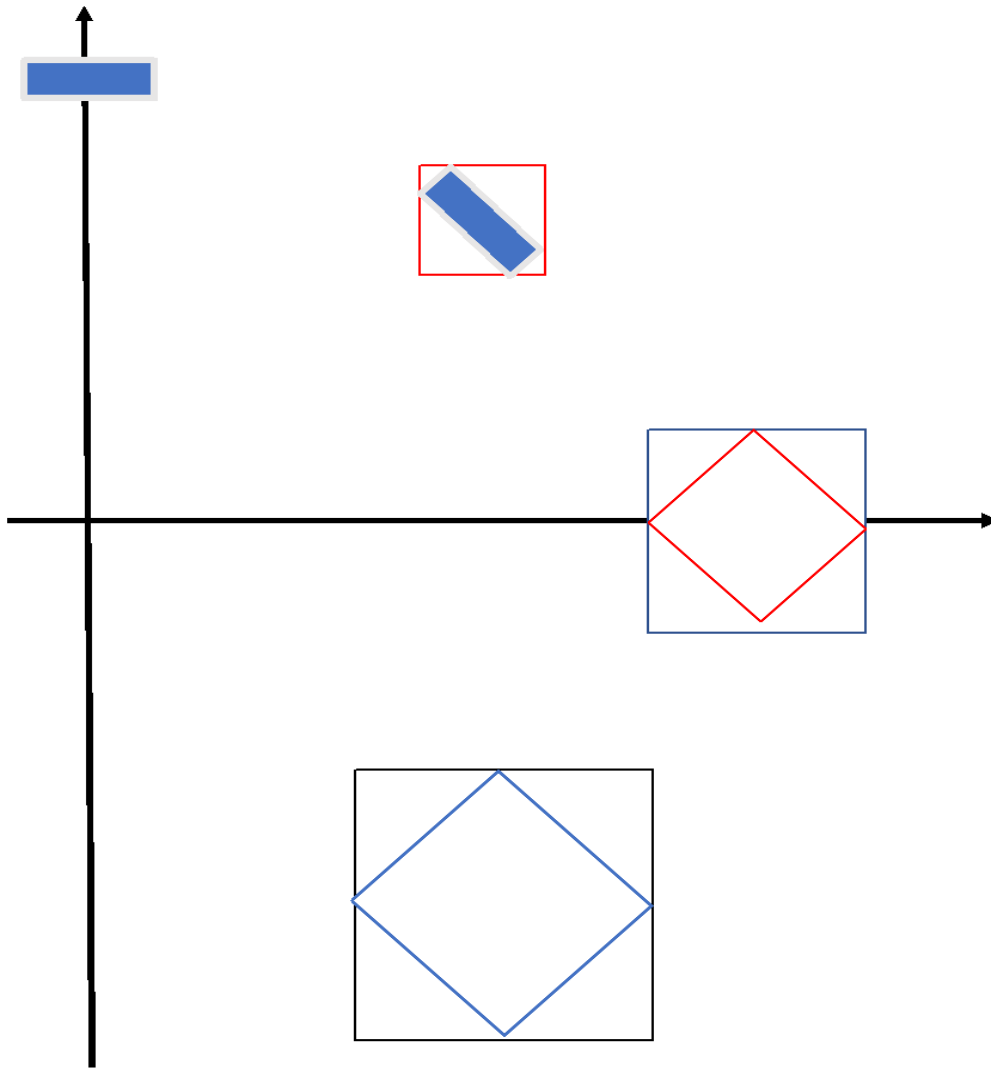


FIGURE 1-2 : Effet d'enveloppement conséquent à la rotation d'un pave

Le deuxième phénomène et celui de dépendance appelé aussi phénomène de multiconcurrences, le phénomène de dépendance suggère que chaque variable est considérée différente, en d'autres termes, ce pessimisme est dû au fait que l'ensemble : $X - Y = \{x - y \text{ , } x \in X | y \in Y\}$ est différent de l'ensemble $\{x - x | x \in X\} = 0$

Exemple: soit $[x] = [-1,1]$ alors :

$$[x] - [x] = [-1,1] - [-1,1] = [-2,2] \neq 0$$

1.6 Conclusion

Ce chapitre a été consacré à la présentation de l'outil fondamental qui est l'arithmétique des intervalles pour la suite de notre travail. En premier lieu, nous nous sommes intéressés à l'arithmétique des intervalles ainsi qu'aux principaux problèmes rencontrés lors de la manipulation des intervalles. Nous avons souligné le fait que nous manipulons des vecteurs intervalles (pavés) engendre un

pessimisme dû au phénomène d'enveloppement ainsi qu'au phénomène de dépendance. Puis, nous avons présenté des outils mettant en œuvre l'arithmétique des intervalles pour manipuler et caractériser de façon approchée et garantie des ensembles de valeurs.

Chapitre II

Méthodes ensemblistes

2.1 Introduction

De nombreuses méthodes en mathématique ou dans les sciences pour l'ingénieur ont été développées pour résoudre des problèmes dont les données qui interviennent sont ponctuelles et dont la solution est aussi ponctuelle, ce type de problème est appelé « problème ponctuel ». Généralement les données qui interviennent dans l'énoncé d'un problème sont entachées d'incertitudes, introduites par exemple par le calculateur lors de la résolution (incertitudes numériques) ou tout simplement par l'environnement expérimental (appareils de mesure, bruit et perturbations sur les mesures...). À cet effet les méthodes développées pour la résolution des problèmes ponctuels ne permettent pas (sauf pour une classe très réduite de problèmes) de trouver de façon garantie un résultat global aussi proche que désiré de la solution du problème, ainsi nous ne savons pas quelle validité à accorder à la solution obtenue. Une reformulation ensembliste du problème ponctuel permettra de concevoir de nouvelles méthodes capables de caractériser de façon garantie et globale le résultat de la solution. Dans cette représentation ensembliste des incertitudes, nous cherchons non pas une solution ponctuelle mais un « ensemble solution ». Un problème qui admet pour solution un ensemble est appelé « problème ensembliste ».

Nous allons en premier lieu, présenter nos motivations qui nous ont permis d'entreprendre l'approche ensembliste dans notre travail, Nous allons constater que l'approche ensembliste offre une meilleure modélisation des incertitudes. Les fondements de base de cette approche sont présentés, tels que les définitions et les opérations entre les ensembles. Enfin on présente le principe de l'inversion ensembliste par arithmétique des intervalles (SIVIA), souvent utilisé pour des problèmes non linéaires.

2.2 Opérations sur les ensembles

Cette première partie sera consacrée à la présentation des diverses opérations que l'on peut effectuer sur des ensembles, en particulier l'intersection, la réunion et le passage au complémentaire

2.2.1 Ensembles et sous-ensembles

Les objets qui constituent un ensemble seront appelés les éléments de cet ensemble. En général, nous désignerons les ensembles avec des majuscules et leurs éléments avec des minuscules.

Notation : Pour signifier que x est un élément d'un ensemble E , nous noterons $x \in E$. Le symbole \in le sigle d'appartenance. Au contraire, si x n'est pas un élément d'un ensemble E , nous noterons $x \notin E$.

Il y a deux moyens d'écrire les ensembles, en extension et en compréhension.

Exemple 2.1. Ensemble écrit en extension

Ensemble des cinq continents : {Afrique, Amérique, Europe, Asie, Océanie}.

Dans le cas d'un ensemble infini on pourra avoir recours à des points de suspension. Par exemple, l'ensemble des entiers naturels peut s'écrire {0, 1, 2 ...}

Exemple 2.2. Ensemble écrit en compréhension

Un ensemble composé de réels vérifiant une certaine inégalité : $x \in \mathbb{R}, x^2 \geq 8$.

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} Et $\{n \in \mathbb{N}, 0 \leq n \leq 9\}$ désignent le même ensemble.

Nous allons maintenant définir quelques ensembles très particuliers.

Définitions : L'ensemble vide est un ensemble qui ne contient aucun élément. Il est unique et est noté \emptyset .

Un singleton est un ensemble constitué d'un seul élément, et donc de la forme

$E = \{x\}$. On prendra soin de distinguer l'élément x de l'ensemble $\{x\}$.

Une paire est un ensemble constitué de deux éléments, et donc de la forme $E = \{x, y\}$.

En guise d'exemples, présentons les ensembles les plus couramment utilisés en mathématiques élémentaires.

Exemple 2.3. Les ensembles numériques usuels

Ensemble des entiers naturels : $\mathbb{N} = \{0, 1, 2, 3, \dots\}$.

Ensemble des entiers relatifs : $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$.

Ensemble des rationnels : $\mathbb{Q} = \frac{p}{q}$

$Q, p \in \mathbb{Z}, q \in \mathbb{N}, q \neq 0$.

Ensemble des nombres réels : \mathbb{R} .

Ensemble des nombres complexes : \mathbb{C} .

Définition : On dit qu'un ensemble A est inclus dans un ensemble B si tous les éléments de A sont éléments de B . On note alors $A \subset B$.

On peut aussi utiliser les formulations A est un sous-ensemble de B ou encore A est une partie de B .

Avec les notations du premier chapitre, l'inclusion peut également s'écrire comme une **implication** :

$$A \subset B \Leftrightarrow (\forall x, x \in A \Rightarrow x \in B) \quad (2-1)$$

2.2.2 Propriétés

1. Pour tous ensemble A , on a $A \subset A$ et $\emptyset \subset A$.

2. Soient A et B deux ensembles.

On a
$$A = B \Leftrightarrow ((A \subset B) \wedge (B \subset A)) \quad (2-2)$$

Cette égalité justifie la méthode dite de double inclusion : Pour montrer que deux ensembles sont égaux, on montre que le premier est inclus dans le second et que réciproquement le second est inclus dans le premier.

2.3 Intersection, réunion

Cette sous-partie sera consacrée à l'étude de deux opérations binaires portant sur les ensembles, l'intersection et la réunion.

2.3.1 Intersection

Définition : On appelle intersection de deux ensembles A et B , l'ensemble noté $A \cap B$ constitué des éléments qui appartiennent à A et à B .

Avec les notations du premier chapitre, l'intersection peut également s'écrire comme une conjonction :

$$A \cap B = \{x, (x \in A) \wedge (x \in B)\} \quad (2-3)$$

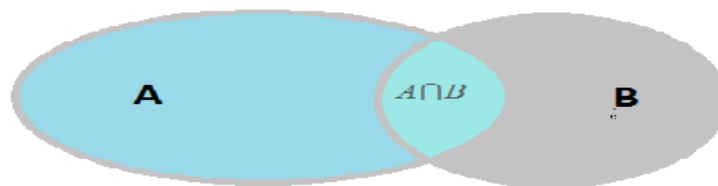


Figure 2.1. Illustration de l'intersection

2.3.2 Réunion

On appelle réunion de deux ensembles A et B l'ensemble noté $A \cup B$ constitué des éléments qui appartiennent à A ou à B .

Avec les notations du premier chapitre, la réunion peut également s'écrire comme une disjonction :

$$A \cup B = \{x, (x \in A) \vee (x \in B)\} \quad (2-4)$$

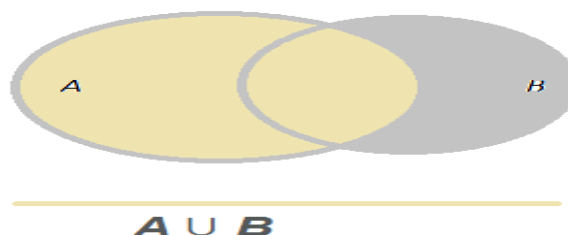


Figure 2.2. Illustration de la réunion

Voici quelques propriétés vérifiées par l'opération de réunion.

Propriétés

Soient A , B et C des ensembles.

On a :

$A \cup B = B \cup A$, la réunion est commutative.

$(A \cup B) \cup C = A \cup (B \cup C)$, la réunion est associative.

$A \subset (A \cup B)$ et $B \subset (A \cup B)$. (2-5)

$A \cup A = A$.

$A \cup \emptyset = A$, \emptyset est un élément neutre pour la réunion.

Si $A \subset B$, alors $A \cup B = B$.

Un élément neutre pour une opération binaire est défini par le fait que le résultat de l'opération entre l'élément neutre et n'importe quel élément est-ce dernier. En quelque sorte il n'influe pas sur le résultat de l'opération.

Comme par exemple 1 pour la multiplication des réels et donc l'ensemble vide pour la réunion.

2.4 Applications

Nous allons dans cette seconde partie définir la notion d'application, puis nous intéresser aux cas particuliers des surjections : injections et bijections.

2.4.1 Premières définitions

Soient E et F deux ensembles.

Définir une application f de E dans F consiste à associer à tout élément x de E un élément unique y de F .

Cet élément y , noté $f(x)$, est appelé l'image de x par f .

L'élément x est un antécédent de y par f .

E est l'ensemble de départ de f et F en est l'ensemble d'arrivée.

Comme lors de la première partie, nous réaliserons régulièrement des schémas afin d'illustrer les concepts nouvellement définis.

Pour représenter une application, le schéma utilisé s'appelle un diagramme sagittal.

En voici un premier à propos de la notion d'**image**.

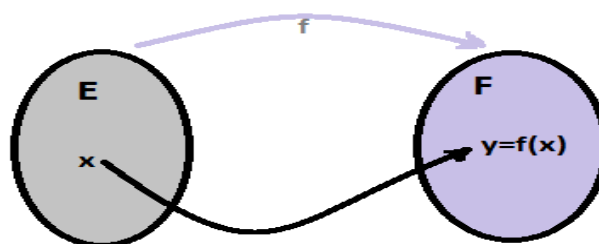


Figure 2.3. Image d'un élément

Ici, y est l'image de x par f , $y=f(x)$.

Bien comprendre que la définition précédente implique que tout élément de l'espace de départ possède une image et que celle-ci est unique.

Et en voici un second pour bien visualiser la définition d'antécédent.

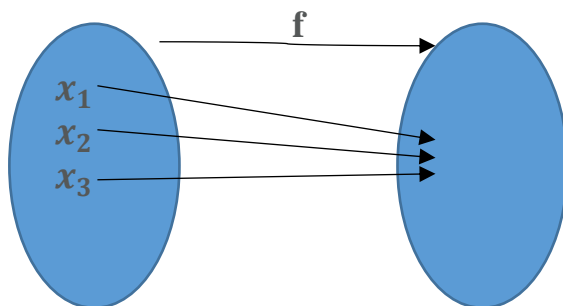


Figure 2.4. Antécédents d'un élément

Dans ce cas de figure, x_1 , x_2 et x_3 sont des antécédents de y par f ,

$$f(x_1) = f(x_2) = f(x_3) = y.$$

Bien comprendre que la définition précédente implique que tout élément de l'espace d'arrivée peut avoir zéro ou plusieurs antécédents.

Pour notre premier exemple concret, nous allons utiliser la très connue fonction carrée.

Exemple 2.4. Antécédents par la fonction carrée

Considérons l'application f de \mathbb{R} dans \mathbb{R} définie par $f(x) = x^2$.

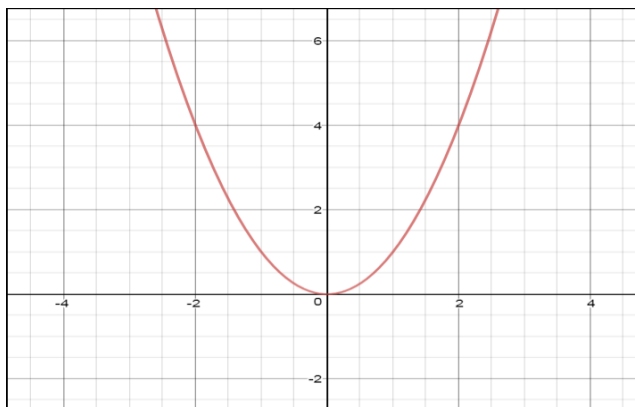


Figure 2.5 : Antécédents par la fonction carrée

On cherche à déterminer les antécédents de n'importe quel réel par f . Pour cela on doit distinguer les trois cas de figure suivants :

Tout réel strictement négatif n'admettra aucun antécédent dans \mathbb{R} par f .

0 admet un unique antécédent.

Tout réel strictement positif admet deux antécédents dans \mathbb{R} par f .

La définition suivante est celle d'une application bien particulière, elle laisse en effet les éléments d'un ensemble inchangés.

Définition : Soit E un ensemble l'application de E dans E qui a un élément x associe x est appelée application identité de E . On la note généralement IdE .

Par définition, on a donc $\forall x \in E, IdE(x) = x$

2.4.2 Composition d'applications

Il est tout à fait possible d'enchaîner les applications, c'est-à-dire en appliquant une seconde aux images provenant d'une première.

Définition :

Soient E, F et G trois ensembles, f une application de E dans F et g une application de F dans G l'application de E dans G notée $g \circ f$ et définie par

$\forall x \in E, (g \circ f)(x) = g(f(x))$ est appelée composée de f par g .

➤ A l'oral on prononce "g rond f".

Représentons cette opération de composition par un diagramme sagittal.

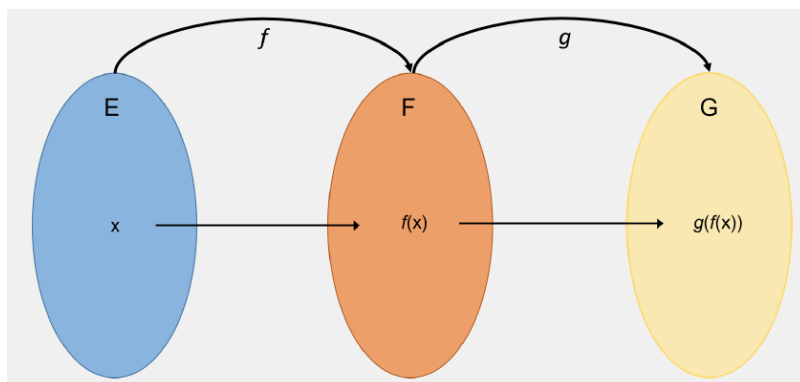


Figure 2.6. Diagramme sagittal d'une composée

2.5 Image directe et image réciproque

On va dans cette sous-partie donner des définitions ne portant plus seulement sur des éléments, mais sur des sous-ensembles des espaces de départ et d'arrivée. Plus précisément, on va voir comment ils sont transformés par l'application.

2.5.1 Définitions :

Soient E et F deux ensembles, et f une application de E dans F .

Soit A un sous-ensemble de E et B un sous-ensemble de F .

On appelle image directe de A par f l'ensemble noté $f(A)$ des images par f des éléments de A . Il s'agit donc d'un sous-ensemble de F dont voici l'expression littérale

$$f(A) = \{y \in F, \exists x \in A, y = f(x)\} = \{f(x), x \in A\} \quad (2-6)$$

On appelle image réciproque de B par f l'ensemble noté $f^{-1}(B)$ des antécédents par f des éléments de B . Il s'agit donc d'un sous-ensemble de E dont voici l'expression littérale

$$f^{-1}(B) = \{x \in E, f(x) \in B\} \quad (2-7)$$

Des diagrammes sagittaux nous permettront de mieux appréhender ces deux définitions un peu plus complexes que les précédentes.

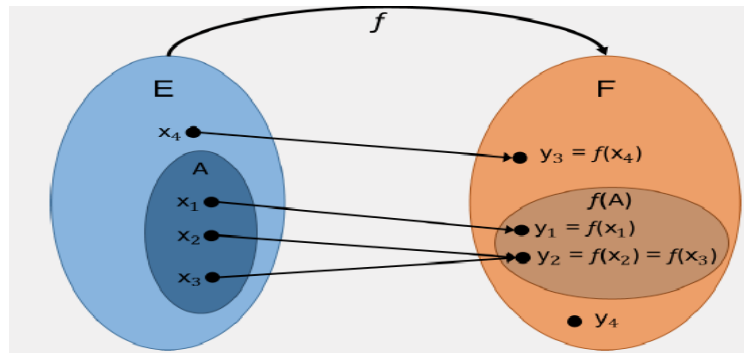


Figure 2.7. Image directe

L'image directe par f du sous-ensemble A de E constitué des éléments x_1 , x_2 et x_3 est le sous-ensemble de F composé des images de ces trois éléments, y_1 et y_2 .

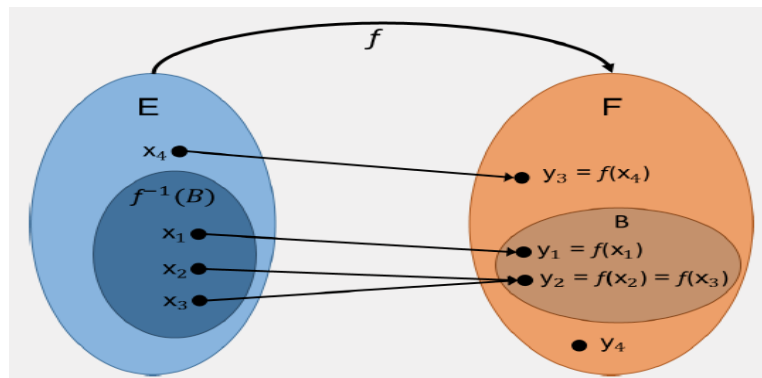


Figure 2.8. Image réciproque

L'image réciproque par f du sous-ensemble B de F constitué des éléments y_1 et y_2 est le sous-ensemble de E composé des antécédents de ces deux éléments,

x_1, x_2 et x_3

Utilisons de nouveau la fonction carrée comme exemple.

Exemple 2.5 :

Images directes et images réciproques par la fonction carrée

Considérons l'application f de \mathbb{R} dans \mathbb{R} définie par $f(x) = x^2$.

Nous allons exprimer quelques images directes et réciproques de sous-ensembles de \mathbb{R} par f . Que le lecteur prenne bien le temps de comprendre chaque cas de figure, en s'aidant si besoin est de la représentation graphique de la fonction carré rappelée dans l'exemple de la sous-partie 2.1.

$$f([1; 2]) = [1; 4].$$

$$f([- 4; - 3]) = [9; 16].$$

$$f([- 1 ; 2]) = [0; 4].$$

$$f^{-1}([0 ; 4]) = [- 2; 2].$$

$$f^{-1}([1 ; 4]) = [- 2 ; - 1] \cup [1 ; 2].$$

$$f^{-1}([- 12 ; - 4]) = \emptyset.$$

$$f^{-1}([- 3; 16]) = [- 4; 4].$$

2.5.2 Bijections

La notion de bijection va regrouper celles de surjection et d'injection.

Définition :

Soient E et F deux ensembles, et f une application de E dans F .

On dit que f est bijective si tout élément de l'ensemble d'arrivée admet un antécédent et un seul par f .

On dit aussi que f réalise une bijection de E sur F .

Réalisons le diagramme sagittal.

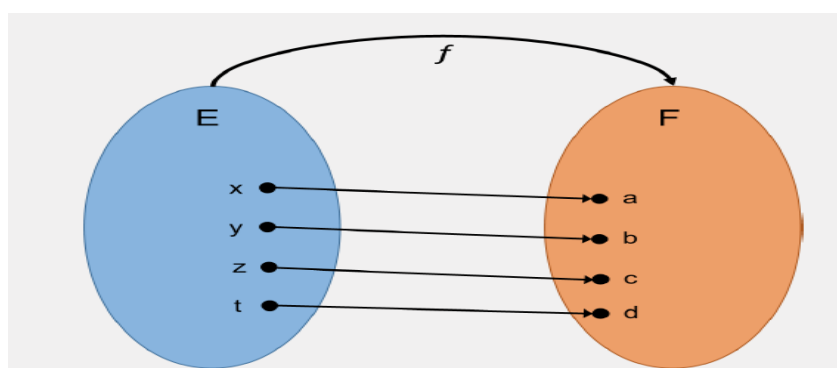


Figure 2.9. Représentation d'une application bijective

L'application f est bijective car tous les éléments de l'ensemble d'arrivée ont exactement un antécédent par f .

Comme nous en avons pris l'habitude, utilisons la fonction carrée comme exemple.

Exemple 2.6 : Bijective de la fonction carrée

L'application f de \mathbb{R} dans \mathbb{R}_+ définie par $f(x) = x^2$ n'est pas bijective car l'élément 4 de l'espace d'arrivée admet deux antécédents par f , à savoir - 2 et 2.

L'application g de \mathbb{R}_+ dans \mathbb{R}_+ définie par $f(x) = x^2$ est par contre bijective.

Obtenir la petite caractérisation suivante d'une application bijective est très simple.

Propriété.

En pratique, on se sert toujours de l'assertion précédente pour montrer qu'une application est bijective.

Si l'on dispose d'une application bijective f , on peut très bien imaginer une application qui à tout élément de l'ensemble d'arrivée associe son unique antécédent par f . Cette remarque conduit à la définition suivante.

Définition :

Soient E et F deux ensembles, et f une application bijective de E dans F .

On appelle application réciproque de f , notée f^{-1} , l'application de F dans E définie pour tout y de F par

$$x = f^{-1}(y) \Leftrightarrow (y = f(x)) \quad (2-8)$$

Le terme inverse est parfois employé à la place de réciproque mais il s'agit d'une incorrection.

L'application réciproque d'une application bijective f réalise donc en quelque sorte le travail opposé à celui de f . Il n'est donc pas surprenant que leur composition soit l'application identité.

Propriété

Soient E et F deux ensembles, f une application bijective de E dans F et f^{-1} son application réciproque.

On a
$$f^{-1} \circ f = Id_E \text{ et } f \circ f^{-1} = Id_F \quad (2-9)$$

Reprenons une dernière fois la fonction carrée en exemple.

Exemple 2.7. Réciproque de la fonction carrée

L'application g de \mathbb{R}_+ dans \mathbb{R}_+ définie par $f(x) = x^2$ est bijective et son application réciproque est l'application g^{-1} de \mathbb{R}_+ dans \mathbb{R}_+ définie par $g^{-1} = \sqrt{x}$.

2.6 ALGORITHME ENSEMBLISTE

L'une des méthodes de caractérisation d'ensembles ayant eu le plus d'applications dans le domaine particulier de l'automatique est la notion d'inversion ensembliste. Cette approche a été formalisée, développée et utilisée pour résoudre des problèmes d'estimations dans [3], [4].

2.6.1 Inversion ensembliste

Considérons un ensemble dans R^n et une fonction $f: R^n, R^m$ Soit Y un sous ensemble de R^m défini comme suit : $X = f(Y)$

L'inversion d'ensembles nous permet de déterminer l'ensemble X , comme image réciproque par la fonction f de l'ensemble Y :

$$X = \{x \in R^n | f(x) \in Y\} = f^{-1}(Y) \quad (2 - 10)$$

On peut résoudre un problème d'inversion ensembliste par le biais de l'algorithme SIVIA (Set Inversion Via Interval Analysis) [7] [8]. SIVIA est un algorithme récursif permettant d'obtenir, à partir d'un pavé initial suffisamment large, deux sous-ensembles encadrant l'ensemble solution, tel que :

$$\underline{\mathbb{S}} < \mathbb{S} < \overline{\mathbb{S}} \quad (2 - 11)$$

L'approximation, ou encadrement, intérieure \mathbb{S} représente l'ensemble des pavés dits acceptables ou faisables. C'est l'ensemble contenant toutes les solutions recherchées. Cependant, il peut y avoir des solutions admissibles qui ne sont pas contenues dans \mathbb{S} . Si on démontre que

$[x] \cap \overline{\mathbb{S}} = \emptyset$; alors le pavé $[x]$ est inacceptable et sera supprimé. Dans le cas où le pavé n'est ni acceptable, ni inacceptable, on dit alors que le pavé $[x]$ est indéterminé. L'algorithme va donc découper ce pavé le long du plus grand coté en deux sous-pavés qui vont être testés à leur tour afin de vérifier s'ils peuvent être gardés ou rejetés. Le processus est itéré à nouveau jusqu'à l'obtention d'un pavé dont la largeur atteint un certain seuil ϵ fixé par l'expérimentateur. Dans tous les autres cas, le pavé sera dit indéterminé [7] [8].

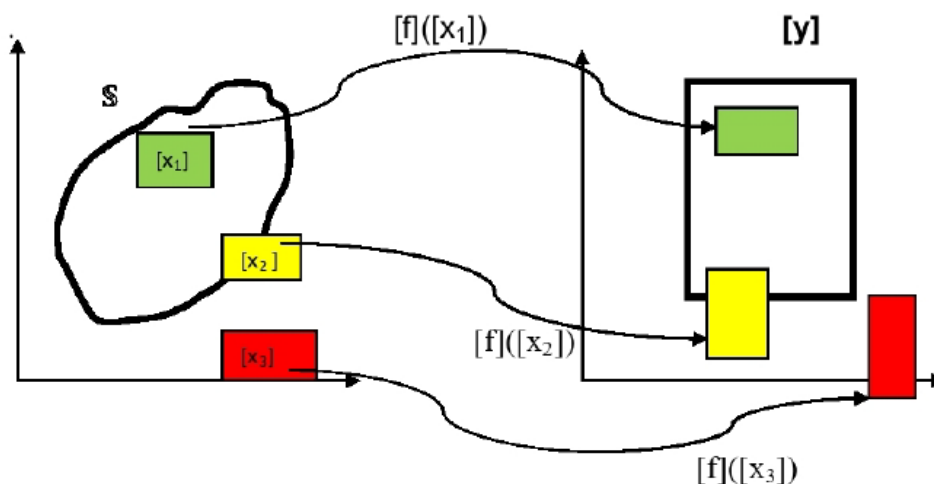


Figure 2-10 Sivia [7]

La description de l'algorithme Sivia est donc :

L'algorithme SIVIA

$$\left\{ \begin{array}{l} \mathbf{SIVIA} (\text{entrée} : [t], [x], \varepsilon; \text{sorties } \underline{\mathbb{S}}, \overline{\mathbb{S}}), \\ \mathbf{1.} \text{ si } [t]([x]) = [0], \quad \text{rejeter } [x]; \\ \mathbf{2.} \text{ si } [t]([x]) = [1], \quad \underline{\mathbb{S}} := \mathbb{S} \cup [x], \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [x]; \\ \mathbf{3.} \text{ si } w([x]) \leq \varepsilon, \quad \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [x]; \\ \mathbf{4.} \text{ bissecter } [x] \text{ en } ([x_1], [x_2]) \\ \quad \mathbf{SIVIA} (e: [t], \quad x_1, \varepsilon; s: \underline{\mathbb{S}}, \overline{\mathbb{S}}); \\ \quad \mathbf{SIVIA} (e: [t], \quad x_2, \varepsilon; s: \underline{\mathbb{S}}, \overline{\mathbb{S}}); \end{array} \right. \quad (2 - 12)$$

2.6.2 Contraction

2.6.2.1 Problème de satisfaction de contraintes

Le problème de satisfaction de contraintes est plus connu par la communauté informatique, spécialisée en intelligence artificielle.

Soit une fonction $f \in \mathbb{R}^n \rightarrow \mathbb{R}^m$ avec X un domaine de variables initiales et $x = (x_1, x_2, \dots, x_n)$ un vecteur variable. Nous considérons un système d'équations.

$$f(x) = 0 \Leftrightarrow \begin{cases} \{f_1(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_j(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_m(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad n \leq m \quad (2-13)$$

Nous appelons problème de satisfaction de contraintes H , la détermination de toutes les solutions du système d'équations à résoudre (2.13), contenues dans le domaine de recherche initial X .

$$\text{On note :} \quad H : (f(x) = 0, x \in X) \quad (2.14)$$

Si S_j est un ensemble de solutions de la contrainte $f_j(x_1, x_2, \dots, x_j) = 0$, alors la solution du problème de satisfaction de contraintes est obtenue par :

$$S = \bigcap_1^n S_j \quad (2.15)$$

Les techniques de contractions ont été élaboré [13] afin de réduire des pavés sans avoir recours parfois aux bisections. Donc, un contracteur C est un opérateur qui permet de contracter ou réduire un pavé de recherche initial sans bisection et en gardant l'intégralité des solutions, c'est à dire éliminer les valeurs inconsistantes. Un contracteur doit donc vérifier les deux propriétés suivantes :

$$\forall [x] \in \mathbb{R}^n \begin{cases} \forall [x] \subseteq X, C_S([x]) \subset [x] \text{ (contractance)} \\ \forall [x] \subseteq X, [x] \cap S \subset C_S([x]) \text{ (complétude)} \end{cases} \quad (2.16)$$

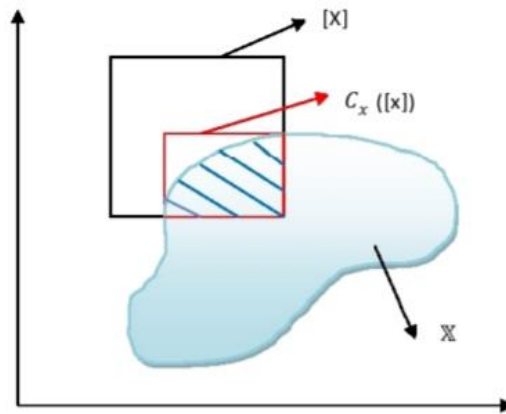


Figure 2.11 – Contraction d'un ensemble [7]

Pour résoudre des problèmes de satisfaction de contraintes non linéaires, on a développé différents types de contracteur : Le contracteur de Newton [17], Le contracteur de Krawczyk [116], Le contracteur de propagation-rétropropagation. On dispose aussi d'un contracteur de Gauss-Seidel valable pour les cas linéaires. Dans le cadre de ce travail, on a utilisé un contracteur par projection.

2.6.2.2 Notion de consistence

Dans un premier temps, nous considérons une seule contrainte \mathcal{F}_j . Soit un pavé $[x] \subset X$. La valeur $x_i \in [x_i]$ est dite consistante avec \mathcal{F}_j , s'il est possible d'instancier les autres composantes x_j de x , de manière que la relation $f_j(x) = 0$ soit satisfaite. En d'autres termes :

$$x_i \in \pi_i(S_j \cap [x]) \quad (2.17)$$

π_i représente la projection de $S_j \cap [x]$ sur $[x_i]$

Exemple

Soient les variables x , y et z suivantes telle que : $x \in [1, 5]$, $y \in [2, 4]$ et $z \in [6, 10]$ et soit la contrainte $z = x + y = [3, 9]$.

On constate que les valeurs 1 pour x et 10 pour z sont dites inconsistantes.

2.6.2.3 Projection de contraintes

Considérons un ensemble S , nous notons $C_S([x])$ le plus petit pavé contenant $[x]$. l'opérateur de contraction C_S est en mesure de remplacer une boîte $[x]$ par un pavé $S \cap [x]$ de plus petite taille tout en conservant l'ensemble solution au complet. Ce procédé que nous allons décrire dans le paragraphe suivant est basé sur le principe de projection de contraintes utilisant des sous solveurs [15].

2.6.2.4 Sous-résolveurs

Un sous résolveur est un algorithme qui permet de calculer une composante x_i du vecteur x en fonction des autres composantes. Ce qui va nous permettre par la suite d'appliquer des contractions par projection inspiré du principe branch and bound et le calcul par intervalle .

Soit $\vec{p} = (p_1, p_2, \dots, p_n)^T$ un vecteur de \mathbb{R}^n .

Un sous ensemble P_1 est de \mathbb{R}^n une contrainte [17] telle que :

$$P_1 = \{\vec{p} \in \mathbb{R}^n / f(\vec{p}) \in [y]\} = f^{-1}([y]) \quad (2.18)$$

où $[y]$ est un intervalle et f une fonction continue. Considérons un pavé $[\vec{p}]$, Le problème de réduction consiste à obtenir un pavé $[x]$ le plus petit possible sans avoir recours à des bisections et qui contient l'ensemble :

$$S_1 = f^{-1}([y]) \cap [\vec{p}] \quad (2.19)$$

On suppose qu'il est possible d'isoler p_i de l'expression $y = f(\vec{p})$, c.à.d. il existe une fonction g telle que :

$$f(\vec{p}) = y \Leftrightarrow p_i = g_i({}^i \vec{p}, y) \quad (2.20)$$

où ${}^i \vec{p} = (p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_n)^T$ et g_i la fonction solution appelée aussi sous résolveur, associée à p_i , soit π_i l'opérateur de projection sur le $i^{\text{ème}}$ axe et $[g_i]$ la fonction d'inclusion associée à g_i

Théorème 1. *Supposons qu'il existe un sous résolveurs g_i d'entrée ip et de sortie p associé au problème de satisfaction de contraintes (2.14) On peut écrire :*

$$\Pi_i(S_1) \subset [g_i]({}^i \vec{p}, [y]) \cap [p_i] \quad (2.21)$$

Si g_i est continu et $[g_i]$ est minimale, alors la relation (2.21) devient égalité.

Supposons maintenant que pour $j \in \{1, \dots, m\}$ et $i \in \{1, \dots, n\}$, il est possible d'isoler la variable p_i dans l'expression $f_j(\vec{p}) = y_j$ c.à.d. qu'il existe une fonction solution g_j^i qui satisfait :

$$f_j(\vec{p}) = y_j \Leftrightarrow p_i = g_j^i({}^i \vec{p}, y) \quad (2.22)$$

En tenant compte du théorème 1, On a :

$$\Pi_i(S_j) \subset [g_j^i]({}^i \vec{p}, [y_j]) \cap [p_i] \quad (2.23)$$

Par la suite,

$\forall j \in \{1, \dots, m\}$ et $i \in \{1, \dots, n\}$, $\pi_i(S_j)$ est inclus dans l'intervalle

$$[h_j^i]([p], [y_j]) \left\{ \begin{array}{l} [g_j^i]([p], [y_i] \cap [p_i] \text{ sig}_j^i \text{ existe} \\ [p_i] \end{array} \right. \quad (2.24)$$

Exemple

Soit la contrainte $p_1 p_2 \in [8, 40]$ et un pavé $[p] = [1, 4] \times [1, 4]$ On peut obtenir un encadrement de S_1 en réduisant $[p]$ comme suit [44].

$p_1 p_2 = y \Leftrightarrow p_1 = y / p_2 \Leftrightarrow p_2 = y / p_1$ pour $p_i \neq 0$ Les fonctions solutions sont :

$$g_1(p_2, y) = y / p_2 \text{ et } g_2(p_1, y) = y / p_1$$

En tenant compte du théorème 1, la projection de S_1 est :

$$\begin{aligned} \pi_1(S_1) &= [g_1]([p_2], y) \cap [p_1] = [y] / [p_2] \cap [p_1] \\ &= [8, 40] [1, 4] \cap [1, 4] = [2, 4] \end{aligned}$$

$$\begin{aligned} \pi_2(S_1) &= [g_2]([p_1], y) \cap [p_2] = [y] / [p_1] \cap [p_2] \\ &= [2, 4] \end{aligned}$$

Par conséquent l'intervalle enveloppe de S_1 est

$$[S_1] = [2, 4] \times [2, 4]$$

L'algorithme SiviaP

La résolution des problèmes d'inversion ensembliste se limite au cas où le nombre de variable est réduit. C'est pour cette raison qu'on a développé des techniques de contractions [5] [6] permettant de réduire les domaines solutions. Si l'on intègre un contracteur par projection à l'algorithme Sivia afin de diminuer le nombre de bisection, on obtient l'algorithme SiviaP.

SiviaP (entrées : $C, [p], [y], \epsilon$; sorties : $\underline{S}, \overline{S}$)

1. $[x] = C([p])$
2. si $[p] = \emptyset$, rejeter $[p]$
3. si $[f]([p]) \subseteq [y]$ alors
 $\underline{S} := \underline{S} \cup [x], \overline{S} := \overline{S} \cup [x]$
4. si $w([p]) < \epsilon$, alors $\overline{S} := \overline{S} \cup [x]$
5. bissecter $[p]$ en $[p_1]$ et $[p_2]$
6. SiviaP ($[p_1]$), SiviaP ($[p_2]$)

2.7 Identifications dans le contexte à erreurs bornées

Résoudre un problème d'inversion ensembliste en utilisant le principe de bisections est un problème NP-dur dont la résolution est limitée aux cas où le nombre de variables est réduit.

Les paramètres estimés par des méthodes classiques sont des valeurs optimales obtenues par optimisation d'un critère comme celui des moindres carrés décrit au chapitre précédent mais leur qualité et fiabilité dépend de l'espérance mathématique et la matrice de covariance de l'estimateur choisi, cela suppose donc qu'on travaille avec des variables aléatoires caractérisées par une densité de probabilité et une covariance. En pratique, ces données sur les erreurs introduites ne sont pas toujours disponibles ce qui conduit à une estimation parfois biaisée voir erronée. Dans le contexte ensembliste, on suppose connaître les bornes sur ces erreurs et donc on n'a besoin d'aucune hypothèse concernant leur distribution statistique. Le résultat de l'identification se présentera sous la forme d'un ensemble de tous les paramètres compatibles avec les bornes des erreurs introduites. On parle d'identification ensembliste dans le contexte à erreurs bornées

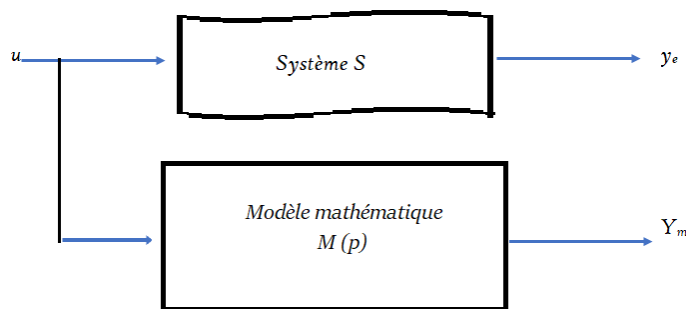


Figure 2.12 – Modèle d'identification

Dans ce chapitre, on s'intéresse à l'identification d'un vecteur P d'un modèle mathématique $M(p)$ de sortie $y_m(t_i)$, à partir de données expérimentales $y_e(t_i)$ disponibles à la sortie d'un système S à tout instant t_i .

D'autre part on définit un vecteur $[y_e(t_i)]$ tel que :

$$y_e(t_i) \in [y_e(t_i)] \quad 2.25$$

Identifier les paramètres du modèle $M(p)$ revient à rechercher l'ensemble des valeurs de p tel que :

$$y_m(t_i) \in [y_e(t_i)] \text{ pour } i = 1, \dots, n \quad 2.26$$

On définit l'erreur de la sortie comme l'écart entre les deux vecteurs de sorties figure 2.12 :

$$e(p) = y_e(t_i) - y_m(t_i) \quad 2.27$$

Si le vecteur $e(p)$ appartient à un vecteur intervalle $[e] = [-\delta_e, \delta_e]$ alors le vecteur $y_m(t_i)$ de $M(p)$ doit appartenir à l'intervalle $[y_m(t_i)] = y_m(t_i) + [-\delta_e, \delta_e]$. Une estimée \hat{p} du vecteur paramètre p est dite acceptable si et seulement si $y_m(t_i)(\hat{p}) \in [y_e]$. Il s'agit donc de déterminer un ensemble de tous les \hat{p} acceptables appartenant à un espace de recherche \mathbb{P} connu a priori.

$$\mathbb{s} = \{p \in \mathbb{P} / y_m(t_i, p) \in [y_e(t_i)]\} \quad 2.28$$

On peut réécrire cette relation avec la notation ensembliste qu'on appelle inversion ensembliste.

$$\mathbb{s} = y_m(t_i, p)^{-1} \cap [y_e(t_i)] \quad 2.29$$

2.7 Conclusion

Ce chapitre a relevé les motivations de l'utilisation de l'approche ensembliste. Celles-ci sont dues principalement à la capacité de mieux prendre en compte les incertitudes sur les données d'un problème et sur les méthodes et les moyens de résolution. En second lieu, nous avons présenté les fondements de base de cette approche et les notions très importantes reliées aux manipulations des ensembles. On a défini ensuite une notion importante qui est l'inversion ensembliste, va nous servir pour notre application dans les chapitres suivants

Chapitre III

La modélisation des robots manipulateurs

3.1 Introduction

Des nombreux domaines civils et militaires intéressent aux système robotique, dans des grandes applications qui sont :

1. La production industriel (usinage, assemblage, soudure, polissage, formage, etc . . .)
2. Les intrusions en milieux hostiles (sous-marin, nucléaire, exploration planétaire, etc . . .)
3. Le transport des biens et des personnes (véhicules intelligents, robots mobiles, etc . . .)
4. Assistance et aide aux personnes (robots personnels, aides techniques, etc . . .)
5. Le secteur sanitaire (chirurgie, rééducation, etc . . .)
6. Le ludique (robots jouets)

Du point de vue de la structure mécanique et de la commande, les robots prennent des formes très diverses.

Pour simuler ou commander le comportement d'un système mécanique articulé (robot), on doit disposer d'un modèle. En fonction de nos objectifs plusieurs types de modélisation sont possibles, les contraintes de la tâche et les performances recherchées. Les modèles mathématiques nécessaires sont :

1. Les modèles géométriques directs et inverses qui expriment la situation de l'organe terminal en fonction des variables articulaires et inversement.
2. Les modèles cinématiques directs et inverses qui expriment la vitesse de l'organe terminal en fonction des variables articulaires et inversement.
3. Les modèles dynamiques qui définissent les équations du mouvement du robot.

Dans ce chapitre, on rappellera les notions de base nécessaires pour l'obtention des différents modèles utilisés en robotique et aussi la commande des robots qui nécessitent le calcul de certains modèles mathématiques tel que le modèle géométrique direct.

La modélisation du robot de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notations ont été proposées. La plus répandue est celle de Denavit-Hartenberg [11], mais cette méthode, développée pour des structures ouvertes simples présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures fermées ou arborescente. C'est pourquoi, on utilise la notation de Khalil et Kleinfinger [13]. Cette méthode est noté D.H.M (Denavit-Hartenberg Modifiée) permet une description homogène avec un nombre minimum de paramètres, des architectures ouvertes simples et complexes des systèmes mécaniques articulés.

3.2 Définitions

- On dit un robot pour tout système mécanique poly-articulé mû par des actionneurs et commandé par un ordinateur qui est destiné à effectuer une grande variété de tâches.
- "Un appareil automatique qui peut effectuer des fonctions normalement effectuées par des humains." traduction du dictionnaire Webster's
- « Appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe ou modifiable. » définition du dictionnaire Petit Larousse
- "Un manipulateur reprogrammable multifonctionnel conçu pour déplacer des matériaux, des outils, des pièces ou des composants spécialisés à travers une série de mouvements programmés pour effectuer une tâche précise." Robot Institut de robotique d'Amérique, 1979.

3.3 Description de la géométrie des robots :

Les éléments qui constituent un robot sont les suivants :

- Organe terminal
- Axe = Articulation
- Corps = Segment
- Effecteur = outil
- Actionneur = Moteur
- Base

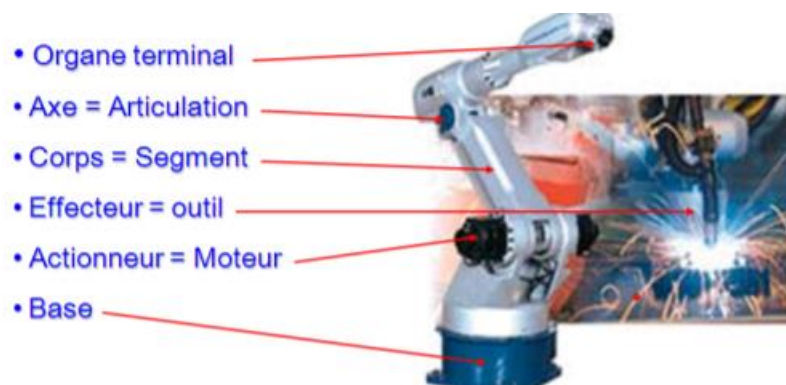


Figure 3.1 Constituants d'un robot [15]

Généralement on distingue cinq parties principales dans un robot :

- a. Système mécanique articulé (S.M.A.) + un organe terminal (voire plusieurs)
- b. Organe terminal
- c. Actionneurs
- d. Capteurs
- e. Système de commande

3.3.1 Le système mécanique articulé (S.M.A.)

Dans le cas des robots manipulateurs le système mécanique ressemble a un mécanisme ayant une structure proche au bras humain. Il permet de plonger ou de remplacer son action. Son fonctionnement est d'amener l'organe terminal dans une position et orientation donnée, d'après les caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps, généralement rigides (ou supposés comme tels), alliés par des liaisons appelées articulations.

Sa mécanisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

a. Articulation

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Soit m le nombre de degré de liberté résultant, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que :

$$0 \leq m \leq 6$$

Lorsque $m = 1$; ce qui est fréquemment le cas en robotique, l'articulation est dite simple : soit rotoïde, soit prismatique.

Articulation rotoïde : c'est une articulation de type pivot, quand note R, réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe (voir la figure suivante).

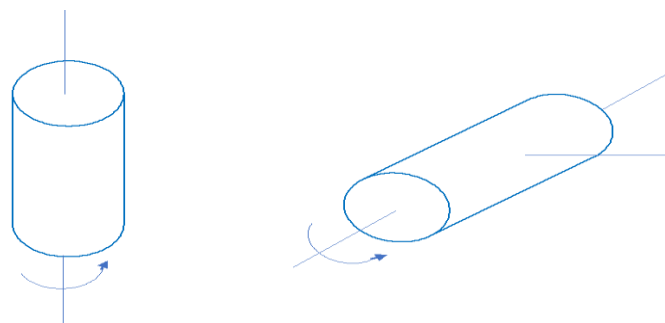


Figure 3-2 Articulation rotoïde

Articulation prismatique : c'est une articulation de type glissière, quand note P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe (voir la figure suivante).

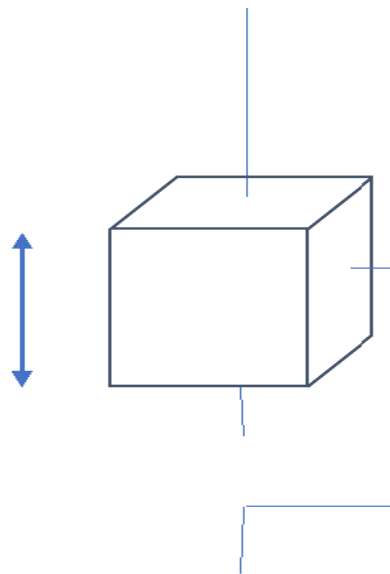


Figure 3-3 Articulation prismatique

b. Organe terminal

Tout un dispositif destiné soit pour la manipulation des objets comme les dispositifs de serrage (pinces à deux ou trois doigts), les dispositifs magnétiques ou à dépression (ventouse), soit pour la transformation (outils de découpe, torche de peinture, torche de soudage). Donc il s'agit d'une interface permettant au robot d'interagir avec son environnement.

c. Les actionneurs

Les actionneurs sont des ensembles du S.M.A qui comporte des moteurs le plus souvent associés à des transmissions. Généralement les actionneurs utilisent des moteurs électriques avec aimant permanent, à courant continu, avec commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée, bien souvent la vitesse de régime élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.

On utilise des actionneurs hydrauliques pour les robots manipulateurs, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

d. Les capteurs

Les capteurs permettent de gérer les relations entre le robot et son environnement. On distingue deux types, capteurs proprioceptifs lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et les capteurs extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

e. Le système de commande

C'est la partie qui synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur.

3.3.2 Architecture des robots

Un robot industriel comporte deux parties essentielles :

Le porteur : Structure mécanique articulée constituée des trois premiers degrés de liberté à partir du bâti. Si on considère un point P de l'extrémité d'un repère R_0 lié au bâti, le rôle du porteur est de fixer la position de P dans R_0 . On définit les différentes liaisons utilisées et qui sont du type pivot notées R ou prismatique notées P (voir tableau de l'Annexe 1) [7]. Le nombre de combinaisons possibles dans un repère tridimensionnel est donc huit. Elles sont représentées dans le tableau de l'Annexe 2 [7].

Le poignet : Son rôle est l'orientation de la pince ou de l'outil terminal porté par le robot. Les deux structures les plus courantes sont données en Annexe 3 [7].

On définit trois grandes classes d'architecture (Annexe 4) [7] en fonction de la répartition des liaisons motorisées du bâti au poignet.

1. Architecture série (ou chaîne cinématique ouverte (Figure 3.4a))
2. Architecture parallèle (ou chaîne cinématique multi boucle (Figure 3.4b))
3. Architecture mixte (série-parallèle ou parallèle-série)

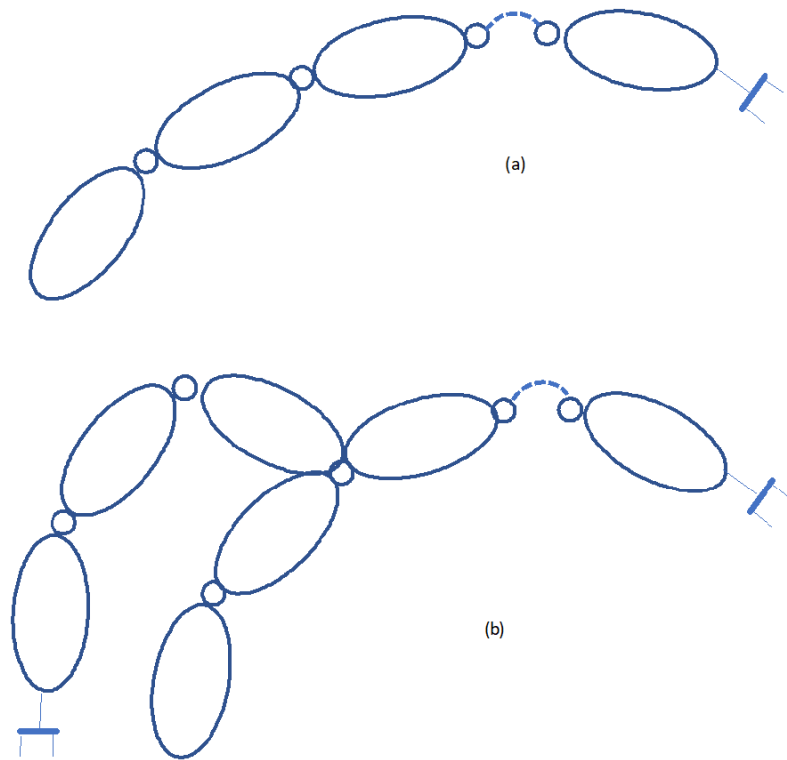


Figure 3.4 Structure ouverte simple (a), Structure arborescente (b)

On définit aussi deux types d'espace relatif au robot.

L'espace articulaire : c'est un espace qui représente la situation de tous ses segments. On utilise des variables articulaires.

L'espace opérationnel : c'est un espace qui représente la situation de l'organe terminal. On utilise des coordonnées cartésiennes, sphériques ou cylindriques. Autres particularités des robots :

La redondance : quand le nombre de degré de liberté de l'organe terminal est inférieur au nombre de degré de liberté de l'espace articulaire (nombre d'articulations motorisées) les qualités requises pour un robot sont la résolution, la précision et la répétabilité.

La singularité : C'est un point de l'espace que le robot ne peut atteindre.

3.4 Degrés de liberté d'un solide dans l'espace :

3.4.1 Définition (degré de liberté, d.d.l.) : Le nombre de d.d.l. d'un mécanisme c'est le nombre de paramètres indépendants qui permettent de définir la position du mécanisme à un instant donné du mouvement.

Combien de degrés de libertés à un solide dans l'espace ? ou encore, combien de paramètres indépendants (nombre minimal) sont-ils nécessaires pour définir la situation (positionnement) du solide dans l'espace (par rapport à un repère de référence) ?

3.4.2 Degrés de libertés :

Degrés de liberté dans l'espace :

- 3 en position
- 3 en orientation

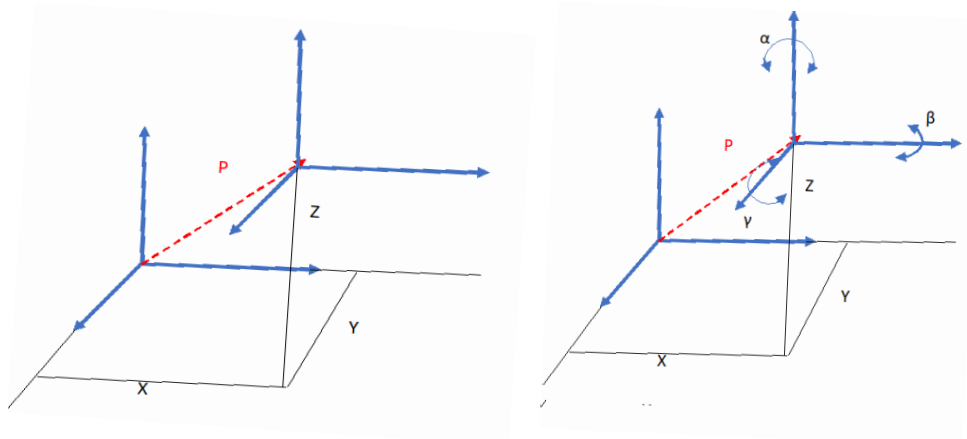


Figure 3.5. a d.d.l. dans l'espace

DDL d'un solide dans l'espace :

Six paramètres indépendants sont obligatoires pour le repérage de la position d'un solide dans l'espace :

La position d'un point noté P est définie par trois paramètres pour repérer un solide (coordonnées cartésiennes, cylindriques, sphériques, ..., dans la base du repère fixe),

L'orientation du solide autour du point P est définie par trois paramètres indépendants déterminent (angles d'Euler, paramètres d'Euler, ...).

Angles d'Euler

Très utilisé en mécanique, l'orientation d'un repère orthonormé lié à un solide par rapport à un repère fixe est définie par trois angles.

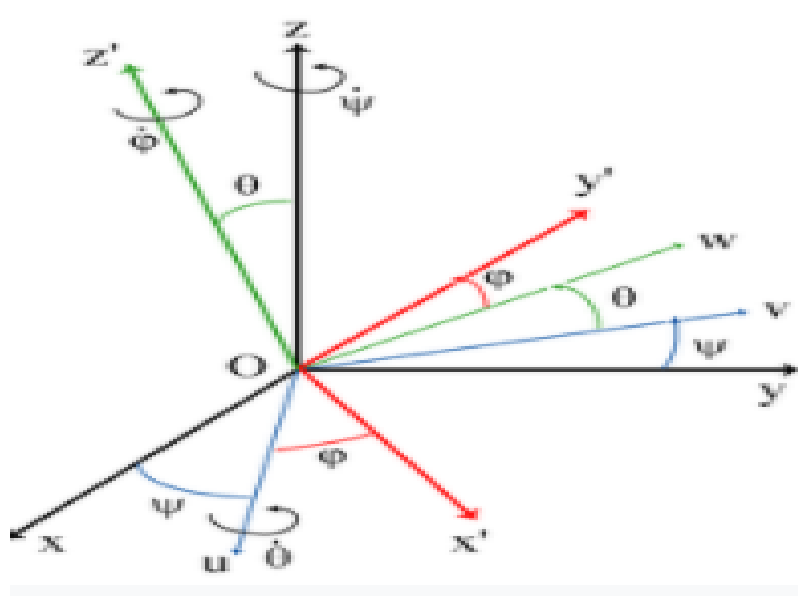


Figure 3.5.b Angle d'Euler [15]

Angles d'Euler ψ , θ et ϕ . Le système fixe est indiqué en noir ($Oxyz$), le système mobile en rouge ($Ox'y'z'$) et la ligne des nœuds en bleu (Ou).

Ces trois rotations sont obtenues en changeant un des trois angles d'Euler et en gardant les deux autres constants. Ces trois rotations sont la précession, la nutation et la rotation propre. On passe du référentiel fixe (O,x,y,z) au référentiel lié au solide (O,x',y',z') par trois rotations successives.

- La précession ψ , autour de l'axe Oz , fait passer de (O,x,y,z) au référentiel (O,u,v,z) (en bleu).
- La nutation θ , autour de l'axe Ou (ligne des nœuds), fait passer de (O,u,v,z) à (O,u,w,z') (en vert).
- La rotation propre ϕ , ou giration, autour de l'axe Oz' , fait passer de (O,u,w,z') au référentiel lié au solide (O,x',y',z') (en rouge).

NB. L'axe Ou est porté par l'intersection des plans Oxy et $Ox'y'$

On peut également passer du référentiel fixe $Oxyz$ au référentiel lié au solide $Ox'y'z'$ par les trois rotations successives suivantes, ayant toutes leurs axes fixes dans le référentiel initial :

- Une rotation d'angle ψ d'axe Oz'
- Une rotation d'angle θ d'axe Ox'
- Une rotation d'angle ϕ d'axe Oz

La transformation entre R_1 et R' s'écrit :

$$R_3 = \text{Rot}(Z', \psi) \text{Rot}(X', \theta) \text{Rot}(Z, \phi) \quad (3.1)$$

$$R_3 = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

3.5 Matrices de transformation homogène :

3.5.1 Coordonnées homogènes

Un point est décrit par ses coordonnées cartésiennes $P_x P_y P_z$ et 1 comme quatrième composante pour une écriture homogène.

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (3.3)$$

Représentation d'une direction (vecteur libre) est représenté par :

$$U = \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} \quad (3.4)$$

Représentation d'un plan

Le plan $\alpha x + \beta y + \gamma z = \rho$ est représenté par un vecteur $Q = [\alpha \ \beta \ \gamma \ \rho]$ tel que pour tout point appartenant à Q on a : $Q.P=0$

$$P.Q = [\alpha \ \beta \ \gamma \ \rho] \cdot \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = 0 \quad (3.5)$$

3.5.2 Transformation homogène

Considérons les trois repères suivants :

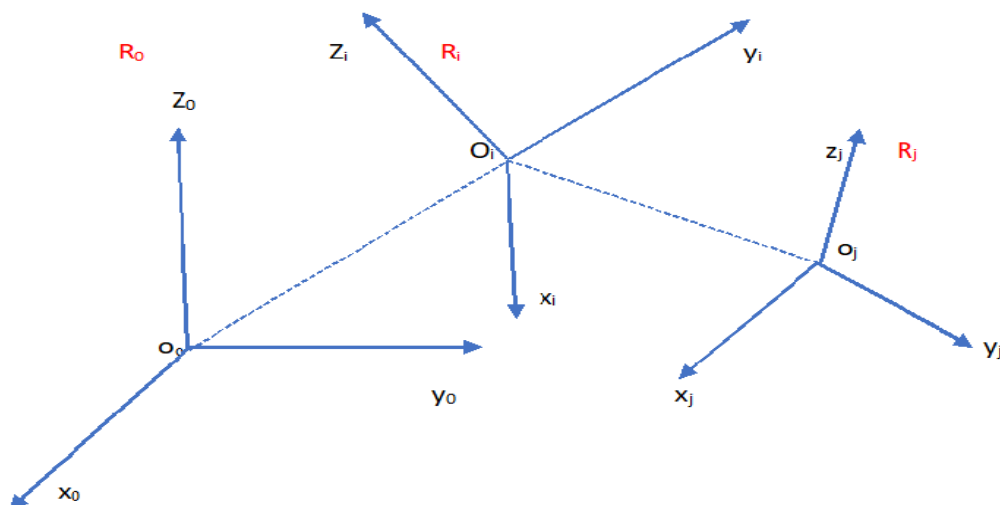


Figure 3.6 -changements de repères

Et soit ${}^i u_j$, ${}^i v_j$ et ${}^i w_j$ les vecteurs unitaires des axes X_j , Y_j et Z_j du repère R_j exprimés dans R_i . Avec ${}^i P_j$ un vecteur exprimant l'origine de repère R_j dans R_i . On définit la matrice de transformation homogène par :

$${}^i T_j = [{}^i u_j, {}^i v_j, {}^i w_j, {}^i P_j] = \begin{bmatrix} u_x & v_x & w_x & P_x \\ u_y & v_y & w_y & P_y \\ u_z & v_z & w_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

La matrice A représente la matrice de rotation ou d'orientation du repère R_i par rapport au repère R_j . La colonne P représente la translation du repère R_i par rapport au repère R_j . Dans le cas d'une translation pure $A = I_3$, tel que I représente la matrice identité.

Propriétés :

1. La matrice A est orthogonale, alors $A^{-1} = A^T$

2. ${}^i T_j^{-1} = {}^j T_i$

3. $Rot(u, \theta)^{-1} = Rot(u, -\theta) = Rot(-u, \theta)$ ²

4. $Trans(u, d)^{-1} = Trans(-u, d) = Trans(u, -d)$ (3.8)

3.5.3 Transformation de vecteurs

Soit un vecteur ${}^i p_1$ définissant le point P_1 dans le repère R_j

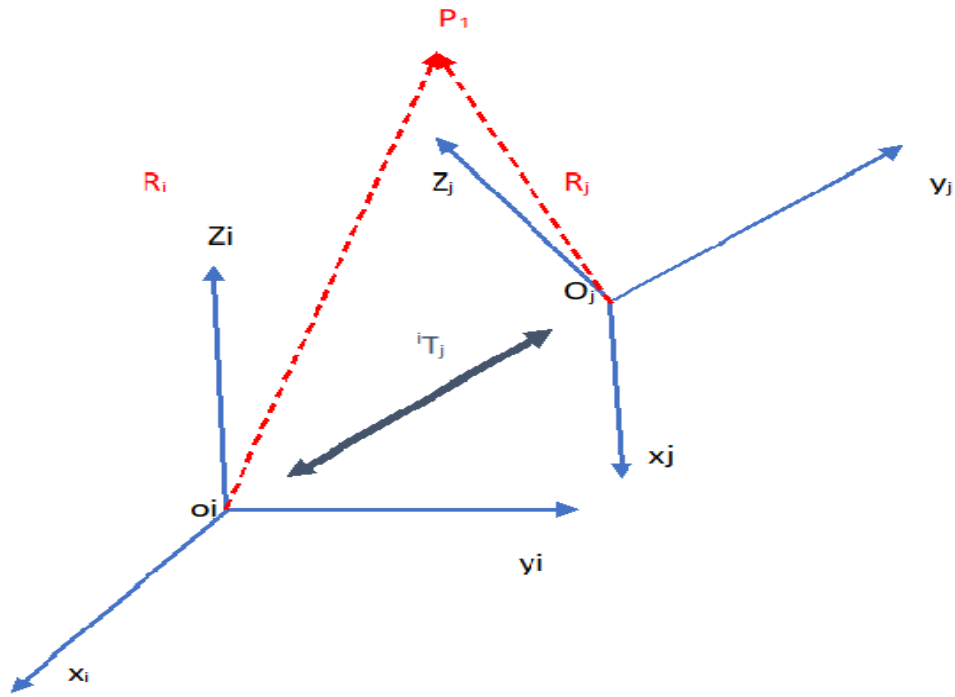


Figure 3.7 – Transformation de point et de vecteur entre 2 repères consécutifs

Pour calculer les coordonnées homogènes du point P_1 dans le repère R_i par l'équation suivante :

$${}^i p_1 = {}^i(O_i P_1) = {}^i u_j^j p_{1x} + {}^i v_j^j p_{1y} + {}^i w_j^j p_{1z} + {}^i T_j^j p_1 \quad (3.9)$$

Donc la matrice ${}^i T_j$ permet de transformer les coordonnées d'un point du repère R_j vers le repère R_i .

3.5.4 Matrice de translation pure homogène

Soit $Trans(a,b,c)$ une transformation qui désigne la translation a , b , et c le long des axes x , y , et z respectivement. La transformation dans ce cas s'exprime par :

$${}^i T_j = Trans(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

On utilise par la suite la notation $Trans(u,d)$ pour désigner une translation d'une valeur d le long de l'axe u .

Propriétés :

$$Trans(a, b, c) = Trans(x, a).Trans(y, b).Trans(z, c)$$

L'ordre des multiplications étant quelconque.

3.5.5 Matrice de rotation homogène

On définit la rotation $Rot(x, \theta)$ la transformation homogène qui s'exprime par :

$${}^i T_j = Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$Rot(x, \theta)$ désigne la rotation ou l'orientation de repère R_i d'un angle θ autour de l'axe x du repère R_j . De la même façon, on définit la rotation autour de y par :

$${}^i T_j = Rot(y, \theta) = \begin{bmatrix} c\theta & 0 & -s\theta & 0 \\ 0 & 1 & 0 & 0 \\ s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Et la rotation autour de z par :

$${}^i T_j = Rot(z, \theta) = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

On peut donc généraliser :

$${}^0 T_k = {}^0 T_1 \cdot {}^1 T_2 \cdot {}^2 T_3 \cdot \dots \cdot {}^{k-1} T_k \quad (3.14)$$

3.5.6 Le torseur cinématique

La vitesse d'un solide dans l'espace peut être décrite en utilisant un torseur cinématique. Ce dernier est défini comme l'application qui à tout point O_i du corps C_i fait correspondre un vecteur V_i appelé champ de vecteur. Si ce vecteur est une vitesse, on parlera de champ des vitesses. Le champ des vitesses est antisymétrique ce qui caractérise un torseur appelé torseur cinématique. On appelle éléments de réduction du torseur cinématique en O_i les éléments définis par :

1. V_i : le moment résultant en O_i , il représente la vitesse absolue de l'origine O_i par rapport à R_0 , tel que :

$$V_i = \frac{d}{dt}(O_0 O_i) \quad (3.15)$$

2. W_i : la résultante du torseur représentant le vecteur de rotation instantanée du corps C_i par rapport à R_0

La connaissance de ces deux éléments permet de calculer la vitesse d'un point O_i par la relation fondamentale suivante :

$$V_j = V_i + W_i \times O_i O_j \quad (3.16)$$

Où le symbole \times désigne le produit vectoriel. Les composantes de V_i et de W_i peuvent être concaténées pour composer le vecteur V_i représentant le torseur cinématique en O_i :

$$V_i = [V_i^T \ W_i^T]^T \quad (3.17)$$

3.6 Modèle géométrique des robots :

3.6.1 Convention de Denavit -Hartenberg modifiée

C'est une méthodologie à suivre pour normaliser, simplifier et rationaliser la modélisation géométrique d'un robot.

Une structure ouverte simple est composée de $n+1$ corps notés C_0, \dots, C_n liés entre eux par n articulations. A chaque corps on associe un repère R_i . Les repères sont numérotés de 0 à n . La $i^{\text{ème}}$ articulation dont la position est notée q_i relie les corps $i-1$ et i .

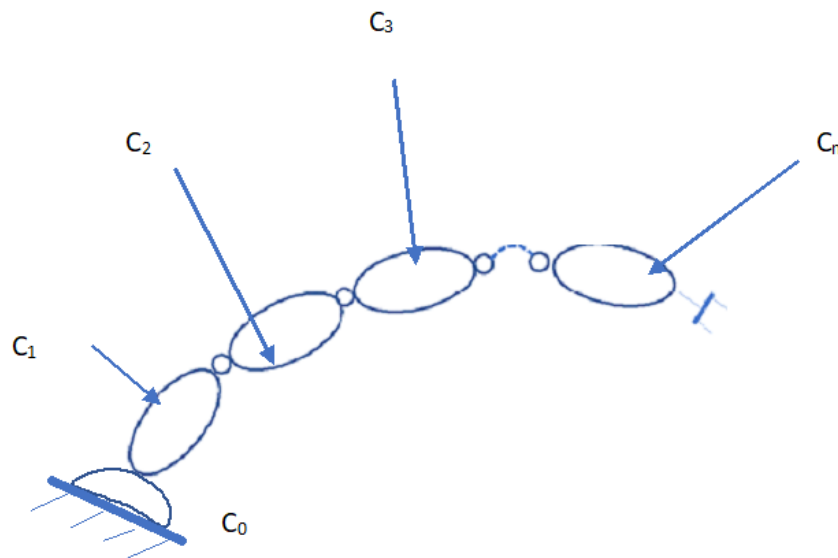


Figure 3.8 – Robot a structure ouverte simple

- **Principe :**

- 1-Fixer un repère à chaque corps du robot.
- 2-Calculer les matrices homogènes entre chaque repère.
- 3-En déduire la matrice homogène entre la base et l'organe terminal.

- **Hypothèses :**

On suppose que le robot est constitué d'un chainage de $n+1$ corps liés entre eux par n articulation rotoides ou prismatiques. A chaque corps, on associe un repère R_i . Les repères sont numérotés de 0 à n . la $i^{\text{ème}}$ articulation, dont la position est notée q_i est le point qui relie les corps $i-1$ et i .

Le repère R_i fixé au corps C_i est définie de sorte que :

- l'axe Z_i est porté par l'axe de l'articulation i .

-l'axes X_i est porté par la perpendiculaire commune aux axes Z_i et Z_{i+1} . Si les axes Z_i et Z_{i+1} sont parallèles ou collinaires, le choix de X_i n'est pas unique.

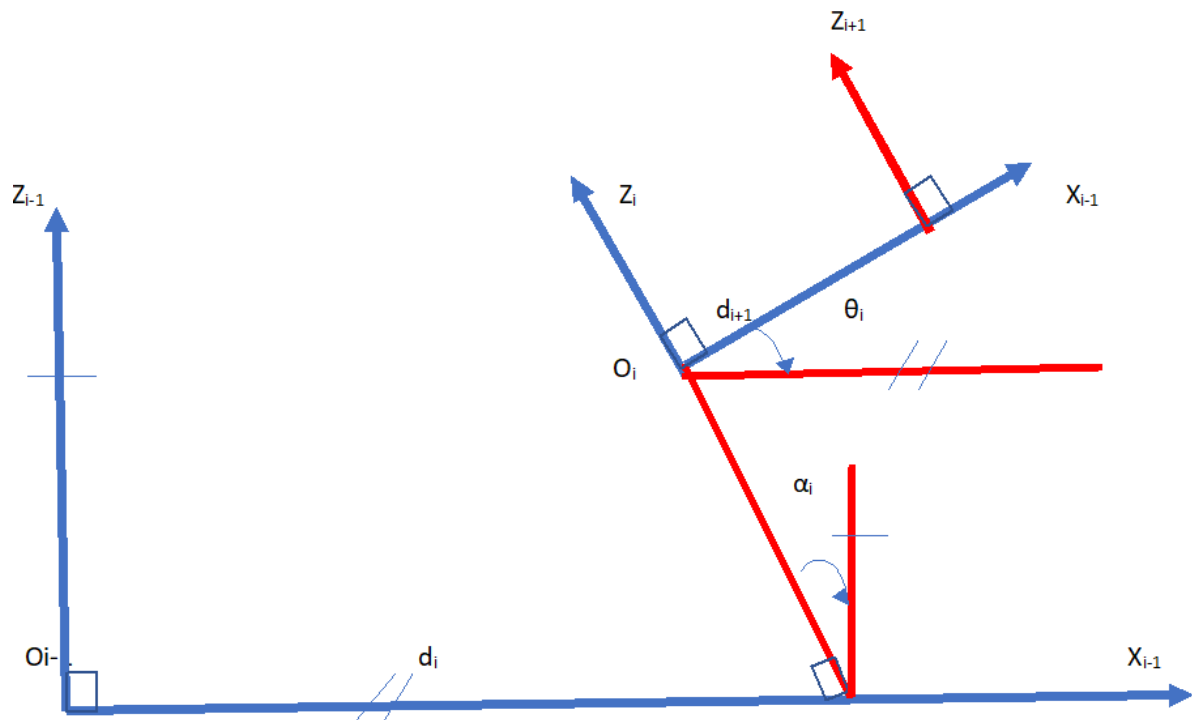


Figure 3.9 -Convention et paramètres de Denavit -Hartenberg modifiée

1. α_i : angle entre les axes Z_{i-1} et Z_i correspondant à une rotation autour de X_{i-1}
2. d_i : distance entre Z_{i-1} et Z_i le long de X_{i-1}
3. θ_i : angle entre l'axe X_{i-1} et X_i correspondant à une rotation autour de Z_i .
4. r_i : distance entre X_{i-1} et X_i le long de Z_i

Remarque : Si l'articulation i est de type prismatique, alors d_i est variable

Si l'articulation i est de type rotoïde, alors θ_i est variable.

La variable articulaire q_i associé à la $i^{\text{ème}}$ articulation est définie par :

$$q_i = \sigma_i \cdot \theta_i + \sigma_i \cdot r_i \quad (3.18)$$

Avec :

$\sigma_i = 0$ si l'articulation j est rotoïdes

$\sigma_i = 1$ si l'articulation est prismatique

La matrice de transformation entre articulation T_i et T_{i-1} est donnée par :

$${}_{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & d_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & d_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & r_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

Avec : $C\theta_i = \text{Cos}(\theta_i)$ et $S\theta_i = \text{Sin}(\theta_i)$.

3.6.2 Présentation des différents modèles

a. Modèle géométrique direct des robots (MGD)

Le modèle géométrique direct d'un robot manipulateur est un ensemble de fonctions qui permet d'exprimer la position ainsi que l'orientation de l'organe terminal d'un robot en fonction des variables articulaires. Cette position est définie par m coordonnées appelées coordonnées opérationnelles et sont notées :

$$X = [X_1 \dots X_m] \quad (3.20)$$

D'autres part, les variables articulaires présentant la configuration du robot sont notées par :

$$q = [q_1 \dots q_m]^T \quad (3.21)$$

Le modèle géométrique direct s'écrit donc :

$$X = f(q) \quad (3.22)$$

f exprime les relations fonctionnelles entre X et q . Pour définir le vecteur X , nous avons plusieurs possibilités selon que l'on combine l'une des méthodes de spécifications de la position avec l'une des méthodes de description de l'orientation. Par exemple avec les cosinus directeurs

$$X = [P_x P_y P_z u_x u_y u_z v_x v_y v_z w_x w_y w_z]^T \quad (3.23)$$

Il existe d'autres méthodes pour modéliser des rotations comme les angles d'Euler, les angles de Roulis-Tangage-Lacet (RTL) ou les paramètres d'Euler (quaternions).

b. Modèle géométrique inverse des robots (MGI)

Le modèle géométrique inverse des robots (MGI) consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles constitue le modèle géométrique inverse.

Nb : Il y a rarement unicité de solution.

Principe :

Soit fT_e la matrice de transformation homogène définissant la situation du robot (repère R_0) dans le repère atelier. Dans le cas général, on peut exprimer fT_e sous la forme :

$${}^fT_e = Z {}^0T_n(q) E \quad (3.24)$$

Avec :

— Z est la matrice de transformation homogène définissant la situation du repère (repère R_0) dans le repère atelier.

— 0T_n est la matrice de transformation homogène du repère terminal R_n dans le repère R_0 , fonction du vecteur des variables articulaires q

— E est la matrice de transformation homogène définissant le repère outil R_E dans le repère R_n .

Lorsque $n \geq 6$, on peut écrire la relation suivante en regroupant dans le membre de droite tous les termes connus :

$${}^0T_n(q) = Z^{-1} {}^fT_e E^{-1} \quad (3.25)$$

Il existe plusieurs méthodes de calcul du MGI : la méthode de Paul [12] qui convient pour la plupart des robots industriels

Principe de PAUL :

Considérons un robot manipulateur dont la matrice de transformation homogène a pour expression :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \quad (3.26)$$

Soit U_0 la situation désirée telle que :

$${}^i T_j = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & v_x & w_x & 0 \\ u_y & v_y & w_y & 0 \\ u_z & v_z & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & v_x & w_x & P_x \\ u_y & v_y & w_y & P_y \\ u_z & v_z & w_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

Méthode de solution

Pour trouver les solutions de l'équation (3.24) Paul a proposé une méthode qui consiste à pré multiplier successivement les deux membres de l'équation par la matrice ${}^jT_{j-1}$

étape 1 : $X = C T_n G_k$

En développant : $C^{-1}X = C^{-1} C T_n G_k$

$$C^{-1}X = T_n G_k$$

$$C^{-1}XG_k^{-1} = T_n G_k G_k$$

$C^{-1}XG_k^{-1} = T_n$ tel que :

$C^{-1}X G_k^{-1}$: connu et T_n inconnu

où $T_n = A_1 A_2 \dots A_n$

⇒ 12 relations avec n inconnus

étape 2 : simplifier les équations. Séparer certains degrés de libertés (si possible)

étape 3 : résoudre les équations simplifiées.

Généralement : ⇒ relations trigonométriques

⇒ relations non-linéaires

⇒ pas toujours facile à résoudre

4 types d'équation souvent rencontrées, l'inconnu de ces types d'équation est θ et il est calculé avec $\text{Atan2}(y, x)$ qui donne l'angle dans son cadran.

3.7 Description géométrique du robot d'application

Modèle géométrique robot plan 3 axes

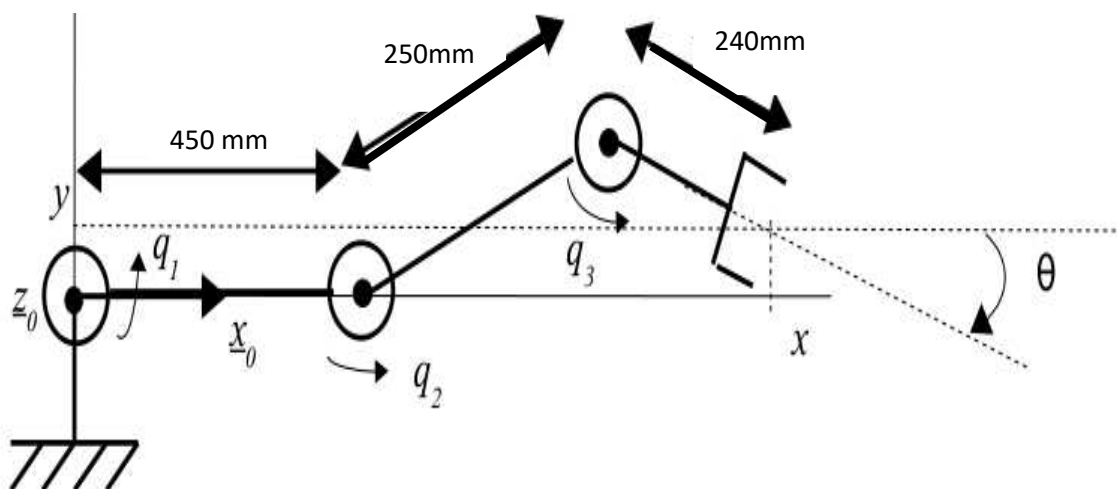


Figure 3.10- robot plant 3 axes

Le modèle géométrique direct est décrit par les équations suivantes :

$$x = r_1 \cos(q_1) + r_2 \cos(q_1 + q_2) + r_3 \cos(q_1 + q_2 + q_3)$$

$$y = r_1 \sin(q_1) + r_2 \sin(q_1 + q_2) + r_3 \sin(q_1 + q_2 + q_3)$$

a. Placement des repères

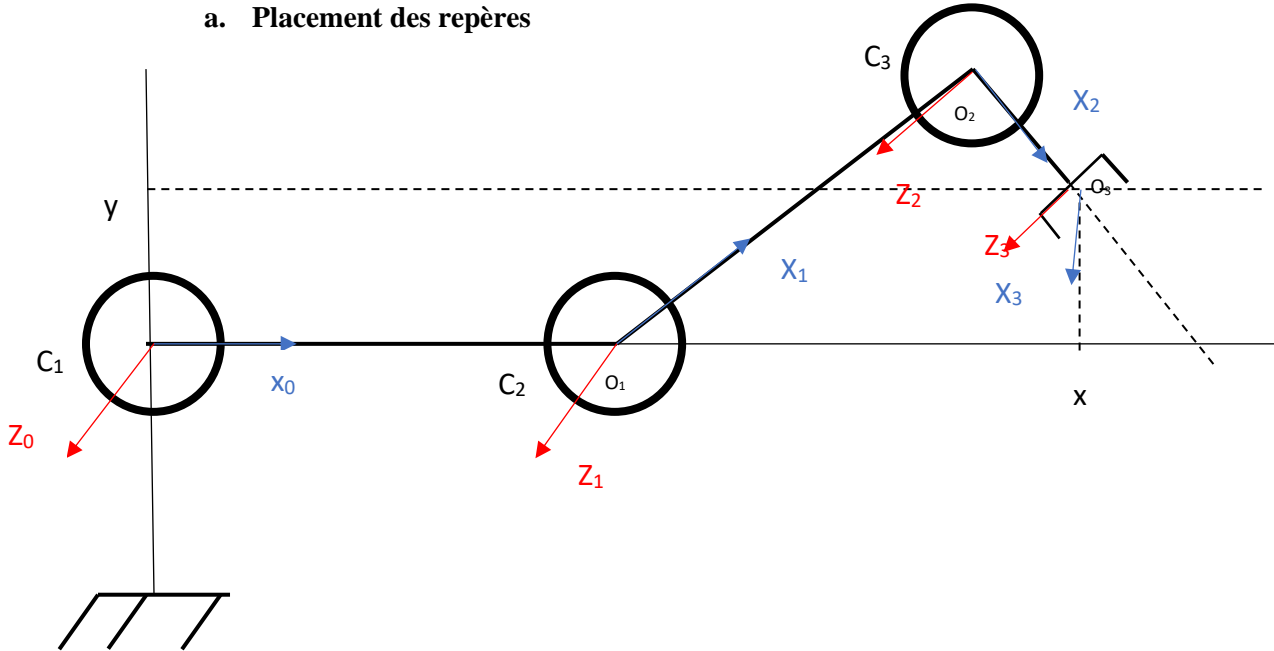


Figure 3.11-placement des repères sur le robot

b. Les paramètres géométriques du robot

Les paramètres articulaires obtenus après application de la convention de Denavit- Hartenberg modifiée sont résumés dans le tableau suivant :

Tableau 1 – Paramètres articulaires

Articulation	θ_i	r_i	d_i	α_i
C ₁	q_1	0	d1	0
C ₂	q_2	0	d2	0
C ₃	q_3	0	d3	0

Les matrices de transformation homogène sont :

$${}^0T_1 = \begin{bmatrix} Cq1 & -Sq1 & d1Cq1 \\ Sq1 & Cq1 & d1Sq1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} Cq2 & -Sq2 & d2Cq2 \\ Sq2 & Cq2 & d2Sq2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} Cq3 & -Sq3 & d3C3 \\ Sq3 & Sq3 & d3S3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_3 = \begin{bmatrix} C(q1 + q2 + q3) & -S(q1 + q2 + q3) & d1Cq1 + d2C(q1 + q2) + d3C(q1 + q2 + q3) \\ S(q1 + q2 + q3) & C(q1 + q2 + q3) & d1Sq1 + d2S(q1 + q2) + d3S(q1 + q2 + q3) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.8 Conclusion

Pour étudier un robot à n articulations, il faut représenter le comportement du ce robot sous la forme d'un modèle mathématique, une telle démarche s'appelle la modélisation. Dans ce chapitre, nous avons défini les éléments essentiels à la modélisation des robots, notamment les différents types d'architectures, les matrice de transformation homogène et la convention de Denavit-Hartenberg qui nous a permis d'obtenir le modèle géométrique direct du robot plan.

Chapitre VI

Application

4.1 Le principe d'application

Le principe de notre application est d'appliquer l'inversion ensembliste sur un robot plan trois axes. Nous choisissons d'étalonner le robot en essayant de déterminer des encadrements pour les longueurs des trois segments. Les valeurs à priori pour ces segments sont :

$$d1 = 450 \text{ mm}$$

$$d2 = 250 \text{ mm}$$

$$d3 = 240 \text{ mm}$$

Nous disposons de mesures expérimentales dans plusieurs cas : un axe mobile et les deux autres immobiles, puis les trois mobiles en même temps. Les simulations ont été exécutées sous Matlab avec la Toolbox Intlab. Dans le but d'augmenter la précision des intervalles et d'améliorer le temps de simulation qui est une des contraintes quand on utilise les méthodes ensemblistes, on améliore l'algorithme Sivia utilisé en lui associant une étape de contraction qui diminue le nombre des bisections, cet Algorithme est SiviaP.

4.2 les résultats de la simulation

Les résultats de simulations obtenus sont :

1/ Axe 1 qui bouge, axe 2 et axe 3 immobiles

$$p0 = [0.3, 0.5] \times [0.2, 0.4] \times [0.2, 0.4]$$

epsilon = 0.01;

erreur additive = 0.01

r_i sont les longueurs des axes qu'on cherche à étalonner, on obtient des intervalles pour chaque valeur r_i

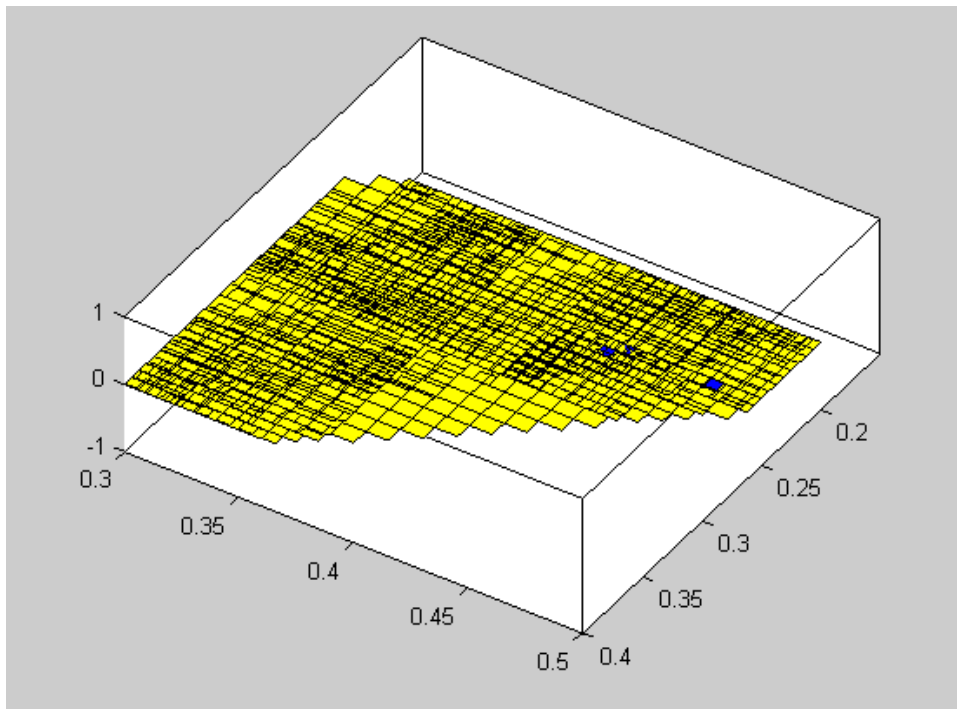


Figure 4.1 – Pavés solutions relatives aux paramètres r1, r2 et r3 obtenu par Sivia

r1= [0.4288, 0.4891]

r2= [0.1999, 0.3]

r3= [0.1999, 0.3]

Elapsed time is 184.793112 seconds.

2/ Axe 2 qui bouge, axe 1 et axe 3 immobiles

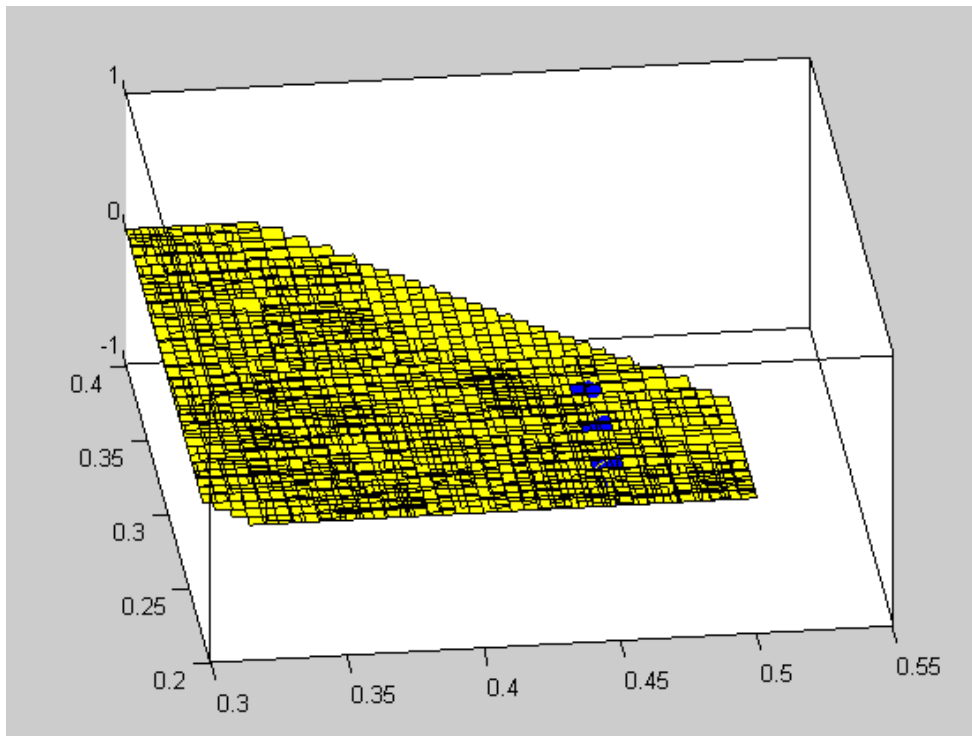


Figure 4.2 – Pavés solutions relatives aux paramètres r1, r2 et r3 obtenu par Sivia

r1= [0.4437, 0.4607]

r2= [0.1999, 0.2876]

r3= [0.1999, 0.2876]

Elapsed time is T= 231.187013 seconds.

3/ Axe 3 qui bouge, axe 1 et axe 2 immobiles

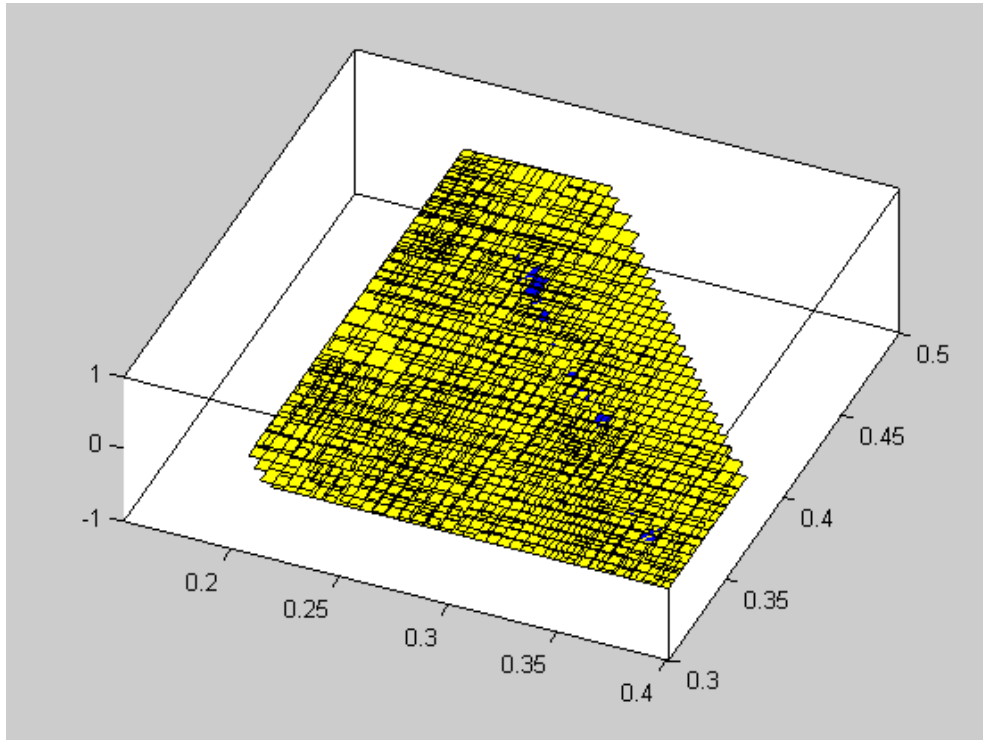


Figure 4.3 – Pavés solutions relatives aux paramètres r1, r2 et r3 obtenu par Sivia

r1= [0.3050, 0.5000]

r2=[0.2062, 0.4]

r3= [0.2292, 0.2472]

Elapsed time is T= 243.207279 seconds.

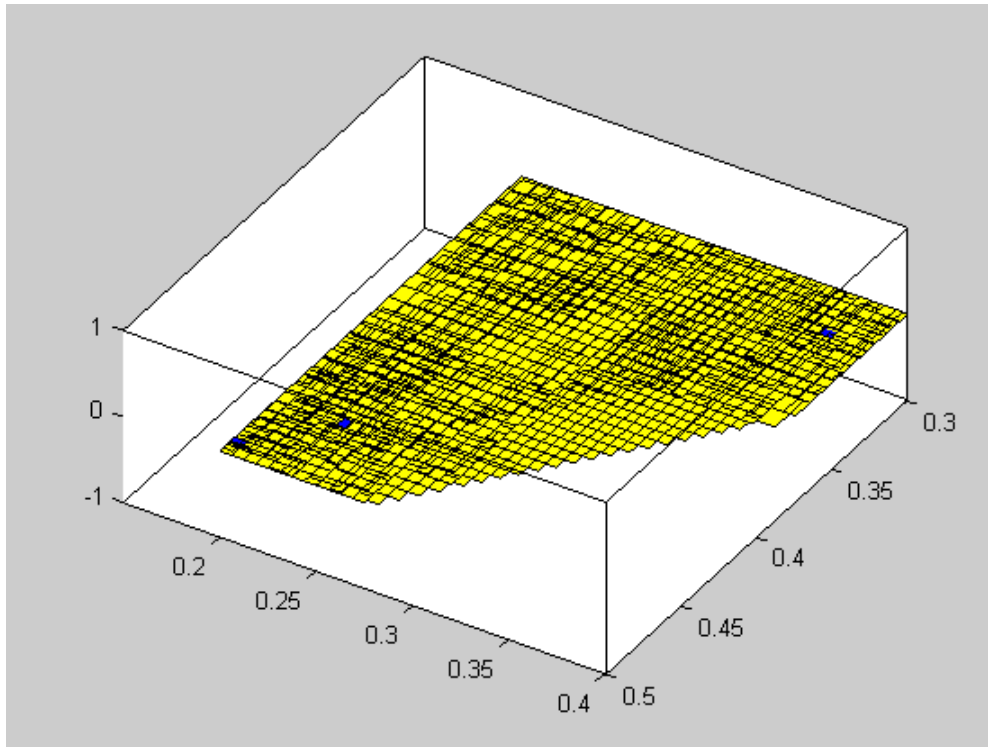


Figure 4.4 – Pavés solutions relatives aux paramètres d'intervalle p obtenu par Sivia

$$r_1 = [0.3054, 0.5000]$$

$$r_2 = [0.1999, 0.3813]$$

$$r_3 = [0.2249, 0.2443]$$

Elapsed time is 404.940596 seconds.

4.3 Modification des 3 axes en mouvement

On teste d'abord l'algorithme Sivia qui nous fournit la figure suivante :

L'intervalle de recherche initial est :

$$p_0 = [0.3, 0.5] \times [0.2, 0.4] \times [0.2, 0.4]$$

epsilon = 0.01;

erreur additive = 0.01

Sivia

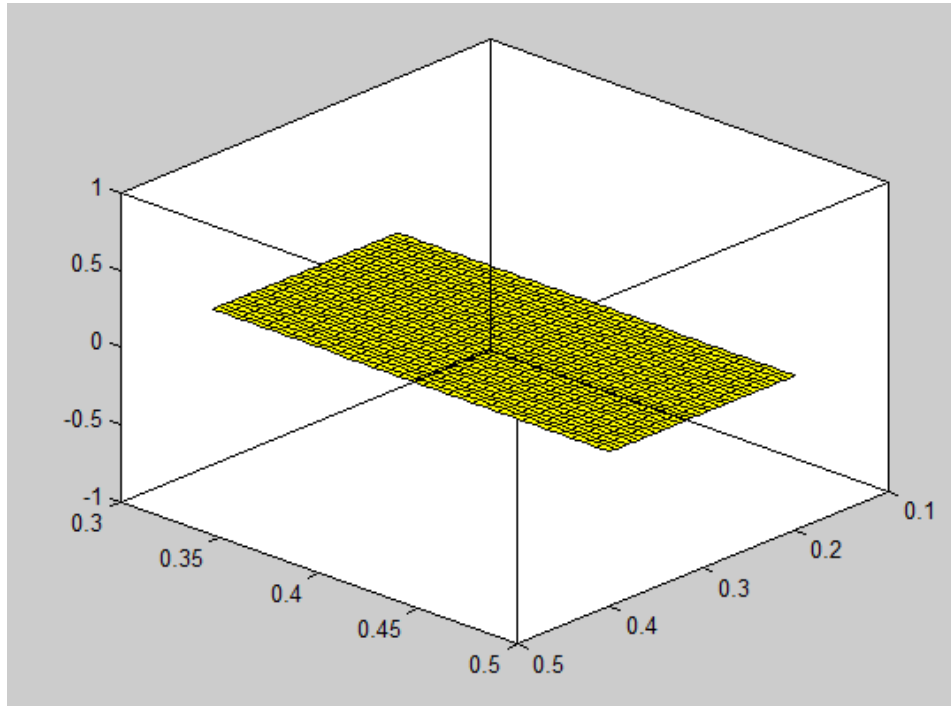


Figure 4.5 – Pavés solutions relatives aux paramètres d'intervalle p après amélioration obtenu par Sivia

Les encadrements relatifs à chaque axe sont les suivants :

$$r_1 = [0.3126, 0.5000]$$

$$r_2 = [0.2062, 0.4000]$$

$$r_3 = [0.2312, 0.2376]$$

Elapsed time is 4420.237812 seconds.

Afin d'améliorer le temps de simulation, on diminue le nombre de bisections en introduisant un contracteur basé sur des projections.

Algorithme SiviaP

SiviaP (Sivia + contracteur)

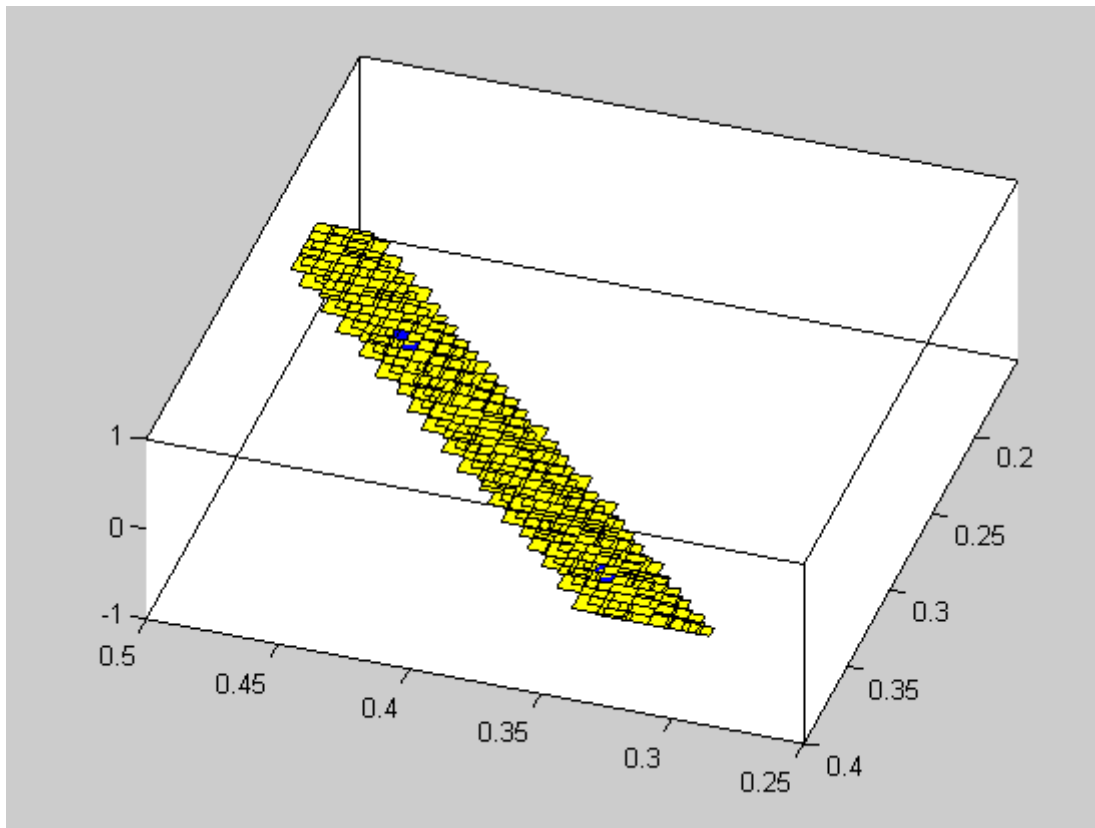


Figure 4.6 – Pavés solutions relatives aux paramètres d'intervalle p obtenu par SiviaP

$$r_1 = [0.3463, 0.4647]$$

$$r_2 = [0.2459, 0.3564]$$

$$r_3 = [0.2276, 0.2468]$$

Elapsed time is 74.332108 seconds.

On constate que le temps est nettement amélioré ainsi que la largeur de l'intervalle.

On peut améliorer la précision

Précision : $\text{eps} = 0.005$

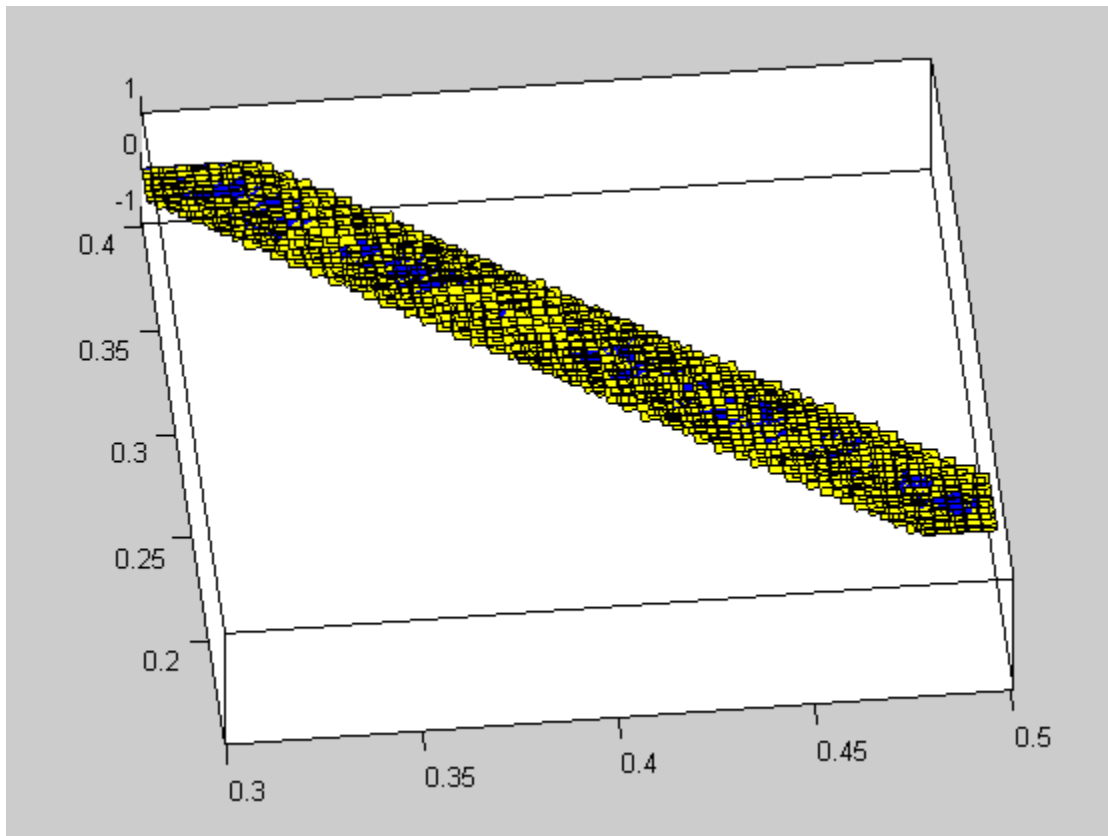


Figure 4.7 – Pavés solutions relatives aux paramètres d'intervalle p obtenu par SiviaP

$r_1 = [0.2999, 0.5000]$

$r_2 = [0.1999, 0.4000]$

$r_3 = [0.2410, 0.2500]$

Elapsed time is 399.553832 seconds.

Afin de confirmer nos résultats, nous avons ensuite testé un solveur Proj2D conçu par Masse Dao et Luc Jaulin [18]

Résultats

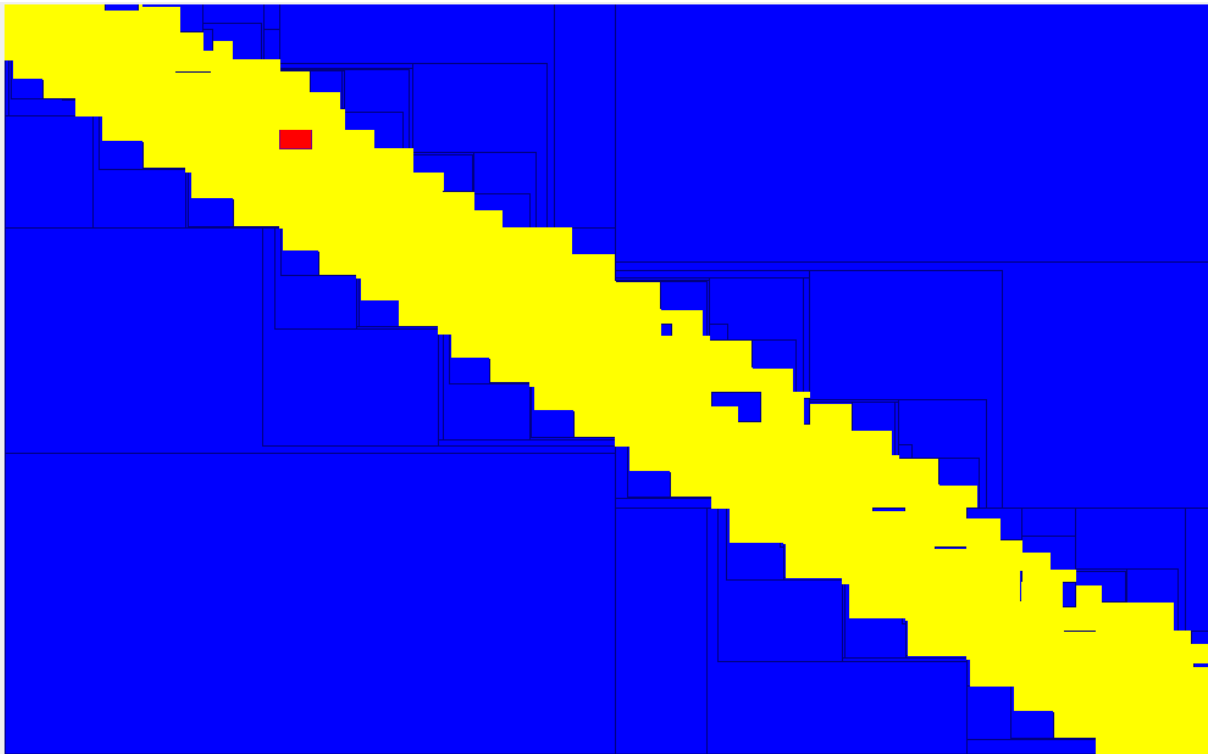


Figure 4.8 – Pavés solutions relatives aux paramètres p_1 , p_2 , p_3 testé par un solveur Proj2D

Les solutions sont représentées par des boîtes rouges.

Les boîtes jaunes sont indéterminées

$p_1 = [0.344980866853, 0.350280347979123]$

$p_2 = [0.361310466292726, 0.366451909929]$

$p_3 = [0.226166498771732, 0.232190388042873]$

Computing Time: 3.198s

Eps=0.01

Puis en améliorant la précision :

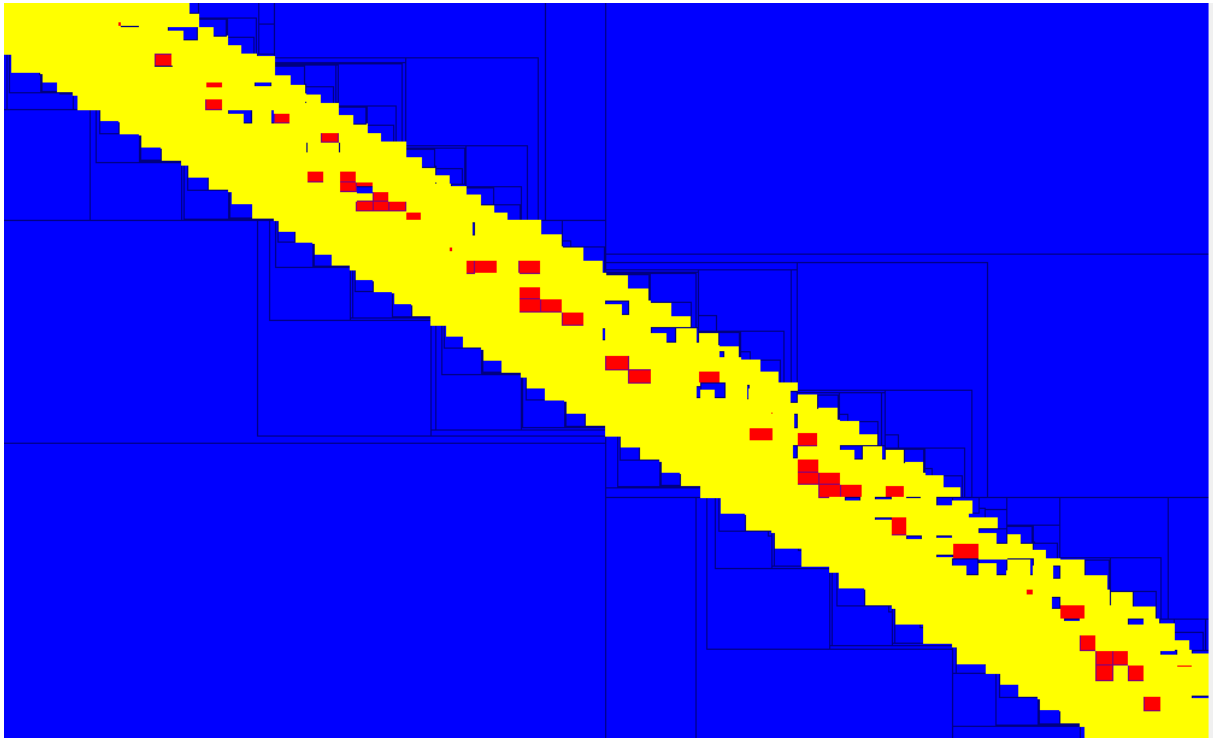


Figure 4.9 – Pavés solutions relatives aux paramètres p_1 , p_2 , p_3 testé par un solveur Proj2D

$p_1 = [0.3, 0.5]$

$p_2 = [0.2, 0.4]$

$p_3 = [0.223341070194639, 0.250573339261123]$

Computing Time: 15.772s

A part le temps qui est très court, il existe beaucoup de similitudes avec ceux obtenue par lignes de commandes

4.4 Conclusion

Cette étude comparative nous a permis de constater que les intervalles obtenus par l'inversion ensembliste sont assez larges et le temps de simulation est long. l'introduction d'un contracteur associé à l'algorithme Sivia nous a permis de réduire le temps de simulation et la longueur de l'intervalle autour de chaque paramètre identifié. Nous pouvons donc conclure que les méthodes ensemblistes nous permettent de déterminer les solutions de manière garantie.

Conclusion générale

Ce travail de recherche a abouti au développement d'une méthode d'amélioration de la précision de la position de l'effecteur final des robots manipulateurs. L'étude réalisée est illustrée sur le robot. L'objectif de ces travaux de thèse est alors de formaliser une méthode de modélisation géométrique adapté aux tâches de grande précision en utilisant la technique d'inversion ensembliste.

Après avoir présenté les fondements de base de cette approche telle que les notions principales reliées à la manipulation des ensembles et illustré quelques représentations ensemblistes, nous avons constaté que la majorité de ces représentations se basent sur un outil de calcul qui est l'arithmétique des intervalles. Puis nous avons abordé les principes de base de cet outil, nous avons souligné l'une des difficultés engendrées par la manipulation des vecteurs intervalles qui était le pessimisme présent dans le résultat des calculs, nous avons présenté les solutions apportées à ce problème. Des outils mettant en application cette arithmétique tels que l'inversion ensembliste par l'arithmétique des intervalles et la caractérisation d'ensembles par contraction ont été illustrés, qui sont généralement utilisés pour caractériser les systèmes d'équations linéaires et non linéaires. Nous avons modélisé le robot 3 axes avec une méthode classique.

Enfin, faire un essai de cette méthode sur l'étalonnage des robots séries semble une perspective intéressante. La possibilité de diminuer le nombre de bisections et l'amélioration de la précision et le temps de simulation avec des méthodes de contractions.

ANNEXE 1

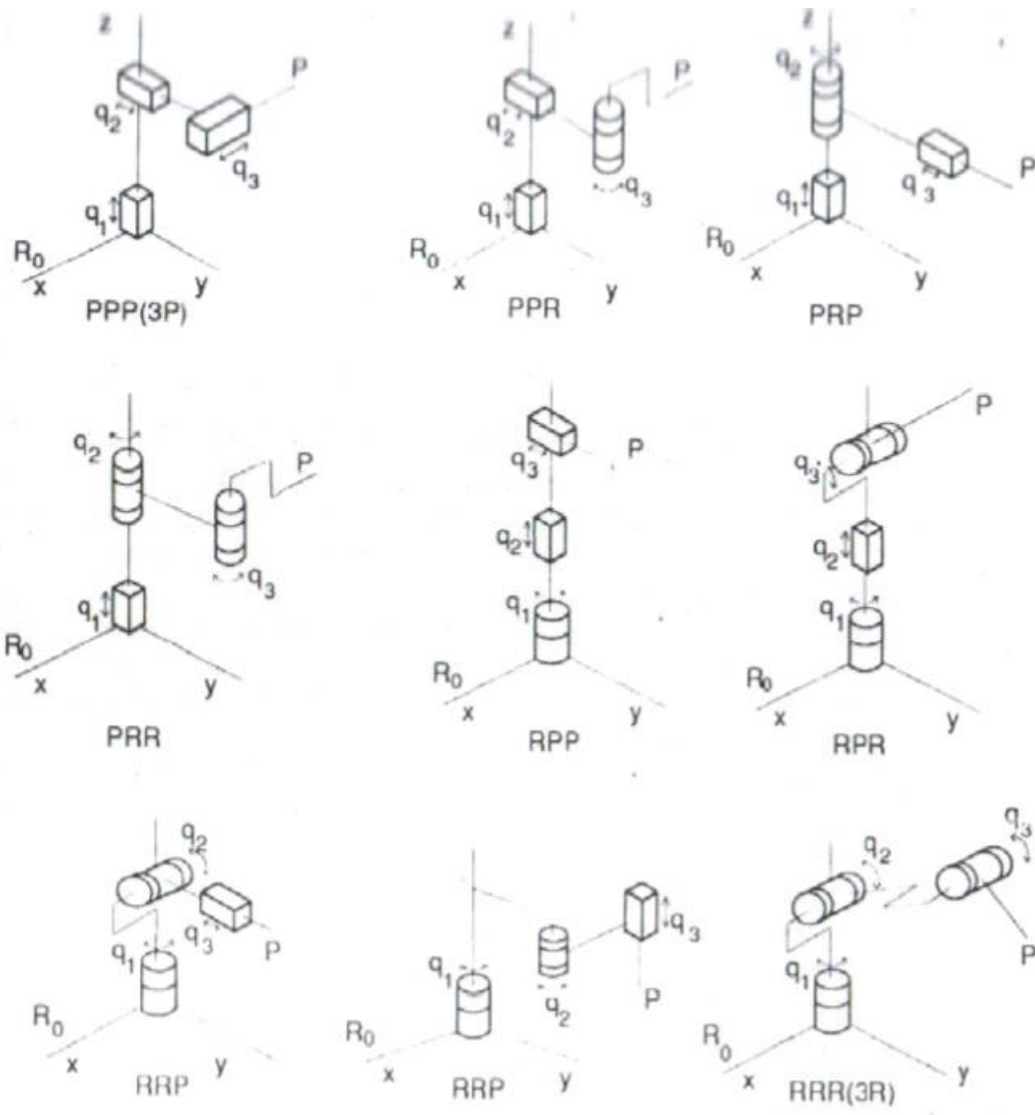
Différentes liaisons mécaniques

Nature de la liaison et position par rapport au repère	Schématisation spatiale	Schématisation plane	Mouvements possibles dans le repère donné
Encastrement			0 0 0 0 0 0
Glissière d'axe (A, \vec{x})			T_x 0 0 0 0 0
Pivot d'axe (A, \vec{z})			0 0 0 0 0 R_x
Pivot glissant d'axe (A, \vec{x})			T_x R_x 0 0 0 0
Hélicoïdale d'axe (A, \vec{x})			<i>comb. inés</i> T_x R_x 0 0 0 0
Rotule de centre A			0 R_x 0 R_y 0 R_z
Linéaire annulaire de centre A et d'axe (A, y)			0 R_x T_y R_y 0 R_z
Appui plan de normale (A, y)			T_x 0 0 R_y T_z 0
Linéaire rectiligne de normale (A, y) et de droite de contact (A, \vec{x})			T_x R_x 0 R_y T_z 0
Ponctuelle de normale (A, \vec{x})			0 R_x T_y R_y T_z R_z

1. <http://jerome.fraisse.pagesperso-orange.fr/interet.htm>

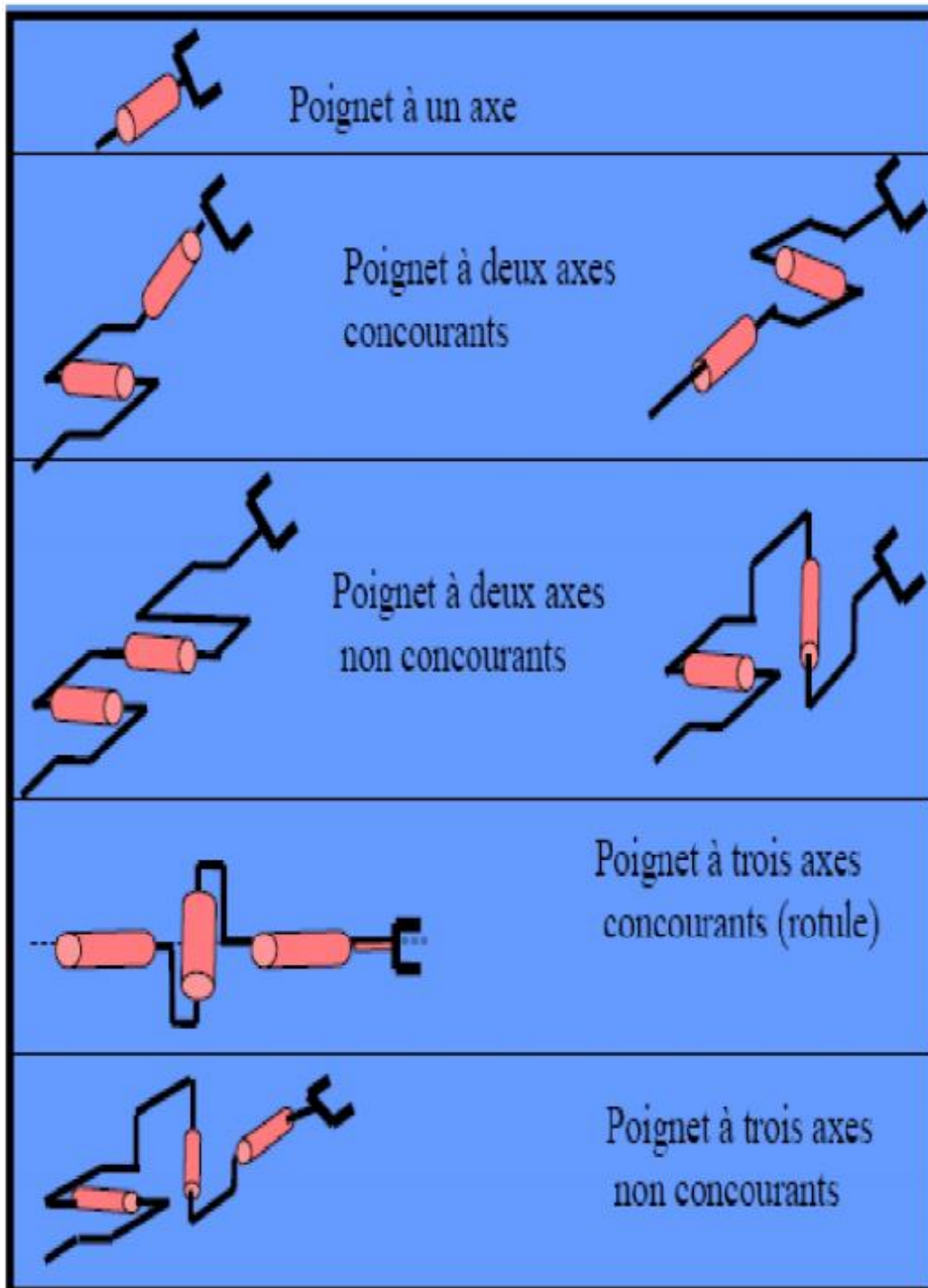
ANNEXE 2

Différentes combinaisons d'architectures du porteur



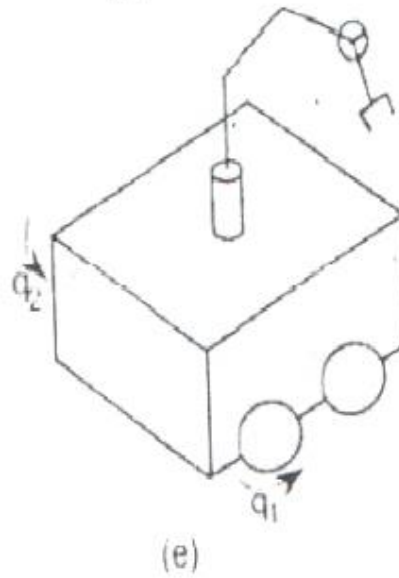
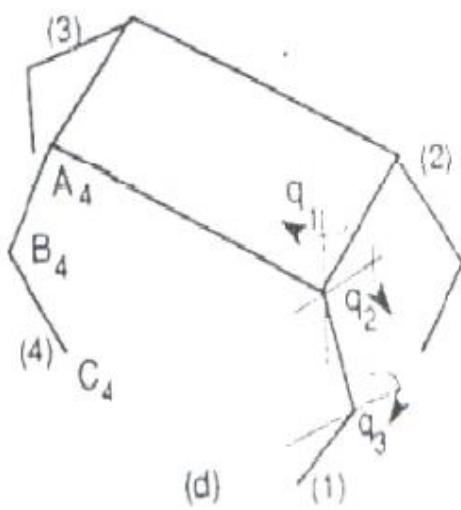
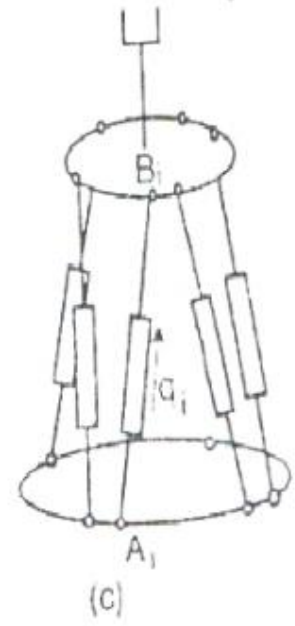
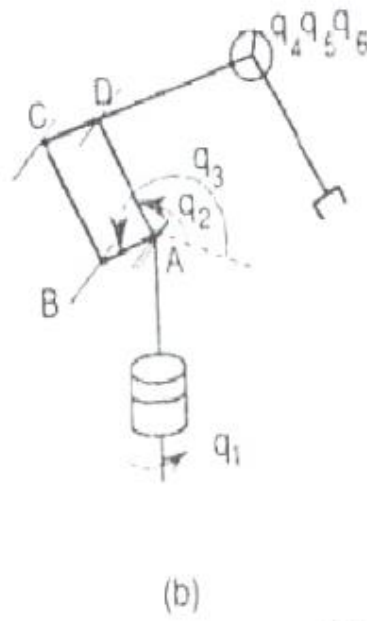
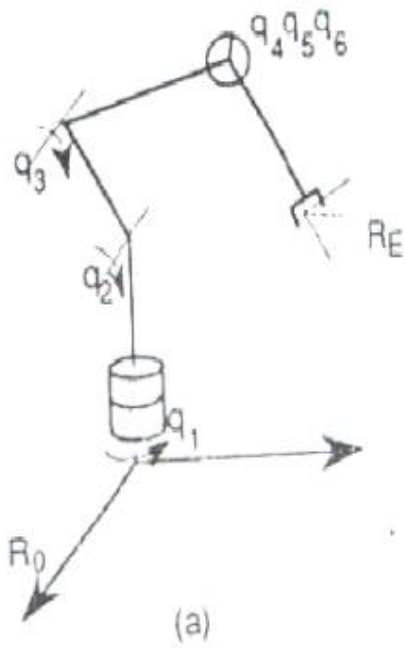
Annexe 3

Architecture des poignets



Annexe 4

Différents types d'architecture des Robots



Bibliographies

[1] www.sophia.inria.fr

[2] www.wekepedia.fr

[3] R. Moore, *Interval arithmetic and automatic error analysis in digital computing*, Phd Thesis, Department of Mathematics, Stanford University, California, Published as Applied Mathematics and Statistics Laboratories Technical Report no.25, 1962.

[4] R. Moore, *Interval analysis*, Prentice Hall, 1966.

[5] Isabelle Braems. *Méthodes ensemblistes garanties pour l'estimation de grandeurs physiques*. Phd of, 2002.

[6] L.Jaulin, *Solution globale et garantie de problèmes ensemblistes : application à l'estimation de problèmes ensemblistes et à la commande*, Phd Thesis, Université de ParisSud, centre d'Orsay, France, 1994.

[7] S. BOURSALI, *METHODES ENSEMBLISTES POUR L'IDENTIFICATION DE PARAMETRES : APPLICATION AU BIOREACTEUR*, thèse de Doctorat, Université Aboubakr Belkaïd– Tlemcen – Faculté de TECHNOLOGIE, 2017

[8] L. Jaulin and E. Walter, *Set inversion via interval analysis for nonlinear bounded-error estimation*, *Automatica*, vol. 29 no.4, pp.1053-1064, 1993.

[9] L. Jaulin, M. L. Kieffer, O. Didrit et E. Walter, *Applied interval analysis with Examples in Parameter and State Estimation*, *Robust Control and Robotics*. Springer verlag, London, 2001.

[10] Xavier Baguenard. *Propagation de contraintes sur les intervalles Application à l'étalonnage des robots*. PhD thesis, INRIA SOPHIA ANTIPOLIS, 2005.

[11] *Cours de robotique fondamentale* David.Daney sophia.inria.fr Projet Coprin INRIA Sophia Antipolis

[12] DEBLAISE D., *Contribution à la modélisation et à l'étalonnage géométriques des manipulateurs parallèles*, thèse de Doctorat, Institut National des Sciences Appliquées, 2006.

[13] [Khalil&Dombre1999] Wisama Khalil et Etienne Dombre « *Modélisation Identification et Commande de robots* » Hermès 1999

[14] GÜRSEL A., *Prediction of geometric errors of robot manipulators with Particle Swarm Optimisation method*, *Robotics and Autonomus Systems*, Volume 54, pp. 956-966, 2006

[15] A.BELLOUFI “*Cours de Robotique*”, Université KASDI MERBAH OURGLA, Algerie (2015).

[16] Eldon Hansen and G William Walster. *Global optimization using interval analysis : revised and expanded*, volume 264. CRC Press, 2003

[17] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Middle East Library. Cambridge University Press, 1990

[18] Luc Jaulin. *Interval constraint propagation with application to bounded-error estimation*. *Automatica*, 36(10) :1547–1552, 2000..