HAL open science

Contributions à l'identification enesembliste ellipsoïdale

Khoi Quoc Tran Dinh

To cite this version:

Khoi Quoc Tran Dinh. Contributions à l'identification enesembliste ellipsoïdale. Automatique / Robotique. Institut National Polytechnique de Grenoble - INPG, 2005. Français. NNT : . tel-00168416

HAL Id: tel-00168416

https://theses.hal.science/tel-00168416v1

Submitted on 14 Sep 2007

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

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

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
__/__/__/__/__/__/__/__/__/__/

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Automatique-Productique

préparé au Laboratoire d'Automatique de Grenoble
dans le cadre de **l'Ecole Doctorale Electronique, Electrotechnique, Automatique, Télécommunications, Signal**

présentée et soutenue publiquement

par

TRAN DINH Khoi Quoc

le 19 juillet 2005

Contributions à l'identification ensembliste ellipsoïdale

Directeur de thèse : *M. Alain BARRAUD*

JURY

Mme	Sylviane GENTIL	Présidente
M.	Dominique MEIZEL	Rapporteur
M.	Alain RICHARD	Rapporteur
M.	Alain BARRAUD	Directeur de thèse
Mme	Suzanne LESECQ	Co-encadrant
M.	Benoît CELSE	Examinateur

Remerciements

Les travaux présentés dans ce mémoire ont été réalisés au Laboratoire d'Automatique de Grenoble (LAG URM 5528) de l'Ecole Nationale Supérieure d'Ingénieur Electricien de Grenoble, l'Institut National Polytechnique de Grenoble. Je remercie tout d'abord ses directeurs successifs, Messieurs Luc DUGARD, Directeur de recherche CNRS et Alain BARRAUD, Professeur à l'Institut National Polytechnique de Grenoble, de m'avoir accueilli au sein de leur laboratoire.

Qu'il me soit permis d'exprimer toute ma gratitude à Madame Suzanne LESECQ, Maître de Conférences à l'Université Joseph Fourier, Habilitée à Diriger des Recherches, et Monsieur Alain BARRAUD, Professeur à l'Institut National Polytechnique de Grenoble qui ont dirigé et encadré cette thèse. Je tiens ici à leur témoigner ma reconnaissance sincère.

Que Madame Sylviane GENTIL, Professeur à l'Institut National Polytechnique de Grenoble, soit remerciée de m'avoir fait l'honneur de présider le jury de thèse.

Que Monsieur Dominique MEIZEL, Professeur à l'Université de Limoges et Monsieur Alain RICHARD, Professeur à l'Université Henri Poincaré-Nancy 1, trouvent ici l'expression de ma profonde gratitude pour m'avoir fait l'honneur de rapporter sur ce travail.

Je tiens à remercier Monsieur Benoît CELSE, Docteur au Département Informatique Scientifique de l'Institut Français de Pétrole, pour l'honneur qu'il m'a fait en acceptant de participer à ce jury.

Je ne pourrais oublier d'exprimer toute ma reconnaissance à tous les chercheurs et enseignants chercheurs ainsi qu'à l'ensemble du personnel technique et administratif du Laboratoire d'Automatique de Grenoble.

Je remercie plus particulièrement mes amis pour la bonne ambiance au laboratoire.

Enfin, je ne peux pas finir sans remercier ma famille et plus particulièrement mes parents et ma femme pour m'avoir soutenu et encouragé durant ces trois années.

Table des matières

Notations et abréviations **5**

Introduction **7**

Chapitre 1 Estimation ensembliste ellipsoïdale **13**

1.1 Fondement du problème d'estimation ensembliste à erreur bornée 13
1.2 Les algorithmes OBE 16
1.2.1 Méthodes basées sur la taille géométrique de l'ellipsoïde 19
1.2.2 Méthodes basées sur la convergence de σ_k^2 24
1.2.3 Synthèse sur les algorithmes OBE 27
1.2.4 Test de cohérence et réduction de bande 28
1.2.5 Généralisation : cas de l'intersection de deux ellipsoïdes 31
1.3 Modèle erreur de sortie pour l'estimation ellipsoïdale 35
1.3.1 Position du problème 35
1.3.2 Reformulation du problème pour un modèle à erreur de sortie 37
1.3.3 Test de cohérence de l'intersection et réduction de bande 41
1.3.4 Algorithme FHCL 43
1.3.5 Algorithme à une seule coupe de Ferreres et M'Saad 44
1.4 Conclusion 46

Chapitre 2 Application de l'algorithme OBE en identification **49**

2.1 Données simulées 49
2.1.1 Description des données 49
2.1.2 Résultats obtenus avec un modèle à erreur d'équation 51
2.1.3 Résultats obtenus avec un modèle à erreur de sortie 57
2.2 Données industrielles de l'IFP 58
2.2.1 Description des données 58
2.2.2 Résultats obtenus 59
2.3 Données industrielles du robot parallèle (LIRMM) 63
2.3.1 Description des données 63
2.3.2 Résultats obtenus 65
2.4 Conclusion 71

Chapitre 3 Approche factorisée dans l'estimation ensembliste ellipsoïdale 73
3.1 Algorithme de Goldfarb et Todd 73
3.1.1 Formulation générale de l'algorithme 73
3.1.2 Forme factorisée 78
3.2 Algorithme OBE factorisé 80
3.2.1 Raisons potentielles de l'instabilité des algorithmes OBE 80
3.2.2 Forme factorisée de l'algorithme OBE normalisé 81
3.2.3 Forme factorisée de l'algorithme OBE général 88
3.2.4 Test de cohérence de l'intersection entre deux ellipsoïdes : expression factorisée de ce test 91
3.3 Conclusion 92

Chapitre 4 Convergence de l'algorithme OBE 95
4.1 Propriétés de convergence des algorithmes OBE 96
4.2 Vérification de la persistance de l'excitation 102
4.2.1 Condition d'une excitation persistante 102
4.2.2 Création d'une excitation adéquate pour satisfaire la convergence de l'algorithme 104
4.3 Choix de la borne pour l'identification 107
4.3.1 Méthodes « classiques » de choix de la borne 108
4.3.2 Algorithme OBE-ABE 109
4.3.3 Méthode basée sur un calcul du Kurtosis de la séquence d'erreur 114
4.4 Résultats expérimentaux 120
4.4.1 Influence de l'excitation : données simulées 120
4.4.2 Influence du choix de la borne : données simulées 123
4.4.3 Influence de l'excitation : données IFP 126
4.4.4 Influence de l'excitation : données du robot parallèle 127
4.4.5 Influence du choix de la borne : données du robot parallèle 130
4.5 Conclusion 132

Chapitre 5 Traitement des données par paquet 135
5.1 Position du problème 136
5.2 Ellipsoïde optimal globalement 138
5.3 Méthodes classiques de traitement d'un paquet de données 140

5.3.1 Une procédure améliorée pour un traitement séquentiel [MAK 96] 140
5.3.2 Un ellipsoïde approximant l'intersection de deux ellipsoïdes 142
5.3.3 Synthèse 143
5.4 Algorithme OBE-paquet 144
5.4.1 Formulation de l'algorithme 144
5.4.2 Prétraitement des données 147
5.4.3 Forme factorisée de l'algorithme OBE-paquet 152
5.4.4 Algorithme OBE-paquet pour un modèle à erreur de sortie 154
5.5 Résultats expérimentaux avec l'algorithme OBE-paquet 156
5.5.1 Résultat sur les données simulées 157
5.5.2 Résultats pour les données de l'IFP 159
5.5.3 Résultats pour les données du robot parallèle 161
5.6 Conclusion 162

Conclusion et perspectives 165

Annexe A Le problème de moindres carrés linéaires 169

Annexe B La méthode du Simplex 173

Annexe C Démonstration du théorème 1.1 dans le cas de l'intersection d'un ellipsoïde et d'une bande 177

Annexe D Démonstration du théorème 4.9 179

Annexe E Application : approche ensembliste ellipsoïdale et détection des défauts 183

Bibliographie 191

Notations

α	1ère coefficient pour calculer le nouvel ellipsoïde
β	2ème coefficient pour calculer le nouvel ellipsoïde
γ	Borne de l'erreur
γ^*	Vraie borne de l'erreur
η	Erreur de sortie
η^M	Borne de l'erreur de sortie
λ	Coefficient de pondération pour calculer le nouvel ellipsoïde (dans l'algorithme SMSA, DH, Tan *et al*., LO, GT)
μ_T	Fonction critère de la trace
μ_V	Fonction critère du déterminant
ν	Erreur de prédiction
σ	Coefficient caractéristique d'un ellipsoïde
υ	Coefficient choisi par l'utilisateur dans algorithme de DH, LO
ς	Paramètre pour le rajustement de la borne dans l'algorithme OBE-ABE
B	Bande
C	Matrice des régresseurs
c	Centre d'un ellipsoïde
d	Vecteur régresseur
e	Erreur d'équation
E	Ellipsoïde
H	Hyperplan frontière d'une bande
h	La distance du centre d'un ellipsoïde à un hyperplan
l	« Longueur » des données (i.e. nombre de données)
l_0	Longueur de la fenêtre temporelle $\mathcal{F}$ dans l'algorithme OBE-ABE
M	Matrice inverse de la matrice P
m	La longueur du paquet de données
p	Dimension du vecteur de paramètres
P	Matrice caractéristique d'un ellipsoïde
q_k	Coefficient pour calculer le nouvel ellipsoïde dans l'algorithme de Fogel et Huang

S	Intersection de plusieurs bandes (l'algorithme OBE) ou de plusieurs demi-espaces (l'algorithme GT)
V	Demi-espace de paramètres
x	Vecteur de paramètres
x^*	Vecteur de paramètres « vrais » (ou théoriques)
y	Sortie mesurée

Remarque : les vecteurs sont par défaut des vecteurs colonnes.

Abréviations

ABE	Automatic Bound Estimation
DH	L'algorithme de Dasgupta et Huang
FH	L'algorithme de Fogel et Huang
FHCl	L'algorithme de Clément pour modèle à erreur de sortie
FM	L'algorithme de Ferreres et M'Saad pour modèle à erreur de sortie
GT	L'algorithme de Goldfarb et Told
i.i.d.	independent identically distributed
LO	L'algorithme de Lozano-Leal et Ortega
OBE	Outer Bounding Ellipsoid
PE	Persistance d'excitation
SISO	Single Input Single Output
SMSA	Set Membership Stochastic Approximation
TAN	L'algorithme de Tan *et al.*

Introduction

L'estimation de paramètres joue un rôle important dans plusieurs domaines de l'automatique en particulier l'identification, la commande de processus ou encore en diagnostic. Elle retient l'attention de nombreux automaticiens et mathématiciens depuis plusieurs dizaines années. Dans le cadre des systèmes dynamiques linéaires de très nombreuses méthodes d'identification ont été développées. On peut les classer succinctement en deux catégories : les méthodes qui donnent une estimée ponctuelle des paramètres, dites ici méthodes « classiques », et les méthodes conduisant à un ensemble, dites méthodes ensemblistes. Aujourd'hui encore, de nouveaux algorithmes voient le jour, avec pour objectif d'améliorer des techniques encore perfectibles ou/et de les simplifier.

Dans les méthodes classiques, les paramètres estimés sont les valeurs optimales obtenues par optimisation d'une certaine fonction critère. Cette optimisation est réalisée habituellement par des algorithmes séquentiels ou itératifs, ou encore mixant ces deux stratégies. La qualité de l'estimateur est alors définie par son espérance mathématique et sa matrice de covariance. Ceci suppose que les différentes erreurs (bruits de mesures, perturbations, erreurs de modélisation,…) ont été assimilées à des variables aléatoires caractérisées par leur moyenne, covariance, ou densité de probabilité. L'estimation obtenue dépend alors de cette caractérisation. En pratique, avec des données réelles, il est peu probable de pouvoir disposer systématiquement d'une telle information. Or, le non-respect des hypothèse statistiques sur ces erreurs peut conduire à des interprétations fausses sur la qualité de l'estimée. L'idée défendue ici, est qu'il est plus réaliste, voire naturel, de supposer connaître uniquement des bornes sur ces erreurs.

Ce concept d'utilisation de la borne sur ces erreurs a créé un nouveau type d'approches en identification, dite approche ensembliste. Cette approche n'a besoin d'aucune hypothèse sur la distribution statistique des erreurs mais uniquement sur sa borne qui est a priori moins difficile à obtenir. Dans ce contexte, le résultat d'identification se présente sous la forme d'un ensemble de tous les paramètres du modèle compatibles avec la borne sur l'erreur. Le vecteur des « vrais » paramètres est alors réputé se trouver à l'intérieur de cet ensemble.

Souvent, l'ensemble exact décrivant l'espace des paramètres est très complexe, même si la dimension de l'espace des paramètres reste limitée. De plus cette complexité ne fait que croître au fur et à mesure que le nombre de données augmente. Calculer directement cet ensemble devient très difficile voire irréalisable. Dans les approches ensemblistes, on cherche donc un ensemble de forme géométrique plus simple qui englobe l'ensemble réel. Les ensembles usuellement utilisés sont les polytopes, les parallèlotopes, les orthotopes ou encore les ellipsoïdes.

Parmi les approches d'estimation ensembliste, les méthodes utilisant les ellipsoïdes présentent des avantages certains. En particulier, l'espace paramétrique (un ellipsoïde) est caractérisé uniquement par son centre et une matrice qui définit sa taille et son orientation. Potentiellement, l'algorithme de mise à jour peut devenir très simple et donc être implanté, si besoin, en temps réel. L'idée d'utiliser des ellipsoïdes a été initialement proposée à la fin des années soixante par Schweppe [SCH 68] pour une estimation d'état d'un modèle. Fogel et Huang [FOG 82] construisent ensuite un algorithme, devenu fondamental, pour l'estimation d'un modèle linéaire en les paramètres. Au cours des années suivantes, plusieurs auteurs différents ont développé cette approche dans de nouvelles directions. Clément [CLE 87], Grötschel et al. [GRO 88], Ferres *et al*. [FER 97], … adaptent cette approche à un modèle ARMAX. Dasgupta et Huang [DAS 87], Nayeri *et al.* [NAY 94] s'intéressent à certaines propriétés de convergence des algorithmes. Durieu *et al*. [DUR 01b], Maksarov et Norton [MAK 02] contribuent à l'estimation d'état et l'intersection de plusieurs ellipsoïdes en même temps.

Néanmoins l'application pratique de ce genre de techniques à l'estimation paramétrique d'un modèle d'un processus industriel, pose entre autre l'éternelle question de la détermination de la borne à utiliser pour l'identification. Il est clair que cette borne est un facteur primordial impactant directement la taille des ellipsoïdes, et in fine l'intervalle estimé des paramètres. Avec un processus réel parfaitement dominé, par exemple dans [SED 98], en analysant toutes les causes d'erreur et les sources de perturbation, on peut, certes, évaluer une borne réaliste sur ces erreurs. Cependant, dans la plupart des cas, cette borne reste généralement plus ou moins mal connue.

Lever cette difficulté est l'un des objectifs de ce travail. Pour étayer les arguments que nous allons développer, nous avons illustré nos résultats sur deux problèmes industriels très différents. Le premier, fourni par l'IFP (Institut Français de Pétrole), est très simple en termes de dimension de l'espace paramétrique et la qualité des données disponibles est médiocre (*i.e.* dynamique pauvre). La difficulté ici, vient de la grande pauvreté informationnelle contenue dans les mesures. Le second, issu d'un robot parallèle du LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier) présente des caractéristiques complètement opposées, à savoir, un « grand » nombre de paramètres (14), des milliers de mesures dont un certain nombre de données aberrantes.

Mais maîtriser l'évaluation de la borne ne résout pas tous les problèmes. Encore faut-il que les algorithmes implantés soient numériquement stables. Ce prérequis n'a que peu d'impact sur des problèmes bien conditionnés, mais peut s'avérer incontournable dans le cas contraire. Or on ne connaît jamais à l'avance la sensibilité qu'induiront les conditions d'identification rencontrées. A cet égard, il est assez surprenant de constater que la quasi-totalité des publications propose des algorithmes qui sont numériquement instables ! Répondre à cette objection est notre seconde principale contribution. Tous les algorithmes développés sont numériquement stables et basés sur des techniques de factorisation orthogonale dont la réputation n'est plus à faire. Outre leurs qualités intrinsèques, les méthodes proposées sont simples à programmer et ne font appel qu'à des fonctions standard d'algèbre linéaire.

Au-delà des deux aspects majeurs de ce travail, on a également abordé les propriétés théoriques que l'on pouvait avancer pour l'ensembles des techniques abordées, en particuliers quels types de convergence il était possible d'énoncer, et sous quelles conditions. L'aspect « persistance de l'excitation » a été rappelé en détail et son évaluation pratique a fait l'objet de propositions algorithmiques concises. Dans la continuité de ce concept, on a proposé une méthodologie pour construire les « meilleures » entrées possibles dans le contexte ensembliste.

L'intérêt de l'aspect séquentiel, sur lequel reposent basiquement les démarches ensemblistes (simplicité, rapidité, temps réel), génère aussi ses propres limites (sous optimalité des résultats). Face à cela, la « recirculation » des données constitue une alternative

qui a fait ses preuves et à laquelle on a fait largement appel. Néanmoins, on a analysé, ici, une autre alternative reposant sur le traitement des données par « paquets » dont l'objectif est de trouver un compromis entre la sous-optimalité du séquentiel pur, et l'inaccessibilité numérique d'un traitement global complet.

Le présent mémoire est organisé de la façon suivante.

Le premier chapitre présente une synthèse des algorithmes ensemblistes du type ellipsoïdal, dits algorithme OBE (Outer Bounding Ellipsoid). Ces algorithmes sont introduits en relation avec l'algorithme général. Le développement de cette méthode d'identification dans le contexte d'un modèle à erreur de sortie est aussi présenté.

Le second chapitre est d'une part consacré à la description des deux applications réelles et d'autre part à la définition des jeux de données simulées, qui seront utilisées tout au long de ce mémoire. C'est aussi l'occasion de faire un bilan comparatif, sur le plan expérimental, des approches dont on a fait précédemment la synthèse.

Le troisième chapitre est consacré à l'implémentation des algorithmes OBE de façon à en garantir la stabilité numérique. Une telle démarche avait été partiellement abordée par Goldfarb et Told en faisant appel à la factorisation LDL^T. Nous proposons ici une approche systématique basée sur la factorisation orthogonale du type QR applicable à toutes les situations et dont l'interprétation en terme d'optimisation quadratique donne une nouvelle vision des formules de mise à jour des ellipsoïdes.

Le quatrième chapitre s'intéresse aux propriétés de convergence des algorithmes. Ceci nous amènera à discuter de la persistance de l'excitation, de la maîtrise de la borne d'erreur et celle des directions engendrées par les régresseurs. Une méthode pour créer une excitation adaptée à cette approche d'identification est présentée. Concernant la question sur la borne, on analyse une approche basée sur la réduction automatique de la borne durant le processus d'identification et un nouvel algorithme reposant sur l'analyse du kurtosis d'une séquence d'erreur pré-estimée. Tous ces résultats sont illustrés en simulation et sur données réelles.

Dans le dernier chapitre, nous nous intéressons au traitement par paquets des données dans le but de trouver un compromis entre « séquentiel » et « global ». La principale difficulté

est la croissance du nombre de paramètres par rapport auxquels devra être résolu un problème d'optimisation non linéaire sous contraintes. L'algorithme proposé est complété par une procédure de prétraitement faisant appel à la méthode du simplexe. Là encore, ces résultats sont illustrés, en particulier sur des données réelles.

La conclusion dresse un bilan des travaux qui ont été effectués et des perspectives de recherche que l'on peut donner à ce travail sont également exposées.

Chapitre 1

Estimation ensembliste ellipsoïdale

L'identification d'un système linéaire en les paramètres par les méthodes « traditionnelles » nécessite une connaissance *a priori* de la caractéristique statistique de l'erreur affectant la sortie pour en tirer le meilleur parti. Pour un procédé réel, la description de cette erreur sous la forme d'une loi de probabilité semble irréalisable lorsque l'erreur provient de plusieurs sources inconnues différentes, par exemple le bruit sur les mesures, l'erreur de modélisation ou bien d'autres perturbations non déterministes… D'autre part, un autre problème pour choisir cette loi provient du nombre de données souvent insuffisant pour construire (ou choisir de manière réaliste) une caractéristique statistique. Une alternative consiste alors à utiliser une borne sur l'erreur, ce qui conduit à une approche ensembliste qui peut sembler attrayante et plus réaliste puisqu'elle ne nécessite pas de connaissance *a priori* ou d'hypothèse statistique sur le bruit affectant les mesures du procédé à identifier. Ce type d'approches a attiré ces deux dernières décennies l'attention de nombreux auteurs. Alors que leur utilisation était fortement limitée dans les années 1980 par les capacités des calculateurs disponibles, la puissance actuelle et croissante de nos machines les remet à l'ordre du jour.

Une différence fondamentale entre ces approches ensemblistes et des méthodes plus « traditionnelles » réside dans le résultat obtenu. En effet, on ne trouve pas un vecteur ponctuel de paramètres associé à un domaine d'incertitude défini par des écarts types (sans aucune certitude d'être dans l'intervalle paramétrique ainsi défini) mais un ensemble contenant de manière sûre le vecteur des « vrais » paramètres. Notons que la notion de vecteur des « vrais » paramètres n'existe qu'en simulation.

Dans ce chapitre, on présente succinctement les idées clefs de l'identification ensembliste. On s'intéresse ensuite aux méthodes englobantes à base d'ellipsoïdes.

1.1 Fondement du problème d'estimation ensembliste à erreur bornée

Pour un système mono-entrée mono-sortie (SISO) linéaire en les paramètres, la sortie y_k, à l'instant k, peut s'écrire sous la forme :

$$y_k = d_k^T x^* + e_k \tag{1.1}$$

où :

- $x^* \in \mathbb{R}^p$ est le vecteur des « vrais » paramètres à estimer ;
- d_k est le régresseur contenant des entrées et des mesures aux instants précédents. d_k est supposé parfaitement connu ;
- e_k est l'erreur affectant la sortie.

Dans l'approche ensembliste, on suppose que l'on connaît uniquement la borne sur cette erreur, c'est-à-dire :

$$|e_k| \leq \gamma_k \tag{1.2}$$

A l'instant k, tous les paramètres du modèle satisfaisant à la fois les équations (1.1) et (1.2) se trouvent l'intérieur d'une bande $\mathbf{B}_k$ définie par :

$$\mathbf{B}_k = \left\{ x \in \mathbb{R}^p : \left(y_k - d_k^T x \right)^2 \leq \gamma_k^2 \right\} \tag{1.3}$$

Dans l'espace des paramètres, $\mathbf{B}_k$ est l'espace restreint entre deux hyperplans parallèles (cf. figure 1.1) :

$$\mathbf{B}_k = \left[\mathbf{H}_k^-, \mathbf{H}_k^+ \right]$$

En conséquence, après k mesures, l'ensemble $\mathbf{S}_k$ contenant les paramètres sera obtenu par l'intersection des k bandes $\mathbf{B}_i$:

$$\mathbf{S}_k = \bigcap_{i=1}^{k} \mathbf{B}_i \tag{1.4}$$

La figure 1.2 illustre l'ensemble $\mathbf{S}_k$ dans un espace à $p = 2$ paramètres après $k = 3$ observations.

En pratique, la forme exacte de l'ensemble convexe $\mathbf{S}_k$ est difficile à décrire. Le nombre de sommets de cet ensemble $\mathbf{S}_k$ augmente en fonction du nombre de paramètres p et du nombre d'observations k. Des techniques de réduction du nombre de sommets peuvent être mises en œuvre ([MIL 82], [BRO 90], [VIC 93],…) mais la technique demeure extrêmement coûteuse en termes de coût calcul et mémoire, et son utilisation se restreint en pratique à des problèmes de petite dimension ($p \leq 5$ environ).

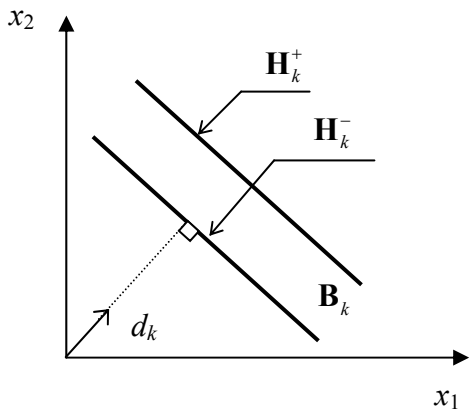


Figure 1.1 : Bande $\mathbf{B}_k$ contenant $x \in \mathbb{R}^2$ compatible avec l'observation et la borne sur le bruit à l'instant k.

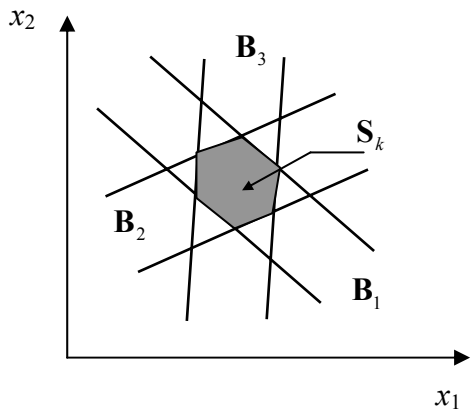


Figure 1.2 : L'ensemble $\mathbf{S}_k$ après 3 observations.

Pour pallier ce problème calculatoire, l'approche ensembliste **englobante** cherche un ensemble de forme géométrique plus simple qui englobe $\mathbf{S}_k$, et ceci sans trop « sur-englober » l'espace $\mathbf{S}_k$ recherché. Les formes géométriques utilisées couramment sont (cf. figure 1.3) :

- les **polytopes** [MO 90] : la forme de l'espace englobant et le nombre de sommets varient à chaque instant. Pour une implantation temps réel, l'utilisation des polytopes est coûteuse en temps calcul ;
- les **parallélotopes** [VIC 93] : le parallélotope est défini comme l'intersection de p bandes dans un espace de paramètres de dimension p. Chaque bande est délimitée par deux hyperplans parallèles. La méthode décrite dans [VIC 93] construit à chaque instant un parallélotope de volume minimum englobant l'intersection du précédant avec la nouvelle bande. Ainsi, la forme « parallélotope » réduit la complexité de l'espace englobant recherché. Cependant, cette méthode reste assez lourde en temps calcul, et n'est donc utilisée que pour des problèmes de petite dimension (la limite $p \leq 5$ semble raisonnable) ;
- les **orthotopes** [MIL 82] : le choix de domaines englobants hyper-rectangulaires tels que les faces du domaine soient parallèles aux axes définissant le domaine paramétrique conduit à un calcul simplifié. Cependant, le pavé trouvé est généralement pessimiste (ou encore « assez lâche », ou surenglobant) ;
- les **ellipsoïdes** [SCH 68] : l'ensemble englobant est donné sous la forme d'un ellipsoïde caractérisé par son centre et une matrice symétrique définie positive. Cette paramétrisation du domaine englobant est très simple.

Les espaces englobants décrits ci-dessus sont calculés récursivement ce qui, pour une implantation temps réel, est un avantage certain. Néanmoins, si quelques précautions qu'on

présentera ultérieurement ne sont pas prises, le domaine obtenu par un calcul récursif peut être extrêmement surenglobant (et donc pessimiste). D'un autre côté, un calcul non récursif est souvent non envisageable d'un point de vue coût calcul.

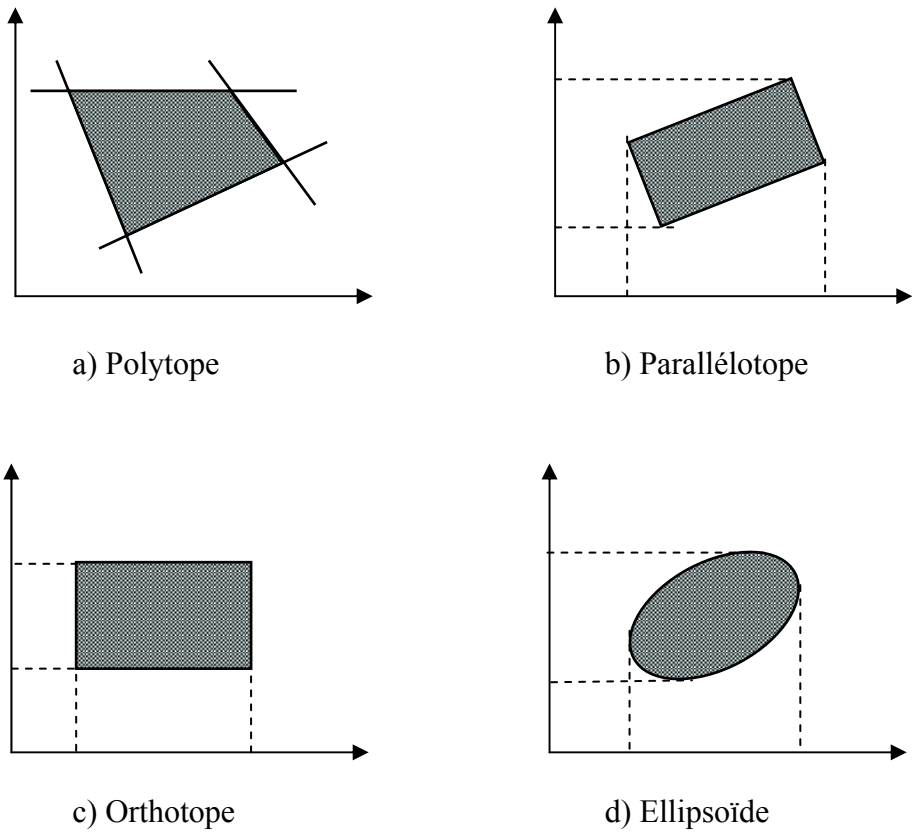


Figure 1.3 : Ensembles englobant dans l'approche ensembliste englobante.

Dans ce mémoire, nous nous intéressons aux algorithmes englobants mettant en œuvre un ellipsoïde. Ces algorithmes sont classiquement réunis dans la famille des algorithmes **OBE** (Outer Bounding Ellipsoid).

1.2 Les algorithmes OBE

Les algorithmes OBE cherchent à chaque instant k, un ellipsoïde $\mathbf{E}_k$ qui englobe l'ensemble $\mathbf{S}_k$. Dans l'espace des p paramètres, l'ellipsoïde $\mathbf{E}_k$ est représenté sous la forme générale suivante :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \left(x - c_k\right)^T P_k^{-1} \left(x - c_k\right) \leq \sigma_k^2 \right\} \tag{1.5}$$

où c_k est le centre de l'ellipsoïde ;

$P_k \in \mathbb{R}^{pxp}$ est une matrice symétrique, définie positive ;

σ_k^2 est un coefficient positif.

Notons que les vecteurs propres de la matrice $\left(\sigma_k^2 P_k\right)$ définissent les axes de l'ellipsoïde, les valeurs propres associées étant liées à la longueur des axes de l'ellipsoïde.

Dans les algorithmes OBE, l'ellipsoïde $\mathbf{E}_k$ est calculé séquentiellement, c'est-à-dire que $\mathbf{E}_k$ va englober l'intersection de la bande $\mathbf{B}_k$ définie par une nouvelle observation (équation 1.3) et de l'ellipsoïde $\mathbf{E}_{k\text{-}1}$ obtenu à l'instant k - 1 (cf. figure 1.4) :

$$\mathbf{E}_k \supset \mathbf{E}_{k-1} \cap \mathbf{B}_k \tag{1.6}$$

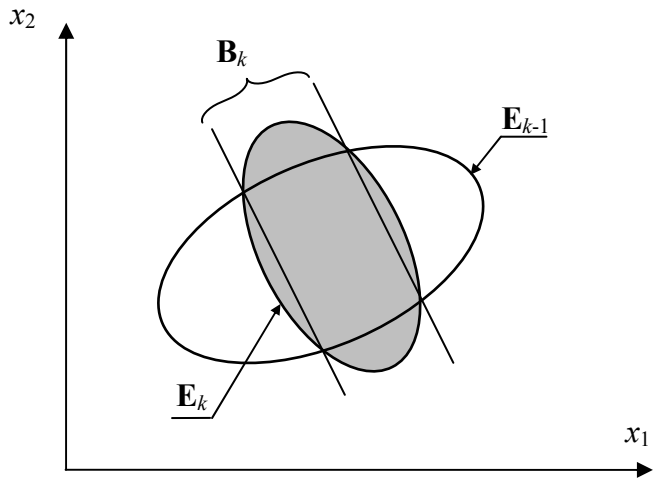


Figure 1.4: $\mathbf{E}_k \supset \mathbf{E}_{k-1} \cap \mathbf{B}_k$ - Calcul séquentiel de l'ellipsoïde dans un espace de $p = 2$ paramètres

Pour satisfaire (1.6), l'ellipsoïde $\mathbf{E}_k$ peut être décrit par :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \boldsymbol{\alpha}_k \left(x - c_{k-1}\right)^T \boldsymbol{P}_{k-1}^{-1} \left(x - c_k\right) + \boldsymbol{\beta}_k \left(\boldsymbol{y}_k - \boldsymbol{d}_k^T x\right)^2 \le \boldsymbol{\alpha}_k \sigma_k^2 + \boldsymbol{\beta}_k \gamma_k^2 \right\} \tag{1.7}$$

où $\boldsymbol{\alpha}_k > 0$, $\boldsymbol{\beta}_k \ge 0$ sont les coefficients permettant de pondérer l'ancienne information avec la nouvelle observation. En développant (1.7) et en identifiant avec (1.5), on obtient la forme classique de l'algorithme OBE :

$$\begin{cases} \boldsymbol{P}_k^{-1} = \boldsymbol{\alpha}_k \boldsymbol{P}_{k-1}^{-1} + \boldsymbol{\beta}_k \boldsymbol{d}_k \boldsymbol{d}_k^T \\ c_k = \boldsymbol{P}_k \left(\boldsymbol{\alpha}_k \boldsymbol{P}_{k-1}^{-1} c_{k-1} + \boldsymbol{\beta}_k \boldsymbol{d}_k \boldsymbol{y}_k\right) \\ \sigma_k^2 = \boldsymbol{\alpha}_k \sigma_{k-1}^2 + \boldsymbol{\beta}_k \gamma_k^2 + c_k^T \boldsymbol{P}_k^{-1} c_k - \boldsymbol{\alpha}_k c_{k-1}^T \boldsymbol{P}_{k-1}^{-1} c_{k-1} - \boldsymbol{\beta}_k \boldsymbol{y}_k^2 \end{cases} \tag{1.8}$$

A partir du lemme d'inversion matricielle, et après quelques calculs, on obtient la forme générale de l'algorithme OBE :

$$\begin{cases} \boldsymbol{P}_k = \dfrac{1}{\boldsymbol{\alpha}_k}\left[\boldsymbol{P}_{k-1} - \dfrac{\boldsymbol{\beta}_k \boldsymbol{P}_{k-1}\boldsymbol{d}_k\boldsymbol{d}_k^T\boldsymbol{P}_{k-1}}{\boldsymbol{\alpha}_k + \boldsymbol{\beta}_k g_k}\right] \\ c_k = c_{k-1} + \boldsymbol{\beta}_k \boldsymbol{P}_k \boldsymbol{d}_k \nu_k \\ \sigma_k^2 = \boldsymbol{\alpha}_k \sigma_{k-1}^2 + \boldsymbol{\beta}_k \gamma_k^2 - \dfrac{\boldsymbol{\alpha}_k \boldsymbol{\beta}_k \nu_k^2}{\boldsymbol{\alpha}_k + \boldsymbol{\beta}_k \boldsymbol{g}_k} \\ \nu_k = \boldsymbol{y}_k - \boldsymbol{d}_k^T c_{k-1} \\ g_k = \boldsymbol{d}_k^T \boldsymbol{P}_{k-1} \boldsymbol{d}_k \end{cases} \tag{1.9}$$

Le nouvel ellipsoïde $\mathbf{E}_k$ contient deux paramètres libres $\boldsymbol{\alpha}_k$ et $\boldsymbol{\beta}_k$ qui définissent en fait sa position, sa taille et son orientation. Compte tenu du choix de ces deux paramètres et de la fonction critère utilisée pour les déterminer, plusieurs algorithmes OBE différents peuvent être écrits. On peut les regrouper dans les deux catégories suivantes [FAV 96] :

- **les méthodes basées sur la taille géométrique de $\mathbf{E}_k$:** dans ce groupe, $\boldsymbol{\alpha}_k$ et $\boldsymbol{\beta}_k$ sont calculés à chaque instant k pour *minimiser* la taille géométrique de $\mathbf{E}_k$. Deux critères sont utilisés :
  - *le critère du déterminant* det *:*

    $$\mu_V(k) = \det\left(\sigma_k^2 \boldsymbol{P}_k\right) \tag{1.10}$$

    Cette fonction critère est en fait le produit des carrés des longueurs des demi-axes de l'ellipsoïde ; elle est donc proportionnelle au volume au carré de l'ellipsoïde ;
  - *le critère de la trace* tr *:*

    $$\mu_T(k) = \mathrm{tr}\left(\sigma_k^2 \boldsymbol{P}_k\right) \tag{1.11}$$

    Dans ce cas, la fonction du critère est la somme des carrés des longueurs des demi-axes de l'ellipsoïde ;
- **les méthodes basées sur « la convergence » de l'argument σ_k^2 :** le choix de $\boldsymbol{\alpha}_k$ et $\boldsymbol{\beta}_k$ assure la convergence de σ_k^2, mais pas de la taille de $\mathbf{E}_k$.

On présente maintenant les différents algorithmes OBE, notamment les formules de $\boldsymbol{\alpha}_k$ et $\boldsymbol{\beta}_k$ pour construire l'ellipsoïde englobant. Les propriétés de convergence des algorithmes OBE seront abordées en détail dans le chapitre 3. Les valeurs de $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$, $\boldsymbol{y}_k$, $\boldsymbol{d}_k$, $\boldsymbol{P}_k$ apparaissant dans chaque algorithme peuvent être différentes, et ne pas correspondre à celles de l'algorithme général (équation 1.9). Cependant, pour éviter d'alourdir les notations, nous avons choisi d'écrire $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$, $\boldsymbol{y}_k$, $\boldsymbol{d}_k$, $\boldsymbol{P}_k$ **en gras** (voir les équations (1.7) à (1.9)) afin d'attirer

l'attention du lecteur sur cette petite difficulté. Dans chaque algorithme, nous indiquerons les relations qui les définissent en fonction des paramètres de l'algorithme général 1.9.

1.2.1 Méthodes basées sur la taille géométrique de l'ellipsoïde

1.2.1.1 Algorithme de Fogel et Huang

Fogel et Huang [FOG 82] ont donné un algorithme de base pour l'estimation des paramètres d'un modèle linéaire dans l'approche ensembliste ellipsoïdale (englobante). Dans leur algorithme, ils utilisent la forme normalisée de l'ellipsoïde, *i.e.* :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \left(x - c_k\right)^T P_k^{-1} \left(x - c_k\right) \leq 1 \right\} \tag{1.12}$$

et la formulation de la bande $\mathbf{B}_k$ donnée dans l'équation (1.3). L'ellipsoïde englobant à l'instant k est calculé séquentiellement par :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \left(x - c_{k-1}\right)^T P_{k-1}^{-1} \left(x - c_k\right) + \frac{q_k}{\gamma_k^2} \left(y_k - d_k^T x\right)^2 \leq 1 + q_k \right\} \tag{1.13}$$

où $q_k \geq 0$ est un coefficient de pondération entre l'ancienne information avec la nouvelle observation. En comparant (1.13) avec (1.7), on en déduit les relations entre les paramètres de ce nouvel algorithme et ceux de l'algorithme OBE général :

$$\boldsymbol{\alpha}_k = \frac{1}{\sigma_{k-1}^2}; \boldsymbol{\beta}_k = \frac{q_k}{\gamma_k^2}; \boldsymbol{P}_k = \frac{P_k}{\sigma_k^2}; \boldsymbol{y}_k \equiv y_k; \boldsymbol{d}_k \equiv d_k \tag{1.14}$$

Le coefficient q_k est calculé en utilisant les critères du déterminant $\mu_V(k)$ (équation 1.10) ou de la trace $\mu_T(k)$ (équation 1.11).

a) Critère de déterminant

Remplaçons dans (1.10) les formules (1.9). En utilisant la relation (1.14) et le lemme d'inversion matricielle, on peut exprimer le critère $\mu_V(k)$ en fonction de $\mu_V(k-1)$:

$$\mu_V(k) = R_V(q_k)\mu_V(k-1)$$

avec $$R_V(q_k) = \frac{\left(1 + q_k - \dfrac{q_k \gamma_k^{-2} \nu_k^2}{1 + q_k \gamma_k^{-2} g_k}\right)^p}{1 + q_k \gamma_k^{-2} g_k}$$

Alors, $\mu_V(k)$ sera minimale pour les valeurs de q_k qui minimisent $R_V(q_k)$. En écrivant $\dfrac{dR_V(q_k)}{dq_k} = 0$, on obtient facilement la condition suivante sur q_k [CLE 87]:

$$a_2 q_k^2 + a_1 q_k + a_0 = 0 \tag{1.15}$$

où
$$\begin{cases} a_2 = (p-1) g_k^2 \\ a_1 = g_k \left[(2p-1)\gamma_k^2 - g_k + \nu_k^2 \right] \\ a_0 = \gamma_k^2 \left[p\left(\gamma_k^2 - \nu_k^2\right) - g_k \right] \end{cases}$$

avec $g_k = d_k^T P_{k-1} d_k ; \nu_k = y_k - d_k^T c_{k-1}$

q_k doit être réel positif pour garantir $\boldsymbol{\beta}_k$ positif, ce qui conduit à :

si $\Delta = a_1^2 - 4a_2 a_0 \geq 0$ et $q_k^* = \dfrac{-a_1 + \sqrt{\Delta}}{2a_2} > 0$, alors $q_k = q_k^*$

sinon $q_k = 0$.

b) Critère de la trace

En remplaçant les formules (1.9), (1.14) dans la fonction critère $\mu_T(k)$ définie dans (1.11) et en utilisant le fait que $\mathrm{tr}(M + uv^T) = \mathrm{tr}M + v^T u$, on peut écrire $\mu_T(k)$ sous la forme [FOG 82] :

$$\mu_T(k) = \left(1 + q_k - \frac{q_k \gamma_k^{-2} \nu_k^2}{1 + q_k \gamma_k^{-2} g_k} \right) \left(\mu_T(k-1) - \frac{q_k h}{1 + q_k \gamma_k^{-2} g_k} \right)$$

avec $\nu_k = y_k - d_k^T c_{k-1}$; $g_k = d_k^T P_{k-1} d_k$; $\tilde{g}_k = d_k^T P_{k-1}^2 d_k$

La valeur optimale de q_k est la solution de :

$$\frac{d\mu_T(k)}{dq_k} = 0 \tag{1.16}$$

Ici encore, q_k doit respecter une condition de positivité ($q_k \geq 0$) et satisfaire (1.16). C'est la racine positive d'un polynôme du 3ème degré :

$$q_k^3 + b_2 q_k^2 + b_1 q_k + b_0 = 0 \tag{1.17}$$

où
$$\begin{cases} b_2 = 3\gamma_k^2 / g_k \\ b_1 = \dfrac{\gamma_k^2 g_k \left[(\gamma_k^2 - \nu_k^2)\mu_T(k-1) - h_k \right] + 2\gamma_k^2 \left[\gamma_k^2 g_k \mu_T(k-1) - (\gamma_k^2 - \nu_k^2) h_k \right]}{g_k^2 \left[g_k \mu_T(k-1) - h_k \right]} \\ b_0 = \dfrac{\gamma_k^4 \left[(\gamma_k^2 - \nu_k^2)\mu_T(k-1) - h_k \right]}{g_k^2 \left[g_k \mu_T(k-1) - h_k \right]} \end{cases}$$

La solution est alors obtenue par :

$$q_k = \begin{cases} 0 \text{ si } b_0 \geq 0 \\ q_k^* \text{ sinon} \end{cases} \tag{1.18}$$

où q_k^* est la racine réelle positive de l'équation (1.16).

c) Remarque

Pour ces deux critères, le cas $q_k = 0$ signifie que l'ellipsoïde n'est pas mis à jour, c'est-à-dire $\mathbf{E}_k = \mathbf{E}_{k\text{-}1}$. Autrement dit, l'observation à l'instant k n'apporte pas de nouvelle information.

1.2.1.2 Famille paramétrée d'ellipsoïdes

C. Durieu *et al.* (par exemple, [DUR 96], [SED 98]) s'intéressent à une famille paramétrée d'ellipsoïdes. Ils construisent un algorithme plus général avec lequel on peut traiter l'intersection de plusieurs ellipsoïdes ou bandes. Dans un souci d'homogénéité avec ce qui précède, on considère ici le cas particulier de l'intersection d'une bande **B** avec un ellipsoïde **E**. Ces deux ensembles sont définis par des formules normalisées. L'ellipsoïde est paramétré par son centre c et la matrice P (ou $M = P^{-1} > 0$) définie positive :

$$\mathbf{E}(c,P) = \left\{ x \in \mathbb{R}^p : (x-c)^T P^{-1} (x-c) \leq 1 \right\} \tag{1.19}$$

ou encore $$\mathbf{E}(c,M^{-1}) = \left\{ x \in \mathbb{R}^p : (x-c)^T M (x-c) \leq 1 \right\} \tag{1.20}$$

La bande d'hyperplan **B** est définie par :

$$\mathbf{B} = \left\{ x \in \mathbb{R}^p : \left(y - d^T x \right)^2 \leq 1 \right\} \tag{1.21}$$

Lorsqu'on applique l'algorithme séquentiellement, à l'instant k (ou pour la $k^{\text{ième}}$ mesure), le nouvel ellipsoïde englobant l'intersection $\mathbf{E}_{k\text{-}1} \cap \mathbf{B}_k$ est calculé par :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \alpha_k (x - c_{k-1})^T M_{k-1} (x - c_k) + (1-\alpha_k)\left(y_k - d_k^T x \right)^2 \leq 1 \right\} \tag{1.22}$$

avec $0 < \alpha_k \leq 1$

En comparant (1.22) avec (1.7), on trouve les relations suivantes :

$$\boldsymbol{\alpha}_k = \frac{\alpha_k}{\sigma_{k-1}^2} ; \boldsymbol{\beta}_k = \frac{1-\alpha_k}{\gamma_k^2} ; \boldsymbol{P}_k = \frac{P_k}{\sigma_k^2} = \left(\sigma_k^2 M_k \right)^{-1} ; \boldsymbol{y}_k = \gamma_k y_k ; \boldsymbol{d}_k = \gamma_k d_k \tag{1.23}$$

On peut ensuite réécrire (1.22) sous la forme :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : (x - c_k)^T L_\alpha (x - c_k) \leq 1 - \delta_\alpha \right\} \tag{1.24}$$

où
$$\begin{cases} L_\alpha = \alpha_k M_{k-1} + (1-\alpha_k) d_k d_k^T \\ c_k = L_\alpha^{-1}\left[\alpha_k M_{k-1} c_{k-1} + (1-\alpha_k) d_k y_k\right] \\ \delta_\alpha = \alpha_k c_{k-1}^T M_{k-1} c_{k-1} + (1-\alpha_k) y_k^2 - c_k^T L_\alpha c_k \end{cases} \tag{1.25}$$

D'après Durieu *et al.* [DUR 01a], deux familles d'ellipsoïdes paramétrés en α_k englobent cette intersection. Elles font apparaître respectivement les matrices P_{k1} et P_{k2} :

$$\mathbf{E}_{k1}(c_k, P_{k1}) \quad \text{avec} \quad P_{k1} = L_\alpha^{-1}$$

$$\mathbf{E}_{k2}(c_k, P_{k2}) \quad \text{avec} \quad P_{k2} = (1-\delta_\alpha) L_\alpha^{-1}$$

[DUR 01a] précise que la deuxième famille conduit à un ellipsoïde plus petit que celui obtenu avec la première famille, au prix de calculs plus compliqués. Notons que certaines propriétés de l'ellipsoïde ne sont établies et démontrées que pour la 1ère famille d'ellipsoïdes paramétrés. Par exemple, dans le calcul de l'intersection simultanée de plusieurs bandes (ou ellipsoïdes), [DUR 01b] montre que les problèmes d'optimisation selon le critère (1.10) et (1.11) sont convexes et que leurs fonctions critère sont deux fois différentiables uniquement pour les ellipsoïdes de la 1ère famille. L'algorithme OBE général correspond quant à lui à celui défini avec la 2ème famille d'ellipsoïdes paramétrés.

Pour les deux familles d'ellipsoïdes, le coefficient α_k est calculé via les deux critères précédemment cités, c'est-à-dire le critère du déterminant ou le critère de la trace. Le lecteur pourra consulter [DUR 98] dans lequel les détails pour les deux familles sont donnés. Notons que l'ellipsoïde obtenu dans la 2ème famille est identique à celui de l'algorithme FH à critère identique (déterminant / trace).

1.2.1.3 Algorithme SMSA

L'algorithme SMSA (Set Membership Set Approximation) proposé par Nayeri *et al.* [NAY 93] présente des résultats (et des tentatives d'amélioration) sur la convergence des algorithmes OBE. L'idée est « d'encourager la convergence » en évitant d'obtenir une matrice $M_k = P_k^{-1}$ tendant vers l'infini, ce qui empêcherait l'analyse théorique de la convergence des méthodes basées sur la taille géométrique de l'ellipsoïde. L'algorithme SMSA est équivalent à celui d'approximation stochastique [LJU 83]. Les formes générales (1.9) de l'ellipsoïde et de la bande sont utilisées. Les coefficients $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$ sont choisis par :

$$\boldsymbol{\alpha}_k = 1 - \lambda_k \; ; \quad \boldsymbol{\beta}_k = \lambda_k \; ; \quad 0 \leq \lambda_k < 1 \tag{1.26}$$

Ici encore, λ_k est calculé à partir des deux critères de la taille géométrique de l'ellipsoïde. Cependant, les propriétés de convergence ne concernent que le cas de la minimisation du volume ([NAY 93], [NAY 94]). Celles-ci sont abordées au chapitre 3.

a) Critère du déterminant

Avec le choix de α_k, β_k dans (1.26), le rapport des volumes de deux ellipsoïdes successifs sera [NAY 93] :

$$R_V(\lambda_k) \triangleq \frac{\mu_V(k)}{\mu_V(k-1)} = \frac{(1-\lambda_k)\left(\frac{1}{1-\lambda_k}\frac{\sigma_k^2}{\sigma_{k-1}^2}\right)^p}{1-\lambda_k+\lambda_k g_k} \tag{1.27}$$

Pour $R_V(\lambda_k^*)$ minimum, λ_k^* est la solution de l'équation du 2ème degré :

$$a_2\lambda_k^2 + a_1\lambda_k + a_0 = 0 \tag{1.28}$$

avec
$$\begin{cases} a_2 = p\left(\gamma_k^2 - \nu_k^2\right) + \left[(1-2p)\gamma_k^2 - \nu_k^2 - \sigma_{k-1}^2\right]g_k + \left[(p-1)\gamma_k^2 + \sigma_{k-1}^2\right]g_k^2 \\ a_1 = 2p\left(\nu_k^2 - \gamma_k^2\right) + \left[(2p-1)\gamma_k^2 + \nu_k^2 + 2\sigma_{k-1}^2\right]g_k - \sigma_{k-1}^2 g_k^2 \\ a_0 = p(\gamma_k^2 - \nu_k^2) - \sigma_{k-1}^2 g_k \end{cases}$$

où $\quad g_k = d_k^T P_{k-1} d_k; \quad \nu_k = y_k - d_k^T c_{k-1}$

[NAY 93] montre que $\lambda_k^* \in [0,1]$ si et seulement si $a_0 < 0$. Dans ce cas, $R_V(\lambda_k^*) < 1$ et λ_k^* est calculé par :

$$\lambda_k^* = \frac{-a_1 + \sqrt{a_1^2 - 4a_2a_0}}{2a_2}$$

A l'inverse, si $a_0 \geq 0$, on met $\lambda_k^* = 0$ et l'ellipsoïde n'est pas mis à jour.

Avec le même critère du déterminant, les coefficients optimaux q_k^*, λ_k^* pour les deux l'algorithme FH et SMSA sont calculés respectivement en résolvant les équations (1.15) et (1.28) ; ils sont en outre différents. Cependant, en les remplaçant dans la formule générale (1.9), on obtient deux ellipsoïdes identiques.

b) Critère de la trace

A partir du choix de α_k, β_k dans (1.26), on peut évidement établir la formule pour calculer λ_k^* qui minimise $\mu_T(k)$. Néanmoins ce critère de la trace ne présente pas de

propriétés de convergences pour l'algorithme SMSA, donc les auteurs ne présentent pas les formules pour calculer λ_k^* dans leurs publications. Par contre, grâce aux propriétés proposées dans [DEL 94] montrant que le rapport $\beta(\lambda_k^*)/\alpha(\lambda_k^*)$ est constant pour des choix différents de $\alpha(\lambda)$ et $\beta(\lambda)$, on peut également obtenir λ_k^* pour l'algorithme SMSA en utilisant la formule (1.18) de l'algorithme FH après quelques calculs.

1.2.2 Méthodes basées sur la convergence de σ_k^2

1.2.2.1 Algorithme de Dasgupta et Huang

Dasgupta et Huang sont les pionniers dans la création d'un nouveau groupe d'algorithmes OBE qui s'intéresse à la convergence de la séquence des σ_k^2. Quelques propriétés de convergence intéressantes sur les paramètres de l'algorithme OBE ont alors été démontrées.

[DAS 87] utilise la forme générale de l'ellipsoïde (1.5) et de la bande d'hyperplan (1.3). L'erreur e_k est supposée bornée par γ pour tous k, *i.e.* $e_k^2 \le \gamma^2, \forall k$. Les coefficients $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$ sont choisis égaux à :

$$\boldsymbol{\alpha}_k = (1-\lambda_k)\ ; \quad \boldsymbol{\beta}_k = \lambda_k\ ; \quad 0 \le \lambda_k < 1 \tag{1.29}$$

La formule générale (1.7) de l'ellipsoïde englobant devient alors :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : (1-\lambda_k)(x-c_{k-1})^T P_{k-1}^{-1}(x-c_{k-1}) + \lambda_k \left(y_k - d_k^T x\right)^2 \le (1-\lambda_k)\sigma_{k-1}^2 + \lambda_k \gamma^2 \right\} \tag{1.30}$$

où le coefficient λ_k dans cet algorithme est calculé pour satisfaire le critère suivant :

$$\sigma_k^2(\lambda_k^*) \le \sigma_k^2(\lambda_k),\ \forall \lambda_k : 0 \le \lambda_k \le \upsilon < 1 \tag{1.31}$$

avec υ un coefficient choisi par l'utilisateur tel que $0 < \upsilon < 1$. Notons que $\upsilon = 1$ signifie que P_k devient singulière. A partir de (1.31), [DAS 87] donne la formule pour λ_k :

$$\lambda_k = \begin{cases} 0 & \text{si } \rho_k \ge 1 \\ \lambda_k^* & \text{sinon} \end{cases} \tag{1.32}$$

avec $\rho_k = \dfrac{\gamma^2 - \sigma_{k-1}^2}{\nu_k^2}$, $\lambda_k^* = \min(\upsilon, \zeta_k)$, $\nu_k = \boldsymbol{y}_k - \boldsymbol{d}_k^T c_{k-1}$

où
$$\zeta_k = \begin{cases} \upsilon & \text{si } \nu_k = 0 \quad \text{(a)} \\ \dfrac{1-\rho_k}{2} & \text{si } g_k = 1 \quad \text{(b)} \\ \dfrac{1}{1-g_k}\left(1-\sqrt{\dfrac{g_k}{\rho_k(g_k-1)+1}}\right) & \text{si } \rho_k(g_k-1)+1>0 \quad \text{(c)} \\ \upsilon & \text{si } \rho_k(g_k-1)+1 \le 0 \quad \text{(d)} \end{cases} \tag{1.33}$$

La démonstration peut être consultée dans [DAS 87].

Remarque

La solution optimale du critère (1.31) satisfait :

$$\left.\frac{d\sigma_k^2}{d\lambda_k}\right|_{\lambda_k^*} = 0 \text{ et } \left.\frac{d^2\sigma_k^2}{d\lambda_k^2}\right|_{\lambda_k^*} > 0 \text{, sinon } \lambda_k^* = 0$$

Dans les cas (a) et (d) de l'équation (1.33) $\left.\dfrac{d\sigma_k^2}{d\lambda_k}\right|_{\lambda_k^*} < 0,\ \forall \lambda_k \in [0,1]$, ce qui signifie que le critère est toujours rempli et que l'optimum est atteint sur le borne de droite de l'intervalle. On prend alors pour υ la valeur choisie par l'utilisateur.

Cet algorithme doit vérifier des conditions (dans 1.33a, b, d) concernant l'erreur de prédiction ν_k avant de calculer λ_k^* et la matrice P_k. Les données « inutiles » (dans le sens où elles ne permettent pas de mettre à jour l'ellipsoïde) sont rapidement détectées, ce qui permet de réduire le temps de calcul. La convergence des paramètres dans cet algorithme est directement liée à la condition d'excitation (voir chapitre 4, paragraphe 4.2).

Enfin, l'algorithme dépend fortement du choix du coefficient υ.

1.2.2.2 Algorithme de Tan *et al.*

Dans [TAN 97], les coefficients α_k, β_k sont choisis tels que :

$$\alpha_k = 1\ ;\quad \beta_k = \lambda_k\ ;\quad \lambda_k \ge 0 \tag{1.34}$$

L'ellipsoïde englobant l'intersection de l'ellipsoïde $\mathbf{E}_{k-1}$ et de la bande $\mathbf{B}_k$ est donné par :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : (x-c_{k-1})^T P_{k-1}^{-1}(x-c_{k-1}) + \lambda_k\left(y_k - d_k^T x\right)^2 \le \sigma_k^2 + \lambda_k \gamma^2 \right\} \tag{1.35}$$

Remplaçons α_k, β_k de (1.34) dans (1.9). Il vient :

$$\sigma_k^2 = \sigma_{k-1}^2 + \lambda_k \gamma^2 - \frac{\lambda_k \nu_k^2}{1+\lambda_k g_k} \tag{1.36}$$

où $\quad g_k = d_k^T P_{k-1} d_k\ ;\ \ \nu_k = y_k - d_k^T c_{k-1}$

Le coefficient λ_k^* est calculé par :

$$\lambda_k^* = \arg\min_{\lambda_k \geq 0} \sigma_k^2 \tag{1.37}$$

λ_k^* est donc la solution de :

$$\frac{\partial \sigma_k^2}{\partial \lambda_k} = 0 \quad \Leftrightarrow \quad \gamma_k^2 - \frac{\nu_k^2}{\left(1+\lambda_k g_k\right)^2} = 0$$

ce qui donne

$$\lambda_k = \begin{cases} 0 & \text{si } |\nu_k| \leq \gamma \text{ ou } g_k = 0 \\ \lambda_k^* & \text{sinon} \end{cases} \quad \text{avec} \quad \lambda_k^* = \frac{1}{\gamma g_k}\left(|\nu_k| - \gamma\right) \tag{1.38}$$

On a de plus $\left.\frac{\partial^2 \sigma_k^2}{\partial \lambda_k^2}\right|_{\lambda_k = \lambda_k^*} > 0$, c'est-à-dire que λ_k^* minimise bien le critère (1.37).

Cet algorithme ne permet de mettre à jour l'ellipsoïde que lorsque l'erreur de prédiction ν_k est plus petite que la borne du bruit γ_k. En fait, cette condition est souvent remplie lorsque le centre de l'ancien ellipsoïde $\mathbf{E}_{k\text{-}1}$ est proche des « vrais » paramètres, sans condition sur la taille de l'ellipsoïde. En conséquence, la procédure d'identification s'arrête « assez tôt » alors que la taille de l'ellipsoïde n'a pas suffisamment diminué, compte tenu de l'information disponible dans les données utilisées.

1.2.2.3 Algorithme de Lozano-Leal et Ortega

Lozano-Leal et Ortega [LOZ 87] utilisent les formes générales de l'ellipsoïde (1.5) et de la bande (1.3). Les coefficient $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$ sont donnés par :

$$\boldsymbol{\alpha}_k = 1 \; ; \quad \boldsymbol{\beta}_k = \lambda_k \; ; \quad 0 \leq \lambda_k < \upsilon \leq 1 \tag{1.39}$$

où υ est un coefficient choisi par l'utilisateur. λ_k est calculé à chaque instant de manière à ce que la séquence des σ_k^2 soit décroissante, *i.e.* $\sigma_k^2 \leq \sigma_{k-1}^2$. [LOZ 87] donne alors la formule pour calculer λ_k :

$$\lambda_k = \begin{cases} 0 & \text{si } |\nu_k| \leq \gamma' \\ \lambda_k^* & \text{sinon} \end{cases}$$

où $$\lambda_k^* = \frac{\upsilon}{1+g_k}\left(1 - \frac{\gamma'}{|\nu_k|}\right) \quad et \quad \gamma' = \gamma\sqrt{1+\upsilon}$$

avec $$g_k = d_k^T P_{k-1} d_k \; ; \quad \nu_k = y_k - d_k^T c_{k-1} \, .$$

La condition de mise à jour de l'ellipsoïde est de ce fait modifiée. Elle dépend du choix du coefficient υ. La démonstration de la convergence de cet algorithme est menée à partir de

considérations sur σ_k^2 comme borne sur une fonction quadratique. La condition d'excitation est bien entendue nécessaire pour garantir la convergence de l'algorithme.

1.2.3 Synthèse sur les algorithmes OBE

Bien qu'il y ait des différences dans le choix des coefficients de pondération, tous les algorithmes OBE ont en définitive une formulation identique pour la mise à jour de l'ellipsoïde. L'objectif visé dans l'évolution de la séquence d'ellipsoïdes induit le choix spécifique des coefficients (α_k, β_k) optimaux. On a donc classé les algorithmes OBE en deux familles en se basant sur cet objectif, à savoir les algorithmes qui tentent de diminuer la taille de l'ellipsoïde et ceux qui s'intéressent à la convergence du paramètre σ_k^2.

Les algorithmes qui visent à réduire la taille géométrique de l'ellipsoïde utilisent souvent la forme normalisée de l'ellipsoïde et de la bande, ceci dans un souci de simplification de l'écriture. Le coefficient σ_k^2 n'apparaît pas dans cette famille. La taille de l'ellipsoïde est mesurée par deux critères différents : le critère du déterminant $\mu_V(k)$ ou le critère de la trace $\mu_T(k)$. Avec le critère du déterminant, le coefficient de pondération est la solution réelle positive d'une équation du 2ème degré. Pour le critère de la trace, ce coefficient est la solution réelle positive d'une équation du 3ème degré. Pour un même critère utilisé, les ellipsoïdes trouvés avec les algorithmes de cette première famille sont identiques bien que les formules de calcul diffèrent.

Les algorithmes basés sur la convergence de σ_k^2 utilisent toujours la forme générale de l'ellipsoïde (et non la forme normalisée) car ils utilisent explicitement σ_k^2. Ces algorithmes demandent une vérification de certaines conditions basées sur l'erreur de prédiction avant toute mise à jour de l'ellipsoïde. De plus, ils dépendent également d'un facteur υ choisi par l'utilisateur. La démonstration de convergence de ces algorithmes est réalisée en considérant σ_k^2 comme la borne d'une fonction quadratique ([DAS 87], [LOZ 87], [TAN 97],…), sous l'hypothèse bien entendu d'une excitation persistante.

La synthèse des différents algorithmes OBE est présentée dans le tableau 1.1.

Algorithme	α_k	β_k	P_k	y_k	d_k	Contrainte	Critère
FH [FOG 82]	$\frac{1}{\sigma_{k-1}^2}$	$\frac{q_k}{\gamma_k^2}$	$\frac{P_k}{\sigma_k^2}$	y_k	d_k	$q_k \geq 0$	$\mu_V(k)$ $\mu_T(k)$
Famille paramétrée [SED 98]	$\frac{\alpha_k}{\sigma_{k-1}^2}$	$\frac{1-\alpha_k}{\gamma_k^2}$	$\frac{P_k}{\sigma_k^2}$	$\gamma_k y_k$	$\gamma_k d_k$	$0 < \alpha_k \leq 1$	$\mu_V(k)$ $\mu_T(k)$
SMSA [NAY 94]	$1 - \lambda_k$	λ_k	P_k	y_k	d_k	$0 \leq \lambda_k < 1$	$\mu_V(k)$ $\mu_T(k)$
DH [DAS 97]	$1 - \lambda_k$	λ_k	P_k	y_k	d_k	$0 \leq \lambda_k \leq \upsilon < 1$	σ_k^2
Tan *et al.* [TAN 97]	1	λ_k	P_k	y_k	d_k	$\lambda_k \geq 0$	σ_k^2
LO [LOZ 87]	1	λ_k	P_k	y_k	d_k	$0 \leq \lambda_k < \upsilon \leq 1$	σ_k^2

Tableau 1.1 : Synthèse des différents algorithmes OBE.

1.2.4 Test de cohérence et réduction de bande

Avant d'appliquer les algorithmes OBE pour construire le nouvel ellipsoïde, il faut analyser la position relative de la bande $\mathbf{B}_k$ par rapport à l'ellipsoïde $\mathbf{E}_{k-1}$ dans l'espace des paramètres. Cette étape est nécessaire pour éviter des calculs inutiles et dans quelques cas, pour obtenir une solution optimale pour l'ellipsoïde englobant $\mathbf{E}_{k-1} \cap \mathbf{B}_k$.

Quatre situations peuvent apparaître :

- *l'intersection de l'ellipsoïde* $\mathbf{E}_{k-1}$ *avec la bande* $\mathbf{B}_k$ *est vide* (Figure. 1.5.a). Ceci implique une incohérence entre les anciennes informations et la nouvelle observation. Cela peut être causé par une donnée aberrante, un choix mal adapté de la borne de l'erreur, ou même une variation des paramètres à estimer… On doit alors stopper le processus d'identification (l'algorithme), réexaminer les données et la borne sur l'erreur ;
- *la bande* $\mathbf{B}_k$ *contient totalement l'ellipsoïde* $\mathbf{E}_{k-1}$ (Figure 1.5.b). On peut trouver clairement que, dans ce cas, $\mathbf{E}_{k-1} \cap \mathbf{B}_k = \mathbf{E}_{k-1}$. On garde donc l'ellipsoïde précédent sans aucun calcul, $\mathbf{E}_k = \mathbf{E}_{k-1}$. Ceci veut dire que la nouvelle observation n'apporte aucune information utile (pour réduire la taille de l'ellipsoïde) ;

- *un des deux hyperplans définissant la bande* $\mathbf{B}_k$ *n'intersecte pas l'ellipsoïde* $\mathbf{E}_{k\text{-}1}$ (Figure 1.5.c). Ce cas peut conduire à une solution sous optimale lorsque l'on utilise directement les algorithmes OBE. On doit donc remplacer l'hyperplan non sécant par un hyperplan parallèle mais tangent à l'ellipsoïde. Avec cette modification, l'intersection entre la bande et l'ellipsoïde reste inchangée, $\mathbf{E}_{k-1} \cap \mathbf{B}_k = \mathbf{E}_{k-1} \cap \mathbf{B}_k^{'}$, mais l'algorithme OBE donne une solution moins pessimiste pour l'ellipsoïde englobant l'intersection ;
- *les deux hyperplans de la bande* $\mathbf{B}_k$ *coupent l'ellipsoïde* $\mathbf{E}_{k\text{-}1}$ (Figure 1.5.d). Dans ce cas, on applique directement l'algorithme OBE avec le critère choisi.

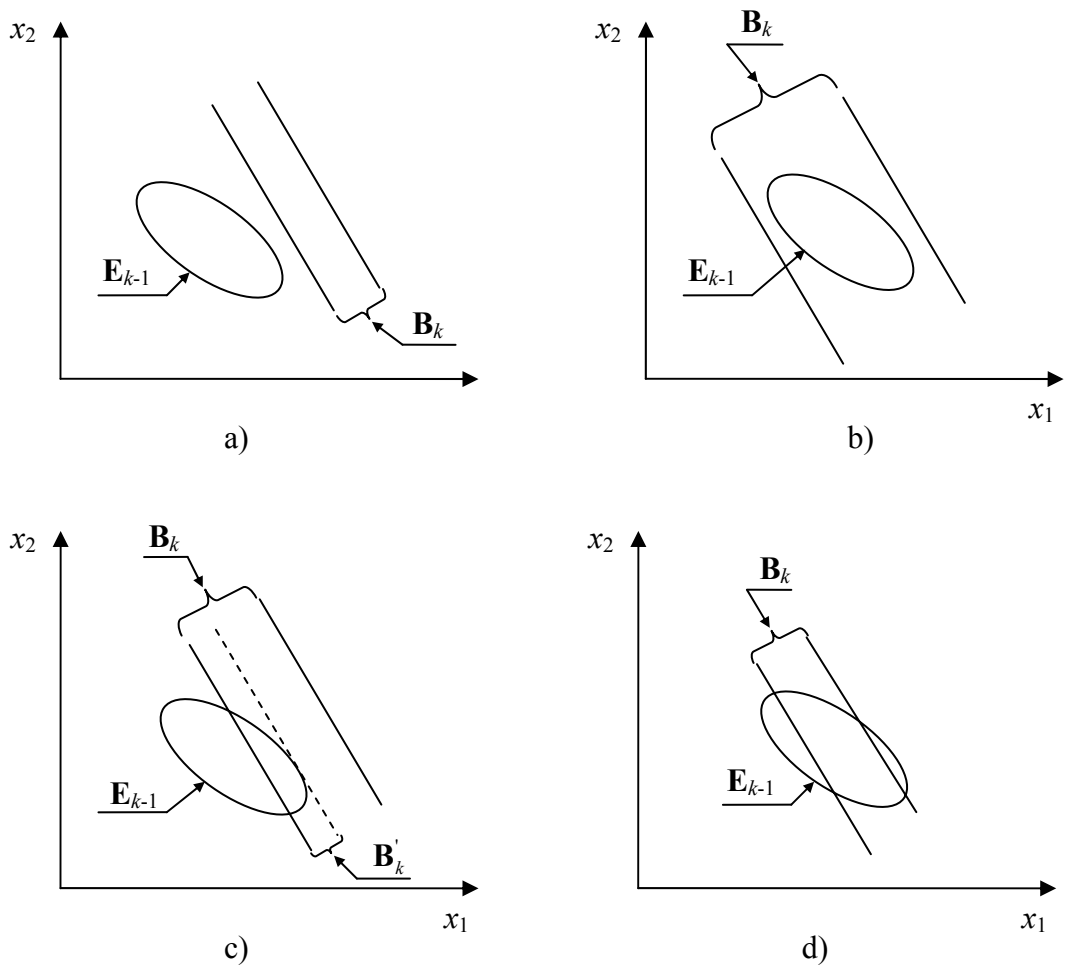


Figure 1.5 : Positions relatives entre la bande et l'ellipsoïde (2 paramètres)

Pour tester la cohérence de l'intersection $\mathbf{E}_{k-1} \cap \mathbf{B}_k$ et réduire si besoin la bande comme indiqué ci-avant, on étudie la position des hyperplans définissant la bande $\mathbf{B}_k$ en relation avec l'ellipsoïde $\mathbf{E}_{k\text{-}1}$. Considérons la bande $\mathbf{B}_k$ dans l'espace des paramètres donnée sous forme normalisée :

$$\mathbf{B}_k = \left\{ x \in \mathbb{R}^p : y_k - 1 \le d_k^T x \le y_k + 1 \right\} \tag{1.40}$$

Cet espace est délimité par deux hyperplans parallèles (figure 1.6) $\mathbf{B}_k = \left[\mathbf{H}_k^-, \mathbf{H}_k^+ \right]$ avec :

$$\begin{aligned} \mathbf{H}_k^- &= \left\{ x \in \mathbb{R}^p : d_k^T x = y_k - 1 \right\} \\ \mathbf{H}_k^+ &= \left\{ x \in \mathbb{R}^p : d_k^T x = y_k + 1 \right\} \end{aligned} \tag{1.41}$$

où le vecteur d_k donne la normale à l'hyperplan $\mathbf{H}_k$. Pour un ellipsoïde normalisé donné par l'équation (1.12) :

$$\mathbf{E}_{k-1} = \left\{ x \in \mathbb{R}^p : (x - c_{k-1})^T P_{k-1}^{-1} (x - c_{k-1}) \le 1 \right\}$$

les deux hyperplans tangents à $\mathbf{E}_{k\text{-}1}$ paramétrés par la direction d_k sont définis par :

$$\begin{aligned} \mathbf{H}_k^{'-} &= \left\{ x \in \mathbb{R}^p : d_k^T x = d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k} \right\} \\ \mathbf{H}_k^{'+} &= \left\{ x \in \mathbb{R}^p : d_k^T x = d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k} \right\} \end{aligned} \tag{1.42}$$

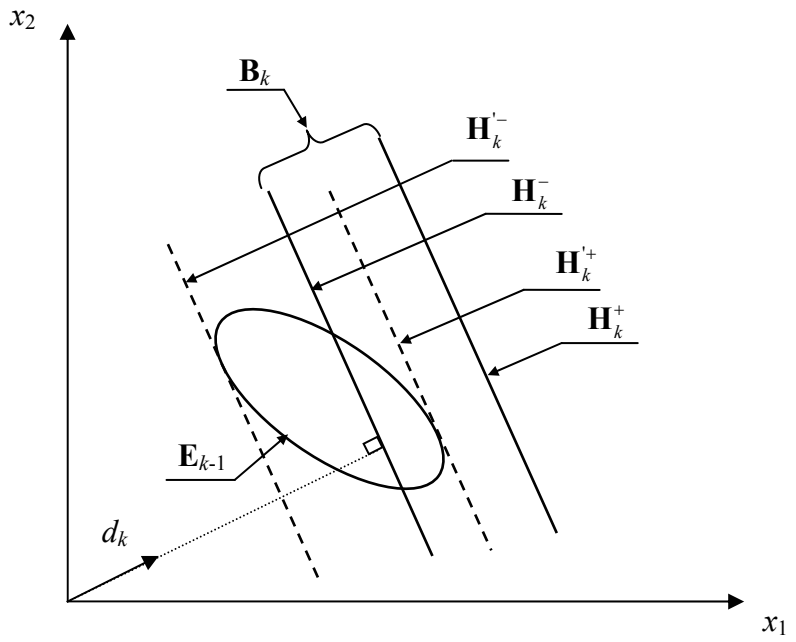


Figure 1.6 : Ellipsoïde et bande (2 paramètres)

Le test de cohérence de l'intersection et de réduction (si besoin est) de la bande est réalisé conformément à une suggestion de Belforte et Bona, reprise dans [CLE 87] :

- **SI** $y_k - 1 \ge d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k}$ ou $y_k + 1 \le d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k}$ alors l'intersection est vide $\mathbf{E}_{k-1} \cap \mathbf{B}_k = \varnothing$. L'espace des paramètres admissibles est vide ;
- **SINON** il existe une intersection, $\mathbf{E}_{k-1} \cap \mathbf{B}_k \ne \varnothing$:

a) Si $y_k - 1 \le d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k}$ **et** $y_k + 1 \ge d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k}$, l'ellipsoïde $\mathbf{E}_{k\text{-}1}$ est totalement inclus dans $\mathbf{B}_k$. On a $\mathbf{E}_k = \mathbf{E}_{k-1}$;

b) Si $y_k - 1 < d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k}$ **ou** $y_k + 1 > d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k}$, un des deux hyperplans de $\mathbf{B}_k$ ne coupe pas $\mathbf{E}_k$. On remplace alors $\mathbf{B}_k$ par $\mathbf{B}_k^{'}$ où :

$$\mathbf{B}_k^{'} = \left\{ x \in \mathbb{R}^p : b_k^{'-} \le d_k^T x \le b_k^{'+} \right\} \tag{1.43}$$

avec
$$\begin{cases} b_k^{'+} = \min\left\{ y_k + 1, d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k} \right\} \\ b_k^{'-} = \max\left\{ y_k - 1, d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k} \right\} \end{cases} \tag{1.44}$$

La nouvelle bande $\mathbf{B}_k^{'}$ est donc obtenue par réduction de $\mathbf{B}_k$. On peut aussi écrire $\mathbf{B}_k^{'}$ sous une forme normalisée :

$$\mathbf{B}_k^{'} = \left\{ x \in \mathbb{R}^p : y_k^{'} - 1 \le d_k^{'T} x \le y_k^{'} + 1 \right\} \tag{1.45}$$

avec
$$\begin{cases} y_k^{'} = \dfrac{b_k^{'+} + b_k^{'-}}{b_k^{'+} - b_k^{'-}} \\ d_k^{'} = \dfrac{2d}{b_k^{'+} - b_k^{'-}} \end{cases} \tag{1.46}$$

On applique ensuite l'algorithme OBE retenu sur $\mathbf{E}_k$ et $\mathbf{B}_k^{'}$ (réduction de bande).

c) Si $y_k - 1 > d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k}$ **et** $y_k + 1 < d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k}$, les deux hyperplans $\mathbf{H}_k^+$ et $\mathbf{H}_k^-$ coupent $\mathbf{E}_{k-1}$. On utilise directement l'algorithme OBE pour construire le nouvel ellipsoïde à partir de $\mathbf{E}_{k-1}$ et $\mathbf{B}_k$ (pas de réduction de bande nécessaire).

1.2.5 Généralisation : cas de l'intersection de deux ellipsoïdes

1.2.5.1 Formulation

Comme on l'a déjà souligné, il existe plusieurs algorithmes OBE différents pour calculer un ellipsoïde englobant l'intersection d'un ellipsoïde $\mathbf{E}_{k-1}$ et d'une bande $\mathbf{B}_k$. Chaque algorithme donne un ellipsoïde qui est optimal au sens d'un critère choisi : suivant l'algorithme utilisé, la forme, la taille et les directions des axes principaux peuvent donc différer.

On s'intéresse maintenant au problème plus général de l'intersection de deux ellipsoïdes. En fait, une bande $\mathbf{B}_k$ définie par l'équation (1.3) peut être considérée comme un

ellipsoïde dégénéré où $P_k^{-1} = M_k = d_k^T d_k$. Il en résulte que l'on peut construire l'ellipsoïde englobant l'intersection de deux ellipsoïdes avec l'un des algorithmes OBE. Soient deux ellipsoïdes $\mathbf{E}_1(c_1, P_1)$ et $\mathbf{E}_2(c_2, P_2)$ sous leur forme normalisée :

$$\mathbf{E}(c_i, P_i) = \left\{ x \in \mathbb{R}^p : (x - c_i)^T P_i^{-1} (x - c_i) \leq 1 \right\}, \; i = 1,2$$

Un ellipsoïde $\mathbf{E}$ englobant l'intersection $\mathbf{E}_1 \cap \mathbf{E}_2$ satisfera :

$$\mathbf{E} = \left\{ x \in \mathbb{R}^p : \alpha (x - c_1)^T P_1^{-1} (x - c_1) + (1 - \alpha)(x - c_2)^T P_2^{-1} (x - c_2) \leq 1 \right\} \tag{1.47}$$

avec $0 \leq \alpha \leq 1$. En développant l'équation (1.47), on obtient :

$$\mathbf{E} = \left\{ x \in \mathbb{R}^p : \; \alpha (x - c_\alpha)^T P_\alpha^{-1} (x - c_\alpha) \leq 1 - \delta_\alpha \right\} \tag{1.48}$$

où
$$\begin{cases} P_\alpha^{-1} = \alpha P_1^{-1} + (1 - \alpha) P_2^{-1} \\ c_\alpha = P_\alpha \left[\alpha P_1^{-1} c_1 + (1 - \alpha) P_2^{-1} c_2 \right] \\ \delta_\alpha = \alpha c_1^T P_1^{-1} c_1 + (1 - \alpha) c_2^T P_2^{-1} c_2 - c_\alpha^T P_\alpha^{-1} c_\alpha \end{cases} \tag{1.49}$$

L'ellipsoïde englobant dépend d'un coefficient α (comme c'était le cas pour les algorithmes OBE précédents). Cependant, en appliquant les critères utilisés dans le cas de l'intersection d'une bande et d'un ellipsoïde, on ne peut pas trouver de formule explicite pour calculer α [DUR 01b].

1.2.5.2 Test de cohérence

Comme pour l'intersection entre un ellipsoïde et une bande, avant tout calcul de l'ellipsoïde englobant, il est nécessaire de vérifier que l'intersection n'est pas vide. En effet, si l'intersection est vide, l'algorithme peut renvoyer un ellipsoïde « englobant » sans que ce dernier n'ait un sens physique. Notons que l'utilisateur ne peut pas détecter ce type de situation s'il n'a pas **au préalable** vérifié que l'intersection est non vide. A titre d'exemple, considérons l'intersection des ellipsoïdes $\mathbf{E}_1(c_1, P_1)$ et $\mathbf{E}_2(c_2, P_2)$ avec $c_1 = [0 ; 0]$, $c_2 = [4 ; 0]$; $P_1 = I_2$ et $P_2 = 4*I_2$. Cette intersection est vide (cf. figure 1.7). Cependant, en utilisant les formules de (1.49), on trouve un ellipsoïde $\mathbf{E}(c, P)$ censé englober cette (non-)intersection, par exemple pour $\alpha = 0.8$, on obtient $c = [0.2353 ; 0]$ et $P = 0.2907*I_2$.

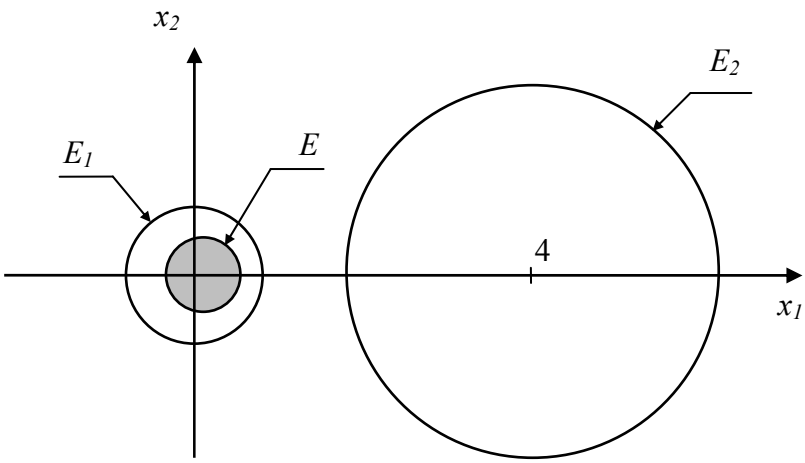


Figure 1.7 : Un exemple de l'intersection fausse de deux ellipsoïdes.

Dans ce travail, nous proposons les théorèmes suivants pour tester l'intersection de deux ellipsoïdes avant tout calcul d'un ellipsoïde englobant [LES 03].

Théorème 1.1

La condition nécessaire et suffisante pour que l'intersection $\mathbf{E}_1 \cap \mathbf{E}_2$ *soit non vide est* $\max(\delta_\alpha) < 1$*, avec* δ_α *définie dans (1.49).*

Remarque : pour $\max(\delta_\alpha) = 1$, l'intersection se réduit à un point assimilable à l'ellipsoïde englobant. En fait, la partie gauche de (1.48) est un terme quadratique. Elle est donc positive. Alors, pour que l'ellipsoïde **E** tel que $\mathbf{E} \supset \mathbf{E}_1 \cap \mathbf{E}_2$ existe, on doit avoir :

$$(1-\delta_\alpha) \geq 0,\ \forall\ \alpha \in [0, 1] \text{ ou encore } \max(\delta_\alpha) \leq 1,\ \forall\ \alpha \in [0,1].$$

Théorème 1.2

La fonction δ_α *est concave et possède un unique maximum en* $\hat{\alpha}$ *avec* $0 < \hat{\alpha} < 1$.

Démonstration

D'abord, on recherche à exprimer la fonction δ_α sous une forme plus simple. En substituant c_α de l'équation (1.49) dans δ_α, on obtient :

$$\delta_\alpha = \alpha c_1^T P_1^{-1} c_1 + (1-\alpha) c_2^T P_2^{-1} c_2 - \left[\alpha c_1^T P_1^{-1} + (1-\alpha) c_2^T P_2^{-1}\right] P_\alpha \left[\alpha P_1^{-1} c_1 + (1-\alpha) P_2^{-1} c_2\right]$$

ou encore

$$\delta_\alpha = \alpha c_1^T P_1^{-1} c_1 + (1-\alpha) c_2^T P_2^{-1} c_2 - \Big[\alpha^2 c_1^T P_1^{-1} P_\alpha P_1^{-1} c_1 + (1-\alpha^2) c_2^T P_2^{-1} P_\alpha P_2^{-1} c_2 + \alpha(1-\alpha) c_1^T P_1^{-1} P_\alpha P_2^{-1} c_2 + \alpha(1-\alpha) c_2^T P_2^{-1} P_\alpha P_1^{-1} c_1\Big]$$

Utilisons le lemme d'inversion matricielle, on développe les termes suivants:

$$P_\alpha P_1^{-1} = \left[P_1\left(\alpha P_1^{-1} + (1-\alpha)P_2^{-1}\right)\right]^{-1} = \frac{1}{\alpha} I - \frac{1-\alpha}{\alpha} P_1\left(\alpha P_2 + (1-\alpha)P_1\right)^{-1}$$

$$P_\alpha P_2^{-1} = \left[P_2\left(\alpha P_1^{-1} + (1-\alpha)P_2^{-1}\right)\right]^{-1} = \frac{1}{\alpha} I - \frac{1-\alpha}{\alpha} P_2\left(\alpha P_2 + (1-\alpha)P_1\right)^{-1}$$

$$P_1^{-1} P_\alpha P_2^{-1} = \left[P_2\left(\alpha P_1^{-1} + (1-\alpha)P_2^{-1}\right)P_1\right]^{-1} = \left(\alpha P_2 + (1-\alpha)P_1\right)^{-1}$$

$$P_2^{-1} P_\alpha P_1^{-1} = \left[P_1\left(\alpha P_1^{-1} + (1-\alpha)P_2^{-1}\right)P_2\right]^{-1} = \left(\alpha P_2 + (1-\alpha)P_1\right)^{-1}$$

Remplaçons ces termes dans δ_α et en simplifiant, on obtient :

$$\delta_\alpha = \alpha(1-\alpha)(c_1 - c_2)^T \left[\alpha P_2 + (1-\alpha) P_1\right]^{-1} (c_1 - c_2) \tag{1.50}$$

Pour conduire aux propriétés du théorème 1.2, dans un premier temps, on change les coordonnées [LES 03] de l'espace $x \to y$ telles que :

$$\begin{cases} \mathbf{E}_1(c_1, P_1) & \Rightarrow \quad \mathbf{E}_{1(y)}(0, \mathrm{I}) \\ \mathbf{E}_2(c_2, P_2) & \Rightarrow \quad \mathbf{E}_{2(y)}\left(c_{2(y)}, P_{2(y)}\right) \end{cases}$$

où $\mathbf{E}_i(c_i, P_i) \triangleq \mathbf{E}_{i(x)}(c_{i(x)}, P_{i(x)})$

$c_{2(y)} = X(c_1 - c_2)$, $P_{2(y)}^{-1} = X^{-T} P_2^{-1} X^{-1}$ avec $P_1^{-1} = X^T X$

et $\mathbf{E}_{i(x)}\left(c_{i(x)}, P_{i(x)}\right), \mathbf{E}_{i(y)}\left(c_{i(y)}, P_{i(y)}\right)$ dénotent l'ellipsoïde $\mathbf{E}_i$ dans les coordonnées x, y respectivement.

On fait ensuite un nouveau changement de coordonnées $y \to z$ tel que :

$$\begin{cases} \mathbf{E}_{1(y)}(0, \mathrm{I}) & \Rightarrow \quad \mathbf{E}_{1(z)}(0, \mathrm{I}) \\ \mathbf{E}_{2(y)}(c_{2(y)}, P_{2(y)}) & \Rightarrow \quad \mathbf{E}_{2(z)}(c_{(z)}, D^{-1}) \end{cases}$$

où $c_{(z)} = V c_{2(y)}$, $\left(P_{2(y)}\right)^{-1} = V^T D V$

et V est une matrice orthogonale, $D > 0$ une matrice diagonale avec $D = \mathrm{diag}(\tau_i)$, $\tau_i > 0$. [LES 03] montre que les intersections suivantes sont équivalentes :

$$\mathbf{E}_1 \cap \mathbf{E}_2 \;\Leftrightarrow\; \mathbf{E}_{1(y)} \cap \mathbf{E}_{2(y)} \;\Leftrightarrow\; \mathbf{E}_{1(z)} \cap \mathbf{E}_{2(z)}$$

L'expression de δ_α dans les coordonnées z devient :

$$\delta_\alpha = \alpha(1-\alpha) c_{(z)}^T \left[\alpha D^{-1} + (1-\alpha) I\right]^{-1} c_{(z)}$$

En dérivant δ_α par rapport à α, on obtient :

$$\frac{\partial \delta_\alpha}{\partial \alpha} = -\sum_{i=1}^{p} \frac{\left(\tau_i^{-1} - 1\right)\alpha^2 - 2\alpha + 1}{\left[\alpha \tau_i^{-1} + \left(1-\alpha\right)^2\right]^2} c_{(z_i)}^2$$

où $\quad c_{(z)} = \begin{bmatrix} c_{(z_1)} & c_{(z_2)} & \dots & c_{(z_p)} \end{bmatrix}^T$

La dérivée seconde donne :

$$\frac{\partial^2 \delta_\alpha}{\partial \alpha^2} = -2\sum_{i=1}^{p} \frac{\tau_i^{-1} c_{(z_i)}^2}{\left[\alpha \tau_i^{-1} + \left(1-\alpha\right)^2\right]^3} < 0$$

On a $\delta_{\alpha=0} = \delta_{\alpha=1} = 0$, $\frac{\partial^2 \delta_\alpha}{\partial \alpha^2} < 0$ et $\delta_\alpha > 0$, $\forall \alpha \in]0,1[$. Alors δ_α est une fonction concave et possède un unique maximum en $\hat{\alpha} \in]0,1[$. □

Théorème 1.3

Si le théorème 1.1 est satisfait, l'ellipsoïde englobant l'intersection $\mathbf{E}_1 \cap \mathbf{E}_2$ *sera* $\mathbf{E}(c,P)$ *avec* $c = c_\alpha$ *et* $P = \left(1-\delta_\alpha\right)P_\alpha$.

Démonstration

Ce théorème est facilement démontré en écrivant (1.45) sous la forme normalisée.

1.3 Modèle erreur de sortie pour l'estimation ellipsoïdale

1.3.1 Position du problème

Dans tous les algorithmes OBE précédents, nous avons supposé un modèle de la forme générale suivante :

$$y_k = d_k^T x^* + e_k \tag{1.51}$$

Le vecteur (régresseur) d_k est supposé connu, c'est-à-dire qu'il ne contient que les entrées, les sorties mesurées ou les termes observables, sans incertitude. L'erreur e_k à chaque instant représente la différence entre la sortie mesurée y_k et la sortie prédite $y_0(k) = d_k^T x^*$ où x^* est le paramètre que l'on doit estimer. Ce type de modèle, dit « erreur d'équation », est susceptible de conduire à une estimation biaisée, dépendant des propriétés statistiques de e_k.

On envisage maintenant l'application des méthodes d'estimation ensembliste ellipsoïdale à un modèle de type « erreur de sortie ». Ce travail a été initialement proposé par Clément [CLE 87]. Ferres et M'Saad [FER 97] en ont donné récemment une variante.

Considérons un modèle dynamique d'un système SISO linéaire en les paramètres décrit par :

$$\begin{cases} y_0(k) + a_1 y_0(k-1) + \ldots + a_n y_0(k-n) = b_0 u_k + b_1 u_{k-1} + \ldots + b_m u(k-m) \\ y_k = y_0(k) + \eta_k \end{cases} \tag{1.52}$$

où $y_0(k)$ est la sortie du modèle à l'instant k ;

y_k est la sortie mesurée ;

η_k présente la différence entre la sortie mesurée et la sortie du modèle, appelée erreur de sortie.

Posons :

$x = \begin{bmatrix} a_1 & \ldots & a_n & b_0 & \ldots & b_m \end{bmatrix}^T \in \mathbb{R}^p$ le vecteur des paramètres à estimer ;

$d_0(k) = \begin{bmatrix} -y_0(k-1) & \ldots & -y_0(k-n) & u_k & \ldots & u_{k-m} \end{bmatrix}^T \in \mathbb{R}^p$ le régresseur ;

avec $p = m + n + 1$. On peut réécrire (1.52) de la façon suivante :

$$\begin{cases} y_0(k) = -\sum_{i=1}^{n} a_j y_0(k-j) + \sum_{j=0}^{m} b_j u_{k-j} \\ y_k = y_0(k) + \eta_k \end{cases} \tag{1.53}$$

où $$y_k = d_0^T(k) x + \eta_k \tag{1.54}$$

On voit que le régresseur $d_0(k)$ ne dépend maintenant que des *sorties passées du modèle* mais pas des sorties mesurées. $d_0(k)$ est donc inconnu. C'est une différence important par rapport au modèle à erreur d'équation. A partir de (1.53), on écrit :

$$y_k = -\sum_{i=1}^{n} a_i y_0(k-i) + \sum_{j=0}^{m} b_j u_{k-j} + \eta_k = -\sum_{i=1}^{n} a_i \left(y_{k-i} - \eta_{k-i} \right) + \sum_{j=0}^{m} b_j u_{k-j} + \eta_k$$

où

$$y_k = -\sum_{i=1}^{n} a_i y_{k-i} + \sum_{j=0}^{m} b_j u_{k-j} + \eta_k + \sum_{i=1}^{n} a_i \eta_{k-i} \tag{1.55}$$

Posons $d_k = \begin{bmatrix} -y_{k-1} & \ldots & -y_{k-n} & u_k & \ldots & u_{k-m} \end{bmatrix}^T$. (1.55) peut se réécrire sous la forme « erreur d'équation » (1.51) :

$$y_k = d_k^T x + e_k$$

avec $$e_k = \eta_k + \sum_{i=1}^{n} a_i \eta_{k-i} \tag{1.56}$$

L'erreur d'équation e_k dépend des termes d'erreur de sortie et des paramètres à estimer, ce qui signifie qu'elle est inconnue. Les algorithmes OBE ne peuvent être appliqués directement. Cependant, on peut définir à chaque instant une borne supérieure sur e_k. Cette borne doit être assez grande pour tenir compte du terme $\sum_{i=1}^{n} a_i \eta_{k-i}$. En conséquence, l'algorithme OBE peut conduire dans ce cas, à une estimation très pessimiste de l'espace admissible pour les paramètres.

1.3.2 Reformulation du problème pour un modèle à erreur de sortie

Dans le modèle à erreur d'équation (1.51), le terme e_k contient à la fois le bruit de mesure et les erreurs de modélisation. Or, on voit dans (1.56) que e_k contient également les paramètres inconnus. Il semble donc impossible de déterminer la borne « exacte » de cette erreur pour pouvoir appliquer l'approche d'estimation ensembliste exposée ci-avant.

Si par contre on s'intéresse directement à un modèle à erreur de sortie (1.52), l'erreur η_k tient compte (de la même façon que pour le modèle à erreur d'équation) du bruit de mesure et des erreurs de modélisation. Il est donc plus aisé de caractériser la séquence $\{\eta_k\}$ dans (1.52) que $\{e_k\}$ définie par (1.56). Dans le contexte de l'estimation ensembliste, on suppose que η_k est bornée, *i.e.* :

$$|\eta_k| \leq \eta_k^M \tag{1.57}$$

ce qui est plus réaliste en pratique qu'une majoration de e_k via (1.56).

Reprenons le modèle (1.53) avec l'hypothèse « erreur de sortie inconnue mais bornée » :

$$\begin{cases} y_k = -\sum_{i=1}^{n} a_i y_0(k-i) + \sum_{j=0}^{m} b_j u_{k-j} + \eta_k & & \text{(a)} \\ y_k = y_0(k) + \eta_k & \forall k \leq l & \text{(b)} \\ |\eta_k| \leq \eta_k^M & & \text{(c)} \end{cases} \tag{1.58}$$

où l est la longueur des données. En utilisant la démarche dans [CLE 87], à partir de (1.58), on obtient les relations suivantes :

$$\begin{cases} \sum_{i=1}^{n} a_i \left(-y_{k-i} - \text{sign}(a_i)\eta_{k-i}^M\right) + \sum_{j=0}^{m} b_j u_{k-j} \le y_k + \eta_k^M \\ \sum_{i=1}^{n} a_i \left(-y_{k-i} + \text{sign}(a_i)\eta_{k-i}^M\right) + \sum_{j=0}^{m} b_j u_{k-j} \ge y_k - \eta_k^M \end{cases} \tag{1.59}$$

où $\text{sign}(a_i) = \begin{cases} 1 & \text{si } a_i \ge 0 \\ 0 & \text{si } a_i < 0 \end{cases}$

Pour se ramener à la forme générale faisant apparaître des hyperplans, on pose :

$$\begin{cases} d_m(k) = \left[\left(-y_{k\text{-}1} - \text{sign}(a_1)\eta_{k\text{-}1}^M\right) \quad \dots \quad \left(-y_{k\text{-}n} - \text{sign}(a_n)\eta_{k\text{-}n}^M\right) \quad u_k \quad \dots \quad u_{k\text{-}m}\right]^T \\ d_M(k) = \left[\left(-y_{k\text{-}1} + \text{sign}(a_1)\eta_{k\text{-}1}^M\right) \quad \dots \quad \left(-y_{k\text{-}n} + \text{sign}(a_n)\eta_{k\text{-}n}^M\right) \quad u_k \quad \dots \quad u_{k\text{-}m}\right]^T \\ x = \left[a_1 \quad \dots \quad a_n \quad b_0 \quad \dots \quad b_m\right]^T \end{cases} \tag{1.60}$$

ce qui conduit pour (1.59) à :

$$\begin{cases} d_m^T(k)x \le y_k + \eta_k^M \\ d_M^T(k)x \ge y_k - \eta_k^M \end{cases} \tag{1.61}$$

Les inéquations dans (1.61) représentent l'espace des paramètres correspondant à l'observation à l'instant k pour le modèle à erreur de sortie considéré. Les remarques suivantes s'imposent :

- les deux hyperplans donnant la bande $\mathbf{B}_k$ dans l'espace des paramètres n'ont pas le même vecteur directeur. Autrement dit, ces deux hyperplans ne sont pas parallèles :

$$\mathbf{B}_k = \left[\mathbf{H}_M^-(k), \mathbf{H}_m^+(k)\right] \text{ où } \begin{cases} \mathbf{H}_M^-(k) = \left\{x \in \mathbb{R}^p : d_M^T(k)x = y_k - \eta_k^M\right\} \\ \mathbf{H}_m^+(k) = \left\{x \in \mathbb{R}^p : d_m^T(k)x = y_k + \eta_k^M\right\} \end{cases} \tag{1.62}$$

- le vecteur directeur d'un hyperplan dépend du signe des paramètres à estimer. Cela signifie que, dans l'espace des paramètres, lorsqu'il y a un changement de signe d'une variable, le vecteur directeur de l'hyperplan considéré change aussi (voir figure 1.8).

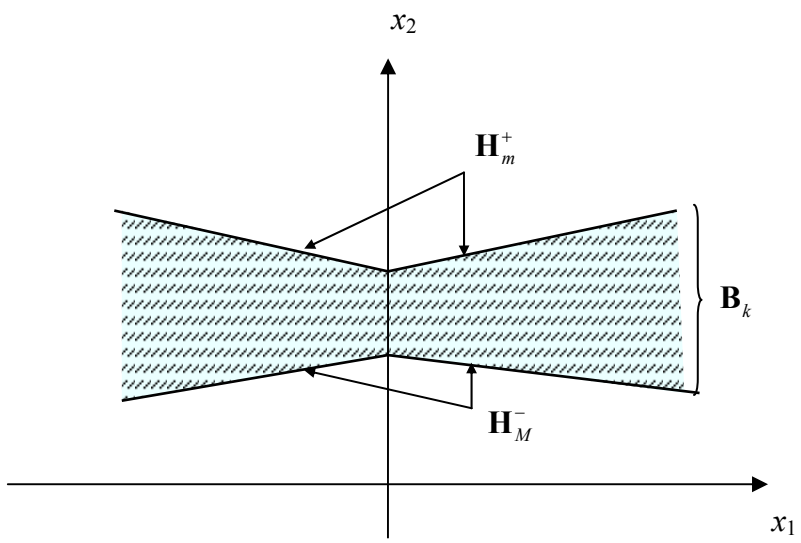


Figure 1.8 : Bande dans l'espace des paramètres pour modèle erreur de sortie ($p = 2$).

Pour simplifier, on suppose connaître *a priori* le signe des paramètres à estimer, ce qui en pratique est peu restrictif. Alors, à chaque instant, dans chaque région de paramètres, la bande $\mathbf{B}_k$ de l'équation (1.62) se restreint à l'espace entre deux hyperplans non parallèles. La figure 1.9 illustre l'intersection de ces bandes après 3 observations.

Notons enfin que l'on peut généraliser ce problème dans le cas où tous les éléments du vecteur régresseur contiennent des erreurs bornées. Le modèle à erreur de sortie avec bruit de mesure inconnu mais borné est un cas particulier où les sorties sont les seuls éléments du régresseur supposés présenter une erreur. On pourrait également considérer une erreur sur les entrées ce qui n'apporte pas de difficulté supplémentaire. Dans ce cas, la sortie mesurée est donnée par [CER 93] :

$$y_k = (d_k + \omega_k)^T x + \eta_k \tag{1.63}$$

où le régresseur est défini par $d_k = \left[d_1(k) \ldots d_i(k) \ldots d_p(k) \right]^T$, avec $d_i(k)$ le $i^{\text{ème}}$ élément à l'instant k et $\omega_k = \left[\omega_1(k) \ldots \omega_i(k) \ldots \omega_p(k) \right]^T$ le vecteur contenant les erreurs sur les éléments correspondants du régresseur.

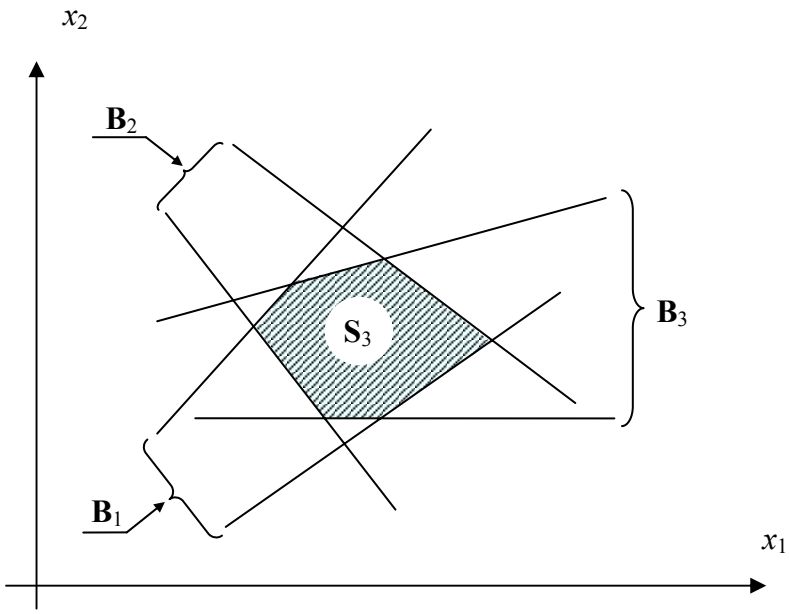


Figure 1.9 : Espace admissible des paramètres pour un modèle à erreur de sortie après 3 observations

Supposons que les erreurs sur le régresseur et l'erreur de sortie soient bornées, *i.e.* :

$$\left|\omega_i(k)\right| \le \omega_i^M(k) \quad et \quad \left|\eta_k\right| \le \eta_k^M \tag{1.64}$$

Des équations (1.63) et (1.64), on obtient :

$$-\eta_k^M - \sum_{i=1}^{p} \omega_i^M(k)\mathrm{sign}(x_i)x_i \le y_k - d_k^T x \le \sum_{i=1}^{p} \omega_i^M(k)\mathrm{sign}(x_i)x_i + \eta_k^M \tag{1.65}$$

Alors :

$$\begin{cases} \sum_{i=1}^{p}\left[d_i(k) - \mathrm{sign}(x_i)\omega_i^M(k)\right]x_i \le y_k + \eta_k^M \\ \sum_{i=1}^{p}\left[d_i(k) + \mathrm{sign}(x_i)\omega_i^M(k)\right]x_i \ge y_k - \eta_k^M \end{cases} \tag{1.66}$$

Posons :

$$\begin{aligned} d_m(k) &= \left[\left(d_1(k) - \mathrm{sign}(x_1)\omega_1^M(k)\right) \quad \ldots \quad \left(d_p(k) - \mathrm{sign}(x_p)\omega_p^M(k)\right)\right]^T \\ d_M(k) &= \left[\left(d_1(k) + \mathrm{sign}(x_1)\omega_1^M(k)\right) \quad \ldots \quad \left(d_p(k) + \mathrm{sign}(x_p)\omega_p^M(k)\right)\right]^T \end{aligned} \tag{1.67}$$

On peut maintenant ramener (1.66) à une formule de la forme (1.61) :

$$\begin{cases} d_m^T(k)x \le y_k + \eta_k^M \\ d_M^T(k)x \ge y_k - \eta_k^M \end{cases}$$

En supposant que l'on connaît *a priori* le signe des paramètres, dans ce cas général, l'espace admissible des paramètres pour chaque observation se restreint à une bande d'espace contenue entre deux hyperplans non parallèles.

Pour alléger les formules, on omet l'indice de temps k dans la présentation qui suit. De plus, sans perte de généralité, on suppose que les hyperplans « frontières » d'une bande sont représentés sous la forme normalisée. Les équations ci-avant deviennent :

$$\begin{cases} d_m^T x \le y+1 \\ d_M^T x \ge y-1 \end{cases} \tag{1.68}$$

et la bande de paramètres admissibles dans (1.62) s'écrit :

$$\mathbf{B}_k = \left[\mathbf{H}_M^-, \mathbf{H}_m^+\right] \text{ avec } \begin{cases} \mathbf{H}_M^- = \left\{x \in \mathbb{R}^p : d_M^T x = y-1\right\} \\ \mathbf{H}_m^+ = \left\{x \in \mathbb{R}^p : d_m^T x = y+1\right\} \end{cases} \tag{1.69}$$

1.3.3 Test de cohérence de l'intersection et réduction de bande

Comme lors de l'utilisation des algorithmes OBE avec « modèle à erreur d'équation », avant tout calcul, on vérifie la position de la bande $\mathbf{B}_k$ par rapport à l'ellipsoïde courant $\mathbf{E}_{k\text{-}1}$ (cf. figure 1.10) pour tester la cohérence de l'intersection et réduire si besoin $\mathbf{B}_k$. On considère la forme générale normalisée de la bande $\mathbf{B}_k$ (1.69) :

- **SI** $y_k - 1 \ge d_M^T c_{k-1} + \sqrt{d_M^T P_{k-1} d_M}$ ou $y+1 \le d_m^T c_{k-1} - \sqrt{d_m^T P_{k-1} d_k}$ alors $\mathbf{E}_{k-1} \cap \mathbf{B}_k = \varnothing$ (figure 1.10a) ;
- **SINON** $\mathbf{E}_{k-1} \cap \mathbf{B}_k \ne \varnothing$ et :

  a) Si $y-1 \le d_M^T c_{k-1} - \sqrt{d_M^T P_{k-1} d_M}$ **et** $y+1 \ge d_m^T c_{k-1} + \sqrt{d_m^T P_{k-1} d_m}$, alors la bande $\mathbf{B}_k$ contient l'ellipsoïde $\mathbf{E}_{k\text{-}1}$, et $\mathbf{E}_k = \mathbf{E}_{k\text{-}1}$ (figure 1.10b) ;

  b) Si $y_k - 1 < d_M^T c_{k-1} - \sqrt{d_M^T P_{k-1} d_M}$ **et** $y+1 < d_m^T c_{k-1} + \sqrt{d_m^T P_{k-1} d_m}$ alors $\mathbf{H}_m^+$ coupe $\mathbf{E}_{k\text{-}1}$ et $\mathbf{H}_M^-$ ne coupe pas $\mathbf{E}_{k\text{-}1}$ (figure 1.10c). On remplace $\mathbf{B}_k$ par $\mathbf{B}_k^{'} = \left[\mathbf{H}_m^{'-}, \mathbf{H}_m^+\right]$ avec :

  $$\mathbf{H}_m^{'-} = \left\{x \in \mathbb{R}^p : d_m^T x = d_m^T c_{k-1} - \sqrt{d_m^T P_{k-1} d_m}\right\} \tag{1.70}$$

  où $\mathbf{H}_m^{'-}$ est l'hyperplan tangent à $\mathbf{E}_{k\text{-}1}$ et parallèle à $\mathbf{H}_m^+$ tel que $\mathbf{B}_k^{'}$ contient l'espace admissible des paramètres. On applique l'algorithme OBE avec $\mathbf{E}_{k\text{-}1}$ et $\mathbf{B}_k^{'}$. $\mathbf{B}_k^{'}$ doit être préalablement normalisée via les relations dans (1.46) ;

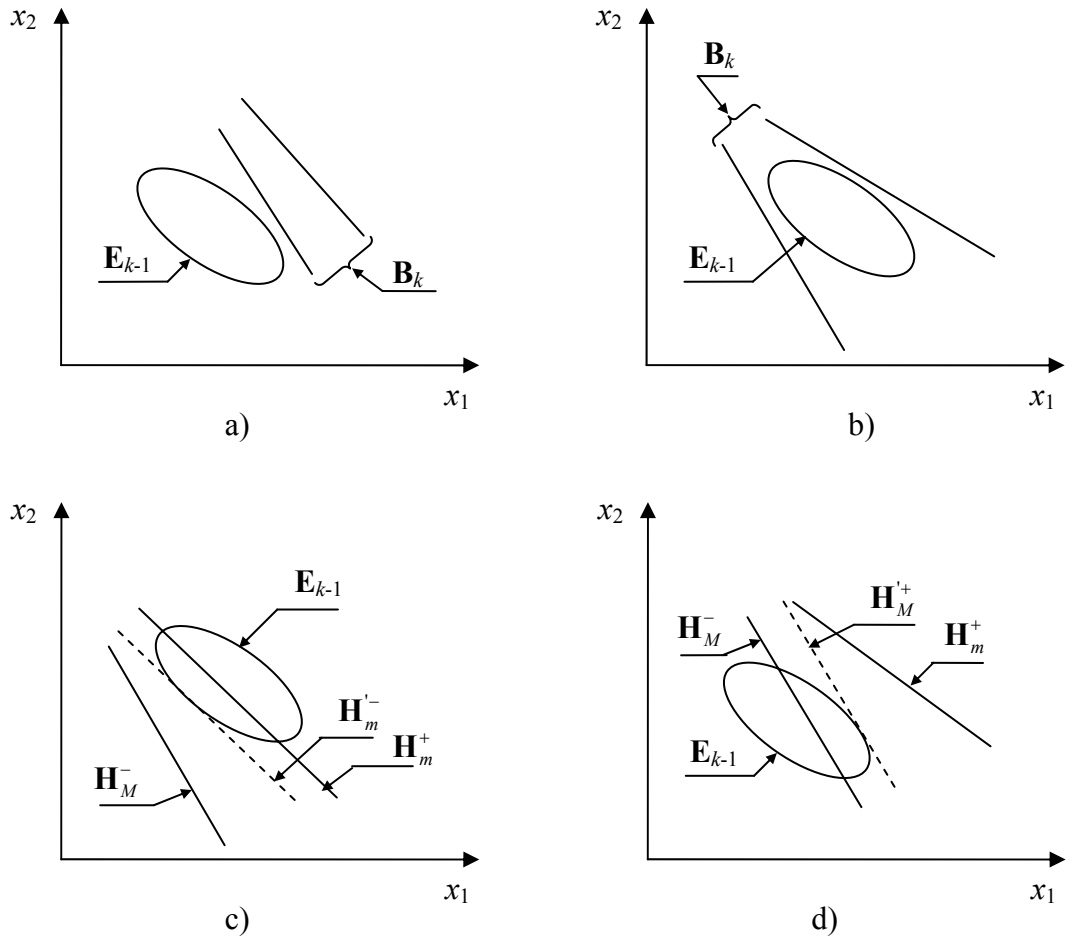


Figure 1.10 : Positions relative de la bande et de l'ellipsoïde (modèle à erreur de sortie)

c) Si $y-1 > d_M^T c_{k-1} - \sqrt{d_M^T P_{k-1} d_M}$ **et** $y+1 \geq d_m^T c_{k-1} + \sqrt{d_m^T P_{k-1} d_m}$ alors $\mathbf{H}_M^-$ coupe $\mathbf{E}_{k\text{-}1}$ et $\mathbf{H}_m^+$ ne coupe pas $\mathbf{E}_{k\text{-}1}$ (figure 1.10d). Remplaçons $\mathbf{B}_k$ par $\mathbf{B}_k^{'} = \left[\mathbf{H}_M^-, \mathbf{H}_M^{'+}\right]$ avec :

$$\mathbf{H}_M^{'+} = \left\{ x \in \mathbb{R}^p : d_M^T x = d_M^T c_{k-1} + \sqrt{d_M^T P_{k-1} d_M} \right\} \tag{1.71}$$

où $\mathbf{H}_M^{'+}$ est l'hyperplan tangent à $\mathbf{E}_{k\text{-}1}$ et parallèle à $\mathbf{H}_M^-$ tel que $\mathbf{B}_\mathbf{k}^{'}$ contient l'espace admissible des paramètres. On normalise $\mathbf{B}_k^{'}$ par (1.46) et on construit ensuite $\mathbf{E}_k \supset \mathbf{E}_{k\text{-}1} \cap \mathbf{B}_k^{'}$ par l'algorithme OBE ;

d) Si $y-1 > d_M^T c_{k-1} - \sqrt{d_M^T P_{k-1} d_M}$ **et** $y+1 < d_m^T c_{k-1} + \sqrt{d_m^T P_{k-1} d_m}$, les deux hyperplans $\mathbf{H}_M^-$ et $\mathbf{H}_m^+$ coupent $\mathbf{E}_{k\text{-}1}$(figure 1.11a). Ces deux hyperplans ne sont pas parallèles ce qui implique que la bande $\mathbf{B}_k$ ne peut pas être définie par une forme quadratique. En conséquence, on ne peut pas appliquer directement l'algorithme OBE. On présente maintenant les algorithmes permettant de résoudre ce problème.

1.3.4 Algorithme FHCL

Cet algorithme est proposé par Clément dans [CLE 87]. Le travail de construction de l'ellipsoïde $\mathbf{E}_k$ comprend deux étapes (cf. figure 1.11). A chaque étape, l'algorithme FHCL appelle un algorithme de type OBE tel que présenté précédemment.

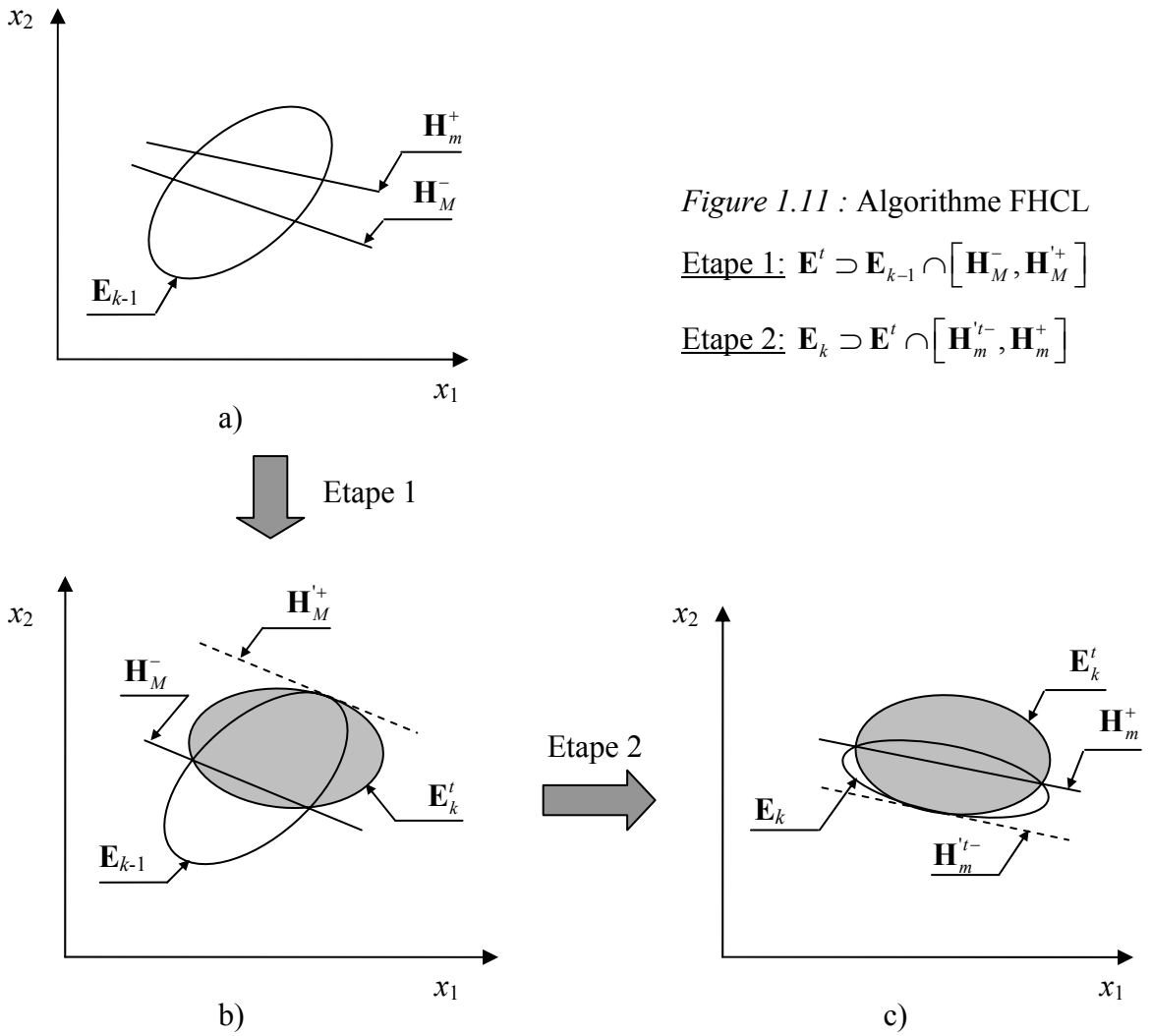


Figure 1.11 : Algorithme FHCL

<u>Etape 1:</u> $\mathbf{E}^t \supset \mathbf{E}_{k-1} \cap \left[\mathbf{H}_M^-, \mathbf{H}_M^{'+}\right]$

<u>Etape 2:</u> $\mathbf{E}_k \supset \mathbf{E}^t \cap \left[\mathbf{H}_m^{'t-}, \mathbf{H}_m^+\right]$

- **Etape** 1 : on cherche l'ellipsoïde $\mathbf{E}^t$ (suivant un critère prédéfini) qui contienne l'intersection $\mathbf{E}_{k\text{-}1} \cap \left[\mathbf{H}_M^-, \mathbf{H}_M^{'+}\right]$ où $\mathbf{H}_M^{'+}$ est défini par (1.71) ;
- **Etape 2** : on construit l'ellipsoïde $\mathbf{E}_k$ qui contienne l'intersection $\mathbf{E}^t \cap \left[\mathbf{H}_m^{'t-}, \mathbf{H}_m^+\right]$ où $\mathbf{H}_m^{'t-}$ est l'hyperplan tangent à $\mathbf{E}^t$ et parallèle avec $\mathbf{H}_m^+$ tel que $\left[\mathbf{H}_m^{'t-}, \mathbf{H}_m^+\right]$ contienne l'espace paramétrique admissible.

Cette décomposition en deux étapes n'est pas optimale, elle peut conduire à un ellipsoïde $\mathbf{E}_k$ sous-optimal (au sens où il est sur-englobant) par rapport à un traitement simultané des deux hyperplans non parallèles.

1.3.5 Algorithme à une seule coupe de Ferreres et M'Saad

Ferreres et M'Saad [FER 97] proposent une méthode pour rechercher un hyperplan $\mathbf{H}_M^{''}$ (figure 1.12) qui soit parallèle à $\mathbf{H}_M^{-}$ tel que $\left[\mathbf{H}_M^{-}, \mathbf{H}_M^{''}\right]$ soit le plus compact et contienne l'espace admissible des paramètres.

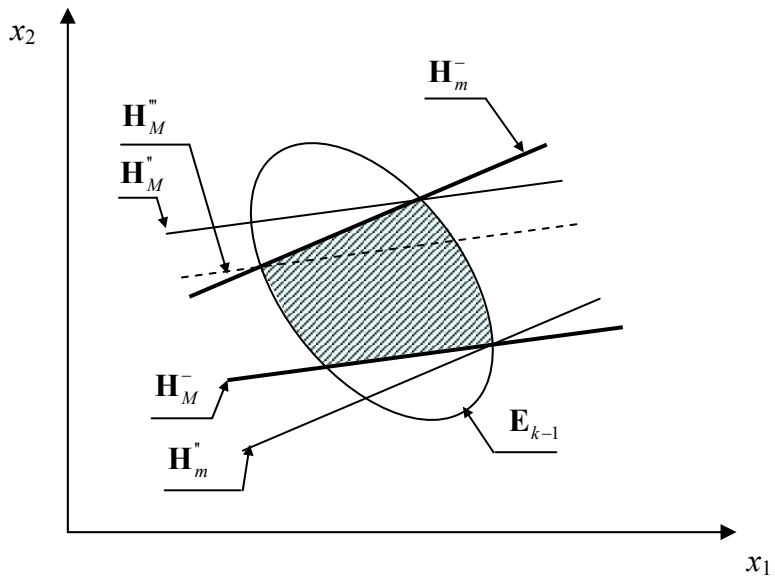


Figure 1.12 : Algorithme de Ferreres et M'Saad

Théorème 1.4 ([FER 97])

Si $\mathbf{H}_M^{-}$, $\mathbf{H}_m^{+}$ *sont tous les deux strictement sécants à* E_{k-1}, *alors l'équation qui représente* $\mathbf{H}_M^{''}$ *est :*

$$\mathbf{H}_M^{''} = \left\{ x \in \mathbb{R}^p : d_M^T x = f_M^{''} \right\} \tag{1.72}$$

où

$$f_M^{''} = d_M^T c_{k-1} + \frac{h_2 p_{12} + \sqrt{\left(p_{22} - h_2^2\right)}\sqrt{\left(p_{11}p_{22} - p_{12}^2\right)}}{p_{22}} \tag{1.73}$$

avec $h_2 = y + 1 - d_m^T c_{k-1}$

$$p_{11} = \sqrt{d_m^T P_{k-1} d_m} \;;\; p_{12} = p_{21} = \sqrt{d_m^T P_{k-1} d_M} \;;\; p_{22} = \sqrt{d_M^T P_{k-1} d_M} \tag{1.74}$$

Démonstration

$\mathbf{H}_M^{''}$ est parallèle à $\mathbf{H}_M^{-}$, donc son vecteur directeur est d_M (voir l'équation (1.72)). On doit chercher $f_M^{''}$ satisfaisant le problème optimal $f_M^{''} = \max\left\{d_M^T x\right\}$ qui admette pour contrainte que x appartienne à $\mathbf{E}_{k\text{-}1}$ et à $\mathbf{H}_m^{+}$. Cela signifie que :

$$\begin{cases} x \in \mathbf{E}_{k-1} & \Leftrightarrow \quad (x-c_{k-1})^T P_{k-1}(x-c_{k-1}) = 1 \\ x \in \mathbf{H}_m^{-} & \Leftrightarrow \quad \mathrm{d}_\mathrm{m}^\mathrm{T} x = y+1 \\ x \in \mathbf{H}_M^{''} & \Leftrightarrow \quad \mathrm{d}_\mathrm{M}^\mathrm{T} x = f_M^{''} \end{cases}$$

On définit une fonction Hamiltonienne H telle que :

$$H = d_M^T x + \chi_1\left[(x-c_{k-1})^T P_{k-1}^{-1}(x-c_{k-1}) - 1\right] + \chi_2\left(d_m^T x - y - 1\right) \tag{1.75}$$

En dérivant H suivant les variables x, χ_1, χ_2, et en posant $h_1 = d_M^T(x-c_{k-1})$, on a :

$$\begin{cases} p_{11} + 2\chi_1 h_1 + \chi_2 p_{12} = 0 \\ p_{21} + 2\chi_1 h_2 + \chi_2 p_{22} = 0 \\ \quad h_1 + 2\chi_1 + \chi_2 h_2 = 0 \end{cases} \tag{1.76}$$

La solution de ce système de 3 équations conduit à :

$$h_1 = \frac{h_2 p_{12} \pm \sqrt{\left(p_{22} - h_2^2\right)}\sqrt{\left(p_{11}p_{22} - p_{12}^2\right)}}{p_{22}} \tag{1.77}$$

Or, on sait que $f_M^{''} = d_M^T x = h_1 + d_M^T c_{k-1}$. Alors, le maximum de $f_M^{''}$ correspond bien à la solution la plus grande pour h_1 :

$$f_M^{''} = d_M^T c_{k-1} + \frac{h_2 p_{12} + \sqrt{\left(p_{22} - h_2^2\right)}\sqrt{\left(p_{11}p_{22} - p_{12}^2\right)}}{p_{22}}. \qquad \square$$

Remarque

- le signe (-) dans (1.77) correspond à l'hyperplan $\mathbf{H}_M^{'''}$ dessiné sur la figure 1.12 ;
- les conditions pour que $\mathbf{H}_M^{-}$ et $\mathbf{H}_m^{+}$ soient tous les deux sécants à $\mathbf{E}_{k\text{-}1}$ garantissent que les termes en racine carrée dans (1.77) soient toujours positifs.

En appliquant un principe identique à celui proposé par [FER 97], on peut définir $\mathbf{H}_m^{''}$ (voir figure 1.12) qui vérifie $\mathbf{H}_m^{''} // \mathbf{H}_m^{+}$ tel que $\left[\mathbf{H}_m^{''}, \mathbf{H}_m^{+}\right]$ soit le plus compact et contienne l'espace admissible des paramètres. On obtient alors un résultat nouveau énoncé dans le théorème 1.5

Théorème 1.5

Si $\mathbf{H}_M^-$, $\mathbf{H}_m^+$ *sont tous les deux strictement sécants à* $\mathbf{E}_{k\text{-}1}$, *alors l'équation qui représente* $\mathbf{H}_m^{''}$ *est :*

$$\mathbf{H}_m^{''} = \left\{ x \in \mathbb{R}^p : d_m^T x = f_m^{''} \right\} \tag{1.78}$$

où

$$f_m^{''} = d_m^T c_{k-1} + \frac{h_1 p_{12} - \sqrt{\left(p_{11} - h_1^2\right)}\sqrt{\left(p_{11}p_{22} - p_{12}^2\right)}}{p_{11}} \tag{1.79}$$

et

$$h_1 = y - d_m^T c_{k-1} - 1$$

Remarques

- les variables p_{ij}, $i, j = 1, 2$ sont définies dans l'équation (1.74) ;
- pour la démonstration, on cherche la solution optimale du problème $f_m^{''} = \min\left\{ d_m^T x \right\}$ satisfaisant la contrainte que x appartienne à $\mathbf{E}_{k\text{-}1}$ et à $\mathbf{H}_M^-$.

En utilisant cette modification des hyperplans, l'algorithme de Ferreres et M'Saad construit l'ellipsoïde $\mathbf{E}_k$ de la façon suivante :

$$\mathbf{E}_k = \mathbf{E}_{k-1} \cap \mathbf{B}_k^{''}$$

avec $\mathbf{B}_k^{''} = \left[\mathbf{H}_M^-, \mathbf{H}_M^{''} \right]$ ou $\mathbf{B}_k^{''} = \left[\mathbf{H}_m^{''}, \mathbf{H}_m^+ \right]$

La bande $\mathbf{B}_k^{''}$ est définie par deux hyperplans parallèles et on peut la réécrire sous la forme normalisée (1.43), ce qui permet d'appliquer directement l'un des algorithmes de type OBE. Cet algorithme construit alors l'ellipsoïde $\mathbf{E}_k$ par une seule coupe (donc en une seule étape). Il faut noter que le résultat obtenu est généralement moins pessimiste (dans le sens où $\mathbf{E}_k$ obtenu est « moins » surenglobant) que celui fourni par l'algorithme FHCL.
Pour améliorer le résultat, on pourrait exploiter le théorème 1.5 et obtenir un autre ellipsoïde englobant $\mathbf{E}_k'$. L'étape suivante consisterait à calculer l'englobant de l'intersection $\mathbf{E}_k \cap \mathbf{E}_k^{'}$.

1.4 Conclusion

Dans ce chapitre, on a présenté différents algorithmes d'estimation ensembliste ellipsoïdale. Cette technique d'identification ne nécessite pas d'hypothèse statistique sur les erreurs attachées à la sortie du système. Seule la borne de l'erreur doit être connue. L'interprétation du résultat obtenu par cette technique d'identification est totalement

différente de ce qu'on obtient par les techniques classiques d'identification puisque l'on calcule un ensemble (ici un ellipsoïde) dont on garantit qu'il contient l'espace admissible de paramètres.

Les algorithmes utilisés dans cette approche d'estimation sont appelés OBE (Outer Bounding Ellipsoid). Ils se répartissent en deux groupes.

Le groupe 1 comprend des algorithmes qui cherchent à réduire la taille géométrique de l'ellipsoïde. Nous avons présenté dans ce groupe trois algorithmes différents :

- l'algorithme de Fogel et Huang (FH) ;
- un algorithme basé sur une famille paramétrée d'ellipsoïdes ;
- l'algorithme SMSA pour l'intérêt sur la convergence du volume de l'ellipsoïde.

Pour chacun de ces algorithmes, la taille de l'ellipsoïde est évaluée via deux critères différents, à savoir, le critère du déterminant ou le critère de la trace. On a montré que ces trois algorithmes sont en fait dans leur principe, identiques. Ils diffèrent simplement dans l'utilisation de la forme des ensembles considérés (complète ou normalisée) et du coefficient de pondération entre l'ensemble obtenu précédemment $\mathbf{E}_{k\text{-}1}$ et les nouvelles informations $\mathbf{B}_k$.

Les algorithmes du groupe 2 se basent sur la convergence du coefficient σ_k^2. Trois d'entre eux ont été présentés :

- l'algorithme de Dasgupta & Huang (DH) ;
- l'algorithme de Tan *et al.* (TAN) ;
- l'algorithme de Lozano-Leal & Ortega.

Ces algorithmes évaluent l'erreur de prédiction γ_k avant de construire un nouvel ellipsoïde. De plus, ils dépendent d'un facteur υ choisi par utilisateur, qui aura une incidence directe sur la convergence de l'algorithme.

Le développement de cette approche d'identification à un contexte « erreur de sortie » a également été présenté. On peut également prendre en compte une erreur bornée sur les régresseurs [FER 97]. La bande considérée dans l'espace des paramètres à chaque instant est délimitée par des hyperplans non parallèles. Elle est en outre « élargie » si on la compare à une formulation de type « erreur d'équation ». Deux algorithmes différents ont été présentés ici :

- l'algorithme de Clément (FHCL) qui construit l'ellipsoïde en deux étapes, chacune de ces étapes faisant appel à un algorithme de type OBE ;
- l'algorithme de Ferreres et M'Saad (FM) permet de construire l'ellipsoïde englobant par une seule coupe (soit une seule étape).

Comme on l'aura noté, toutes ces approches sont séquentielles et donc sous optimales par rapport à une prise en compte simultanée de toutes les données. L'expérience a largement montré l'intérêt de « rejouer » les données plusieurs fois de suite en prenant comme condition initiale pour le passage suivant, les conditions finales atteintes au passage précédent. C'est la notion de « **recirculation** » que l'on exploitera largement dans les chapitres suivants.

L'évaluation et la comparaison des différents algorithmes OBE seront réalisées sur les résultats de l'identification dans le chapitre 2 avec des données simulées et des données réelles.

Chapitre 2
Application de l'algorithme OBE en identification

On a vu au cours du chapitre précédent qu'il existe plusieurs algorithmes OBE différents lorsque l'on effectue une estimation ensembliste ellipsoïdale. On a classé ces algorithmes en deux groupes principaux. Le premier comprend les algorithmes basés sur la réduction de la taille géométrique de l'ellipsoïde. Les algorithmes du second groupe s'intéressent quant à eux, à la convergence du coefficient σ_k^2 (voir formule 1.9). Ces algorithmes OBE sont construits pour un modèle à erreur d'équation mais on peut les étendre à un modèle de type erreur de sortie.

Pour évaluer et comparer ces différents algorithmes lors de leur utilisation en identification ensembliste ellipsoïdale, on se propose de les appliquer tout d'abord à des données simulées. Ainsi, on peut vérifier les propriétés de ces algorithmes mais également choisir d'une manière « optimale » les différents paramètres de réglage de la procédure d'identification. L'algorithme permettant d'obtenir la « meilleure » identification (au sens où l'ellipsoïde obtenu est de taille minimale) est ensuite appliqué à des données réelles issues respectivement d'un procédé industriel de l'Institut Français du Pétrole [HEI 03] (IFP) et d'un robot parallèle du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM).

2.1 Données simulées

2.1.1 Description des données

Le modèle retenu est du 1^{èr} ordre. Sa fonction du transfert est :

$$H(z) = \frac{Y(z)}{U(z)} = \frac{bz^{-1}}{1-az^{-1}} = \frac{0.05z^{-1}}{1-0.95z^{-1}} \tag{2.1}$$

Ce procédé a un temps de montée t_M = 2.2 T avec T = -t_e / ln$a$ et t_e est la période d'échantillonnage. L'entrée utilisée pour la simulation est une Séquence Binaire Pseudo Aléatoire (SBPA). Ses caractéristiques satisfont [LAN 93] :

$$\begin{cases} sN_{SBPA}t_e > t_M \\ l \geq s(2^{N_{SBPA}} - 1)t_e \\ T_{SBPA} = st_e \end{cases} \tag{2.2}$$

où N_{SBPA} est le nombre de digits du registre à décalage, l est la longueur de l'essai et T_{SBPA} est la période d'horloge de la SBPA. Pour notre étude, on a choisi $N_{SBPA} = 4$, $s = 20$, la longueur est donnée par $l = 20 \times (2^4\text{-}1) \times t_e = 300t_e$, ce qui permet de satisfaire la condition (2.2). L'amplitude de la SBPA est normalisée à 1.

La sortie simulée pour un modèle à erreur d'équation est créée par :

$$y_k = 0.95y_{k-1} + 0.05u_k + e_k \tag{2.3}$$

où l'erreur e_k est un bruit uniforme variant dans l'intervalle $\left[-\gamma^*;\gamma^*\right]$. Par la suite, ce jeu de données est nommé SIMU-ARX-i (« i » signifie un**I**forme, ceci permettra de distinguer ces jeux de données de ceux contenant un bruit **G**aussien, noté « g », dans le chapitre 4).

Pour un modèle à erreur de sortie, la sortie simulée est engendrée par l'équation :

$$\begin{cases} y_0(k) = 0.95y_0(k-1) + 0.05u_k \\ y_k = y_0(k) + \eta_k \end{cases} \tag{2.4}$$

où y_0 est la sortie théorique du modèle (sans bruit), η_k est un bruit uniforme dans l'intervalle $\left[-\eta_k^M;\eta_k^M\right]$. Ce jeu de données est nommé par la suite SIMU-OE-i.

En modifiant le niveau du bruit ajouté à la sortie, on obtient plusieurs jeux de données différents. Il va de soi que la borne du bruit aura une incidence directe sur la taille de l'ensemble estimé. Le niveau de bruit en sortie est évalué via le Rapport Signal sur Bruit (RSB) moyen défini par :

$$RSB \text{ (en dB)} = 20\log_{10}\frac{\sqrt{\sum_{k=1}^{l} y_k^2}}{\sqrt{\sum_{k=1}^{l} e_k^2}}$$

Pour les données simulées, on choisit $\gamma^* = \eta^{*M} = 0.003$ ce qui conduit à $RSB \approx 50$ dB pour les deux jeux de données SIMU-ARX-i et SIMU-OE-i.

2.1.2 Résultats obtenus avec un modèle à erreur d'équation

Nous réalisons l'identification sur le jeu de données SIMU-ARX-i. D'après (2.3), le vecteur des « vrais » paramètres à estimer est $x^* = \begin{bmatrix} 0.95 & 0.05 \end{bmatrix}^T$. Le régresseur à l'instant k est calculé à partir des observations précédentes $d_k = \begin{bmatrix} y_{k-1} & u_{k-1} \end{bmatrix}^T$. L'ellipsoïde initial est choisi volontairement très grand pour garantir qu'il contienne bien l'espace de paramètres admissible $\sigma_0^2 P_0 = 10^6 I_2$. Le centre de l'ellipsoïde initial est pris à l'origine des coordonnées, i.e. $c_0 = \mathbf{0}$.

Le Tableau 2.1 présente les résultats de la procédure d'identification ensembliste ellipsoïdale pour quatre algorithmes différents (à savoir, « FH » Fogel et Huang, « DH » Dasgupta et Huang, « TAN » Tan *et al.*, « LO » Lozano-Leal et Ortega) et après une seule circulation des données (ce qui signifie qu'on a utilisé une seule fois le jeu de données). a correspond au pôle du système tandis que b est le paramètre du numérateur. La borne sur l'erreur γ_k est choisie constante et égale à la vraie borne du bruit introduit dans la simulation $\gamma_k = \gamma = \gamma^* = 0.003, \forall k = 1:l$. Les algorithmes du groupe 1 conduisent au même ellipsoïde. C'est pour cette raison que nous avons choisi d'utiliser l'algorithme de Fogel et Huang (FH) pour représenter ce groupe. Pour les algorithmes DH et LO, le facteur υ (dans les formules 1.31 et 1.39) est choisi égal à 0.8. Le critère utilisé est μ_V (critère du déterminant). A partir de l'ellipsoïde trouvé, un intervalle sur chaque paramètre estimé est obtenu par la projection de l'ellipsoïde sur les axes paramétriques (coordonnées) correspondants (voir Figure 2.1). Il est calculé par :

$$x \in \left[x_{\min} ; x_{\max} \right] \text{ avec } \begin{cases} x_{\min} = c - \Delta \\ x_{\max} = c + \Delta \end{cases} \text{ et } \Delta = \sqrt{\operatorname{diag}(\sigma^2 P)} \tag{2.5}$$

Il va de soi que cette projection peut fausser l'analyse que l'on fait du domaine admissible puisque cela revient en fait à inscrire l'ellipsoïde trouvé dans un parallélotope, ce dernier pouvant être très surenglobant (cf. Figure 2.1).

Les résultats obtenus montrent que les intervalles estimés contiennent toujours les « vrais » paramètres du procédé. La taille de l'ellipsoïde (qui influe sur les intervalles trouvés) obtenu par les algorithmes du groupe 1 (ici FH) est très inférieure à celle de l'ellipsoïde obtenu avec les algorithmes du groupe 2 (DH, TAN, LO). Cette constatation était prévisible car l'objectif des algorithmes du groupe 2 n'est pas la réduction systématique de la taille de l'ellipsoïde,

mais la décroissance de σ_k^2. Néanmoins si P_0 est plus « raisonnable », la différence de résultats a tendance à se réduire significativement.

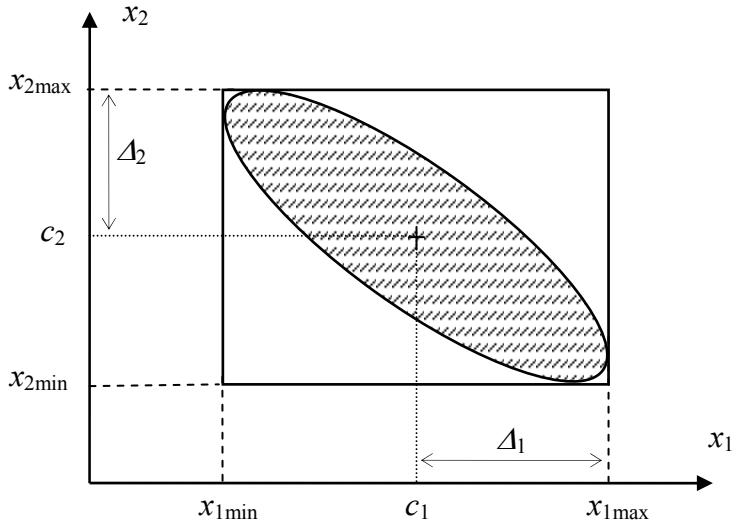


Figure 2.1 : Ellipsoïde et sa projection sur les axes paramétriques.

Algo.	Param.	[min ; max]	Centre de l'intervalle	Δ	μ_V	R(%)
FH	*a*	[0.9496 ; 0.9509]	0.9502	$6.51\ 10^{-4}$	$1.85\ 10^{-14}$	14.7
	b	[0.0497 ; 0.0502]	0.0499	$2.76\ 10^{-4}$		
DH	*a*	[0.9094 ; 0.9914]	0.9504	$4.10\ 10^{-2}$	$6.14\ 10^{-10}$	12.4
	b	[0.0380 ; 0.0618]	0.0499	$1.19\ 10^{-2}$		
TAN	*a*	[0.0509 ; 1.3912]	0.9504	$4.41\ 10^{-1}$	$1.04\ 10^{-3}$	8.7
	b	[-0.0499 ; 0.1498]	0.0499	$9.99\ 10^{-2}$		
LO	*a*	[-60.8 ; 62.7]	0.9495	61.79	$1.12\ 10^{6}$	14.3
	b	[-39.9 ; 40. 1]	0.0510	40.04		

Tableau 2.1: Résultats de l'identification pour les données SIMU-ARX-i

La Figure 2.2 montre l'évolution de μ_V. On peut constater que pour l'algorithme DH, le volume augmente ponctuellement pendant le processus d'identification, ce qui montre clairement que cet algorithme ne tente pas de diminuer (si possible) à chaque nouvelle donnée le volume de l'ellipsoïde calculé. Cette situation apparaît lorsque le coefficient de pondération λ_k prend la valeur υ choisie *a priori* par l'utilisateur (et non la valeur ξ_k calculée par l'algorithme).

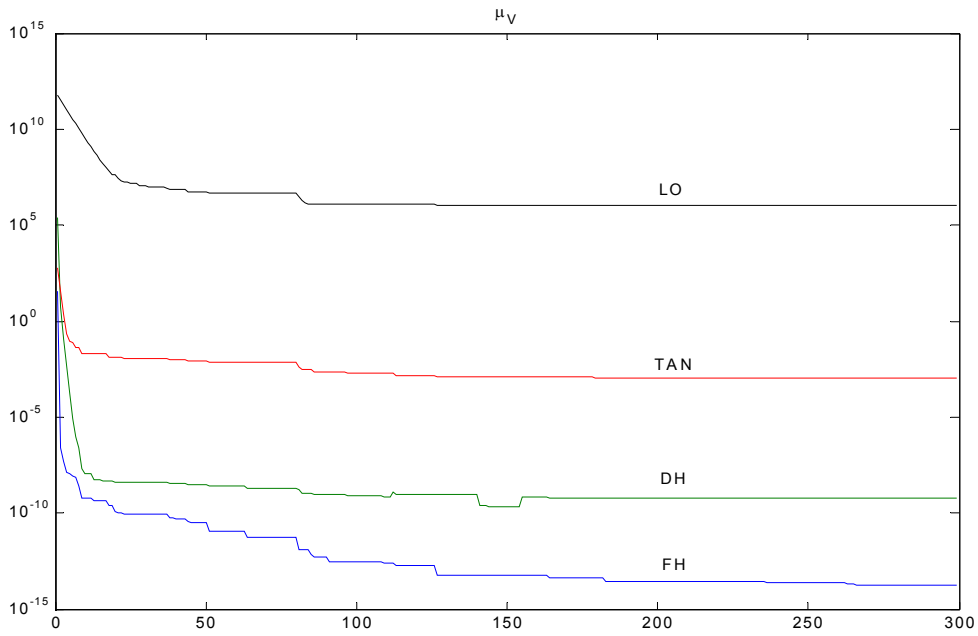


Figure 2.2 : Evolution du volume de l'ellipsoïde lors de la 1ère circulation des données.

La Figure 2.3 reporte l'évolution du coefficient σ_k^2 pour les quatre algorithmes considérés. Il est clair que l'objectif de l'algorithme FH n'est pas une décroissance (lorsqu'elle est possible) de σ_k^2 alors que cette décroissance est garantie par les trois algorithmes du groupe 2. Rappelons que le volume de l'ellipsoïde est donné par $\mu_V(k) = \det\left(\sigma_k^2 P_k\right)$.

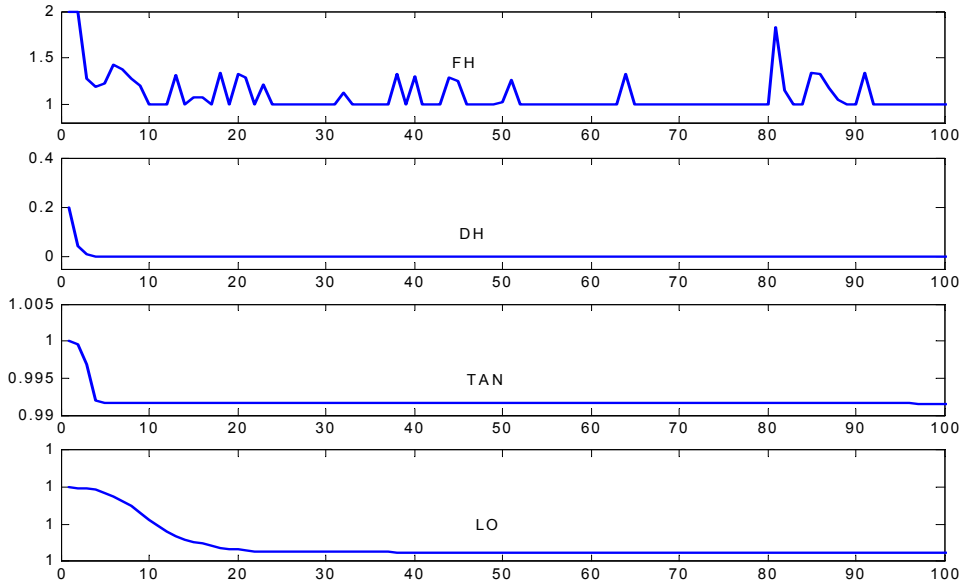


Figure 2.3 : Evolution de σ_k^2 lors de la 1ère circulation des données.

On définit le taux de données utilisées (c'est-à-dire apportant de l'information et conduisant à une modification de l'ellipsoïde) R par :

$$R = 100 * \frac{\text{Nombre d'observations utilisées}}{\text{Longueur du vecteur de données } (l)} \quad [\%]$$

On voit dans le Tableau 2.1 que ce taux avec l'algorithme TAN est plus faible qu'avec les autres algorithmes. En fait, on s'aperçoit qu'après quelques pas de l'algorithme, le centre de l'ellipsoïde est très proche des « vrais » paramètres. L'erreur de prédiction ν_k satisfait alors $\nu_k^2 \leq \gamma^2$ ce qui conduit à un arrêt prématuré de l'algorithme. Pour l'algorithme LO, cette condition est modifiée par $\nu_k^2 \leq \gamma^2(1+\upsilon)$. Le taux R dépend donc du facteur υ choisi par l'utilisateur.

L'évolution des coordonnées du centre de l'ellipsoïde obtenu pour la 1ère circulation des données est présentée dans la Figure 2.5 pour les différents algorithmes considérés. Le pôle a est représenté dans la partie haute de la figure tandis que l'évolution du paramètre du numérateur b est tracée sur la partie basse de la figure.

En fait, dans l'approche ensembliste, le centre de l'ellipsoïde est l'un des vecteurs de paramètres admissibles. Néanmoins, son analyse permet de comprendre la convergence de la procédure d'identification vers le domaine admissible « optimal », compte tenu des données disponibles, de la borne de bruit choisie, et de l'aspect séquentiel de l'algorithme. La figure 2.4 montre que pour les algorithmes TAN et LO du groupe 2, le centre de l'ellipsoïde n'oscille pas et converge rapidement vers le vecteur de paramètres x^* théoriques ayant servis à simuler les données (et qui est donc connu). La condition sur l'erreur de prédiction ν_k avant de construire le nouvel ellipsoïde permet de garantir cette convergence sans oscillation. Autrement dit, dans ces deux algorithmes, l'ellipsoïde précédent est conservé si la bande contient le centre de l'ellipsoïde ; il est modifié dans le cas contraire. Ces algorithmes sont donc capables de poursuivre des variations de paramètre. En revanche, l'algorithme FH du premier groupe voit le centre de l'ellipsoïde « bouger » assez fortement car l'objectif poursuivi ici est la réduction de la taille de l'ellipsoïde sans tenir compte de la position du centre de l'ellipsoïde précédemment calculé.

En conséquence, si le but poursuivi est de rechercher un espace de paramètres admissible le plus petit possible, les algorithmes du premier groupe sont évidemment mieux adaptés.

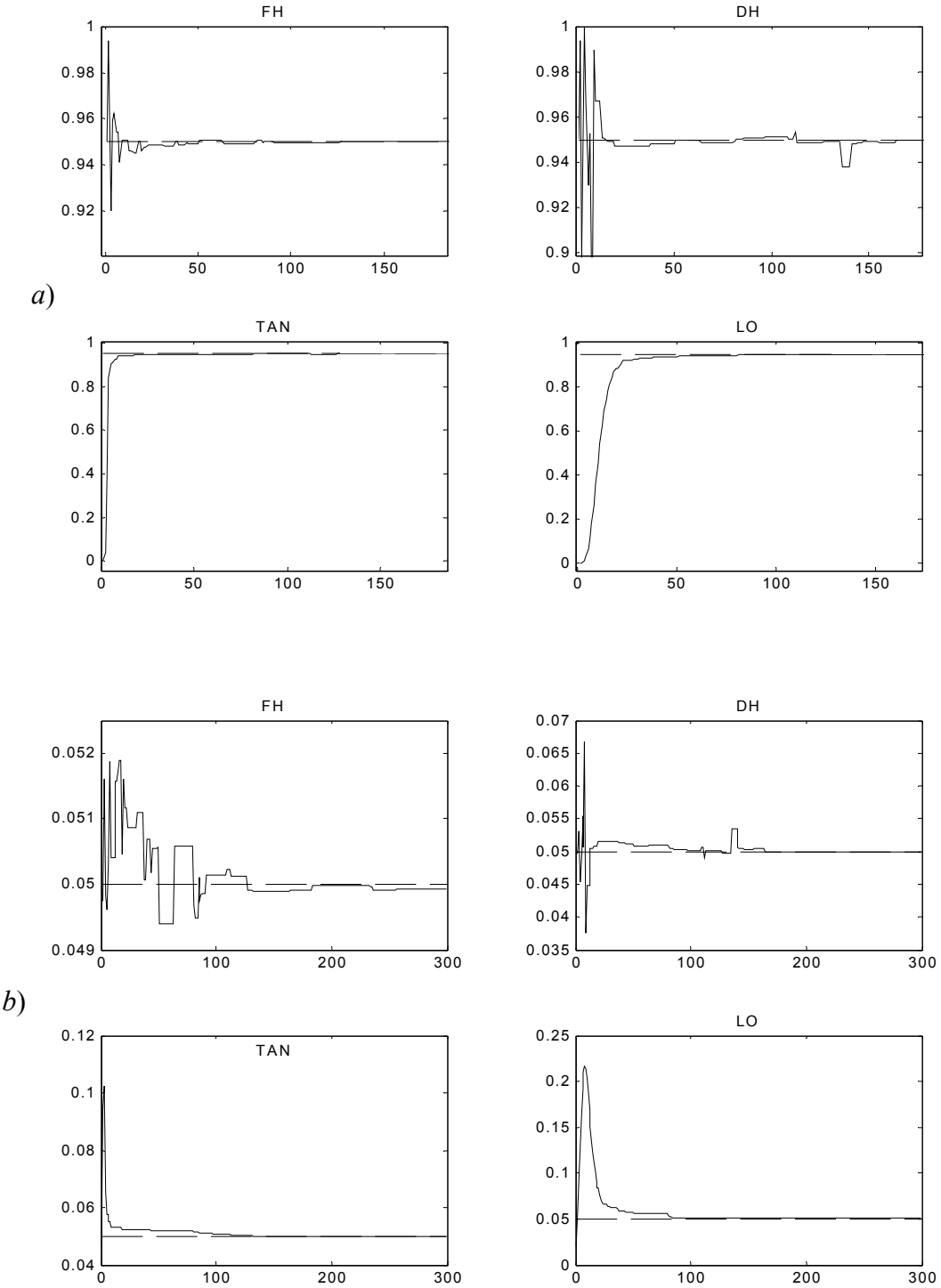


Figure 2.4 : Evolution des paramètres estimés.

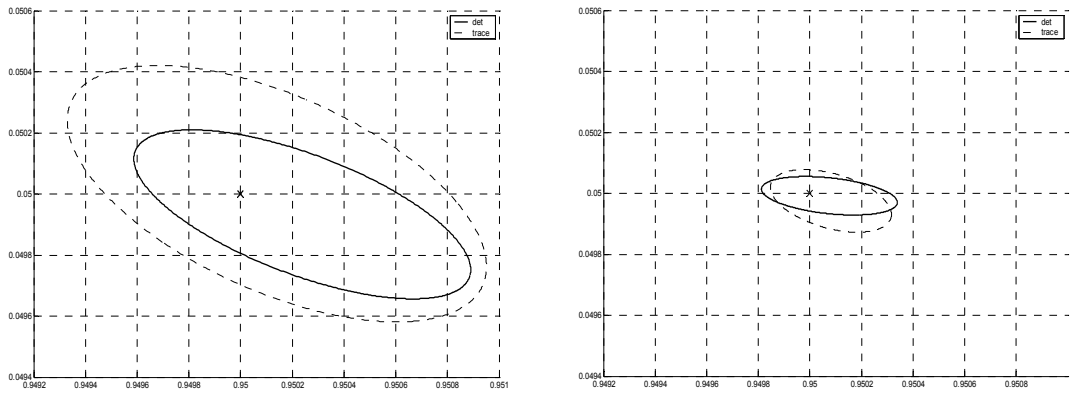
Dans la suite de ce travail, on s'attachera à calculer un espace admissible ellipsoïdal le plus petit possible, ce qui nous conduit donc à nous intéresser aux algorithmes du premier groupe qui cherchent à diminuer la taille de cet espace de paramètre. Rappelons que pour chaque algorithme de ce groupe, deux critères différents (critère du déterminant μ_V et critère

de la trace μ_T) permettent d'évaluer la taille de l'ellipsoïde. La comparaison de ces deux critères pour les données simulées considérées est maintenant présentée dans la figure 2.5 et le tableau 2.2.

Critère	**Param.**	**[min ; max]**	**Centre**	Δ	**det($\sigma^2 P$)**	**tr($\sigma^2 P$)**	***R*(%)**
μ_V	*a*	[0.9496 ; 0.9509]	0.9502	$6.51\ 10^{-4}$	$1.85\ 10^{-14}$	$5.01\ 10^{-7}$	14.7
	b	[0.0497 ; 0.0502]	0.0499	$2.76\ 10^{-4}$			
μ_T	*a*	[0.9493 ; 0.9509]	0.9501	$8.11\ 10^{-4}$	$7.81\ 10^{-14}$	$8.33\ 10^{-7}$	19.1
	b	[0.0496 ; 0.0504]	0.0500	$4.19\ 10^{-4}$			

Tableau 2.2 : Résultats de l'identification (algorithme FH) avec les deux critères considérés.

La figure 2.5 montre que la taille de l'ellipsoïde trouvé avec le critère du déterminant est plus petite bien que le nombre de données utilisées soit moindre (cf. la valeur du paramètre *R*(%) qui donne le pourcentage de mesures effectivement utilisées pour mettre à jour l'ellipsoïde). En revanche, le critère de la trace donne (généralement) un ellipsoïde plus « rond » dans le sens où ce dernier n'est pas exagérément allongé suivant l'un des axes de l'ellipsoïde. Cette constatation est logique puisque le déterminant est une mesure directe de la taille de l'ellipsoïde sans maîtriser sa forme alors que le critère de la trace tend à minimiser la somme des demi-axes au carré de l'ellipsoïde. Rappelons que le critère de la trace conduit à la résolution d'une équation du 3ème degré pour déterminer le coefficient de pondération permettant le calcul du nouvel ellipsoïde.



Pas de recirculation 20 recirculations

Figure 2.5 : Ellipsoïdes obtenus (algorithme FH) avec les deux critères.

2.1.3 Résultats obtenus avec un modèle à erreur de sortie

Comme on l'a déjà vu, l'estimation ensembliste ellipsoïdale peut être appliquée à un modèle à erreur de sortie. Dans cette partie, on utilise le jeu données SIMU-OE-i construit avec l'approche « erreur de sortie ».

On utilise tout d'abord un modèle à erreur d'équation. La borne utilisée est $\gamma_k = \gamma = \eta^{*M} = 0.003$. Rappelons que η^{*M} est la vraie borne de l'erreur de sortie introduite dans la simulation. On constate qu'après quelques calculs, l'intersection entre l'ellipsoïde et la bande devient vide, comme prévu dans section 1.3.1. Cette situation apparaît car la borne utilisée est plus petite que la borne induite par le modèle « erreur d'équation » : elle ne tient pas compte du terme $\sum_{i=1}^{n} a_i \eta_{k-i}$.

Le Tableau 2.3 présente les résultats obtenus avec le critère du déterminant. Pour le modèle à erreur d'équation, on prend cette fois $\gamma = 2\eta^{*M}$. Pour un modèle à erreur de sortie, on choisit la vraie borne du bruit introduit lors de la création des données, i.e. $\eta_k^M = \eta^{*M}, \forall k = 1:l$. Rappelons que :

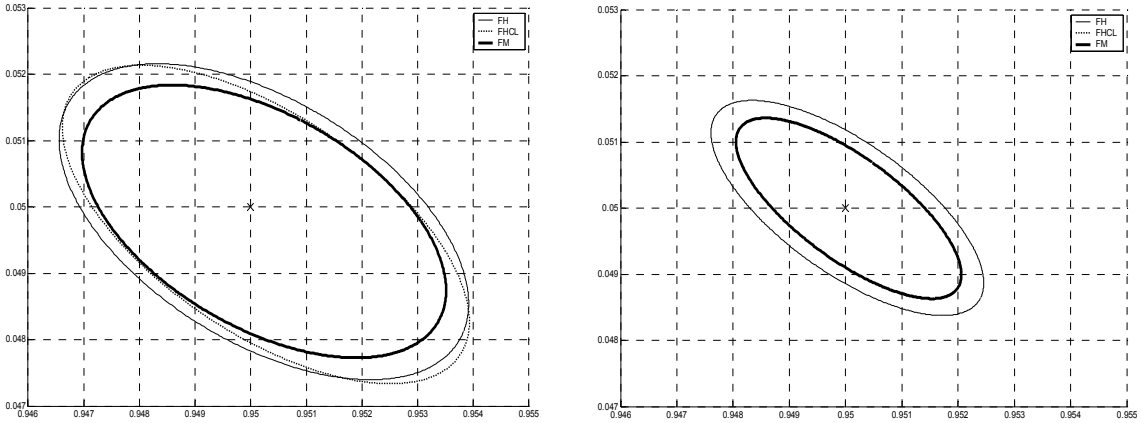
- « FH » correspond à l'algorithme de Fogel et Huang pour modèle à erreur d'équation, voir la section 1.2.1.1. ;
- « FHCL » correspond à l'algorithme de Clément pour modèle à erreur de sortie, voir la section 1.3.4. ;
- « FM » correspond à l'algorithme de Ferreres et M'Saad pour modèle à erreur de sortie, voir la section 1.3.5.

Les ellipsoïdes obtenus sans recirculation des données et avec 20 recirculations sont présentés dans la figure 2.6.

Algo.	**Borne**	**Param.**	**[min ; max]**	**Centre**	Δ	μ_V	***R*(%)**
FH	$\gamma = 2\eta^{*M}$	*a*	[0.9466 ; 0.9539]	0.9502	3.68 10^{-3}	5.59 10^{-11}	17.1
		b	[0.0474 ; 0.0522]	0.04978	2.38 10^{-3}		
FHCL	$\eta^M = \eta^{*M}$	*a*	[0.9466 ; 0.9539]	0.9503	3.66 10^{-3}	4.81 10^{-11}	22.7
		b	[0.0473 ; 0.0521]	0.0497	2.39 10^{-3}		
FM	$\eta^M = \eta^{*M}$	*a*	[0.9469 ; 0.9535]	0.9502	3.27 10^{-3}	3.36 10^{-11}	17.7
		b	[0.0477 ; 0.0518]	0.0498	2.05 10^{-3}		

Tableau 2.3 : Résultats de l'identification pour les données SIMU-OE-i (pas de recirculation).

La comparaison des résultats obtenus pour les deux modèles considérés est assez délicate (voir les tableaux 2.2 et 2.3). On note que pour le modèle à erreur d'équation, le centre de l'ellipsoïde obtenu ne coïncide pas avec les valeurs théoriques des paramètres. Notons néanmoins que les valeurs théoriques appartiennent bien à l'ensemble admissible obtenu. Les mêmes remarques peuvent être faites dans le cas du modèle à erreur de sortie. Notons d'autre part que la borne utilisée pour les deux modèles est différente et qu'on ne connaît pas (*a priori*) la borne exacte pour l'équation à erreur d'équation.



pas de recirculation 20 recirculations

Figure 2.6 : Ellipsoïdes pour les algorithmes FH, FHCL, FM.

On constate également que l'algorithme FM (Fererreres et M'Saad [FER 97]) donne un ellipsoïde un peu plus petit que celui de FHCL puisqu'il construit l'ellipsoïde avec une seule coupe (voir les paragraphes 1.3.4 et 1.3.5). Néanmoins, après quelques recirculations des données, les ellipsoïdes obtenus sont quasiment identiques (voir figure 2.6, dessin de droite). On voit donc ici l'intérêt de la recirculations des données (réutilisation des données pour améliorer le résultat obtenu, c'est-à-dire ici diminuer la taille de l'ellipsoïde) qui permet de diminuer l'influence de l'aspect séquentiel de l'algorithme employé.

2.2 Données industrielles de l'IFP

2.2.1 Description des données

L'entrée u et la sortie y des données industrielles de l'IFP sont présentées dans la figure 2.7. Il semble clair que l'entrée n'est pas bien excitée. Ceci prédit une difficulté pour l'estimation des paramètres. Notons que l'on n'a aucune information sur l'erreur attachée au

procédé, mais on sait que des saturations existent.

2.2.2 Résultats obtenus

Nous choisissons un modèle du 1er ordre, i.e. $y_k = ay_{k-1} + bu_{k-1}$. Nous allons tout d'abord utiliser un modèle de type erreur d'équation. Les résultats de l'identification ensembliste ellipsoïdale obtenus avec les critères du déterminant et de la trace sont présentés dans les tableaux 2.4 et 2.5. L'algorithme utilisé est celui de Fogel et Huang (voir paragraphe 1.2.1.1). On peut voir qu'après la 1ère circulation des données, les intervalles des paramètres obtenus sont grands. Ils contiennent même des valeurs n'ayant aucun sens physique (par exemple $a_{\max} > 1$, $b_{\min} < 0$). Néanmoins après 10 recirculations, ces intervalles sont réduits et deviennent exploitables. Le taux de données utilisées est cette fois très faible (2.4% pour le critère du déterminant et 1.8% pour le critère de la trace). L'ellipsoïde $\mathbf{E}_{k-1}$ « tombe » dans la bande $\mathbf{B}_k$ dans la plupart de cas. Les Tableaux 2.4 et 2.5 montrent aussi que les intervalles [min ; max] obtenus par les deux critères sont presque identiques.

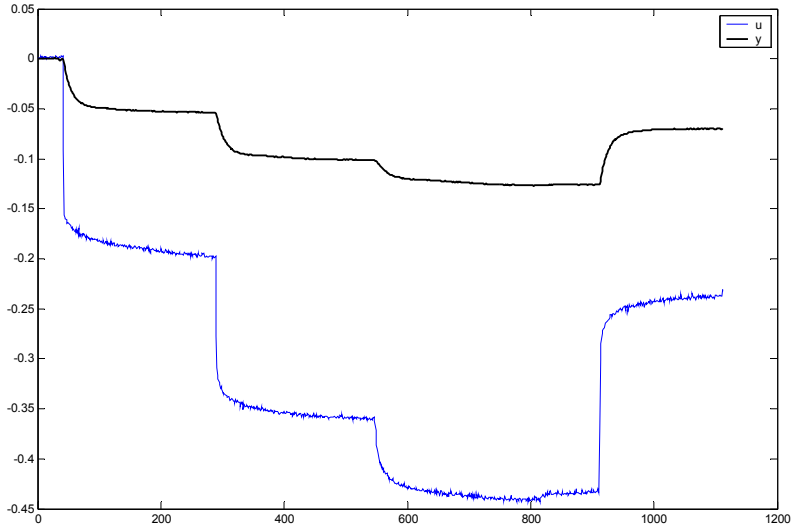


Figure 2.7 : Données industrielles de l'IFP.

Nb recirculations	param.	[min ; max]	centre	Δ	μ_V	R(%)
1	a	[0.7967 ; 1.0205]	0.9086	0.1119	4.85 10^{-7}	13.7
	b	[-0.0133 ; 0.0637]	0.0252	0.0385		
10	a	[0.8479 ; 0.9462]	0.8971	0.0491	1.08 10^{-7}	2.4
	b	[0.0145 ; 0.0441]	0.0293	0.0148		

Tableau 2.4 : Résultats de l'identification (critère du **déterminant**).

Nb recirculations	param.	[min ; max]	centre	Δ	μ_T	R(%)
1	*a*	[0.8016 ; 1.0189]	0.9103	0.1087	0.0133	12.8
	b	[-0.0135 ; 0.0628]	0.0246	0.0382		
10	*a*	[0.8559 ; 0.9408]	0.8984	0.0425	0.0019	1.8
	b	[0.0172 ; 0.0425]	0.0299	0.0127		

Tableau 2.5 : Résultats de l'identification (critère de la **trace**).

Pour l'estimation, on a utilisé une borne d'erreur estimée à partir d'une technique d'identification de type moindres carrés. Notons que cette méthode sera présentée dans le chapitre 4. Cette technique conduit à choisir $\gamma_k = \gamma = \max\left(\left|e_{mc}\right|\right) = 0.0022$. Remarquons qu'aucune intersection vide entre l'ellipsoïde $\mathbf{E}_{k\text{-}1}$ et la bande $\mathbf{B}_k$ n'apparaît ce qui signifie que cette borne est « acceptable » dans le sens où elle n'est pas sous-estimée. La figure 2.8 illustre l'évolution du volume de l'ellipsoïde μ_V pendant la 1ère circulation des données. On constate que ce volume décroît. Les paliers observés (sur lesquels il n'y a pas de diminution du volume) correspondent à des données pour lesquelles $\mathbf{E}_{k\text{-}1}$ est complètement contenu dans $\mathbf{B}_k$.

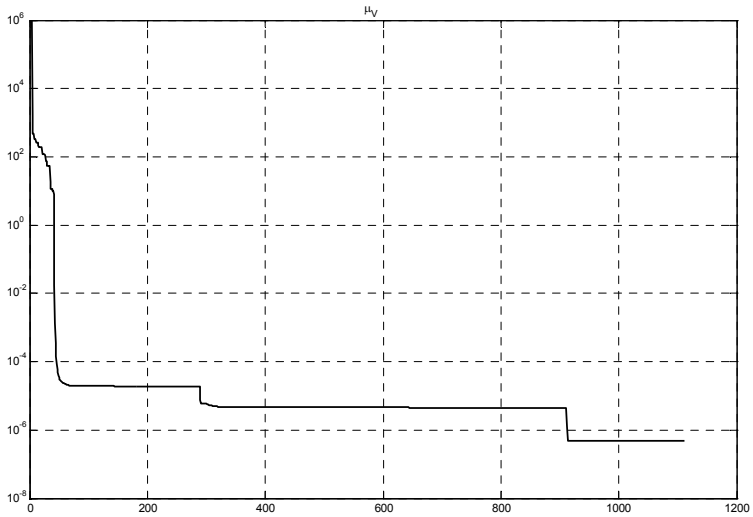


Figure 2.8 : Evolution du critère μ_V lors de la 1ère circulation.

En comparant avec les figures 2.7 et 2.8, on constate que la taille de l'ellipsoïde diminue uniquement lorsque l'entrée est correctement excitée : les paliers n'apportent pas d'information pour la réduction de la taille de l'ellipsoïde. En fait, une analyse graphique de la bande $\mathbf{B}_k$ montre que sa direction est quasiment inchangée lorsque l'on se situe sur les « paliers » des données d'entrée/sortie. Cela induit un ellipsoïde « allongé » (voir figure 2.9)

même lorsque le critère de la trace est utilisé. La projection de cet ellipsoïde sur les axes paramétriques conduit à un intervalle [min ; max] pessimiste : une analyse plus fine des résultats obtenus devrait tenir compte de l'inclinaison de l'ellipsoïde obtenu. Cependant, la recirculation avec le même jeu de données conduit à un meilleur résultat en terme d'intervalle [min ; max] et de taille de l'ellipsoïde. La figure 2.9 présente l'ellipsoïde trouvé après 1 et 10 circulations des données. Notons néanmoins que le rapport entre la longueur des axes peut atteindre un facteur 20.

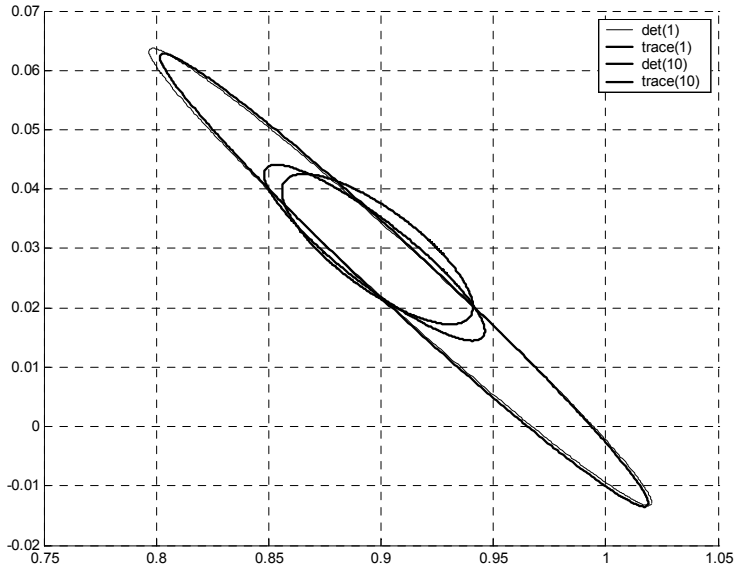
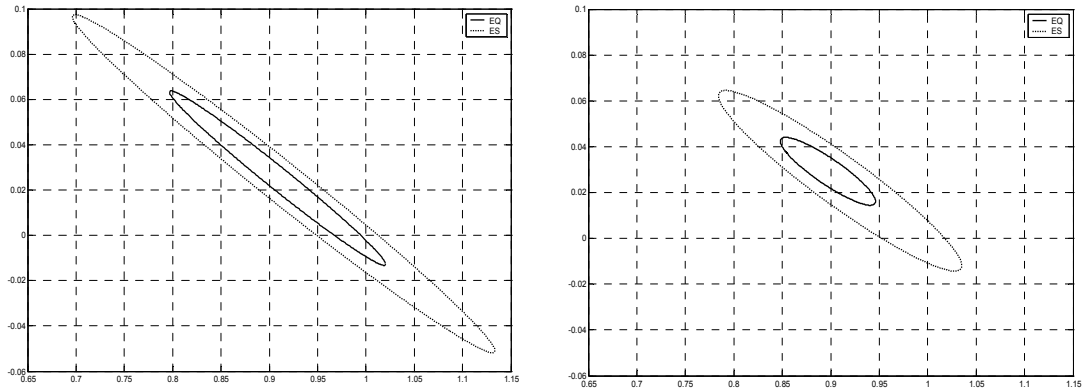


Figure 2.9 : Ellipsoïdes trouvés par les deux critères.

On utilise ensuite pour cette identification un modèle à erreur de sortie. Evidement, l'utilisation de la borne précédente $\gamma_k = \gamma = \max\left(\left|e_{mc}\right|\right) = 0.0022$ conduit à un ellipsoïde plus grand que celui obtenu avec le modèle à erreur d'équation (voir figure 2.10). La raison en est fort simple puisque la bande $\mathbf{B}_k$ est élargie et qu'on a maintenant deux hyperplans non parallèles. L'algorithme utilisé est celui de FM (Ferreres et M'Saad, voir paragraphe 1.3.5).

Dans un souci de comparaison des résultats obtenus avec les modèles « erreur d'équation » (algorithme FH, paragraphe 1.2.1.1) et « erreur de sortie » (algorithme FM, paragraphe 1.3.5) respectivement notés EQ et ES dans le tableau 2.6, la borne sera réglée manuellement (et heuristiquement puisque l'on n'a pas d'information a priori sur les erreurs présentes). Les paramètres obtenus sont présentés dans le tableau 2.6. (X) correspond à une intersection vide entre un ellipsoïde et une bande, ce qui conduit à l'arrêt de l'algorithme considéré.



pas de recirculation 10 recirculations

Figure 2.10 : Ellipsoïdes obtenus pour les algorithmes FH et FM ($\gamma = \eta^M = \max\left(\left|e_{mc}\right|\right)$).

Borne	Modèle	1 circulation	10 recirculations des données		
		μ_V	paramètre *a* [min ; max]	paramètre *b* [min ; max]	μ_V
0.0018	EQ	$1.93\ 10^{-7}$	[0.8700 ; 0.9180]	[0.0236 ; 0.0371]	$1.01\ 10^{-8}$
	ES	$3.37\ 10^{-6}$	[0.8000 ; 1.0169]	[-0.0084 ; 0.0596]	$1.52\ 10^{-6}$
0.0012	EQ	$2.42\ 10^{-8}$	(X)	(X)	(X)
	ES	$6.03\ 10^{-7}$	[0.8464 ; 0.9497]	[0.0131 ; 0.0449]	$1.36\ 10^{-7}$
0.0008	EQ	(X)	(X)	(X)	(X)
	ES	$9.50\ 10^{-8}$	[0.8816 ; 0.9030]	[0.0299 ; 0.0322]	$1.20\ 10^{-10}$

Tableau 2.6 : Comparaison des résultats pour les modèles EQ et ES.

Le choix heuristique de la borne permet d'obtenir des résultats plus satisfaisants (en terme de taille de l'ellipsoïde et de sa projection sur les axes de l'espace paramétrique) pour le modèle à erreur de sortie que ceux obtenus avec la borne précédemment prédéterminée $\gamma_k = \gamma = \max\left(\left|e_{mc}\right|\right) = 0.0022$. Notons que ce choix manuel conduit pour le modèle EQ à des intersections vides lorsque la borne retenue n'est plus assez « grande » pour garantir une intersection non vide entre l'ellipsoïde courant et la bande d'espace considérée. Néanmoins, d'un point de vue industriel, un choix heuristique de la borne ne peut être acceptable que s'il est basé sur des connaissances a priori des erreurs présentes sur les mesures d'entrée/sortie. Ces connaissances devraient tenir compte par exemple, des biais et dérives des capteurs, des non linéarités connues, etc.

Rappelons que les données industrielles IFP ne sont pas correctement excitées : on a

constaté que la direction de la bande $\mathbf{B}_k$ dans l'espace de paramètres est quasiment inchangée d'un instant d'échantillonnage à l'autre. Il s'en suit l'obtention d'un ellipsoïde « allongé », quel que soit le critère utilisé. L'espace admissible de paramètres obtenu dépend donc fortement à la fois de la borne choisie, du modèle utilisé et de l'excitation de l'entrée.

2.3 Données industrielles du robot parallèle (LIRMM)

2.3.1 Description des données

Le robot parallèle considéré appartient à la famille H4. Il a été réalisé au LIRMM[1] et la figure 2.11 en donne une photographie. Il comprend 4 moteurs, les avant-bras et la nacelle équipée éventuellement d'une pince. Le prototype peut atteindre 10g d'accélération et une vitesse supérieure de 5m/s au niveau de la nacelle.

Figure 2.11 : Robot parallèle H4 du LIRMM

Le modèle du robot a été établi à partir de considérations physiques. [VIV 03] suppose que l'inertie des avant-bras est légitimement considérée comme une partie de l'inertie du moteur et que les effets des bras (fabriqués en carbone) sont négligés.

Posons $\boldsymbol{\Gamma}_{mot} \in \mathbb{R}^4$ le vecteur des couples actionneurs. Les équations de la dynamique s'écrivent :

$$\boldsymbol{\Gamma}_{mot} = I_{mot}\ddot{q} + J^T M\left(\ddot{\theta} - G\right) + F_v\dot{q} + F_s \operatorname{sign}(\dot{q}) \tag{2.6}$$

où « sign(x) » correspond à la fonction signe de la variable x. $G = \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix}^T$ où g, l'accélération de pesanteur, est un vecteur constant. Les différentes matrices constantes

[1] L'auteur souhaite remercier Messieurs Philippe Poignet et Andrès Vivas qui ont fourni ce bref descriptif du robot ainsi que les données qui ont permis la validation sur un problème de grande dimension (14 paramètres) des techniques proposées dans ce travail.

intervenant dans le modèle sont respectivement la matrice des inerties, la matrice liée à la nacelle et les différents coefficients de frottement :

$$I_{mot} = \begin{bmatrix} I_{mot1} & 0 & 0 & 0 \\ 0 & I_{mot2} & 0 & 0 \\ 0 & 0 & I_{mot3} & 0 \\ 0 & 0 & 0 & I_{mot4} \end{bmatrix} ; \qquad M = \begin{bmatrix} M_{nac} & 0 & 0 & 0 \\ 0 & M_{nac} & 0 & 0 \\ 0 & 0 & M_{nac} & 0 \\ 0 & 0 & 0 & I_{nac} \end{bmatrix}$$

$$F_v = \begin{bmatrix} F_{v1} & 0 & 0 & 0 \\ 0 & F_{v2} & 0 & 0 \\ 0 & 0 & F_{v3} & 0 \\ 0 & 0 & 0 & F_{v4} \end{bmatrix} ; \qquad F_s = \begin{bmatrix} F_{s1} & 0 & 0 & 0 \\ 0 & F_{s2} & 0 & 0 \\ 0 & 0 & F_{s3} & 0 \\ 0 & 0 & 0 & F_{s4} \end{bmatrix}$$

où :

- $I_{mot\,\mathrm{i}}$ (i = 1 : 4) est l'inertie du $i^{\text{ème}}$ moteur. Elle inclut l'inertie des avant-bras ;
- M_{nac} est la masse de la nacelle et I_{nac} est son inertie.

Les variables dans l'équation (2.6) sont :

- $\dot{q} \in \mathbb{R}^4$ le vecteur des vitesses articulaires ;
- $\ddot{q} \in \mathbb{R}^4$ le vecteur des accélérations articulaires ;
- $\ddot{\theta} = \begin{bmatrix} \ddot{\theta}_x & \ddot{\theta}_y & \ddot{\theta}_z & \ddot{\theta}_\omega \end{bmatrix}^T$ le vecteur des accélérations cartésiennes ;

Notons que $J = J(\theta, q) \in \mathbb{R}^4$ est la matrice Jacobienne.

En posant $J^T = [J_{43}\ J_4]$ où J_{43} correspond aux trois premières colonnes de J^T et J_4 à la dernière et en développant (2.6), le modèle dynamique du robot peut s'écrire sous la forme d'une relation linéaire en les paramètres :

$$\begin{bmatrix} \Gamma_{mot1} \\ \Gamma_{mot2} \\ \Gamma_{mot3} \\ \Gamma_{mot4} \end{bmatrix} = \left[\begin{bmatrix} \ddot{q}_1 & & & 0 \\ & \ddot{q}_2 & & \\ & & \ddot{q}_3 & \\ 0 & & & \ddot{q}_4 \end{bmatrix} \quad J_{43} \begin{bmatrix} \ddot{\theta}_x \\ \ddot{\theta}_y \\ \ddot{\theta}_z - g \end{bmatrix} \quad J_4 \ddot{\theta}_\omega \quad \begin{bmatrix} \dot{q}_1 & & & 0 \\ & \dot{q}_2 & & \\ & & \dot{q}_3 & \\ 0 & & & \dot{q}_4 \end{bmatrix} \quad \mathrm{sign} \begin{bmatrix} \dot{q}_1 & & & 0 \\ & \dot{q}_2 & & \\ & & \dot{q}_3 & \\ 0 & & & \dot{q}_4 \end{bmatrix} \right] x \qquad (2.7)$$

avec

$$x = [I_{mot1}\ I_{mot2}\ I_{mot3}\ I_{mot4}\ M_{nac}\ I_{nac}\ F_{v1}\ F_{v2}\ F_{v3}\ F_{v4}\ F_{s1}\ F_{s2}\ F_{s3}\ F_{s4}]^T \in \mathbb{R}^{14} \qquad (2.8)$$

La relation (2.7) s'écrit également :

$$\Gamma_{mot} = Cx \quad \in \mathbb{R}^4$$

avec $C \in \mathbb{R}^{4\mathrm{x}14}$ la matrice des observations (le régresseur) et $x \in \mathbb{R}^{14}$ le vecteur des 14 paramètres physiques à estimer. Dans les équations (2.7), seuls les couples moteurs $\Gamma_{mot\,\mathrm{i}}$ et

les positions articulaires q_i sont directement mesurés. Les valeurs des accélérations $\ddot{\theta}$ ne sont pas disponibles. Elles sont calculées par :

$$\ddot{\theta} = J\ddot{q} + \dot{J}\dot{q}$$

où $\dot{J}$ est la dérivée par rapport au temps de J. Cette dérivée est calculée par un algorithme de différence centrée.

L'équation (2.7) représente donc les relations entre des variables mesurées, des variables déduites des mesures et les paramètres du modèle. Pour alléger les notations, on omet l'indice temporel. Les $4N$ observations sont enregistrées consécutivement puis réorganisées par bloc. Le bloc 1 correspond aux observations pour le moteur 1, le bloc 2 pour le moteur 2 et ainsi de suite. Ainsi, le modèle du robot devient :

$$Y = Cx + e \tag{2.9}$$

avec :

$$Y = \begin{bmatrix} \begin{bmatrix} \Gamma_{mot1}(1) \\ \dots \\ \Gamma_{mot1}(N) \end{bmatrix} \\ \dots \\ \begin{bmatrix} \Gamma_{mot4}(1) \\ \dots \\ \Gamma_{mot4}(N) \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{4*N} \text{ le vecteur des couples mesurés,}$$

$C \in \mathbb{R}^{(4*N)\text{x}14}$ la matrice d'observation ;

$e \in \mathbb{R}^{4*14}$ le vecteur d'erreurs supposées additives sur la sortie du modèle.

Chaque vecteur ligne de la matrice C correspond au régresseur du modèle général dans (1.1) tandis que le vecteur colonne (par exemple colonne i, i =1 : 14) contient les $i^{\text{ème}}$ éléments de tous les régresseurs. Notons que le $i^{\text{ème}}$ élément d'un régresseur est attaché au $i^{\text{ème}}$ paramètre à estimer. La figure (2.12) montre la sortie (les couples mesurés du robot) et quelques colonnes de la matrice C pour le modèle (2.9) du robot.

2.3.2 Résultats obtenus

Le robot parallèle conduit à un modèle linéaire par rapport aux 14 paramètres définis dans l'équation (2.9) :

$$Y = Cx + e$$

où la matrice C contient des termes mesurés ou calculés (voir les accélérations $\ddot{\theta}$). L'erreur e contient le bruit de mesure, l'erreur de modélisation et les erreurs de calcul. L'analyse de la colonne 5 de la figure 2.12 montre qu'il existe des points aberrants. Si l'on conserve toutes les mesures (sans analyse préalable des données pour en extraire les points aberrants), la borne e sera pessimiste.

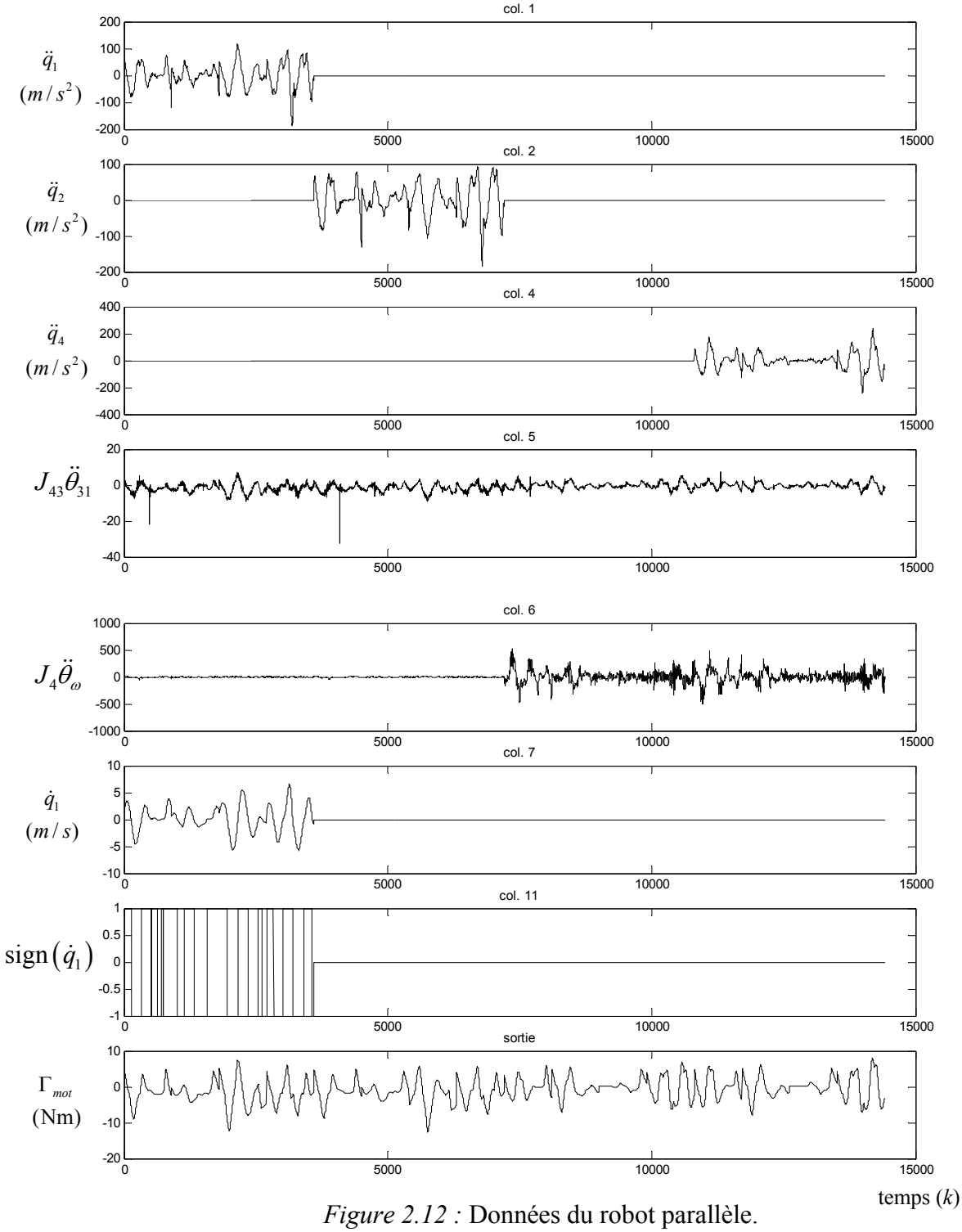


Figure 2.12 : Données du robot parallèle.

Pour ces raisons, on propose de choisir la borne de manière heuristique. On utilise tout d'abord une borne estimée à partir d'une technique d'identification de type moindres carrés. On obtient alors γ_0 = 29.9Nm. Rappelons que cette méthode sera présentée dans le chapitre 4. On réduit ensuite la borne jusqu'à l'obtention d'une intersection vide entre une bande et un ellipsoïde. Cette démarche heuristique conduit à choisir γ = 6Nm. Le tableau 2.7 présente les résultats de l'identification ensembliste ellipsoïdale après 60 recirculations[2] des données. A partir de 60 recirculations, la taille de l'ellipsoïde ne diminue plus. Le caractère « X » correspond à un manque d'information a priori sur la valeur du paramètre à estimer. L'algorithme utilisé est celui de FH (Fogel et Huang) présenté au paragraphe 1.2.1.1.

Paramètre	**[min ; max]**	**Centre**	**Δ(%)**	***a priori***
I_{mot1}	[-0.0251 ; 0.1158]	0.0454	155.2	0.012
I_{mot2}	[0.0418 ; 0.0580]	0.0499	16.28	0.012
I_{mot3}	[-0.0504 ; 0.0998]	0.0247	304.1	0.012
I_{mot4}	[-0.0469 ; 0.1035]	0.0283	266. 2	0.012
M_{nac}	[0.1395 ; 0.2893]	0.2144	34.92	1.0
I_{nac}	[-0.03613 ; 0.03524]	-0.00045	799.2	0.0008
F_{v1}	[-2.9720 ; 3.7908]	0.40939	825.9	X
F_{v2}	[-0.2689 ; 1.2673]	0.4992	153.9	X
F_{v3}	[-2.8062 ; 2.9045]	0.0491	581.0	X
F_{v4}	[-2.6674 ; 2.7023]	0.0174	1539.1	X
F_{s1}	[-8.0750 ; 9.8545]	0.8898	1007.5	X
F_{s2}	[-1.2478 ; 1.4495]	0.1008	1337.5	X
F_{s3}	[-16.212 ; 16.868]	0.3282	5039.5	X
F_{s4}	[-14.437 ; 15.390]	0.47671	3128.4	X

Tableau 2.7 : Résultat de l'identification pour les données du robot parallèle (μ_V = 3.13 10^{-9}, t_CPU = 456s).

Δ(%) correspond à l'équation (2.5) (calcul en pourcentage par rapport au centre), « Centre » est le centre de l'ellipsoïde et [min ; max] est la projection de l'ellipsoïde sur les

[2] Rappelons que la technique de recirculation des données permet de palier artificiellement au manque d'efficacité inhérent à l'algorithme séquentiel utilisé.

différents axes de l'espace de paramètres. On constate que l'intervalle sur chaque paramètre n'est pas toujours cohérent puisqu'il contient par exemple des valeurs négatives pour des paramètres qui physiquement doivent être positifs. Une des principales raisons est que la borne utilisée pour l'identification ensembliste ellipsoïdale est surestimée. Notons également que le nombre de données disponibles est très grand ($4N$ = 14416) alors que le taux de données utilisées est faible puisque l'on a $R(\%)$ = 7.5% pour la première circulation des données et $R(\%)$ = 0.3% pour la 60ème circulation des données. Ce faible taux implique que les données prises dans l'ordre « bloc 1 – bloc 2 – bloc 3 – bloc 4 » pour une identification simultanée de tous les paramètres aboutit souvent à une situation pour laquelle l'ellipsoïde $\mathbf{E}_{k-1}$ est contenu dans la bande $\mathbf{B}_k$. Enfin, compte tenu de la dimension du problème et du nombre de données à traiter, le temps de calcul (t_CPU = 456s) est sans commune mesure avec celui constaté pour les données simulées (paragraphe 2.1) ou de l'IFP (paragraphe 2.2), ce qui dans un contexte industriel peut être rédhibitoire.

Reconsidérons maintenant les colonnes de la matrice C de la figure 2.12. En analysant les formules (2.7) et (2.9), on constate que les données du robot ont été réorganisées en quatre blocs consécutifs possédant chacun N lignes. Pour le 1er bloc, seules les colonnes numéros 1, 5, 6, 7 et 11 portent des informations, les autres colonnes restant à zéro. Cela signifie donc que pour ce premier bloc, certains paramètres ne sont pas sensibilisés. On peut alors obtenir un sous-modèle avec seulement cinq paramètres à estimer. Notons également que les colonnes correspondant aux paramètres x_1, x_7, x_{11} sont elles-mêmes à zéro pour les autres blocs. Ces trois paramètres sont donc estimés via les données du bloc 1. Il en est de même avec les données des blocs 2 à 4. Cependant, pour les paramètres communs à plusieurs blocs, il faut envisager une relation entre les intervalles trouvés lors d'une estimation de sous-modèles correspondant à un traitement par bloc des données. On réécrit maintenant le modèle du robot en quatre sous-modèles :

$$\begin{cases} Y_a = C_a x_a + e_a \\ Y_b = C_b x_b + e_b \\ Y_c = C_c x_c + e_c \\ Y_d = C_d x_d + e_d \end{cases} \tag{2.10}$$

avec :

$$\begin{cases} x_a = \begin{bmatrix} x_1 & x_5 & x_6 & x_7 & x_{11} \end{bmatrix}^T = \begin{bmatrix} I_{mot1} & M_{nac} & I_{nac} & F_{v1} & F_{s1} \end{bmatrix}^T \\ x_b = \begin{bmatrix} x_2 & x_5 & x_6 & x_8 & x_{12} \end{bmatrix}^T = \begin{bmatrix} I_{mot2} & M_{nac} & I_{nac} & F_{v2} & F_{s2} \end{bmatrix}^T \\ x_c = \begin{bmatrix} x_3 & x_5 & x_6 & x_9 & x_{13} \end{bmatrix}^T = \begin{bmatrix} I_{mot3} & M_{nac} & I_{nac} & F_{v3} & F_{s3} \end{bmatrix}^T \\ x_d = \begin{bmatrix} x_4 & x_5 & x_6 & x_{10} & x_{14} \end{bmatrix}^T = \begin{bmatrix} I_{mot4} & M_{nac} & I_{nac} & F_{v4} & F_{s4} \end{bmatrix}^T \end{cases} \tag{2.11}$$

et

- pour le premier bloc, $k = 1 : N$ et C_a contient les colonnes 1, 5, 6, 7, 11 ;
- pour le deuxième bloc, $k = \mathrm{N} + 1 : 2N$ et C_b contient les colonnes 2, 5, 6, 8, 12 ;
- pour le troisième bloc, $k = 2N + 1 : 3N$ et C_c contient les colonnes 3, 5, 6, 9,13 ;
- pour le quatrième bloc, $k = 3N + 1 : 4N$ et C_d contient les colonnes 4, 5, 6, 10, 14.

L'identification des paramètres en traitant les données par bloc (et en utilisant des sous-modèles) est donc beaucoup plus facile, la taille du problème à traiter ayant fortement diminuée (on est passé de 14 à 5 paramètres pour chacun des sous-modèles). Les résultats obtenus sont présentés dans le Tableau 2.8 après 60 circulations des données. La borne utilisée est identique pour chaque bloc, $\gamma_a = \gamma_b = \gamma_c = \gamma_d = \gamma = 6\mathrm{Nm}$. Elle est également égale à la borne utilisée lors de l'identification simultanée des 14 paramètres présentée ci-avant. L'algorithme retenu ici est celui de Fogel et Huang (voir paragraphe 1.2.1.1).

Paramètre	**[min ; max]**	**Centre**	**Δ(%)**	***a priori***
I_{mot1}	[0.0063 ; 0.0805]	0.0434	85.4	0.012
M_{nac}	[-0.1299 ; 0.6809]	0.2755	147. 2	1.0
I_{nac}	[-0.1994 ; 0.2864]	0.0435	558.4	0.0008
F_{v1}	[-1.6799 ; 2.3432]	0.3316	606.6	X
F_{s1}	[-2.9494 ; 5.1822]	1.1164	364. 2	X

Tableau 2.8.a : Résultats de l'identification sur le bloc 1 (t_CPU = 39s)

On constate que les intervalles [min ; max] obtenus pour les paramètres n'apparaissant que dans un bloc sont nettement plus petits que ceux obtenus lors du traitement simultané (voir Tableau 2.7). Pour les paramètres communs (x_5, x_6), les intervalles trouvés sont plus « larges », mais on peut s'intéresser à leur intersection puisque ces paramètres recherchés pour le système physique complet doivent satisfaire simultanément les différents sous-modèles considérés. Cela conduit à :

$$[x_{5\min} ; x_{5\max}] = [0.1783 ; 0.2654], [x_{6\min} ; x_{6\max}] = [-0.0231 ; 0.0255].$$

Bien que les intervalles [min ; max] obtenus lors de l'identification par bloc soient diminués, ils restent très grands et inexploitables d'un point de vue pratique. La principale raison est que la borne utilisée est trop lâche. On aurait pu choisir une borne différente lors de l'identification de chaque sous-modèle, ce qui peut conduire à des intervalles plus petits.

Paramètre	**[min ; max]**	**Centre**	**Δ(%)**	***a priori***
I_{mot2}	[0.0459 ; 0.0536]	0.0498	7.7	0.012
M_{nac}	[0.1783 ; 0.2654]	0.2218	19.6	1.0
I_{nac}	[-0.1517 ; 0.0707]	-0.0405	274.6	0.0008
F_{v2}	[0.1513 ; 0.8469]	0.4991	69.7	X
F_{s2}	[-0.5456 ; 0.8017]	0.1281	526.0	X

Tableau 2.8.b : Résultats de l'identification sur le bloc 2 (t_CPU = 39s)

Paramètre	**[min ; max]**	**Centre**	**Δ(%)**	***a priori***
I_{mot3}	[-0.0287 ; 0.0845]	0.0279	203.0	0.012
M_{nac}	[-1.6839 ; 2.1368]	0.2265	843.6	1.0
I_{nac}	[-0.0231 ; 0.0259]	0.0014	174.5	0.0008
F_{v3}	[-1.6858 ; 1.7145]	0.0144	11838	X
F_{s3}	[-7.8833 ; 9.3493]	0.7329	1175.5	X

Tableau 2.8.c : Résultats de l'identification sur le bloc 3 (t_CPU = 40s)

Paramètre	**[min ; max]**	**Centre**	**Δ(%)**	***a priori***
I_{mot4}	[-0.0237 ; 0.0871]	0.0317	174.6	0.012
M_{nac}	[-1.7673 ; 1.9843]	0.1085	1729.4	1.0
I_{nac}	[-0.0233 ; 0.0255]	0.0011	2237.5	0.0008
F_{v4}	[-1.6388 ; 1.4257]	-0.1066	1437.7	X
F_{s4}	[-6.1945 ; 9.2293]	1.5174	508.2	X

Tableau 2.8.d : Résultats de l'identification sur le bloc 4 (t_CPU = 39s)

En conclusion, l'exploitation des données par bloc conduit à une estimation des paramètres du robot parallèle plus rapide puisque les sous-modèles utilisés n'ont que cinq paramètres. Cette démarche pose de nouveau le problème du choix de la borne d'erreur, et ce, pour chaque bloc, ce qui peut être fastidieux, et de toute façon irréaliste dans un contexte

industriel lorsque ce choix est réalisé heuristiquement, voire « manuellement ».

2.4 Conclusion

Dans ce chapitre, pour évaluer et comparer les algorithmes OBE, nous avons d'abord réalisé l'identification ensembliste ellipsoïdale sur des données simulées. Le modèle retenu est du 1^{èr} ordre et un bruit uniforme a été rajouté. Les résultats obtenus montrent clairement les propriétés de chaque algorithme. Les algorithmes du groupe 1 sont ensuite retenus pour le travail réalisé dans cette thèse, l'objectif étant d'obtenir un ensemble admissible de paramètres aussi petit que possible.

On s'est ensuite intéressé à l'identification de paramètres à partir de données réelles fournies par l'IFP (2 paramètres à identifier) et issues d'un robot parallèle du LIRMM (14 paramètres à identifier, ce qui constitue un problème de grande dimension pour les approches ensemblistes). On a constaté que les données IFP ne sont pas correctement excitées, ce qui conduit à un ellipsoïde « allongé », quel que soit le critère considéré (trace ou déterminant). Pour les données du robot, on a constaté des données visiblement aberrantes dans les régresseurs. Si on n'y prend pas garde, cela conduit à une borne utilisée pour la procédure d'identification ensembliste ellipsoïdale très grande, le domaine admissible de paramètres devenant alors difficilement exploitable, compte tenu de sa taille. Nous avons proposé de traiter les données du robot en 4 blocs, ce qui implique de redéfinir le modèle complet en quatre sous-modèles ayant chacun des paramètres indépendants et des paramètres communs. L'identification par bloc montre ici de nombreux avantages comme par exemple, une réduction sensible du temps de calcul (on est passé de l'identification simultanée 14 à 5 paramètres), un ensemble admissible de paramètre plus cohérent avec la connaissance physique que l'on a du procédé, un libre choix de la borne pour chaque bloc de données…

Dans tous les cas (données simulées ou réelles), les résultats obtenus par les techniques d'identification ensembliste ellipsoïdale dépendent essentiellement de la borne choisie sur l'erreur. Le choix de cette borne est encore plus difficile pour des données réelles puisque l'on ne possède pas, en général, suffisamment d'information permettant d'éviter de faire un choix trop pessimiste, conduisant alors à un ensemble englobant surdimensionné et par là-même non exploitable. En outre, un réglage heuristique ou manuel de cette borne n'est pas scientifiquement satisfaisant car il est difficilement justifiable. Notons enfin, que ce point

délicat contribue certainement à freiner le transfert de ces techniques d'identification dans l'industrie.

Le problème du choix de la borne sur l'erreur sera abordé dans le chapitre 4 dans lequel on proposera une technique semi-automatique de choix de cette borne.

Chapitre 3

Approche factorisée dans l'estimation ensembliste ellipsoïdale

Les algorithmes OBE présentés dans le chapitre précédent constituent une approche d'estimation ensembliste réaliste car l'ensemble englobant calculé est simple à manipuler et à mettre à jour, le coût calcul est réduit si on le compare à d'autres approches ensemblistes. Cependant, en analysant les formules permettant la construction séquentielle de l'ensemble englobant, on constate un risque d'instabilité numérique pouvant conduire à un résultat qui n'a pas de sens physique.

Dans ce chapitre, on s'intéresse à la stabilisation numérique de l'algorithme OBE. Nous allons construire un algorithme OBE factorisé où la matrice caractéristique de l'ellipsoïde (matrice $P = M^{-1}$ qui doit être symétrique définie positive lorsque l'ellipsoïde n'est pas dégénéré) est décomposée par une factorisation de Cholesky. D'un point de vue numérique, cette forme factorisée garantira la stabilité numérique de l'algorithme.

On s'intéresse tout d'abord à l'algorithme particulier de type OBE proposé par Goldfarb et Todd [GOL 82]. On cherche à chaque instant un ellipsoïde qui englobe l'intersection entre l'ellipsoïde précédent et un demi-espace de paramètres délimité par un hyperplan. Pour garantir la stabilité numérique, cet algorithme a été donné sous forme factorisée en utilisant la technique de décomposition LDL^{T}.
On présente ensuite la forme générale de l'algorithme OBE factorisé.

3.1 Algorithme de Goldfarb et Todd

3.1.1 Formulation générale de l'algorithme

Le problème traité ici consiste à chercher un espace satisfaisant les contraintes :

$$\mathbf{S} = \left\{ x \in \mathbb{R}^p : d_k^T x \le b_k, \;\; k = 1:l \right\} \tag{3.1}$$

Dans l'espace des paramètres de dimension p, $\mathbf{S}$ est l'intersection de l demi-espaces $\mathbf{V}_k$,

chacun étant délimité par un hyperplan $\mathbf{H}_k$ avec :

$$\mathbf{H}_k = \left\{ x \in \mathbb{R}^p : d_k^T x = b_k \right\} \tag{3.2}$$

La Figure (3.1) illustre l'ensemble **S** pour l = 3 hyperplans dans un espace de p = 2 paramètres.

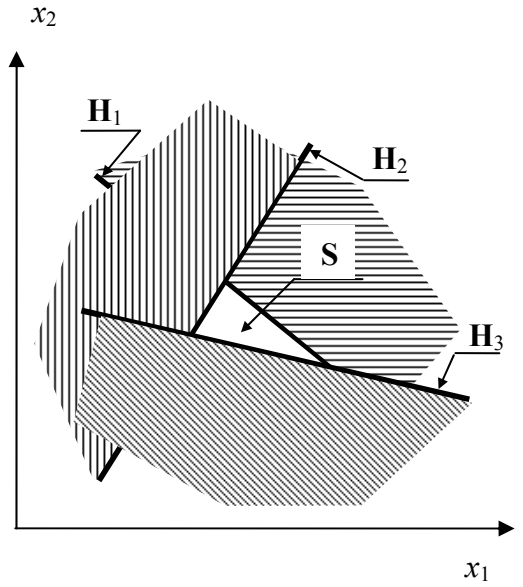


Figure 3.1

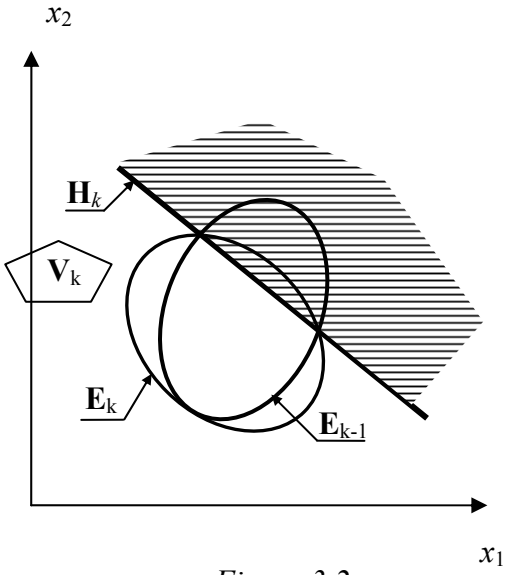


Figure 3.2

Une description directe de l'espace **S** est difficile car le nombre de sommets et la forme de l'ensemble dépendent des inéquations de contrainte et de leur nombre. Goldfarb et Todd [GOL 82] proposent de construire séquentiellement un ensemble englobant sous la forme d'un ellipsoïde (voir la figure 3.2). Ce problème s'avère en fait être un cas particulier de celui traité par l'algorithme OBE. En effet, pour la 1ère famille d'algorithmes OBE, et en utilisant le critère du déterminant, lorsque l'un des deux hyperplans définissant la bande est hors de l'ellipsoïde (il n'intersecte pas l'ellipsoïde), on est ramené au contexte de Goldfarb et Todd. L'ellipsoïde de volume minimum étant unique, les ellipsoïdes trouvés par les deux algorithmes dans ce cas sont identiques. L'algorithme de Goldfarb et Todd sera noté par la suite « GT ».

Considérons tout d'abord l'intersection entre une boule unité $\overline{\mathbf{E}}_{k\text{-}1}$, centrée en zéro, et un demi-espace $\overline{\mathbf{V}}_k$ avec :

$$\overline{\mathbf{E}}_{k-1} = \left\{ \overline{x} \in \mathbb{R}^p : \overline{x}^T \overline{x} \leq 1 \right\} \tag{3.3}$$

$$\overline{\mathbf{V}}_k = \left\{ \overline{x} \in \mathbb{R}^p : \overline{d}_k^T \overline{x} \leq \overline{b}_k \right\} \tag{3.4}$$

Supposons que $\overline{h}$ est la distance entre le centre de $\overline{\mathbf{E}}_{k-1}$ et l'hyperplan $\overline{\mathbf{H}}_k$ qui est la frontière de $\overline{\mathbf{V}}_k$. Cette distance est mesurée par :

$$\overline{h} = -\frac{\overline{b}}{\sqrt{\overline{d}_k^T \overline{d}_k}} \tag{3.5}$$

Notons que $\overline{h} < 0$ signifie que l'origine des coordonnées se situe dans le demi-espace $\overline{\mathbf{V}}_k$. Alors :

a) si $\overline{h} \leq$ -1: $\overline{\mathbf{E}}_{k\text{-}1} \subset \overline{\mathbf{V}}_k \Rightarrow \overline{\mathbf{E}}_k \supset \overline{\mathbf{E}}_{k\text{-}1} \cap \overline{\mathbf{V}}_k = \overline{\mathbf{E}}_{k\text{-}1}$

b) si $\overline{h} > 1$: $\overline{\mathbf{E}}_{k\text{-}1} \cap \overline{\mathbf{V}}_k = \varnothing$

c) si -1<$\overline{h} \leq 1$: $\overline{\mathbf{E}}_{k\text{-}1} \cap \overline{\mathbf{V}}_k \neq \varnothing$

D'après [GOL 82] et dans le cas c), l'ellipsoïde $\overline{\mathbf{E}}_k$ qui englobe l'intersection $\overline{\mathbf{E}}_{k\text{-}1} \cap \overline{\mathbf{V}}_k$ va passer par les intersections entre $\overline{\mathbf{E}}_{k-1}$ et $\overline{\mathbf{H}}_k$ et le point $\overline{M}\left(-\overline{d}_k / \sqrt{\overline{d}_k^T \overline{d}_k}\right)$. $\overline{M}$ est en fait le point tangent de l'ellipsoïde $\overline{\mathbf{E}}_{k-1}$ avec l'hyperplan se situant dans $\overline{\mathbf{V}}_k$ et parallèle à $\overline{\mathbf{H}}_k$ (voir Figure 3.3). $\overline{\mathbf{E}}_k$ est décrit par :

$$\overline{\mathbf{E}}_k = \left\{ \overline{x} \in \mathbb{R}^p : \left(\overline{x} - \overline{c}_k\right)^T \overline{P}_k^{-1} \left(\overline{x} - \overline{c}_k\right) \leq 1 \right\} \tag{3.6}$$

avec :

$$\begin{cases} \overline{c}_k = -\dfrac{\lambda_k \overline{d}_k}{\sqrt{\overline{d}_k^T \overline{d}_k}} \\ \overline{P}_k = \delta \left(\mathrm{I} - \kappa \dfrac{\overline{d}_k \overline{d}_k^T}{\overline{d}_k^T \overline{d}_k} \right) \\ \delta = \dfrac{\left(1\text{-}\lambda_k\right)^2 \left(1 + \overline{h}\right)}{\left(1 + \overline{h} - 2\lambda_k\right)} ; \kappa = \dfrac{2\lambda_k}{1 + \overline{h}} \end{cases} \tag{3.7}$$

Le centre $\overline{c}_k$ se situe sur une droite dont le vecteur directeur est défini par $\overline{d}_k$. $\overline{P}_k$ est une matrice symétrique, définie positive. L'ellipsoïde $\overline{\mathbf{E}}_k$ dépend du coefficient $\lambda_k \geq 0$ libre. L'algorithme GT cherche λ_k tel que le volume de l'ellipsoïde $\overline{\mathbf{E}}_k$ soit minimum

Le critère du déterminant μ_V est proportionnel au carré du volume de $\overline{\mathbf{E}}_k$. Il est évalué par :

$$\mu_V = \det(\overline{P}_k) = \delta^p \left(1 - \kappa\right) \tag{3.8}$$

En dérivant μ_V par rapport à λ_k, et en prenant $\left.\frac{d\mu_V}{d\lambda_k}\right|_{\lambda_k^*} = 0$, on obtient :

$$\lambda_k^* = \frac{1+p\bar{h}}{1+p} \tag{3.9}$$

Les paramètres intermédiaires intervenant dans (3.7) deviennent alors :

$$\delta = \frac{p^2\left(1-\bar{h}^2\right)}{p^2-1}; \quad \kappa = \frac{2\left(1+p\bar{h}\right)}{\left(1+p\right)\left(1+\bar{h}\right)} \tag{3.10}$$

Le coefficient λ_k^* est non négatif et satisfait l'équation (3.9). D'après [GOL 82], pour avoir un volume de l'ellipsoïde $\bar{\mathbf{E}}_k$ plus faible que celui de $\bar{\mathbf{E}}_{k-1}$, la distance $\bar{h}$ de l'origine **0** à l'hyperplan $\bar{\mathbf{H}}_k$ doit satisfaire $\bar{h} \geq -1/p$.

Réalisons maintenant une transformation linaire de l'espace $\bar{x} \to x$ par :

$$x = c_{k-1} + J\,\bar{x} \tag{3.11}$$

où J est une matrice de dimension $p \times p$, non singulière. $\bar{\mathbf{E}}_{k-1}$ dans l'espace x devient l'ellipsoïde (voir Figure 3.3) :

$$\mathbf{E}_{k-1} = \left\{x \in \mathbb{R}^p : \left(x - c_{k-1}\right)^T P_{k-1}^{-1}\left(x - c_{k-1}\right) \leq 1\right\} \tag{3 .12}$$

où $P_{k-1} = JJ^T$. Le demi-espace $\bar{\mathbf{V}}_k$ dans le système de coordonnées x s'écrit :

$$\mathbf{V}_k = \left\{x \in \mathbb{R}^p : d_k^T x \leq b_k\right\} \tag{3.13}$$

avec $d_k = J^{-T}\bar{d}_k$; $b_k = \bar{b}_k + d_k^T c_{k-1}$.

Avec cette transformation, on est revenu au problème général. L'ensemble **S** est contenu dans l'ellipsoïde $\mathbf{E}_k$ calculé séquentiellement, $\mathbf{E}_k \supset \mathbf{E}_{k-1} \cap \mathbf{V}_k$. D'après [GOL 82], $\mathbf{E}_k$ est exactement l'image de $\bar{\mathbf{E}}_k$ (3.6) dans espace x. Lors de la transformation, on multiplie le volume de tous les ellipsoïdes par la constante $|\det J|$. Les choix de P_k et c_k permettant de minimiser la taille de $\mathbf{E}_k$ doivent satisfaire $P_k = J\bar{P}_k J^T$ et $c_k = c_{k-1} + J\,\bar{c}_k$ respectivement. Goldfarb et Todd donnent le théorème 3.1 ci-après.

Théorème 3.1

Parmi tous les ellipsoïdes qui englobent l'intersection de $\mathbf{E}_k$ *et* $\mathbf{V}_k$ *définis dans (3 .12) et (3.13), celui de volume minimum est donné par :*

$$c_k = c_{k-1} - \lambda_k \frac{P_{k-1} d_k}{\sqrt{d_k^T P_{k-1} d_k}} \tag{3.14}$$

$$P_k = \delta \left(P_{k-1} - \kappa \frac{P_{k-1} d_k d_k^T P_{k-1}}{d_k^T P_{k-1} d_k} \right) \tag{3.15}$$

avec

$$\lambda_k = \frac{1 + p\bar{h}}{1+p} \; ; \; \delta = \frac{p^2 \left(1 - \bar{h}^2\right)}{p^2 - 1}; \quad \kappa = \frac{2\left(1 + p\bar{h}\right)}{\left(1+p\right)\left(1+\bar{h}\right)} \tag{3.16}$$

De plus, le rapport des volumes des ellipsoïdes $\mathbf{E}_k$ *sur* $\mathbf{E}_{k\text{-}1}$ est :

$$R_V = \frac{\left(\det P_k\right)^{1/2}}{\left(\det P_{k-1}\right)^{1/2}} = \left(\delta^n \left(1-\kappa\right)\right)^{1/2} = \left(\frac{p^2}{p^2-1}\right)^{(n-1)/2} \frac{n}{n+1} \left(1 - \bar{h}^2\right)^{(n-1)/2} \left(1 - \bar{h}\right) \tag{3.17}$$

où R_V *s'accroît pour* $-1 < \bar{h} \le -1/p$ *et décroît pour* $-1/p \le \bar{h} \le 1$.

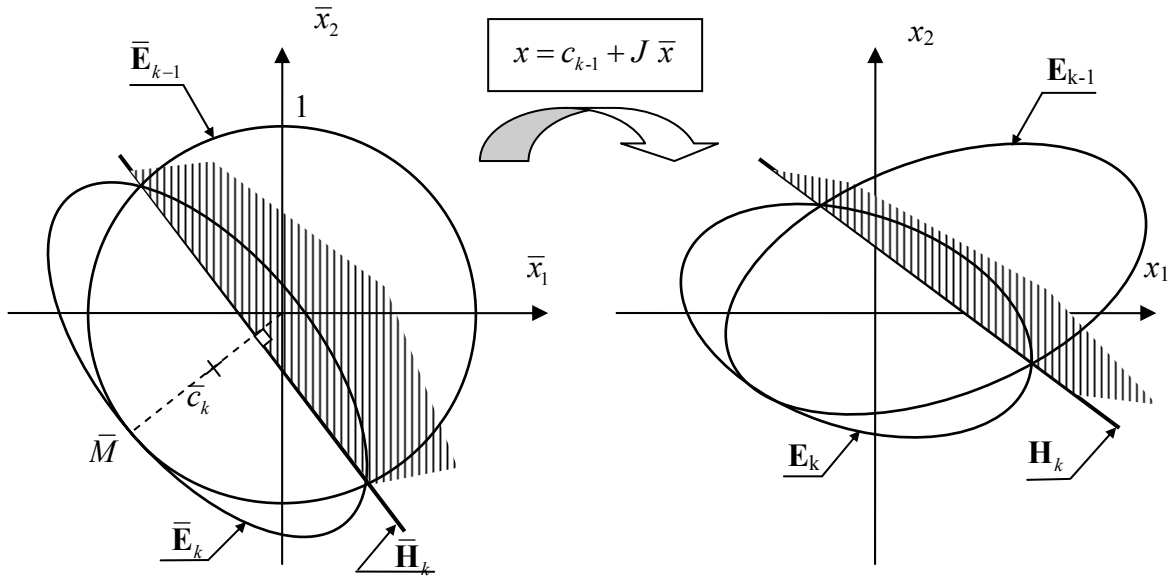


Figure 3.3 : Transformation d'ellipsoïde de coordonnées $\bar{x} \to x$.

Remarque

$\bar{h}$ est la distance entre l'origine **0** et l'hyperplan $\bar{\mathbf{H}}_k$ dans l'espace $\bar{x}$. En utilisant la transformation (3.11), $\bar{h}$ est calculé dans le système de coordonnées *x* par :

$$\bar{h} = \frac{d_k^T c_{k-1} - b_k}{\sqrt{d_k^T P_{k-1} d_k}} \tag{3.18}$$

Le théorème 3.1 indique clairement que la possibilité de réduire le volume de l'ellipsoïde dépend de la position relative entre l'hyperplan $\mathbf{H}_k$ et l'ellipsoïde $\mathbf{E}_{k\text{-}1}$. Il donne aussi le rapport des volumes. Notons qu'en considérant le cas particulier de l'algorithme OBE où l'un des deux hyperplans définissant la bande coupe $\mathbf{E}_{k\text{-}1}$, Fogel et Huang [FOG 82] ont déjà montré ce dernier résultat.

3.1.2 Forme factorisée

Dans l'algorithme GT ci-avant, pour construire le nouvel ellipsoïde $\mathbf{E}_k$, on doit calculer à chaque instant la racine carrée d'un terme quadratique, à savoir $\sqrt{d_k^T P_{k-1} d_k}$. En outre, en raison d'erreurs liées à la précision finie de la machine, le terme $\bar{h}$ peut ne plus être positif, ou/et la matrice P peut perdre son caractère « symétrique, définie positive » dans (3.15). En conséquence, l'algorithme peut tomber en défaut lors du calcul numérique. Goldfarb et Todd ont proposé une forme factorisée afin de stabiliser numériquement cet algorithme.

On sait qu'une matrice symétrique définie positive P peut toujours être décomposée en :

$$P = LDL^T \tag{3.19}$$

avec :

- L une matrice triangulaire inférieure avec une diagonale unité (tous les termes diagonaux sont égaux à 1) ;
- D une matrice diagonale telle que $D = diag(\tau_1, ..., \tau_p), \quad \tau_i > 0, \forall i = 1 : p$.

En utilisant cette factorisation de P, on ne réalise pas directement les calculs donnés dans le théorème 3 .1 qui peuvent induire l'instabilité numérique. Goldfarb et Todd donnent l'algorithme factorisé suivant.

Théorème 3.2

Les paramètres de l'ellipsoïde englobant apparaissant dans les équations (3.14) à (3.16) peuvent être calculés par :

$$\begin{cases} c_k = c_{k-1} - \lambda_k w \\ L_k = L_{k-1}\tilde{L} \\ D_k = \delta \tilde{D} \end{cases} \tag{3.20}$$

avec $\quad w = L_{k-1} v \; ; \quad v = D_{k-1}\tilde{d}_{k-1} \big/ \sqrt{\varpi} \; ; \quad \varpi = \tilde{d}_k^T D_{k-1} \tilde{d}_k \; ; \quad \tilde{d}_k = L_{k-1}^T d_k$ (3.21)

où $\tilde{L}$, $\tilde{D}$ *satisfont :*

$$\tilde{L}^T \tilde{D} \tilde{L} = D_{k-1} - \kappa v v^T \tag{3.22}$$

Les éléments de la matrice diagonale $\tilde{D} = diag(\tilde{\tau}_1, ..., \tilde{\tau}_p)$ *et de la matrice triangulaire inférieure* $\tilde{L} = \{\tilde{l}_{ij}\}$ *avec* $\tilde{l}_{ij} = v_i \zeta_j$, $j < i$ *sont calculés par la relation de récurrence ci-après. Posons :*

$$t_{p+1} = 1 - \kappa v^T D_{k-1}^T v = 1 - \kappa = \left(\frac{p-1}{p+1}\right)\left(\frac{1-\bar{h}}{1+\bar{h}}\right)$$

Pour $j = p, p\text{-}1, ..., 1$, *on calcule :*

$$\begin{aligned} t_j &= t_{j+1} + \kappa v_j^2 / \tau_j \\ \tilde{\tau}_j &= \tau_j t_{j+1} / t_j \\ \zeta_j &= -\kappa v_j / (\tau_j t_{j+1}) \end{aligned} \tag{3.23}$$

Démonstration

En remplaçant la relation (3.21) dans (3 .20), on a :

$$c_k = c_{k-1} - \lambda_k L_{k-1} v = c_{k-1} - \lambda_k L_{k-1} D_{k-1} L_{k-1}^T d_k \Big/ \sqrt{d_k^T L_{k-1} D_{k-1} L_{k-1}^T d_k}$$

Sachant que $L_{k-1} D_{k-1} L_{k-1}^T = P_{k-1}$, on obtient la formule de c_k donnée dans l'équation (3.14). Pour P_k, on a :

$$P_k = L_k D_k L_k^T = L_{k-1} \tilde{L} \delta \tilde{D} \tilde{L}^T L_{k-1}^T = L_{k-1} \delta \left(D_{k-1} - \kappa v v^T \right) L_{k-1}^T$$

$$\Leftrightarrow \quad P_k = \delta L_{k-1} \left(D_{k-1} - \kappa \frac{D_{k-1} L_{k-1}^T d_k d_k^T L_{k-1} D_{k-1}}{d_k^T L_{k-1} D_{k-1} L_{k-1}^T d_k} \right) L_{k-1}^T$$

$$\Leftrightarrow \quad P_k = \delta \left(P_{k-1} - \kappa \frac{P_{k-1} d_k d_k^T P_{k-1}}{d_k^T P_{k-1} d_k} \right) \text{ de l'équation (3.15).}$$ □

Remarques

- les éléments des matrices $\tilde{L}$, $\tilde{D}$ sont calculés en utilisant l'algorithme de Gill, Murray et Saunders [GOL 82] ;
- les coefficients λ_k, δ, κ dans l'équation (3.21) sont calculés en utilisant (3.16). Dans ce cas, la distance $\bar{h}$ devient :

$$\bar{h} = \frac{\left(d_k^T c_{k-1} - b_k \right)}{\sqrt{\varpi}} \tag{3.24}$$

- dans cet algorithme factorisé, le terme $d_k^T P_{k-1} d_k$ disparaît. Par contre, on doit calculer le coefficient $\varpi = \tilde{d}_k^T D_{k-1} \tilde{d}_k$. Comme $D_{k\text{-}1}$ est une matrice diagonale positive, ϖ sera strictement positif. On évite ainsi les erreurs en calculant des termes quadratiques ;
- avec la mise en forme factorisée du théorème 3.2, la stabilité numérique de l'algorithme de Goldfarb et Todd est garantie. Cependant, cet algorithme est seulement un cas particulier de l'approche ensembliste ellipsoïdale que l'on a présenté au chapitre 1. Nous allons donc chercher une forme factorisée pour les algorithmes OBE.

3.2 Algorithme OBE factorisé

3.2.1 Raisons potentielles de l'instabilité des algorithmes OBE

Considérons d'abord la forme normalisée (bande et ellipsoïde) de l'algorithme OBE donné dans les formules (1.20) et (1.21). D'après (1.24), l'ellipsoïde $\mathbf{E}_k$ est calculé par :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \left(x - c_k\right)^T M_k \left(x - c_k\right) \le 1,\ M_k = P_k^{-1} \right\} \tag{3.25}$$

où

$$\begin{aligned} M_k &= L_\alpha / (1 - \delta_\alpha) \\ c_k &= L_\alpha^{-1} \left[\alpha_k M_{k-1} c_{k-1} + (1 - \alpha_k) d_k y_k \right] \\ L_\alpha &= \alpha_k M_{k-1} + (1 - \alpha_k) d_k d_k^T \\ \delta_\alpha &= \alpha_k c_{k-1}^T M_{k-1} c_{k-1} + (1 - \alpha_k) y_k^2 - c_k^T L_\alpha c_k \end{aligned} \tag{3.26}$$

Supposons que l'intersection $\mathbf{E}_{k-1} \cap \mathbf{B}_k$ ne soit pas vide. Alors, en théorie, on doit avoir $M_k > 0$ ou $0 \le \delta_\alpha < 1$. Cependant, la formule (3.26) ne peut en aucun cas garantir numériquement ces propriétés. Le calcul numérique des termes $c^T Mc$, $c^T Lc$ ou L_α^{-1} peut de ce fait induire une erreur. De plus, l'analyse de la formule de δ_α montre la présence du signe « - » qui ne permet pas de garantir sa non négativité. Si de telles situations se produisent, l'algorithme peut tomber en défaut incontrôlable : la matrice M_k peut devenir non définie positive et on peut alors avoir un accroissement artificiel de la taille de l'ellipsoïde bien que α_k prenne sa valeur optimale α_k^*. Un tel exemple est donné dans [LES 02]. $M_{k\text{-}1}$ est une matrice de Hilbert, engendrée par la commande Matlab $M_{k-1} = \text{hilb}(n)$ avec n = 8 sa dimension. Le conditionnement de $M_{k\text{-}1}$ vaut 1.5 10^{10}, ce qui ne présente aucune difficulté d'ordre numérique pour une précision machine de 2.2 10^{-16} (standard IEEE double précision). Le vecteur d_k est constitué des n premières composantes de la dernière colonne de l'inverse de la matrice Hilbert de dimension n + 1. $c_{k\text{-}1}$ est un vecteur dont toutes les composantes valent 1

et y_k = 1. La vérification de la condition d'intersection entre la bande et l'ellipsoïde donné dans la section 1.2.4 montre que cette dernière n'est pas vide. Cela revient à dire que $0 < \delta_\alpha < 1$ quel que soit α dans $0 \le \alpha < 1$ pour $M_k > 0$. Cependant, l'application de la formule (3.26) avec α_k = 0.001 donne ici δ_α = -47.6 pour les calculs fait dans l'environnement Matlab. Ceci illustre clairement le problème d'instabilité numérique de cet algorithme.

Pour l'algorithme OBE général donné dans les formules (1.8) et (1.9), la matrice caractérisant l'ellipsoïde $\mathbf{E}_k$ est $\sigma_k^2 P_k$ où :

$$\sigma_k^2 = \alpha_k \sigma_{k-1}^2 + \beta_k \gamma_k^2 - \varphi_k \tag{3.27}$$

$$\varphi_k = \alpha_k c_{k-1}^T P_{k-1}^{-1} c_{k-1} + \beta_k y_k^2 - c_k^T P_k^{-1} c_k$$

avec

$$= \frac{\alpha_k \beta_k \nu_k^2}{\alpha_k + \beta_k g_k} \quad avec \quad g_k = d_k^T P_{k-1} d_k$$

Le calcul numérique des termes quadratiques $c^T P^{-1} c$ ou $d^T P d$ dans φ_k peut aussi conduire aux défauts indiqués ci-avant. Nous allons donc établir une forme factorisée pour stabiliser les algorithmes OBE.

3.2.2 Forme factorisée de l'algorithme OBE normalisé

3.2.2.1 La forme factorisée « information »

L'originalité de la démarche consiste à observer que la détermination de $\mathbf{E}_k$ peut être considérée comme la solution d'un problème quadratique [LES 02]. Considérons le problème d'optimisation :

$$\hat{x} = \arg \min_x \left[f(x) = \alpha_k \left\| x - c_{k-1} \right\|_{M_{k-1}}^2 + \left(1 - \alpha_k\right) \left\| y_k - d_k^T x \right\|_2^2 \right] \tag{3.28}$$

$\hat{x}$ est la solution obtenue pour un gradient nul :

$$\left. \frac{\partial f(x)}{\partial x} \right|_{x=\hat{x}} = 2 \left\{ \left[\alpha_k M_{k-1} + \left(1 - \alpha_k\right) d_k d_k^T \right] \hat{x} - \left[\alpha_k M_{k-1} c_{k-1} + \left(1 - \alpha_k\right) d_k y_k \right] \right\} = 0$$

$$\Rightarrow \quad \hat{x} = L_\alpha^{-1} \left[\alpha_k M_{k-1} c_{k-1} + \left(1 - \alpha_k\right) d_k y_k \right] \tag{3.29}$$

avec $\quad L_\alpha = \alpha_k M_{k-1} + \left(1 - \alpha_k\right) d_k d_k^T$

En comparant (3.29) avec la formule de c_k dans (3.26), on constate que $\hat{x}$ est identique

au centre de l'ellipsoïde $\mathbf{E}_k$. De plus, $\frac{\partial^2 f(x)}{\partial x^2} = 2L_\alpha > 0,\ \forall\alpha\colon 0<\alpha \le 1$ implique que $\hat{x}$ ou c_k est l'unique solution du problème d'optimisation $f(x)$ de l'équation (3.28).

On peut réécrire le critère $f(x)$ en utilisant des formes factorisées :

$$\begin{cases} M_{k-1} = X_{k-1}^T X_{k-1} \\ \tilde{X}_{k-1} = \sqrt{\alpha_k}\, X_{k-1} \\ v = \sqrt{1-\alpha_k}\, d_k \\ w = \sqrt{1-\alpha_k}\, y_k \end{cases} \tag{3.30}$$

où $X_{k\text{-}1}$ est la factorisation de Cholesky de $M_{k\text{-}1}$, ($X_{k\text{-}1}$ est donc triangulaire supérieure). En conséquence, $f(x)$ devient :

$$\begin{aligned} f(x) &= \left\| \tilde{X}_{k-1}x - \tilde{X}_{k-1}c_{k-1} \right\|_2^2 + \left\| v^T x - w \right\|_2^2 = \left\| \begin{bmatrix} \tilde{X}_{k-1} \\ v^T \end{bmatrix} x - \begin{bmatrix} \tilde{X}_{k-1}c_{k-1} \\ w \end{bmatrix} \right\|_2^2 \\ &= \left\| Ax - b \right\|_2^2 \end{aligned} \tag{3.31}$$

Le problème d'optimisation $f(x)$ s'écrit donc comme un problème de moindres carrées. La solution $\hat{x}$ de $f(x)$ est calculée par la technique de factorisation orthogonale [STE 93] (voir annexe A) :

$$\begin{cases} f(x) = \left\| Ax - b \right\|_2^2 = \left\| Ux - b_1 \right\|_2^2 + \left\| b_2 \right\|_2^2 \\ \hat{x} = U^{-1} b_1 \\ f(\hat{x}) = \left\| b_2 \right\|_2^2 \end{cases} \tag{3.32}$$

Les paramètres U, b_1, b_2 sont obtenus en factorisant la matrice $\Pi = \begin{bmatrix} A & b \end{bmatrix}$ par une matrice orthogonale Q (factorisation « QR ») :

$$\begin{cases} \Pi = \begin{bmatrix} A & b \end{bmatrix} \\ Q \perp\colon Q\Pi = \begin{bmatrix} U & b_1 \\ 0 & b_2 \end{bmatrix} = \tilde{\Pi} \end{cases} \tag{3.33}$$

où $\Pi \in \mathbb{R}^{(p+1)\times(p+1)}$, $U \in \mathbb{R}^{p\text{x}p}$ est une matrice triangulaire supérieure, $b_1 \in \mathbb{R}^p$ et b_2 est un scalaire. A partir de l'équation (3.32), on peut établir la forme factorisée de l'algorithme OBE normalisé.

Théorème 3.3 : forme factorisée « information » de l'algorithme OBE normalisé

Les paramètres de l'ellipsoïde $\mathbf{E}_k$ *sont construits comme suit :*

- Créer la matrice $\Pi = \begin{bmatrix} \tilde{X}_{k-1} & \tilde{X}_{k-1}c_{k-1} \\ v^T & w \end{bmatrix}$

- Mettre Π *sous la forme triangulaire supérieure par* $Q\perp$:

$$Q\Pi = \begin{bmatrix} U & r \\ 0 & s \end{bmatrix} = \tilde{\Pi} \tag{3.34}$$

- Calculer le centre de l'ellipsoïde $c_k = U^{-1}r$

- La factorisation de M_k *est donné par* $X_k = \dfrac{U}{\sqrt{1-s^2}}$

Démonstration

Par définition, la solution $\hat{x}$ du critère $f(x)$ est le centre de l'ellipsoïde. D'après (3.31), on obtient $c_k = \hat{x} = U^{-1}r$. A partir de (3.34), on a :

$$\Pi^T\Pi = \begin{bmatrix} \tilde{X}_{k-1}^T & v \\ c_{k-1}^T\tilde{X}_{k-1}^T & w \end{bmatrix}\begin{bmatrix} \tilde{X}_{k-1} & \tilde{X}_{k-1}c_{k-1} \\ v^T & w \end{bmatrix} \tag{3.35}$$

et

$$\tilde{\Pi}^T\tilde{\Pi} = \begin{bmatrix} U^T & 0 \\ r^T & s \end{bmatrix}\begin{bmatrix} U & r \\ 0 & s \end{bmatrix} \tag{3.36}$$

Q étant une matrice orthogonale, on a :

$$\tilde{\Pi}^T\tilde{\Pi} = \Pi^T Q^T Q\Pi = \Pi^T\Pi$$

En développant (3.35) et (3.36) et en les identifiant, on obtient :

$$U^TU = \tilde{X}_{k-1}^T\tilde{X}_{k-1} + vv^T = \alpha_k M_{k-1} + (1-\alpha_k)d_k d_k^T = L_\alpha$$

$$r^Tr + s^2 = c_{k-1}^T\tilde{X}_{k-1}^T\tilde{X}_{k-1}c_{k-1} + w^2 = \alpha_k c_{k-1}^T M_{k-1}c_{k-1} + (1-\alpha_k)y_k^2$$

Mais $\quad r = Uc_k \Rightarrow \; r^Tr = c_k^T U^T U c_k = c_k^T L_\alpha c_k$

Alors $\quad s^2 = \alpha_k c_{k-1}^T M_{k-1}c_{k-1} + (1-\alpha_k)y_k^2 - c_k^T L_\alpha c_k \equiv \delta_\alpha$

De plus $\quad M_k = X_k^T X_k = \dfrac{L_\alpha}{1-\delta_\alpha} = \dfrac{U^TU}{1-s^2} \Rightarrow X_k = \dfrac{U}{\sqrt{1-s^2}}$. □

Remarques

- avec cet algorithme factorisé, $\delta_\alpha = s^2 \geq 0, \;\; \forall\alpha: 0 \leq \alpha < 1$. La propriété théorique de δ_α est alors remplie exactement. Reprenons l'exemple de la section 3.2.1, avec cette forme factorisée, on obtient $\delta_\alpha = 1.6997\ 10^{-2}$;

- les calculs de c_k et de la matrice $M_k = X_k^T X_k$ sont indépendants. Cette situation est sans équivalence dans (3.26) parce qu'il faut impérativement connaître c_k pour calculer δ_α, puis P_k. Elle constitue une des propriétés particulières de l'approche factorisée ;
- pour évaluer la valeur optimale de α_k^*, on doit chercher la solution d'une équation du 2ème degré (critère du déterminant) ou du 3ème degré (critère de la trace), comme pour l'algorithme OBE non factorisé. Ces équations font apparaître les grandeurs :

$$g_k = d_k^T M_{k-1}^{-1} d_k \,;\; \tilde{g}_k = d_k^T \left(M_{k-1}^{-1}\right)^2 d_k \,;\;\; \mu_T(k-1) = \mathrm{tr}\left(M_{k-1}^{-1}\right)$$

Remarquons que le calcul numérique direct de ces formules ne garantit pas $g_k > 0,\ \tilde{g}_k > 0$ et $\mu_T(k-1) > 0$. De plus, la matrice M_k n'est pas directement disponible dans l'algorithme factorisé puisque son calcul n'est pas nécessaire. On va donc également utiliser une forme factorisée pour calculer ces grandeurs :

$$\begin{cases} \mu_T(k-1) = \|Z\|_F^2, \;\; Z : X_{k\text{-}1}^T Z = I \\ g_k = \|\omega\|_2^2, \;\; \omega : X_{k\text{-}1}^T \omega = d_k \Leftrightarrow \omega = Zd \\ \tilde{g}_k = \|\varpi\|_2^2, \;\; \varpi : X_{k\text{-}1} \varpi = \omega \Leftrightarrow \varpi = Z^T \omega \end{cases} \tag{3.37}$$

où $\|A\|_F = \left[\sum a_{ij}^2\right]^{1/2}$ est la norme de Frobenius.

En mettant l'algorithme OBE normalisé sous forme factorisée, tout doute quant au calcul numérique de δ_α disparaît. Tous les termes quadratiques sont calculés par la norme au carré d'un vecteur, donc strictement positifs.

3.2.2.2 La forme factorisée « covariance »

Le calcul de α^* (voir paragraphe 1.2.1) et l'estimation d'un paramètre variable ont mis en évidence le fait qu'il aurait été plus naturel de travailler avec la matrice $P_k = M_k^{-1}$ plutôt que M_k. Comme on l'a souligné précédemment, si la matrice M_k s'apparente à une matrice d'information, par dualité, la matrice $P_k = M_k^{-1}$ est assimilable à une matrice de covariance. On présente maintenant la forme factorisé « covariance » de l'algorithme OBE qui donne une formulation factorisée portant directement sur P_k. Enonçons d'abord la proposition 3.1.

Proposition 3.1

Les paramètres de l'ellipsoïde $\mathbf{E}_k$ *dans (3.25) peuvent être calculés par :*

$$\begin{cases} P_k = (1-\delta_\alpha)N_\alpha \\ c_k = c_{k-1} + \dfrac{1-\alpha_k}{\alpha_k + (1-\alpha_k)g_k} P_{k-1} d_k \nu_k \\ N_\alpha = \dfrac{1}{\alpha_k}\left[P_{k-1} - \dfrac{1-\alpha_k}{\alpha_k + (1-\alpha_k)g_k} P_{k-1} d_k d_k^T P_{k-1} \right] \\ \delta_\alpha = \dfrac{\alpha_k (1-\alpha_k)}{\alpha_k + (1-\alpha_k)g_k} \nu_k^2 \end{cases} \tag{3.38}$$

où $\nu_k = y_k - d_k^T c_{k-1};\ g_k = d_k^T P_{k-1} d_k$

Démonstration

D'après (3.26), en posant $N_\alpha = L_\alpha^{-1}$ et avec la relation $P_k = M_k^{-1}$, on a :

$$N_\alpha = \left[\alpha_k P_{k-1}^{-1} + (1-\alpha_k) d_k d_k^T \right]^{-1} \tag{3.39}$$

Utilisons le lemme d'inversion matricielle dans l'équation (3.39) pour en déduire N_α en (3.38). Replaçons en suite $L_\alpha^{-1} = N_\alpha$ dans c_k (3.26), on obtient :

$$\begin{aligned} c_k &= \frac{1}{\alpha_k}\left[P_{k-1} - \frac{1-\alpha_k}{\alpha_k + (1-\alpha_k)g_k} P_{k-1} d_k d_k^T P_{k-1} \right]\left[\alpha_k P_{k-1}^{-1} c_{k-1} + (1-\alpha_k) d_k y_k \right] \\ &= c_{k-1} + \frac{1-\alpha_k}{\alpha_k + (1-\alpha_k)g_k} P_{k-1} d_k \left[\frac{1-\alpha_k}{\alpha_k}\left(\frac{\alpha_k}{1-\alpha_k} + g_k \right) y_k - d_k^T P_k P_k^{-1} c_{k-1} - \frac{1-\alpha_k}{\alpha_k} g_k y_k \right] \\ &= c_{k-1} + \frac{1-\alpha_k}{\alpha_k + (1-\alpha_k)g_k} P_{k-1} d_k \nu_k \end{aligned}$$

Considérons maintenant la formule de δ_α dans (3.26). En posant le dernier terme de cette formule comme ρ_α, i.e. $\rho_\alpha \triangleq c_k^T L_\alpha c_k$, et en utilisant les formules de c_k, L_α dans (3.26), on peut écrire :

$$\begin{aligned} \rho_\alpha &= \left[\alpha_k c_{k-1}^T P_{k-1}^{-1} + (1-\alpha_k) d_k^T y_k \right] N_\alpha \left[\alpha_k P_{k-1}^{-1} c_{k-1} + (1-\alpha_k) d_k y_k \right] \\ &= \alpha_k^2 c_{k-1}^T P_{k-1}^{-1} N_\alpha P_{k-1}^{-1} c_k + 2\alpha_k (1-\alpha_k) y_k c_k^T P_k^{-1} N_\alpha d_k + (1-\alpha_k)^2 d_k^T N_\alpha d_k y_k^2 \end{aligned} \tag{3.40}$$

En utilisant de nouveau le lemme d'inversion matricielle, on développe les termes suivants :

- $N_\alpha P_{k-1}^{-1} = \left[\alpha_k I + (1-\alpha_k) P_{k-1} d_k d_k^T \right]^{-1} = \dfrac{I}{\alpha_k} - \dfrac{1-\alpha_k}{\alpha_k} \dfrac{P_{k-1} d_k d_k^T}{\alpha_k + (1-\alpha_k) g_k}$

- $P_{k-1}^{-1} N_\alpha = \left[\alpha_k I + (1-\alpha_k) d_k d_k^T P_{k-1} \right]^{-1} = \dfrac{I}{\alpha_k} - \dfrac{1-\alpha_k}{\alpha_k} \dfrac{d_k d_k^T P_{k-1}}{\alpha_k + (1-\alpha_k) g_k}$

En remplaçant ces termes dans (3.40) et en utilisant N_α dans (3.38), après quelques manipulations classiques, on en déduit :

$$\rho_\alpha = \alpha_k c_{k-1}^T P_{k-1}^{-1} c_{k-1} - \frac{\alpha_k(1-\alpha_k)}{\alpha_k + (1-\alpha_k) g_k}\left(d_k^T c_{k-1}\right)^2 + \frac{2\alpha_k(1-\alpha_k) d_k^T c_{k-1} y_k}{\alpha_k + (1-\alpha_k) g_k} + \frac{(1-\alpha_k)^2 g_k y_k^2}{\alpha_k + (1-\alpha_k) g_k} \quad (3.41)$$

Enfin, on remplace (3.41) dans la formule de δ_α (3.26), on trouve :

$$\delta_\alpha = \frac{\alpha_k(1-\alpha_k)}{\alpha_k + (1-\alpha_k) g_k} v_k^2 . \qquad \square$$

Remarques

- les paramètres de l'ellipsoïde englobant $\mathbf{E}_k$ dans (3.38) ne dépendent que de la matrice P_k mais non de son inverse. C'est-à-dire que l'on n'a pas besoin d'effectuer d'inversion ;
- la formule de δ_α dans (3.38) avec $0 < \alpha \leq 1$ garantit théoriquement sa non négativité. Cependant, cette grandeur dépend encore du terme quadratique $g_k = d_k^T P_{k-1} d_k$ qui peut être faussé lors des calculs numériques.

Utilisons maintenant la factorisation de Cholesky de la matrice P, i.e.

$$P = \bar{X}^T \bar{X} \quad (3.42)$$

où $\bar{X}$ est triangulaire supérieure ($\bar{X}$ ici pour distinguer de X, matrice factorisée de M), on peut établir la forme factorisée « covariance » suivante.

Théorème 3.4 : forme factorisée « covariance » de l'algorithme OBE normalisé

Les paramètres de l'ellipsoïde $\mathbf{E}_k$ *sont construits comme suit :*

- *Créer la matrice* $\Gamma = \begin{bmatrix} \frac{1}{\sqrt{1-\alpha_k}} & 0 \\ \frac{\bar{X}_{k-1}}{\sqrt{\alpha_k}} d_k & \frac{\bar{X}_{k-1}}{\sqrt{\alpha_k}} \end{bmatrix}$ (3.43)

- *Mettre* Γ *sous la forme triangulaire supérieure par* $Q\perp$ *tel que :*

$$Q\Gamma = \begin{bmatrix} \bar{s} & \bar{r} \\ 0 & \bar{U} \end{bmatrix} = \tilde{\Gamma};\ \bar{s} \text{ scalaire},\ \bar{r} \in \mathbb{R}^{1xp},\ U \in \mathbb{R}^{pxp} \quad (3.44)$$

- *Calculer* $c_k = c_{k-1} + \bar{r}^T v_k / \bar{s}$

- *Calculer* $\delta_\alpha = v_k^2 / \bar{s}^2$

- *Obtenir* $\bar{X}_k = \sqrt{1 - \frac{v_k^2}{\bar{s}^2}} \bar{U}$

Démonstration

A partir de (3.43), et avec la propriété dans (3.42), i.e. $P_{k-1} = \overline{X}_{k-1}^T \overline{X}_{k-1}$, on a :

$$\Gamma^T \Gamma = \begin{bmatrix} \dfrac{1}{1-\alpha_k} + \dfrac{g_k}{\alpha_k} & \dfrac{d_k^T P_{k-1}}{\alpha_k} \\ \dfrac{P_{k-1} d_k}{\alpha_k} & \dfrac{P_{k-1}}{\alpha_k} \end{bmatrix} \tag{3.45}$$

En utilisant la factorisation (3.44), et Q étant une matrice orthogonale, il vient :

$$\tilde{\Gamma}^T \tilde{\Gamma} = \begin{bmatrix} \overline{s}^2 & \overline{sr} \\ \overline{r}^T \overline{s} & \overline{r}^T \overline{r} + \overline{U}^T \overline{U} \end{bmatrix} = \Gamma^T Q^T Q \Gamma = \Gamma^T \Gamma \tag{3.46}$$

En identifiant (3.45) et (3.46) élément par élément, on obtient :

a) $$\overline{s}^2 = \frac{1}{1-\alpha_k} + \frac{g_k}{\alpha_k} = \frac{\alpha_k + (1-\alpha_k) g_k}{\alpha_k (1-\alpha_k)} \tag{3.47}$$

En comparant (3.38) et (3.47), on en déduit $\delta_\alpha = \nu_k^2 / \overline{s}^2$;

b) $$\overline{r}^T \overline{s} = \frac{P_{k-1} d_k}{\alpha_k} \tag{3.48}$$

Alors de (3.38) :

$$c_k = c_{k-1} + \frac{\alpha_k (1-\alpha_k)}{\alpha_k + (1-\alpha_k) g_k} \frac{P_{k-1} d_k}{\alpha_k} \nu_k = c_{k-1} + \frac{1}{\overline{s}^2} \overline{r}^T \overline{s} \nu_k = c_{k-1} + \overline{r}^T \nu_k / \overline{s}$$

c) $$\overline{r}^T \overline{r} + \overline{U}^T \overline{U} = \frac{P_{k-1}}{\alpha_k} \Rightarrow \overline{U}^T \overline{U} = \frac{P_{k-1}}{\alpha_k} - \frac{(\overline{r}^T \overline{s})(\overline{sr})}{\overline{s}^2}$$

Utilisons (3.47) et (3.48), il vient :

$$\overline{U}^T \overline{U} = \frac{1}{\alpha_k} \left[P_{k-1} - \frac{1-\alpha_k}{\alpha_k + (1-\alpha_k) g_k} P_{k-1} d_k d_k^T P_{k-1} \right] = N_\alpha$$

En outre d'après (3.38) :

$$P_k = (1-\delta_\alpha) N_\alpha = (1-\delta_\alpha) \overline{U}^T \overline{U} = \overline{X}_k^T \overline{X}_k$$

alors $\overline{X}_k = \sqrt{1-\delta_\alpha}\, \overline{U} = \sqrt{1 - \dfrac{\nu_k^2}{\overline{s}^2}}\, \overline{U}$. □

Remarques

- dans cet algorithme factorisé, le terme quadratique dans la formule de δ_α disparaît. La propriété théorique de δ_α est alors remplie numériquement ;
- similairement à la forme factorisée du théorème 3.3, les calculs de c_k et de la matrice

$P_k = \bar{X}_k^T \bar{X}_k$ dans cet algorithme factorisé sont indépendants ;

- les grandeurs pour évaluer la valeur optimale de α_k^* (voir la section 1.2.1) sont facilement obtenues, i.e. :

$$g_k = \left(\bar{X}_{k-1} d_k\right)^T \left(\bar{X}_{k-1} d_k\right); \quad \hbar_k = \left(\bar{X}_{k-1} d_k\right)^T \bar{X}_{k-1} \bar{X}_{k-1}^T \left(\bar{X}_{k-1} d_k\right); \quad \mu_T(k-1) = \mathrm{tr}\left(\bar{X}_{k-1}^T \bar{X}_{k-1}\right)$$

3.2.3 Forme factorisée de l'algorithme OBE général

Les formes factorisées de l'algorithme OBE normalisé nous ont conduit à proposer une forme factorisée pour l'algorithme OBE général, évitant ainsi à l'utilisateur de passer par une étape de normalisation. Comme dans l'algorithme OBE normalisé, on va construire la forme factorisée « standard » et « covariance » pour l'algorithme OBE général.

3.2.3.1 La forme factorisée « standard »

Rappelons la formule générale de l'algorithme OBE donnée dans l'équation (1.8) :

$$\begin{cases} M_k = \alpha_k M_{k-1} + \beta_k d_k d_k^T \\ c_k = M_k^{-1}\left(\alpha_k M_{k-1} c_{k-1} + \beta_k d_k y_k\right) \\ \sigma_k^2 = \alpha_k \sigma_{k-1}^2 + \beta_k \gamma_k^2 - \varphi_k \\ \varphi_k = \alpha_k c_{k-1}^T M_{k-1} c_{k-1} + \beta_k y_k^2 - c_k^T M_k c_k \end{cases} \tag{3.49}$$

On va montrer qu'on peut mettre sous forme factorisée cet algorithme en utilisant le même raisonnement que celui suivi pour l'algorithme OBE normalisé dans le paragraphe 3.2.2.1. En posant $M = X^T X$ où X est la factorisation de Cholesky de M, l'algorithme factorisé est énoncé dans le théorème 3.5.

Théorème 3.5 : forme factorisée « information » de l'algorithme OBE général

Les paramètres de l'ellipsoïde $\boldsymbol{E}_k$ *sont construits par :*

- *Créer la matrice* $\Pi = \begin{bmatrix} \sqrt{\alpha_k} X_{k-1} & \sqrt{\alpha_k} X_{k-1} c_{k-1} \\ \sqrt{\beta_k} d_k^T & \sqrt{\beta_k} y_k \end{bmatrix}$

- *Mettre* Π *sous la forme triangulaire supérieure par* $Q \perp$:

$$Q\Pi = \begin{bmatrix} U & r \\ 0 & s \end{bmatrix} = \tilde{\Pi} \tag{3.50}$$

- *Calculer* $\begin{cases} c_k = U^{-1} r \\ X_k = U \\ \sigma_k^2 = \alpha_k \sigma_{k-1}^2 + \beta_k \delta_k^2 - s^2 \end{cases}$

Démonstration

Similairement au paragraphe 3.2.2.1, en considérant le problème d'optimisation :

$$\hat{x} = \arg \min_x \left[f(x) = \alpha_k \left\| x - c_{k-1} \right\|_{M_{k-1}}^2 + \beta_k \left\| y_k - d_k^T x \right\|_2^2 \right]$$

on obtient $\hat{x}$ identique au centre c_k de l'ellipsoïde $\mathbf{E}_k$ dans l'algorithme OBE général (3.49). Utilisons la forme factorisée de la matrice M, on peut réécrire $f(x)$ selon :

$$\begin{aligned} f(x) &= \left\| \sqrt{\alpha_k} X_{k-1} x - \sqrt{\alpha_k} X_{k-1} c_{k-1} \right\|_2^2 + \left\| \sqrt{\beta_k} d_k^T x - \sqrt{\beta_k} y_k \right\|_2^2 \\ &= \left\| \begin{bmatrix} \sqrt{\alpha_k} X_{k-1} \\ \sqrt{\beta_k} d_k^T \end{bmatrix} x - \begin{bmatrix} \sqrt{\alpha_k} X_{k-1} c_{k-1} \\ \sqrt{\beta_k} y_k \end{bmatrix} \right\|_2^2 = \left\| Ax - b \right\|_2^2 \end{aligned}$$

Alors, en construisant la matrice Π comme dans (3.50) et en la factorisant, d'après (3.32), on déduit :

$$c_k = \hat{x} = U^{-1} r$$

A partir des formules de Π et $\tilde{\Pi}$ dans (3.50), on a :

$$\Pi^T \Pi = \begin{bmatrix} \alpha_k M_{k-1} + \beta_k d_k d_k^T & \alpha_k M_{k-1} c_{k-1} + \beta_k d_k y_k \\ \alpha_k c_{k-1}^T M_{k-1} + \beta_k d_k^T y_k & \alpha_k c_{k-1}^T M_{k-1} c_{k-1} + \beta_k y_k^2 \end{bmatrix} \tag{3.51}$$

et

$$\tilde{\Pi}^T \tilde{\Pi} = \begin{bmatrix} U^T & 0 \\ r^T & s \end{bmatrix} \begin{bmatrix} U & r \\ 0 & s \end{bmatrix} = \begin{bmatrix} U^T U & U^T r \\ r^T U & r^T r + s^2 \end{bmatrix} \tag{3.52}$$

Q étant une matrice orthogonale, il vient $\tilde{\Pi}^T \Pi = \Pi^T Q^T Q \Pi = \Pi^T \Pi$.

Alors, en identifiant (3.51) et (3.52), on obtient :

$$U^T U = \alpha_k M_{k-1} + \beta_k d_k d_k = M_k = X_k^T X_k$$

$$\Rightarrow \quad X_k = U$$

De plus

$$\begin{cases} r^T r + s^2 = \alpha_k c_{k-1}^T M_{k-1} c_{k-1} + \beta_k y_k^2 \\ r^T r = c_k^T U^T U c_k = c_k^T M_k c_k \end{cases}$$

$$\Rightarrow \quad s^2 = \alpha_k c_{k-1}^T M_{k-1} c_{k-1} + \beta_k y_k^2 - c_k^T M_k c_k = \varphi_k \tag{3.53}$$

De (3.53) et en comparant avec (3.49), on déduit σ_k^2 dans (3.50). □

Remarque

On remarque qu'un signe « - » existe dans la formule de σ_k^2. Si l'intersection $\mathbf{E}_{k-1} \cap \mathbf{B}_k$ est non vide, σ_k^2 est évidement positif, en théorie. Une erreur numérique est toujours latente

dans le calcul de φ_k donné dans la formule (3.49). Cependant, avec (3.50), φ_k est remplacé par s^2 qui garantit sa positivité.

3.2.3.2 La forme factorisée « covariance »

Dans ce paragraphe, on va construire la forme factorisée pour l'algorithme OBE général où intervient la matrice P_k plutôt que M_k. Rappelons que ce type de formule a été présenté dans (1.9) où :

$$\begin{cases} P_k = \dfrac{1}{\alpha_k}\left[P_{k-1} - \dfrac{\beta_k P_{k-1} d_k d_k^T P_{k-1}}{\alpha_k + \beta_k g_k}\right] \\ c_k = c_{k-1} + \beta_k P_k d_k \nu_k \\ \sigma_k^2 = \alpha_k \sigma_{k-1}^2 + \beta_k \gamma_k^2 - \varphi_k \end{cases} \tag{3.54}$$

avec $\quad g_k = d_k^T P_{k-1} d_k \;;\; \nu_k = y_k - d_k^T c_{k-1} \;;\; \varphi_k = \dfrac{\alpha_k \beta_k \nu_k^2}{\alpha_k + \beta_k g_k}$

En utilisant la factorisation Cholesky de la matrice P, *i.e.*

$$P = \overline{X}^T \overline{X} \tag{3.55}$$

on a établi la forme factorisée « covariance » de l'algorithme OBE général énoncée dans le théorème 3.6.

Théorème 3.6 : forme factorisée « covariance » de l'algorithme OBE général

Les paramètres de l'ellipsoïde $\mathbf{E}_k$ *sont construits selon les étapes :*

- *Créer la matrice* $\Gamma = \begin{bmatrix} \dfrac{1}{\sqrt{\beta_k}} & 0 \\ \dfrac{\overline{X}_{k-1}}{\sqrt{\alpha_k}} d_k & \dfrac{\overline{X}_{k-1}}{\sqrt{\alpha_k}} \end{bmatrix}$ (3.56)

- *Mettre* Γ *sous la forme triangulaire supérieure par* $Q\perp$ *tel que:*

$$Q\Gamma = \begin{bmatrix} \overline{s} & \overline{r} \\ 0 & \overline{U} \end{bmatrix} = \tilde{\Gamma};\ \overline{s} \text{ scalaire},\ \overline{r} \in \mathbb{R}^{1xp},\ U \in \mathbb{R}^{pxp} \tag{3.57}$$

- *Calculer* $c_k = c_{k-1} + \alpha_k \beta_k \nu_k \overline{r}^T \overline{s}$

- *Calculer* $\varphi_k = \nu_k^2 / \overline{s}^2$ et $\sigma_k^2 = \alpha_k \sigma_{k-1}^2 + \beta_k \delta_k^2 - \varphi_k$

- *Obtenir* $\overline{X}_k = \overline{U}$

Démonstration

Construisons les deux matrices $\Gamma^T\Gamma$, $\tilde{\Gamma}^T\tilde{\Gamma}$ et compte tenu du fait que $\tilde{\Gamma}^T\tilde{\Gamma} = \Gamma^T Q^T Q\Gamma = \Gamma^T\Gamma$, en identifiant leurs éléments, on obtient :

a) $$\overline{s}^2 = \frac{\alpha_k + \beta_k g_k}{\alpha_k \beta_k} \tag{3.58}$$

alors $\varphi_k = v_k^2 / \overline{s}_k^2$

b) $$\overline{r}^T \overline{s} = \frac{P_{k-1} d_k}{\alpha_k} \tag{3.59}$$

En comparant (3.59) avec (3.54), on obtient :

$$c_k = c_{k-1} + \beta_k P_k d_k v_k = c_{k-1} + \alpha_k \beta_k v_k \overline{r}^T \overline{s}$$

c) $$\overline{r}^T \overline{r} + \overline{U}^T \overline{U} = \frac{P_{k-1}}{\alpha_k}$$

En utilisant les résultats dans (3.58) et (3.59), on déduit :

$$\overline{U}^T \overline{U} = \frac{1}{\alpha_k}\left[P_{k-1} - \frac{\beta_k P_{k-1} d_k d_k^T P_{k-1}}{\alpha_k + \beta_k g_k}\right] = P_k = \overline{X}_k^T \overline{X}_k$$

alors $\overline{X} = \overline{U}$. □

3.2.4 Test de cohérence de l'intersection entre deux ellipsoïdes : expression factorisée de ce test

Le test de l'intersection de deux ellipsoïdes $E_1(c_1, M_1 = P_1^{-1})$ et $E_2(c_2, M_2 = P_2^{-1})$ a été abordé dans la section 1.2.5. Il est évalué via le coefficient δ_α par :

$$\delta_\alpha = \alpha c_1^T M_1 c_1 + (1-\alpha) c_2^T M_2 c_2 - c_\alpha^T M_\alpha c_\alpha \tag{3.60}$$

$\mathbf{E}\left(c_\alpha, M_\alpha / \left(1-\delta_\varepsilon\right)\right)$ est l'ellipsoïde englobant l'intersection $E_1 \cap E_2$ avec :

$$\begin{cases} M_\alpha = \alpha M_1 + \left(1-\alpha\right) M_2 \\ c_\alpha = M_\alpha^{-1}\left[\alpha P_1^{-1} c_1 + \left(1-\alpha\right) P_2^{-1} c_2\right] \end{cases} \tag{3.61}$$

Le calcul direct de δ_α peut être entaché d'erreurs numériques qui conduisent à une fausse conclusion sur ce test. On propose donc une approche factorisée pour calculer δ_α.

Théorème 3.7

Le coefficient δ_α dans (3.60) peut être trouvé par l'approche factorisée suivante :

- *Construire la matrice* $\Pi = \begin{bmatrix} \sqrt{\alpha_k} X_1 & \sqrt{\alpha_k} X_1 c_1 \\ \sqrt{1-\alpha_k} X_2 & \sqrt{1-\alpha_k} X_2 c_2 \end{bmatrix}$
- *Mettre* Π *sous la forme triangulaire supérieure par* $Q\perp$:

$$Q\Pi = \begin{bmatrix} U & r \\ 0 & s \end{bmatrix} = \tilde{\Pi}$$

- *Calculer* $\delta_\alpha = \|s\|_2^2$

La démonstration est identique à celle du théorème 3.3. On obtient alors également le centre de l'ellipsoïde $c_\alpha = U^{-1} r$ et la factorisation de Cholesky $X_\alpha = \dfrac{U}{\sqrt{1-s^2}}$ de la matrice $M_\alpha = X_\alpha^T X_\alpha$, si l'intersection existe. Notons que l'obtention de δ_α, c_α et M_α est indépendante. On peut donc réaliser le test sur la cohérence de l'intersection sans calculer les paramètres d'un ellipsoïde englobant qui, si l'intersection est vide, n'a pas de sens.

3.3 Conclusion

Dans ce chapitre, on a tout d'abord rappelé un algorithme particulier de la famille des algorithmes OBE [GOL 82]. L'objectif de cet algorithme est différent de celui des algorithmes du chapitre 1. Un point intéressant de cet algorithme est la détection de la position de la bande ce qui permet de réduire le volume de l'ellipsoïde englobant. Pour éviter les erreurs numériques potentielles lors du calcul des termes quadratiques, Goldfarb et Todd ont proposé une forme factorisée de cet algorithme utilisant une décomposition LDL^T de la matrice P.

Les points faibles des algorithmes OBE lors des applications numériques ont été analysés. Une des raisons de leur instabilité numérique provient du calcul des termes quadratiques qui peuvent devenir négatifs si on exploite directement les formules de construction de l'ellipsoïde englobant données dans le chapitre 1.

La forme factorisée des algorithmes OBE normalisés et généraux permet de pallier cette difficulté. L'originalité de cette démarche est de reformuler l'algorithme OBE sous la forme d'un problème de moindres carrés. En outre, les calculs des différents paramètres de l'ellipsoïde englobant et les coefficients clefs de l'algorithme sont indépendants. La forme factorisée « covariance » se montre plus favorable car elle donne directement la matrice P qui est utilisée à chaque instant pour calculer le coefficient de pondération α_k^* et l'intervalle si besoin dans lequel se situent les paramètres cherchés. Notons que nous avons utilisé une

factorisation de Cholesky de la matrice *M*, i.e. $M = X^T X$, et de la matrice *P*, i.e. $P = \bar{X}^T \bar{X}$ où $X, \bar{X}$ sont des matrices triangulaires supérieures.

Les algorithmes OBE ainsi factorisés sont numériquement stables. En outre, les propriétés théoriques de certains coefficients et paramètres sont naturellement satisfaites.

Chapitre 4

Convergence de l'algorithme OBE

Dans ce chapitre, on aborde le problème de convergence des algorithmes OBE. Rappelons que le résultat fourni par cet algorithme dans le cadre de ce travail d'identification ensembliste ellipsoïdale est un ensemble englobant donné sous la forme d'un ellipsoïde. Le terme « convergence » dans ce cas porte une signification différente de celle des méthodes traditionnelles qui produisent une valeur « ponctuelle » du vecteur de paramètre, associée généralement à un écart type. Une analyse bibliographique de l'algorithme OBE (quelle que soit sa forme, voir le chapitre 1) montre que la condition de convergence dépend non seulement des propriétés des données mais aussi de la valeur choisie pour la borne sur l'erreur.

La condition d'excitation persistante des données est une condition indispensable pour assurer la convergence de l'algorithme OBE. Les méthodes pour vérifier cette condition seront présentées dans ce chapitre. Nous proposons aussi des moyens permettant d'améliorer cette condition de convergence sur les données réelles issues du robot parallèle du LIRMM (voir le chapitre 2 pour la description de ce procédé, des données et les résultats obtenus avec les algorithmes de la littérature).

L'algorithme OBE (quel qu'il soit) nécessite la connaissance *a priori* d'une borne sur l'erreur qui d'un point de vue pratique est souvent délicate à choisir. Cette borne influe directement sur la largeur de la bande $\mathbf{B}_k$, donc sur la taille de l'ellipsoïde obtenu à la fin de la procédure d'identification. Cette borne doit remplir deux conditions antagonistes : d'une part elle doit être petite pour permettre d'aboutir à un ellipsoïde de petite taille ; d'autre part, elle doit être assez grande pour ne pas rejeter « trop » de données déclarées aberrantes. Enfin, et c'est là la difficulté rencontrée par l'utilisateur de ces techniques, elle doit être aussi cohérente que possible avec la « vraie » borne de l'erreur.

Un algorithme appelé OBE-ABE et proposé par LIN *et al*. [LIN 98] propose une solution qui permet de réduire automatiquement la borne pendant le processus d'identification. Néanmoins, cet algorithme ne fonctionne que sous des hypothèses

restrictives et souvent non réalistes d'un point de vue industriel. Nous proposons dans ce chapitre une méthode de choix semi-automatique de la borne basée sur une analyse préalable de la séquence d'erreur estimée. La méthode retenue utilise une mesure du Kurtosis. L'algorithme qui en découle est nommé « OBE-Kurtosis ». L'évaluation des différentes méthodes (OBE-ABE et OBE-Kurtosis) sera réalisée sur les données simulées et celles du robot du LIRMM.

4.1 Propriétés de convergence des algorithmes OBE

Les algorithmes OBE cherchent un ensemble englobant l'espace admissible de paramètres sous la forme d'un ellipsoïde. Le centre c_k de l'ellipsoïde $\mathbf{E}_k$ peut être considéré comme le vecteur de paramètres estimés du procédé à l'instant k, même si en toute rigueur n'importe quel point du domaine obtenu peut remplir ce rôle. La forme et la taille de l'ellipsoïde sont liées à la matrice $\sigma_k^2 P_k$. La « convergence » d'un algorithme OBE peut être explorée sous deux aspects : la convergence du centre de l'ellipsoïde vers le « vrai » paramètre x^* (connu uniquement pour des données simulées) et la convergence de l'ensemble englobant vers un point (i.e. un ensemble de taille nulle). Comme on l'a déjà vu dans le chapitre 1, les algorithmes OBE se classent en deux groupes suivant la fonction critère utilisée. On va maintenant analyser pour les algorithmes de chacun de ces deux groupes, leur propriété de convergence.

Dans le groupe 1, l'analyse bibliographique de l'algorithme FH et de l'algorithme basé sur une famille paramétrée (voir le chapitre 1) qui construisent l'ellipsoïde avec une même méthodologie (recherche directe de l'ellipsoïde $\mathbf{E}_k$ qui englobe l'intersection $\mathbf{E}_{k-1} \cap \mathbf{B}_k$), ne fournit pas de propriétés de convergence. Bien que les fonctions critères de ces deux algorithmes (minimisation du déterminant ou de la trace de la matrice $\sigma_k^2 P_k$) aient été interprétées géométriquement, aucune démonstration quant à la convergence de l'ellipsoïde englobant n'est exprimée. En fait, [FOG 87] indique la capacité de convergence de l'ellipsoïde vers un point sous l'hypothèse que $\{e_k\}$ soit un bruit blanc. Or, [NAY 94] montre que cette propriété de convergence n'est pas correcte dans ce cas (i.e. $\{e_k\}$ est bruit blanc n'induit pas la convergence de l'algorithme FH).

Dix ans après l'apparition des premiers algorithmes du groupe 1, [NAY93] (algorithme SMSA avec le critère du déterminant) qui utilise une autre méthodologie de construction de

l'ellipsoïde (normalisation de la matrice covariance dans l'algorithme SM-WRLS [NAY 93] pour éviter que cette matrice tende vers l'infini), a présenté quelques propriétés de convergence intéressantes. Ces propriétés seront généralisées pour tous les algorithmes du groupe 1 utilisant le critère du déterminant car on montre que les ellipsoïdes trouvés sont identiques. Dans la suite de ce chapitre, on présente la méthodologie de construction de l'algorithme SMSA (déjà partiellement abordée dans le chapitre 1) et ses propriétés de convergence.

Historiquement, les algorithmes du groupe 2 dont l'objectif est la réduction du coefficient σ_k^2 sont les premiers à aborder les problèmes de convergence. La première propriété, concernant le centre de l'ellipsoïde, est présentée dans le théorème 4.1.

Théorème 4.1

Supposons que $x^* \in \mathbf{E}_0$, *alors* $x^* \in \mathbf{E}_k$, $\forall k$.

Remarque

Dans la démonstration, [DAS 87], [TAN 97], [LOZ 87] utilisent la fonction de Lyapunov :

$$\psi_k = \left(x^* - c_k\right)^T P_k^{-1}\left(x^* - c_k\right) = \Delta x_k^T P_k^{-1} \Delta x_k \tag{4.1}$$

Le calcul de $\sigma_k^2 = \alpha_k \sigma_{k-1}^2 + \beta_k \gamma_k^2 + c_k^T P_k^{-1} c_k - \alpha_k c_{k-1}^T P_{k-1}^{-1} c_{k-1} - \beta_k y_k^2$ de (1.8) conduit à :

$$\psi_k - \sigma_k^2 \le \psi_{k-1} - \sigma_{k-1}^2 \tag{4.2}$$

En outre, on a $\psi_{k-1} \le \sigma_{k-1}^2 \quad \Leftrightarrow \quad x^* \in \mathbf{E}_{k-1}$. Alors, si $x^* \in \mathbf{E}_0$, cela implique que $\psi_0 \le \sigma_0^2$ et de l'équation (4.2), il vient $\psi_k \le \sigma_k^2$, par conséquent $x^* \in \mathbf{E}_k$.

Cette propriété (théorème 4.1) est également vérifiée pour les algorithmes du groupe 1 car la démonstration ne nécessite pas l'expression des coefficients de pondération α_k^*, β_k^*. Cette propriété est en fait facilement obtenue par une analyse géométrique.

Les algorithmes OBE du groupe 2 requièrent une condition sur la persistance de l'excitation (PE) pour pouvoir établir les propriétés de convergence. Cette condition est définie ci-après.

Définition 4.1

Une excitation est dite persistante (PE) si pour tout k_0, il existe $N_1 \in \mathbb{N}$ et $\varepsilon_1, \varepsilon_2 > 0$ tels que :

$$0 < \varepsilon_1 I \leq \sum_{k=k_0}^{k_0+N_1} d_k d_k^T \leq \varepsilon_2 I < \infty, \quad \forall k \tag{4.3}$$

Remarque

Si la condition PE dans (4.3) est vérifiée, alors il existe $\varepsilon_3, \varepsilon_4 > 0$ tels que :

$$0 < \varepsilon_3 I \leq P_k \leq \varepsilon_4 I < \infty \tag{4.4}$$

pour tout *k*. Les propriétés communes de convergence des algorithmes de ce groupe sont maintenant résumées dans le théorème 4.2.

Théorème 4.2

Supposons que la condition PE (4.3) soit satisfaite. Alors :

(i) $\lim_{k\to\infty} \Delta_k^2 \in [0, C_\Delta]$ *avec* $\Delta_k^2 \triangleq \left\| x^* - c_k \right\|_2^2$; (4.5)

(ii) $\lim_{k\to\infty} \left\| c_k - c_{k-1} \right\|_2 = 0$; (4.6)

(iii) $v_k^2 = \left(y_k - d_k^T c_{k-1} \right)^2$ *borné lorsque* $k \to \infty$; (4.7)

(iv) $\lim_{k\to\infty} \lambda_k^* = 0$. (4.8)

Remarques

La démonstration [DAS 87], [TAN 97], [LOZ 87] de ce théorème provient d'une analyse de la fonction quadratique donnée dans l'équation (4.1). La propriété la plus important est la « convergence » de la distance au carrée Δ_k^2 entre le centre de l'ellipsoïde et le vecteur de « vrais » paramètres dans l'intervalle $[0, C_\Delta]$. La limite C_Δ dépend de chaque algorithme car les coefficients λ_k^* choisis diffèrent, par exemple, $C_\Delta(\text{DH}) = \varepsilon_4 \gamma^2$, $C_\Delta(\text{Tan}) = \dfrac{4N_1\gamma^2}{\varepsilon_1}$ (où « DH » est l'algorithme de Dasgupta et Huang [DAS 87]; « Tan » celui de Tan *et al.* [TAN 97]).

On constate que l'intervalle de convergence du centre de l'ellipsoïde dépend de la borne γ utilisée durant le processus d'identification. La propriété (4.7) est en fait un corollaire déduit de (4.5). Dans l'algorithme DH [DAS 87] (voir chapitre 1, paragraphe 1.2.2.1),

$\lim_{k\to\infty} \nu_k^2 \in \left[0, \gamma^2\right]$. Les propriétés (4.6) et (4.8) impliquent que l'ellipsoïde ne change plus ou encore que la mise à jour de l'ellipsoïde s'arrête lorsque l'algorithme a convergé. A cette époque, ces propriétés (à notre connaissance) n'étaient pas démontrées pour les algorithmes du groupe 1.

Dasgupta et Huang [DAS 87] montrent en outre que le coefficient σ_k^2 converge exponentiellement vers l'intervalle $\left[0, \gamma^2\right]$. Autrement dit, la borne utilisée en identification devient la borne sur le coefficient σ_k^2. La vitesse de convergence dépend du coefficient υ choisi *a priori* par l'utilisateur.

Les algorithmes du groupe 2 présentent des propriétés de convergence intéressantes. Néanmoins, aucune relation sur la convergence de la taille de l'ellipsoïde n'est exprimée ce qui est un manque certain pour l'objectif que l'on s'est fixé dans ce travail. En effet, on recherche un ensemble de taille minimale et la convergence (vers cette valeur minimale) de la taille de l'ensemble englobant est une nécessité vraiment primordiale.

On s'intéresse maintenant à l'algorithme SMSA du groupe 1 (voir chapitre 1, paragraphe 1.2.1.3). Cet algorithme est obtenu à partir de l'algorithme SM-WRLS (Set-Membership Weighted Recursive Least Squares) dans lequel une matrice « covariance » est normalisée par la somme des coefficients de pondération antérieurs [NAY 94]. Cette normalisation conduit l'algorithme SMSA à être un cas particulier de la forme générale de l'algorithme OBE donné dans (1.9). Le choix du coefficient de pondération λ_k^* suivant le critère du déterminant (voir la section 1.2.1.3) permet de montrer les propriétés de convergence [NAY 93] données dans le théorème 4.3.

Théorème 4.3

Supposons que la condition PE donnée dans (4.3) soit vérifiée. S'il y a un nombre infini d'observations utiles pour mettre à jour l'ellipsoïde, alors l'algorithme SMSA converge dans le sens où :

(i) $$\lim_{k\to\infty} \left\| c_k - c_{k-1} \right\|_2 = 0 \ ; \tag{4.9}$$

(ii) $$\lim_{k\to\infty} \lambda_k^* = 0 \ ; \tag{4.10}$$

(iii) $$\lim_{k\to\infty} \nu_k^2 \in \left[0, \gamma_k^2\right] \ ; \tag{4.11}$$

(iv) $$\lim_{k\to\infty} \Delta_k^2 < \frac{4}{\varepsilon_1} \lim_{k\to\infty} \sum_{i=k+1}^{k+N_1} \gamma_i^2 \tag{4.12}$$

A la différence des algorithmes du groupe 1, la démonstration du théorème 4.3 provient de l'analyse des conditions de mise à jour de l'ellipsoïde, i.e. $a_0 \geq 0$ dans (1.28). De plus, l'algorithme SMSA nécessite une séquence infinie d'observations. On constate alors que les propriétés de convergence des algorithmes du groupe 2 (concernant la convergence du centre de l'ellipsoïde) sont maintenant valables pour ceux du groupe 1. La condition PE (4.3) est indispensable pour obtenir ces propriétés.

Notons enfin que la propriété sur la convergence du volume de l'ellipsoïde a été démontrée dans [NAY 94].

Théorème 4.4

Supposons que la condition PE de l'équation (4.3) soit vérifiée. S'il y a un nombre infini d'observations utiles pour mettre à jour l'ellipsoïde, alors l'algorithme SMSA converge dans le sens où :

$$\lim_{k\to\infty} \mu_V(k) = 0 \tag{4.13}$$

où $\mu_V(k) = \det\left(\sigma_k^2 \boldsymbol{P}_k\right)$ *est défini dans (1.10) et est proportionnel au volume au carré de l'ellipsoïde englobant.*

La démonstration peut être consultée dans [NAY 94]. Le point crucial dans cette démonstration est l'existence d'une sous-séquence **infinie** de données utilisées pour la mise à jour de l'ellipsoïde. La convergence dans ce cas implique que la séquence d'ellipsoïdes converge vers le « vrai » vecteur de paramètres x^*.

A partir du théorème 4.4, et en analysant les caractéristiques de la séquence d'erreur $\{e_k\}$, [LIN 98] a démontré le théorème 4.5 concernant la borne de l'erreur permettant de garantir la convergence de l'algorithme SMSA.

Théorème 4.5

Supposons que γ^* *connu soit la vraie borne de la séquence d'erreur* $\{e_k\}$. *Choisissons* γ_k *égal à* γ^* *pour tout k dans le processus d'identification. Choisissons* $\varsigma > 0$ *et posons :*

$$D_{\varsigma^+} = \left[\gamma^* - \varsigma, \gamma^*\right]; \quad D_{\varsigma^-} = \left[-\gamma^*, -\gamma^* + \varsigma\right] \tag{4.14}$$

Si la condition d'excitation persistante (4.3) est remplie, et s'il existe $\delta > 0$ tel que :

$$\wp\left(e_k \in D_{\varsigma^+}\right) > \delta \;\; et \;\; \wp\left(e_k \in D_{\varsigma^-}\right) > \delta, \; \forall k \tag{4.15}$$

où $\wp(.)$ dénote une fonction de probabilité. Alors l'algorithme SMSA converge.

Dans ce théorème, la condition de convergence est explicitée en détail : la convergence de l'algorithme SMSA est directement liée aux caractéristiques de l'erreur e_k et à la borne utilisée. La condition (4.15) peut être interprétée par « **l'erreur e_k doit visiter régulièrement sa borne** ». Le théorème 4.5 a conduit au corollaire donné ci-après [LIN 98].

Corollaire

Supposons que la condition d'excitation persistante (4.3) soit remplie. Si la borne utilisée en identification est surestimée, i.e. s'il existe ε tel que $\gamma_k^2 > e_k^2 + \varepsilon^2, \forall k = 1:l$, alors l'ellipsoïde ne converge pas vers un point.

En résumé, l'algorithme SMSA appliqué avec le critère du déterminant permet de donner des résultats systématiques pour le problème de convergence de l'algorithme OBE, à savoir :

- une condition de convergence du centre de l'ellipsoïde ;
- une condition de convergence du volume de l'ellipsoïde.

Or les ellipsoïdes trouvés par tous les algorithmes du groupe 1 sont identiques. En conséquence, les propriétés de convergence de l'algorithme SMSA sont applicables à tous les algorithmes OBE basés sur le critère de minimisation du déterminant.

L'algorithme SMSA doit donc être considéré comme un « pont » entre les algorithmes du groupe 1 et du groupe 2 dans le sens où les résultats de convergence qui ont été établis pour SMSA peuvent être généralisés aux autres algorithmes du groupe 1.

On ne s'intéressera dans la suite de ce chapitre qu'à la convergence des algorithmes OBE du groupe 1, avec le critère du déterminant.

Pour conclure ce paragraphe, les conditions de convergence sont résumées par :

(cd-i) $x^* \in \mathbf{E}_0$ *où* $\mathbf{E}_0$ *est l'ellipsoïde initial ;*

(cd-ii) *l'excitation est persistante d'après la relation (4.3) ;*

(cd-iii) *l'erreur e_k visite régulièrement la borne utilisée lors de la procédure d'identification, ceci d'après la condition (4.15) ;*

[BAI 99] montre en outre que l'interaction entre d_k et e_k est également importante. Idéalement, le vecteur régresseur d_k doit balayer toutes les directions dans l'espace considéré et l'erreur e_k doit atteindre ses bornes $\left[-\gamma_k, \gamma_k\right]$ au moins une fois dans chaque direction. *Si toutes ces conditions sont parfaitement satisfaites, l'ellipsoïde converge vers un singleton*. Rappelons que la condition sur les directions du régresseur d_k n'était pas satisfaite pour les données de l'IFP.

En pratique, les conditions pour que l'ellipsoïde converge vers un point ne sont jamais vérifiées, la raison principale étant que les fichiers de données ont une longueur finie. On peut alors considérer que l'algorithme OBE converge dans le sens où x^* appartient à l'ellipsoïde final et que l'ellipsoïde a été réduit à la taille la plus petite possible.

4.2 Vérification de la persistance de l'excitation

4.2.1 Condition d'une excitation persistante

La persistance de l'excitation (PE) donnée dans la définition 4.1, équation (4.3), est l'une des deux conditions principales permettant de garantir la convergence de l'ellipsoïde telle que définie au paragraphe précédent. La vérification directe de la condition PE (4.3) est souvent délicate. Elle fait en outre apparaître deux réels $\varepsilon_1, \varepsilon_2 > 0$ laissés à la discrétion de l'utilisateur. Pour un système mono-entré/mono-sortie (SISO) linéaire en les paramètres :

$$y_k + a_1 y_{k-1} + \ldots + a_n y_{k-n} = b_1 u_{k-1} + \ldots + b_m u_{k-m} \tag{4.16}$$

Anderson *et al* dans [AND 82] ont reformulé la condition PE à l'aide du théorème 4.6.

Théorème 4.6 ([AND 82])

Considérons un système SISO (4.16). Supposons que $A(z) = z^n + a_1 z^{n-1} + \ldots + a_n$ *et* $B(z) = b_1 z^{n-1} + \ldots + b_m z^{n-m}$ *soient premiers entre eux et que* $\{y_k\}$ *soit une séquence bornée. Alors pour tous* $N_1 \geq 2n + m - 2$ *, si :*

$$0 < \varepsilon_5 I \leq \sum_{k=k_0}^{k_0+N_1} \begin{bmatrix} u_{k+n} \\ \ldots \\ u_{k-m+1} \end{bmatrix} \begin{bmatrix} u_{k+n} & \ldots & u_{k-m+1} \end{bmatrix} \leq \varepsilon_6 I \tag{4.17}$$

alors $$0 < \varepsilon_1 I \leq \sum_{k=k_0}^{k_0+N_1} d_k d_k^T \leq \varepsilon_2 I$$

avec $\varepsilon_5, \varepsilon_6 > 0$ *et* $\forall k_0$.

Ce théorème traduit le fait que pour un système SISO, la condition PE d'une séquence d'entrée implique la condition PE des régresseurs si $A(z)$ et $B(z)$ sont premiers entre eux. Ce théorème permet de choisir une séquence d'entrée appropriée pour l'identification ensembliste ellipsoïdale étudiée dans ce travail. [AND 82] donne également une vérification de la condition PE basée uniquement sur la sortie du système.

Théorème 4.7 ([AND 82])

Considérons un système SISO (4.16). Supposons que $A(z) = z^n + a_1 z^{n-1} + \ldots + a_n$ *et* $B(z) = b_1 z^{n-1} + \ldots + b_m z^{n-m}$ *soient premiers entre eux et que* $\{u_k\}$ *soit une séquence bornée. Alors pour tout* $N_1 \geq 2n + m - 2$*, si :*

$$0 < \varepsilon_7 I \leq \sum_{k=k_0}^{k_0+N_1} \begin{bmatrix} y_{k+m} \\ \ldots \\ y_{k-n+1} \end{bmatrix} \begin{bmatrix} y_{k+m} & \ldots & y_{k-n+1} \end{bmatrix} \leq \varepsilon_8 I \tag{4.18}$$

alors $$0 < \varepsilon_1 I \leq \sum_{k=k_0}^{k_0+N_1} d_k d_k^T \leq \varepsilon_2 I$$

avec $\varepsilon_3, \varepsilon_4 > 0$ *et* $\forall k_0$.

Dans le cas général, si les conditions des théorèmes 4.6 et 4.7 ne sont pas remplies, on doit vérifier la condition PE en utilisant l'équation (4.3). Par exemple, pour les données du robot du LIRMM, on ne peut pas se ramener à un modèle de la forme (4.16), ce qui impose donc d'utiliser la relation (4.3). Quoi qu'il en soit, la vérification de la condition PE nécessite une « comparaison » de matrices et l'utilisation de deux réels positifs à choisir. Ainsi, considérons le cas général (4.3) :

$$\varepsilon_1 I \leq \sum_{k=k_0}^{k_0+N_1} d_k d_k^T \leq \varepsilon_2 I$$

Posons $$\Phi = \begin{bmatrix} d_{k_0}^T \\ \ldots \\ d_{k_0+N_1}^T \end{bmatrix}, \quad \Phi \in \mathbb{R}^{N_1 x p} \tag{4.19}$$

où d_k est le régresseur à l'instant k. On obtient facilement que :

$$\Phi^T\Phi = \begin{bmatrix} d_{k_0} & \dots & d_{k_0+N_1} \end{bmatrix} \begin{bmatrix} d_{k_0}^T \\ \dots \\ d_{k_0+N_1}^T \end{bmatrix} = \sum_{k=k_0}^{k_0+N_1} d_k d_k^T \tag{4.20}$$

Réalisons une décomposition en valeurs singulières de la matrice Φ, ce qui donne :

$$\Phi = UDV^T \tag{4.21}$$

où $U \in \mathbb{R}^{N\text{x}N_1}$, $V \in \mathbb{R}^{pxp}$ sont des matrices orthogonales et :

$$D = \begin{bmatrix} \tau_1 & 0 & & 0 \\ 0 & \dots & & \\ & & \tau_r & \\ 0 & & & 0 \end{bmatrix}, \quad D \in \mathbb{R}^{N_1xp}$$

avec $\tau_1 \geq \tau_2 \geq \dots \geq \tau_r \geq 0$ les valeurs singulières de la matrice Φ. Si Φ est de rang plein, i.e. $r = p$, on a :

$$\tau_p^2 I \leq \Phi^T\Phi = V \begin{bmatrix} \tau_1^2 & & 0 \\ & \dots & \\ 0 & & \tau_p^2 \end{bmatrix} V^T \leq \tau_1^2 I \tag{4.22}$$

De (4.20) et (4.22), on déduit :

$$\varepsilon_1 I = \tau_p^2 I \leq \Phi^T\Phi = \sum_{k=k_0}^{k_0+N_1} d_k d_k^T \leq \tau_1^2 I = \varepsilon_2 I \tag{4.23}$$

En conclusion, si la matrice Φ définie dans (4.19) est de rang plein, la condition de persistance de l'excitation est remplie. Cependant, cette vérification ne donne pas une information sur la « qualité » effective de la persistance de l'excitation.

En pratique, la condition PE est vérifiée en calculant le conditionnement $cond(\Phi) \geq 1$ de la matrice Φ. $cond(\Phi)$ mesure le rapport entre les valeurs singulières maximum τ_1 et minimum τ_r non nulle. Notons que pour Φ de rang plein, $\tau_r = \tau_p \neq 0$. $cond(\Phi)$ permet dans tous les cas d'évaluer la qualité de l'excitation (persistante ou non).

4.2.2 Création d'une excitation adéquate pour satisfaire la convergence de l'algorithme

Rappelons que pour que l'algorithme OBE converge, dans le cas idéal, le vecteur régresseur d_k doit balayer toutes les directions de l'espace paramétrique considéré, puisque d_k

est le vecteur orthogonal à la bande $\mathbf{B}_k$. A partir d'une analyse géométrique, on perçoit que l'angle entre deux régresseurs **successifs** (noté θ_{dd_+}) est un indicateur sur la potentialité de convergence de l'algorithme. Si deux bandes consécutives sont orthogonales (leurs régresseurs étant également orthogonaux ce qui pour ces deux régresseurs signifie qu'on a un conditionnement parfait), la réduction de la taille de l'ellipsoïde englobant est plus rapide. Cela est illustré dans la figure 4.1 sur laquelle on constate bien qu'une réduction rapide de la taille de l'ellipsoïde est obtenue lorsque les bandes successives sont (quasiment) orthogonales deux à deux, ce qui signifie que la condition d'excitation persistante PE (4.3) est également parfaitement remplie.

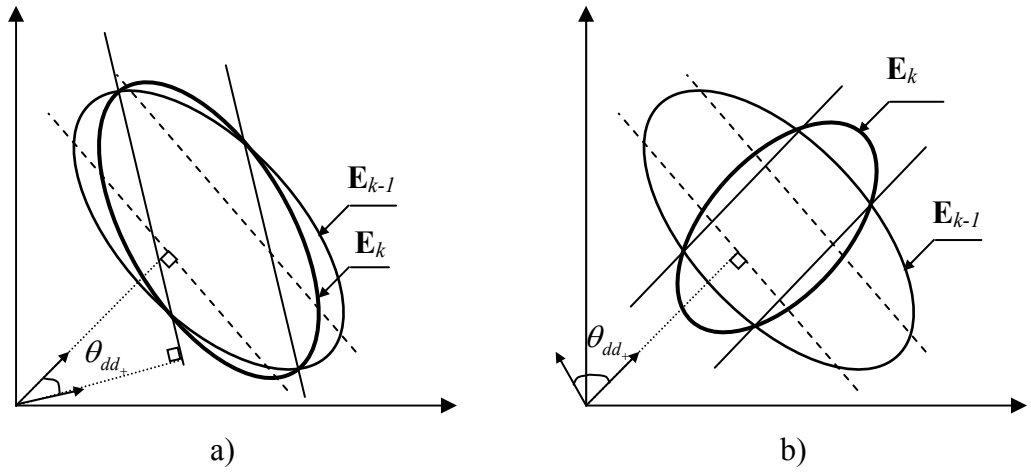


Figure 4.1 : Mise à jour de l'ellipsoïde et angle entre deux régresseurs successifs θ_{dd_+}.

La difficulté est donc de créer un fichier de données contenant des régresseurs successifs (quasiment) orthogonaux, ce qui permet de garantir la condition PE (4.3). Considérons le système SISO linéaire en les paramètres (4.16), le vecteur régresseur étant $d_k = \begin{bmatrix} y_{k-1} & \dots & y_{k-n} & u_{k-1} & \dots & u_{k-m} \end{bmatrix}^T$. Nous proposons une procédure « en ligne » permettant de construire un fichier de données remplissant au mieux la condition sur les bandes énoncée ci-dessus.

Procédure 4.1

Supposons qu'à l'instant k, on connaisse les observations et les ellipsoïdes antérieurs. La procédure de construction « en ligne » du régresseur est :

1. *former le régresseur* $d_k = \begin{bmatrix} y_{k-1} & \dots & y_{k-n} & u_{k-1} & \dots & u_{k-m} \end{bmatrix}^T$;

2. *prédire la sortie* $\hat{y}_k = d_k^T \hat{x}_{k-1}$ *où* $\hat{x}_{k-1} \equiv c_{k-1}$ *est le centre de l'ellipsoïde* $\mathbf{E}_{k-1}$ *;*
3. *construire le régresseur prédit* $\hat{d}_{k+1} = \left[\hat{y}_k \quad \ldots \quad y_{k+1-n} \quad u_k \quad \ldots \quad u_{k+1-m}\right]^T$ *et calculer* u_k *tel que* $d_k \perp \hat{d}_{k+1}$ *;*
4. *si* $|u_k| > \max(u)$*, où* $\max(u)$ *est la valeur maximale de u applicable au procédé, saturer* u_k *à la valeur* $\max(u)$ *;*
5. *ajouter* u_k *à la séquence d'entrée.*

Remarques

- le produit scalaire $(\bullet)$ de deux vecteurs *a*, *b* est défini par :

$$a \bullet b = \|a\|_2 \|b\|_2 \cos\theta_{ab} = a^T b \tag{4.24}$$

où θ_{ab} est l'angle entre les deux vecteurs. Alors pour $d_k \perp \hat{d}_{k+1}$, on doit avoir $\theta_{d_k \hat{d}_{k+1}} = 90°$ ou $\cos\theta_{d\hat{d}_+} = 0$. $\hat{u}_k$ est alors **une** solution de l'équation $d_k^T \hat{d}_{k+1} = 0$. On aurait pu également utiliser le produit vectoriel qui fait apparaître $\sin\theta_{d\hat{d}_+} = 1$;

- cette procédure ne permet **pas de garantir que l'on balaye tout l'espace paramétrique** ;
- avec u_k calculé par la procédure 4.1 et y_k mesuré sur le procédé, les deux régresseurs d_k, et d_{k+1} réels ne sont pas parfaitement orthogonaux. Ceci est principalement dû à la saturation imposée (si besoin) à u_k, au bruit qui entache y_k, etc. Cependant, l'angle θ_{dd_+} sera augmenté autant que faire se peut, ce qui permet de garantir la condition PE.

En pratique, cette procédure coûte cher en temps calcul car l'identification ensembliste ellipsoïdale et le calcul de l'entrée à appliquer sont effectués conjointement et dépendent l'un de l'autre. Pour pallier ces difficultés, on propose la procédure 4.2 qui sera effectuée hors-ligne.

Procédure 4.2

1. *exciter le procédé avec une séquence SBPA et faire l'acquisition des données ;*
2. *(pré)-estimer le vecteur de paramètres du modèle* $\hat{x}$ *sur ce fichier de données en utilisant une méthode quelconque d'identification (par exemple, moindres carrés) ou des connaissances a priori ;*
3. *calculer hors ligne la séquence* $\{u_k\}$ *telle que* $\hat{d}_k \perp \hat{d}_{k+1}$ *où :*

$$\hat{d}_k = \left[\hat{y}_{k-1} \quad \ldots \quad \hat{y}_{k-n} \quad u_{k-1} \quad \ldots \quad u_{k-m}\right]^T \text{ et } \hat{y}_j = \hat{d}_j^T \hat{x} \ ;$$

4. *si* $|u_k| > \max(u)$*, où* $\max(u)$ *est la valeur maximale de u applicable au procédé, saturer* u_k *à la valeur* $\max(u)$ *;*
5. *appliquer la séquence* $\{u_k\}$ *obtenue au procédé et réaliser sur le nouveau fichier de données l'identification ensembliste ellipsoïdale.*

Avec la procédure 4.2, on crée une séquence d'entrée qui produit deux régresseurs prédictifs orthogonaux. Comme pour la procédure 4.1, deux régresseurs successifs réels ne sont pas parfaitement orthogonaux mais l'angle $\theta_{d_k d_{k+1}}$ entre eux sera augmenté autant que faire se peut. Ceci sera vérifié en simulation. Ici encore, on ne garantit pas de balayer toutes les directions.

Lorsque le jeu de données est disponible et n'a pas été obtenu suivant l'une des procédures 4.1 ou 4.2 (c'est le cas par exemple, des données du robot parallèle), on peut améliorer la vitesse de convergence de l'algorithme OBE en réorganisant « raisonnablement » l'ordre des régresseurs (si cela est possible compte tenu du modèle retenu pour l'identification). On essaie alors d'obtenir un angle θ_{dd_+} maximisé (idéalement proche de 90°) pour les premiers régresseurs, ce qui revient en fait à choisir (et réorganiser) les lignes de la matrice Φ (4.19) les plus indépendantes possibles les unes des autres.

4.3 Choix de la borne pour l'identification

Les techniques (d'identification ou d'observation) à « erreur inconnue mais bornée » visent à s'affranchir des hypothèses parfois irréalistes (et la plupart du temps non vérifiables) sur la distribution de l'erreur. Cet objectif cache en fait une difficulté latente puisqu'elles s'avèrent être extrêmement sensibles au choix de la borne γ_k. En outre, comme on l'a indiqué dans le paragraphe 4.1, la borne choisie pour la procédure d'identification est l'un des deux facteurs primordiaux influençant la convergence de l'algorithme OBE.

En théorie, la borne γ_k sur l'erreur e_k peut avoir une valeur différente à chaque instant k. Rappelons que cette borne intervient directement sur la largeur de la bande $\mathbf{B}_k$, donc sur la taille de l'ellipsoïde $\mathbf{E}_k$. Cependant, déterminer γ_k à chaque instant est irréaliste, surtout dans un contexte industriel. En outre, on ne peut pas toujours caractériser l'erreur e_k, ce qui rend impossible le choix optimal (au sens où il n'est pas surestimé) de γ_k. Pour s'affranchir de cette difficulté, en pratique, on utilise une borne constante γ pour toutes les erreurs, i.e. $\gamma_k = \gamma$,

$\forall k = 1:l$. La borne γ est évidement, en général, plus grande que la « vraie » borne γ_k. En conséquence, ce choix de la borne conduit à des bandes **B** plus lâches. Ceci est l'une des raisons qui induit un ellipsoïde englobant ne pouvant pas converger vers un point (ellipsoïde de taille nulle).

Même si l'on accepte de prendre une borne constante pour toutes les erreurs, le choix de γ reste délicat. En effet, si γ est sous-estimé, on est confronté à de nombreuses inconsistances des mesures se manifestant par une intersection vide entre l'ellipsoïde $\mathbf{E}_{k\text{-}1}$ et la bande $\mathbf{B}_k$, ce qui conduit à arrêter l'algorithme prématurément, ou bien à obtenir un ellipsoïde « trop petit ». Pour des données simulées, on a même constaté que ce cas peut conduire à un ellipsoïde ne contenant pas le « vrai » vecteur de paramètres. Si par contre γ est surestimé, on constate un gel de la mise à jour de l'ellipsoïde. Ce dernier est de « grande » taille conduisant à une interprétation des paramètres pouvant ne plus avoir de sens physique (par exemple, on sait que l'un des paramètres x_i est positif, et l'ellipsoïde trouvé possède des x_i négatifs).

Le problème du choix de la borne n'est, en général, pas abordé dans littérature du domaine, les auteurs se contentant de donner la valeur retenue pour γ sans exposer la méthodologie ayant conduit à ce choix. Dans la suite, nous présentons des méthodes dites « classiques » de choix de la borne γ. L'algorithme OBE-ABE (OBE-Automatic Bound Estimation) est ensuite présenté. Il permet de réduire automatiquement la borne durant le processus d'identification. Enfin, nous proposons une méthode de choix de la borne basée sur une analyse préalable de l'erreur pré-estimée à partir d'une mesure du Kurtosis.

4.3.1 Méthodes « classiques » de choix de la borne

La démarche la plus naturelle consiste à exploiter des connaissances *a priori* sur les caractéristiques de l'erreur e_k. Cette information peut être obtenue entre autre, par une analyse des documents constructeurs des capteurs. Néanmoins, il faut également tenir compte de l'erreur de modélisation, ce qui s'avère délicat, et bien souvent impossible. En conséquence, la démarche la plus communément utilisée en estimation ensembliste est d'essayer de caractériser e_k (voir par exemple la thèse de doctorat [SED 98] dans laquelle une analyse fine des e_k pour l'application considérée est menée). On prend ensuite :

$$\gamma = \max\left(\left|e_k\right|\right), \quad \forall k = 1:l \tag{4.25}$$

Dans le cas où l'on n'a pas d'information pour caractériser e_k, on doit choisir heuristiquement (plus exactement « manuellement et par tâtonnement » !) une valeur pour la borne γ et la faire varier suivant des critères plus ou moins subjectifs. Par exemple, si le pourcentage d'intersections vides entre la bande et l'ellipsoïde devient « important », la borne retenue γ est déclarée sous-estimée. On doit donc l'augmenter et recommencer le processus d'identification. A l'inverse, si la borne est surestimée, le taux de données utilisées (voir paragraphe 2.1.2) pour mettre à jour l'ellipsoïde sera trop « faible » et la taille de l'ellipsoïde obtenu ne diminue plus. L'ellipsoïde obtenu est alors surdimensionné. La borne retenue γ est déclarée surestimée. On doit donc la diminuer et recommencer le processus d'identification.

On voit donc qu'on doit essayer heuristiquement plusieurs valeurs de la borne jusqu'à obtenir une borne « acceptable », compte tenu du taux de données utilisées et du pourcentage d'intersections vides choisis par l'utilisateur. Dans un contexte industriel, cette démarche par tâtonnement n'est pas acceptable : elle est un **réel frein au transfert industriel** des méthodes ensemblistes. De plus, elle coûte cher puisque l'on doit réinitialiser l'algorithme d'identification à chaque modification de γ.

La procédure 4.3 propose une autre méthode de choix de la borne γ. Cette démarche présente l'avantage de ne pas utiliser un réglage par tâtonnement de γ. Néanmoins, elle conduit très souvent à une valeur surestimée de la borne, qui en pratique est de nouveau diminuée par tâtonnement ! Elle nécessite en outre une première estimation $\hat{x}$ (ponctuelle et non ensembliste) du vecteur de paramètres.

Procédure 4.3

1. *déterminer par une technique quelconque (y compris en utilisant des informations a priori) une estimation* $\hat{x}$ *du vecteur de paramètres ;*
2. *calculer la séquence d'erreurs estimées* $\{\hat{e}_k\}$ *où* $\hat{e}_k = y_k - d_k^T \hat{x}$ *;*
3. *prendre* $\gamma_{clas} = max\left(\left|\hat{e}_k\right|\right)$.

4.3.2 Algorithme OBE-ABE

L'algorithme OBE-ABE proposé par Lin *et al.* [LIN 98] permet de réduire automatiquement la borne sur l'erreur pendant la procédure d'identification ce qui, d'un point de vue pratique, est particulièrement attrayant puisque cela semble résoudre le problème du choix de la borne.

4.3.2.1 Algorithme

L'algorithme OBE-ABE est en fait l'utilisation conjointe de l'algorithme SMSA[1] (voir paragraphe 1.2.1.3) et d'une procédure (dite ABE) de réduction de la borne de l'erreur γ. On présente ici cet algorithme lorsqu'il est associé à l'algorithme SMSA. Rappelons que dans ce cas, les coefficients de pondération sont $\alpha_k = 1-\lambda_k, \beta_k = \lambda_k$ où $\lambda_k^* > 0$ implique que l'observation à l'instant k est sélectionnée pour mettre à jour l'ellipsoïde.

Algorithme OBE-ABE

1. *Choisir a priori $\varsigma > 0$ et un nombre entier l_0 ;*
2. *pour $k = 1 : l$, l étant la longueur du fichier de données :*

> ***si*** $\lambda_k^* > 0$, *calculer l'ellipsoïde englobant par l'algorithme SMSA (OBE)*
>
> ***sinon***
>
> ***si*** *la fenêtre temporelle $\mathcal{F}$ de longueur l_0 dans laquelle $\lambda_j^* = 0,\ \forall j \in \mathcal{F}$ existe :*
>
> *diminuer la borne suivant* $\gamma_k^2 = \begin{cases} \gamma_{k-1}^2 - d_J & si\ d_J > 0 \\ \gamma_{k-1}^2 & sinon \end{cases}$ (4.26)
>
> *avec* $d_J = \dfrac{\sigma_{J-1}^2 g_J}{p} - \varsigma\left(2\gamma_{k-1} - \varsigma\right);\ J = \underset{j\in\mathcal{F}}{argmax}\left\{v_j^2\right\}$
>
> ***sinon***
>
> *passer à l'instant suivant*
>
> ***fin si***
>
> ***fin si***

On a : $\sigma_k^2 = \left(1-\lambda_k\right)\sigma_{k-1}^2 + \lambda_k\gamma_k^2 - \dfrac{\left(1-\lambda_k\right)\lambda_k v_k^2}{\left(1-\lambda_k\right)+\lambda_k g_k}$; $g_k = d_k^T P_{k-1} d_k$; $v_k = y_k - d_k^T c_{k-1}$;

$\mathcal{F} = \left[k - l_0 + 1, k\right]$ *et p, la dimension du vecteur paramètre.*

Une illustration géométrique de l'algorithme OBE-ABE est présentée dans la figure 4.2. Dans la fenêtre $\mathcal{F}$, l'ellipsoïde n'est pas mis à jour ($\mathbf{E}_{k-1}$ est contenu dans $\mathbf{B}_k$), ou encore aucune donnée n'est sélectionnée puisqu'elle n'apporte pas de nouvelle information. Cela correspond à une borne courante surestimée. La partie « ABE » de l'algorithme réduit alors cette borne.

[1] ou de tout autre algorithme OBE du groupe 1 associé au critère du déterminant.

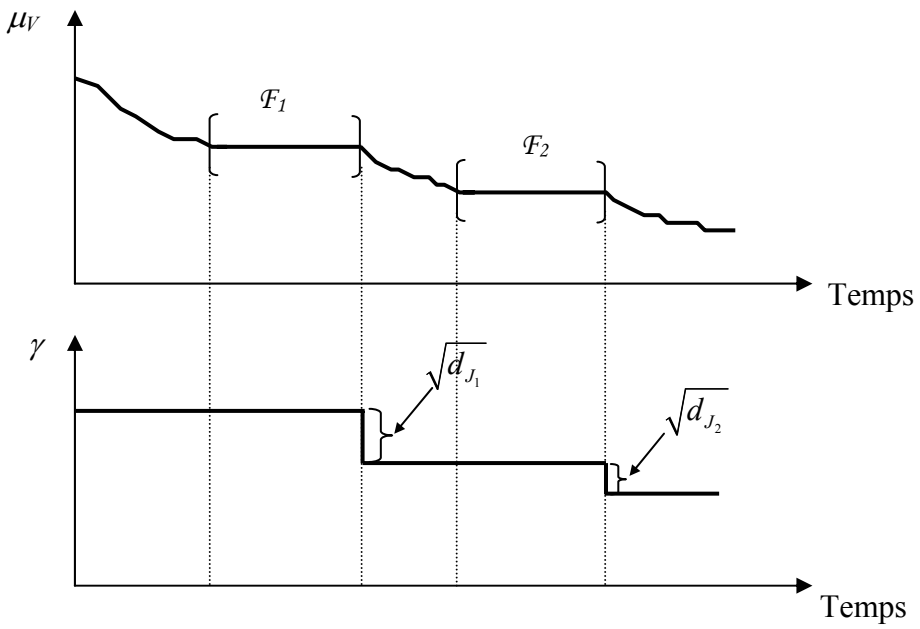


Figure 4.2 : Illustration de l'algorithme OBE-ABE.

L'algorithme OBE-ABE est initialisé avec une borne γ_0 choisie *a priori*, et surestimée. L'ellipsoïde $\mathbf{E}_0$ doit être grand pour contenir le vecteur de « vrais » paramètres. Cet algorithme dépend des paramètres ς, l_0 choisis par l'utilisateur. Les caractéristiques de l'algorithme OBE via les coefficients de pondération α_k et β_k influent sur la vitesse de réajustement de la borne.

4.3.2.2 Propriétés de l'algorithme OBE-ABE

Les propriétés de convergence de l'algorithme OBE-ABE sont données dans le théorème 4.8 ci-après [LIN 98].

Théorème 4.8

Supposons que $\{e_k\}$ *soit une séquence i.i.d. admettant pour bornes inconnues* γ^* *et* $-\gamma^*$. *Si la condition d'excitation persistante PE (4.3) est vérifiée, et si pour* $\varsigma > 0$ *donné, il existe un nombre* $\delta > 0$ *tel que :*

$$\wp\left(e_k \in D_{\varsigma^+}\right) > \delta \quad et \quad \wp\left(e_k \in D_{\varsigma^-}\right) > \delta, \ \forall k \tag{4.15}$$

où $D_{\varsigma^+}, D_{\varsigma^-}$ *sont définis dans (4.14), alors l'algorithme OBE-ABE converge.*

Démonstration

Le lecteur pourra consulter la démonstration dans [LIN 98].

Remarque

On constate que l'algorithme SMSA (ou n'importe quel algorithme OBE du groupe 1 avec critère du déterminant) et l'algorithme OBE-ABE requièrent les mêmes conditions sur la persistance de l'excitation et sur la borne de l'erreur pour garantir la convergence telle que définie précédemment. La différence principale entre eux est que l'algorithme OBE-ABE est initialisé avec une borne surestimée et qu'il la réduit automatiquement tandis que l'algorithme SMSA (ou OBE du groupe 1) utilise une borne qui n'évolue pas durant tout le processus d'identification.

On s'intéresse maintenant au choix des paramètres l_0, et ς ainsi qu'à leur influence sur la convergence de l'algorithme. Le paramètre l_0 définit la longueur de la fenêtre temporelle $\mathcal{F}$ à la fin de laquelle on réalise un réajustement de γ_k. l_0 doit tout d'abord être suffisamment « grand » pour assurer que dans cette fenêtre, l'ellipsoïde n'est plus mis à jour du fait d'une borne courante surestimée et non de données (localement) mal excitées. Naturellement, l_0 est limité par la longueur l du fichier de données et le besoin de réajuster la borne au plus vite afin de tendre vers une borne ni surestimée, ni sous-estimée. Un choix de l_0 trop « petit » peut entraîner un effet pervers de la diminution automatique de la borne γ_k : cette dernière peut devenir sous-estimée ce qui peut conduire l'algorithme SMSA (ou OBE) à s'arrêter pour cause d'intersection vide entre la bande $\mathbf{B}_k$ et l'ellipsoïde $\mathbf{E}_{k\text{-}1}$. [JOA 97] propose d'utiliser une fenêtre $\mathcal{F}$ de longueur variable. On initialise l'algorithme OBE-ABE avec une « petite » fenêtre et on augmente graduellement l_0. Cette démarche permet de diminuer rapidement la borne γ_k (souvent trop surestimée) au début de la procédure d'identification puis de réduire de façon moins importante cette borne au cours du temps. [JOA 97] propose une formule expérimentale pour la longueur de la fenêtre $\mathcal{F}$:

$$l_0(k) = -4p\log\gamma_k + l_0(k-1) \tag{4.27}$$

où p est la dimension du vecteur de paramètres.

Le deuxième paramètre influençant la réduction de γ_k (donc la convergence de l'algorithme OBE-ABE) est ς. Le réajustement d_J (4.23) de γ_k^2 est :

$$d_J = \frac{\sigma_{J-1}^2 g_J}{p} - \varsigma\left(2\gamma_{k-1} - \varsigma\right) = d_{J_1} - d_{J_2} \tag{4.28}$$

d_{J_1} dépend en particulier de la longueur des axes de l'ellipsoïde courant (via les grandeurs σ_{J-1} et g_J) tandis que d_{J_2} dépend de la borne précédente et du paramètre ς. Au

début de processus d'identification, lorsque l'ellipsoïde est de « grande dimension », d_J est grand et dominé par d_{J_1}. Quand l'ellipsoïde devient « petit », le terme d_J se trouve influencé par d_{J_2}, donc par ς. Ainsi, ce paramètre a une influence sur le réajustement fin de la borne γ_k. Notons que d_{J_2} est une fonction du 2ème degré de ς.

[LIN 98] a proposé une méthode de choix de ces deux paramètres l_0, ς. On prend tout d'abord une valeur « petite » positive pour ς. Avec ς choisi, on estime la « queue » de la distribution des $\{e_k\}$ noté δ dans l'équation (4.15). La longueur l_0 de la fenêtre $\mathcal{F}$ est ensuite choisie telle que $\left[1-(1-\delta)^{l_0}\right]^{l/l_0} \approx 1$. Le terme $\wp_\delta \triangleq 1-(1-\delta)^{l_0}$ apparaît dans la démonstration de [LIN 98] comme étant la probabilité pour que γ_k devienne la vraie borne sur l'erreur e_k. Ainsi le choix de l_0 et ς dans cette méthode nécessite une information sur la queue de la distribution des $\{e_k\}$.

En pratique, dans un souci de simplicité et de robustesse, **nous proposons de choisir ς comme un pourcentage de la borne γ_k**. Ainsi, la décroissance de γ_k entraîne la décroissance de ς, donc du terme d_{J_2} dans (4.28). En conséquence, on obtient un réajustement de plus en plus fin. **Une autre amélioration proposée concerne le moment où l'on applique la nouvelle borne**. Dans l'algorithme OBE-ABE originel, on applique γ_k après la fenêtre $\mathcal{F}$, donc juste après le réajustement de la borne. En fait, dans la fenêtre $\mathcal{F}$, aucune observation n'apporte d'information puisque l'ellipsoïde n'est pas mis à jour puisque la borne précédente γ_{k-1} est surestimée. Il semble donc logique d'utiliser γ_k non pas à la fin de $\mathcal{F}$ mais au début de cette fenêtre en effectuant un retour en arrière (correspondant à un nouveau traitement des données de $\mathcal{F}$ mais avec la borne γ_k). Cela permet d'utiliser les données de la fenêtre $\mathcal{F}$ pour mettre à jour l'ellipsoïde et d'augmenter la vitesse de convergence de l'algorithme.

L'algorithme OBE-ABE semble être une excellente solution pour pallier le problème du choix de la borne dans l'approche ensembliste ellipsoïdale. La condition fondamentale permettant de garantir sa convergence (avec notamment $\gamma_k \to \gamma^*$) est que l'erreur e_k ait souvent une valeur proche de sa « vraie » borne. En outre, comme on l'a déjà signalé, la vitesse de convergence de cet algorithme dépend fortement de l_0 et ς dont le choix reste délicat et souvent empirique.

Par conséquent, l'utilisateur, après avoir été confronté au choix direct de la borne γ pour les méthodes dites « classiques » se trouve maintenant à devoir régler ces deux paramètres l_0 et ς, ce qui somme toute n'est pas plus simple !

4.3.3 Méthode basée sur un calcul du Kurtosis de la séquence d'erreur

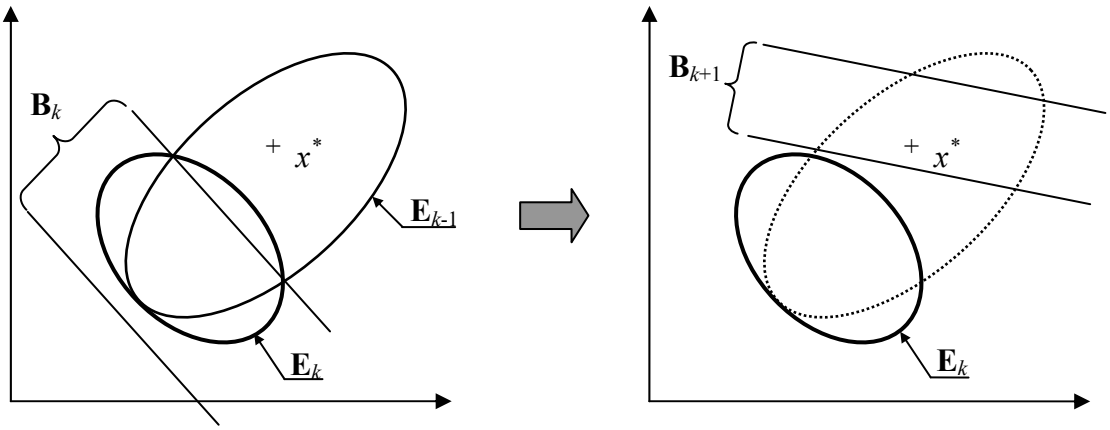
En pratique, les démarches proposées aux paragraphes 4.3.1 et 4.3.2 respectivement sont souvent inexploitables car :

- le bruit e_k n'atteint pas « assez souvent » la borne γ utilisée en identification même si cette dernière est juste (c'est le cas par exemple lorsque e_k suit une distribution gaussienne et que $\gamma = \max(|e_k|)$). En conséquence, l'algorithme OBE ne converge pas vers l'ellipsoïde de taille minimale. L'utilisation de l'algorithme OBE-ABE dans ce cas peut conduire à une borne sous-estimée ;
- on est en présence de mesures aberrantes qui n'ont pas été éliminées lors de l'inspection du fichier de données. Ces mesures aberrantes produisent des valeurs de e_k artificiellement grandes. La technique « classique » de choix de γ conduit alors à une borne surestimée pour la majeure partie des données. Une diminution de γ à l'aide de l'algorithme OBE-ABE produit quant à elle des intersections vides ($\mathbf{E}_{k\text{-}1} \cap \mathbf{B}_k = \varnothing$) fréquentes.

Dans le cas où des mesures aberrantes existent, le calcul d'un ellipsoïde englobant consistant implique d'éliminer strictement ces données aberrantes. Une inspection préalable du fichier de données est donc indispensable, avec le risque d'éliminer des données portant de l'information.

Une autre démarche plus naturelle et mise en oeuvre par plusieurs utilisateurs de l'approche ensembliste ellipsoïdale est de détecter les intersections vides entre la bande et l'ellipsoïde, puis d'éliminer les données correspondantes. Après avoir retiré ces données réputées aberrantes (puisqu'elles ont conduit à un instant donné à une intersection vide), on doit réinitialiser l'algorithme afin d'éviter une contribution erronée de ces données sur l'ellipsoïde obtenu. Notons néanmoins que le calcul séquentiel de l'ellipsoïde implique qu'une intersection vide à l'instant k ne signifie pas pour autant que l'observation considérée soit aberrante au sens strict du terme. Ceci est illustré dans la figure 4.3 où x^* représente le vecteur des « vrais » paramètres. A l'instant k, $\mathbf{B}_k$ ne contient pas x^* car on est en présence d'une mesure aberrante. Cependant, la figure 4.3.a montre que $\mathbf{E}_k \supset \mathbf{E}_{k-1} \cap \mathbf{B}_k \neq \varnothing$, donc on ne peut pas détecter cette mesure aberrante. A l'inverse, à l'instant $k+1$, la mesure n'est pas

aberrante ($x^* \in \mathbf{B}_{k+1}$) mais $\mathbf{E}_k \cap \mathbf{B}_{k+1} = \varnothing$ car $\mathbf{E}_k$ a été déplacé lors de l'étape d'identification précédente. Cette situation a été observée sur des données simulées. Elle doit également pouvoir se produire pour des données réelles.



a) $x^* \notin \mathbf{B}_k$; $\mathbf{E}_{k-1} \cap \mathbf{B}_k \neq \varnothing$ b) $x^* \in \mathbf{B}_{k+1}$; $\mathbf{E}_k \cap \mathbf{B}_{k+1} = \varnothing$

Figure 4.3 : Détection fausse de l'observation aberrante ; (x^* est le vecteur des vrais paramètres)

En conséquence, cette démarche peut induire des confusions : des données aberrantes sont utilisées pour calculer l'ellipsoïde (figure 4.3.a) tandis que des données non aberrantes sont rejetées (figure 4.3.b). En outre, elle est coûteuse en temps calcul, en particulier lorsque la longueur du fichier de données est importante et/ou lorsque l'on applique le principe de recirculation des données.

Nous proposons dans la suite de ce manuscrit une méthode visant à éliminer plus rapidement les données réputées aberrantes et à obtenir une borne sur l'erreur basée sur un calcul du Kurtosis de la séquence d'erreur.

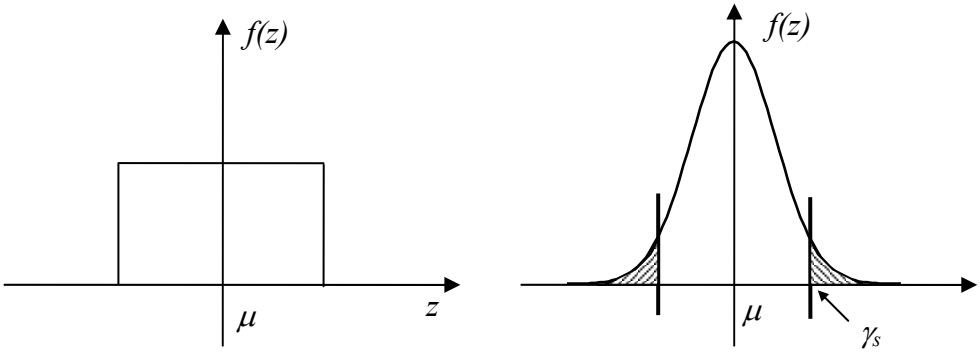
4.3.3.1 Définition du Kurtosis d'une distribution

Le (coefficient de) Kurtosis d'une distribution z correspond au moment d'ordre 4 :

$$kur(z) = \frac{E\left[\left(z-\mu\right)^4\right]}{\sigma^4} \tag{4.29}$$

où μ est la moyenne, σ est l'écart type et $E(.)$ désigne l'opérateur espérance mathématique. Le Kurtosis permet en fait de mesurer le degré « d'écrasement » de la distribution (la distribution est-elle « plate » ou « piquée » ?). Il donne une idée approximative sur la forme de la queue d'une distribution. Notons que le Kurtosis d'une distribution gaussienne vaut 3, celui d'une distribution uniforme est de 1.8 et une distribution exponentielle a un Kurtosis de 6. La Figure

4.4 représente des distributions uniforme (a) et gaussienne (b). Une distribution telle que $kur(z) > 3$ a une queue de distribution moins « lourde » que la distribution gaussienne.



a) Distribution uniforme, *kur*(*z*) = 1.8 b) Distribution gaussienne, *kur*(*z*) = 3

Figure 4.4

Dans le cadre de notre étude, pour une séquence $\{e_k\}$, plus le Kurtosis estimé est grand, plus la probabilité pour que e_k atteigne un voisinage la borne $\max(|e_k|)$ est faible. Ainsi, l'analyse du Kurtosis de la séquence $\{e_k\}$, permet d'évaluer si e_k visite souvent sa borne.

En traçant l'histogramme de la séquence d'erreur $\{e_k\}$ (supposée approximativement symétrique), on peut tronquer les deux queues de l'histogramme en utilisant le seuil γ_s. On obtient alors une nouvelle séquence notée $\left\{e_k^{tr}\right\}$ dont l'histogramme est moins « pointu », conduisant à une diminution du Kurtosis. Le seuil γ_s devient naturellement la borne de la séquence tronquée. Il est aisé de constater que la séquence tronquée $\left\{e_k^{tr}\right\}$ visite plus souvent sa borne que la séquence initiale $\{e_k\}$. Pour quelques distributions particulières, on peut établir en fonction du seuil γ_s choisi, une relation entre le Kurtosis de la distribution tronquée et le taux de données éliminées. Le théorème 4.9 donne cette relation dans le cas d'une distribution gaussienne.

Théorème 4.9

Le Kurtosis d'une distribution gaussienne tronqué z_{tr} et le taux de données éliminées R_e sont donnés par :

$$kur(z_{tr}) = \frac{3 - \sqrt{\frac{2}{\pi}}\alpha q e^{-\alpha^2/2}\left(\alpha^2 + 3\right)}{\left(1 - \sqrt{\frac{2}{\pi}}\alpha q e^{-\alpha^2/2}\right)^2} \tag{4.30}$$

$$R_e(\%) = 100\left(1 - \frac{1}{q}\right) \tag{4.31}$$

avec $$q = \frac{1}{erf\left(\alpha / \sqrt{2}\right)} \;;\; erf\left(\alpha / \sqrt{2}\right) = \int_{-\alpha\sigma}^{\alpha\sigma} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} dt \tag{4.32}$$

où $\gamma_s = \alpha\sigma$ *est le seuil utilisé pour tronquer les données z.*

Démonstration

Elle peut être consultée dans l'annexe D.

Les relations (4.30) et (4.31) sont tracées sur la figure 4.5 lorsque α varie de 0.5 à 5. La distribution initiale est gaussienne d'écart type $\sigma = 1$ et de moyenne $\mu = 0$.

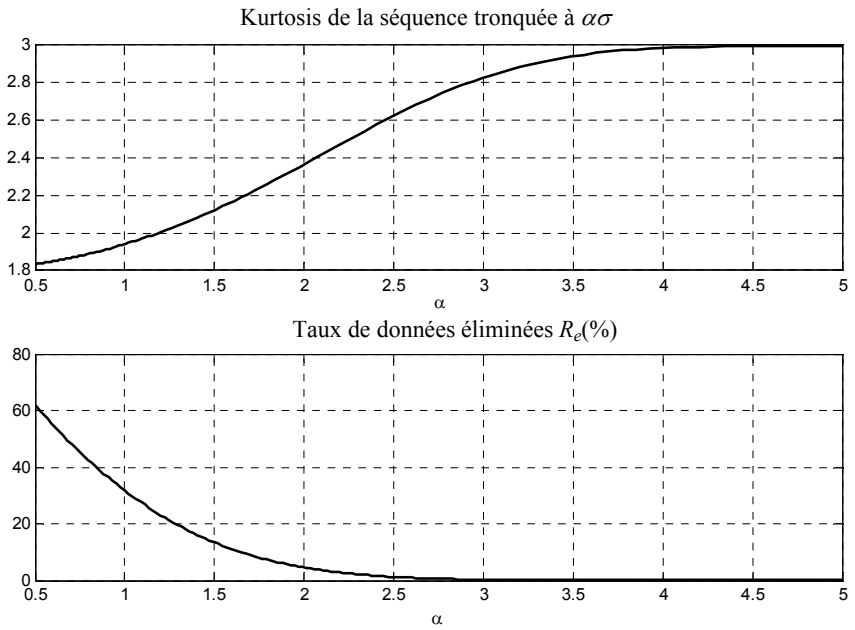


Figure 4.5 : Kurtosis d'une distribution gaussienne tronquée et taux de données éliminées R_e(%) en fonction de α.

4.3.3.2 Troncature de la séquence d'erreur

On applique maintenant la démarche de troncature, à base de la mesure du Kurtosis, sur la séquence d'erreur $\{e_k\}$ afin de prétraiter les données avant l'identification ensembliste ellipsoïdale. Supposons tout d'abord que $\{e_k\}$ dans (1.1) est disponible. En analysant la valeur du Kurtosis de $\{e_k\}$, on peut prédire la possibilité d'atteindre la borne max($|e_k|$) de l'erreur e_k. Dans le cas de mesures aberrantes (mais idéalement pas trop nombreuses !), on va avoir des

valeurs importantes de l'erreur situées sur les queues de l'histogramme de la séquence $\{e_k\}$. Il en résulte une valeur de $kur\{e_k\}$ « grande ». En tronquant les queues de l'histogramme de $\{e_k\}$, et en rejetant les données correspondant à ces erreurs, on obtient les résultats suivants :

- toutes les données réputées aberrantes et présentant de « grandes » erreurs e_k sont rapidement éliminées ;
- le seuil γ_s servant à tronquer l'histogramme de l'erreur devient la borne de la séquence d'erreur tronquée $\left\{e_k^{tr}\right\}$;
- le Kurtosis de $\left\{e_k^{tr}\right\}$ a été réduit par rapport à $kur\left(\left\{e_k\right\}\right)$. L'histogramme de $\left\{e_k^{tr}\right\}$ est plus « plat » et plus compact, ce qui implique que la borne γ_s sera plus fréquemment visitée pour la séquence tronquée que la borne $\max\left(\left|e_k\right|\right)$ de la séquence initiale $\{e_k\}$. L'utilisation du jeu de données ainsi prétraité induira un ellipsoïde « plus consistant » dans le sens où les conditions de convergence de l'algorithme seront mieux remplies puisque la borne sur l'erreur sera plus souvent visitée ;
- plus la valeur du Kurtosis de la séquence initiale $kur\left(\left\{e_k\right\}\right)$ est diminuée par troncature, plus on élimine des données réputées aberrantes ;
- en pratique, dans une séquence d'erreur $\{e_k\}$, le nombre d'erreurs e_k se situant aux deux queues de l'histogramme est souvent faible. En tronquant ces erreurs, on réduit considérablement la valeur numérique de la borne en rejetant très peu de données. Ainsi, avec cette technique, on élimine une minorité de données réputées aberrantes tout en garantissant un ellipsoïde englobant de faible dimension, compte tenu des données conservées.

Notons que si l'on élimine un pourcentage fixé *a priori* de données contenant de « grandes erreurs », sans considérer la valeur du Kurtosis de la séquence $\{e_k\}$, on peut rencontrer les situations suivantes :

- s'il y a « trop » de mesures aberrantes dans le fichier de données (ce qui n'est évidemment pas souhaitable !), la séquence tronquée présente une borne γ dont la valeur numérique est grande, ce qui conduit à un ellipsoïde de grande dimension ;
- si l'histogramme de $\{e_k\}$ est presque « plat » (par exemple séquence quasiment uniforme), on peut éliminer des données utiles.

Dans le cas de données réelles, la séquence $\{e_k\}$ n'est pas disponible. Une séquence d'erreur pré-estimée $\{\hat{e}_k\}$ est utilisée à la place de $\{e_k\}$. $\{\hat{e}_k\}$ peut être calculée à partir de l'écart entre la sortie mesurée et la sortie du modèle obtenue pour un vecteur de paramètres estimé $\hat{x}$ par une méthode quelconque (connaissances a priori ou technique d'identification ne nécessitant pas la connaissance de la borne sur l'erreur). Ici encore, on peut penser que les grandes valeurs des erreurs causées par les mesures aberrantes se situent sur les queues de l'histogramme de $\{\hat{e}_k\}$. En conséquence, l'application à la séquence $\{\hat{e}_k\}$ de la technique de troncature de l'histogramme des erreurs telle que présentée ci-avant, permet d'éliminer les mesures réputées aberrantes présentes dans le fichier de données. Le seuil γ_s devient la borne sur l'erreur pour le fichier de données prétraité.

Les résultats de l'identification ensembliste ellipsoïdale pour des données simulées et réelles montrent que cette borne ne conduit pas à des intersections vides entre la bande $\mathbf{B}_k$ et l'ellipsoïde $\mathbf{E}_{k\text{-}1}$. Cela signifie donc que la borne γ_s n'est pas sous-estimée pour le fichier de données prétraité.

Le choix de la borne γ_s à partir de l'analyse du Kurtosis de $\{\hat{e}_k\}$ (ou de $\{e_k\}$) ainsi que le prétraitement associé du fichier de données sont résumés dans la procédure 4.4.

Procédure 4.4

1. *Choisir la valeur (a priori) du Kurtosis à atteindre kur_0 ainsi qu'une limite $R_{e\max}$ pour le taux de données éliminées ;*
2. *déterminer par une technique quelconque (y compris en utilisant des informations a priori) une estimation $\hat{x}$ du vecteur de paramètres ;*
3. *calculer l'erreur estimée $\hat{e}_k = y_k - d_k^T \hat{x}$;*

   si $kur\left(\{\hat{e}_k\}\right) < kur_0$, *alors* $\gamma_{kur} = \max\left(|\hat{e}_k|\right), \quad \forall k = 1:l$

   sinon

   choisir un seuil γ_s et tronquer les erreurs tel que $kur\left(\{\hat{e}_k^{tr}\}\right) \approx kur_0$ *;*

   si $R_e \le R_{e\max}$, $\gamma_{kur} = \gamma_s$ *;*

   sinon *choisir un autre seuil γ_{kur} jusqu'à obtenir* $R_e = R_{e\max}$ *;*

   fin si *;*

   fin si
4. *éliminer du fichier de données les valeurs réputées aberrantes.*

Dans ce travail, nous avons choisi (empiriquement) $kur_0 = 2.5$, $R_{e\max} = 5\%$. Ces valeurs garantissent que l'erreur tronquée visite souvent la borne et qu'il n'y a pas « trop » de données éliminées. On constate dans la figure 4.5 qu'avec une distribution gaussienne $N(0,1)$, en éliminant 5% des données, on obtient $kur\left(\left\{e_k^{tr}\right\}\right) \approx 2.4$.

L'avantage principal de cette méthode est d'éliminer de manière automatique les données réputées aberrantes (car elles possèdent une erreur réelle e_k ou estimée $\hat{e}_k$ située dans les queues de l'histogramme de l'erreur) et d'obtenir conjointement une borne pour la procédure d'identification ensembliste ellipsoïdale. Le fichier de données ainsi pré-traité ne contient maintenant que des données pour lesquelles on peut garantir que la séquence d'erreurs visite souvent sa borne et que cette dernière n'est pas surestimée. Cela conduit à un ellipsoïde englobant consistant (compte tenu des données conservées) pour lequel les conditions de convergence de l'algorithme sont remplies.

Notons néanmoins que cette méthode dépend du vecteur de paramètres pré-estimé $\hat{x}$. Une solution simple pour réduire cette dépendance est de relâcher la valeur du Kurtosis choisie kur_0 et de diminuer le taux de données éliminées $R_{e\max}$.

4.4 Résultats expérimentaux

Les influences de l'excitation et du choix de la borne sur la convergence de l'algorithme OBE sont vérifiées en pratique dans cette partie. Un algorithme du groupe 1 avec le critère du déterminant est utilisé. Des donnés simulées et les données réelles (IFP et robot parallèle du LIRMM) sont exploitées.

4.4.1 Influence de l'excitation : données simulées

Pour comparer l'influence de l'excitation, trois fichiers de données simulées sont créés. Les entrées sont respectivement :

- un signal en échelon ;
- une séquence SBPA ;
- une entrée créée par la procédure 4.2 présentée au paragraphe 4.2.2, telle que deux régresseurs successifs soient orthogonaux.

Le modèle du système simulé est linéaire, du 1er ordre (2.1) :

$$H(z) = \frac{Y(z)}{U(z)} = \frac{bz^{-1}}{1 - az^{-1}} = \frac{0.05z^{-1}}{1 - 0.95z^{-1}}$$

Le bruit ajouté à la simulation e_k est uniforme avec pour borne théorique $\gamma^* = 0.003$. La condition de persistance de l'excitation (PE) (4.3) semble remplie pour les trois fichiers de données ($cond(échelon) = 129.8$, $cond(SBPA) = 6.2$, $cond(procédure\ 3.2) = 5.5$). En fait, la condition 4.3 dépend également du choix de N_1. Les résultats sont obtenus ici avec $N_1 = l$ (toutes les données). Si l'on prend un bloc de 20 régresseurs, on obtient $cond(échelon) = 48.6$, $cond(SBPA) = 7.3$ et $cond(procédure\ 3.2) = 6.2$. L'angle entre deux régresseurs successifs θ_{dd_+} est représenté sur la figure 4.6 pour les trois fichiers de données considérés.

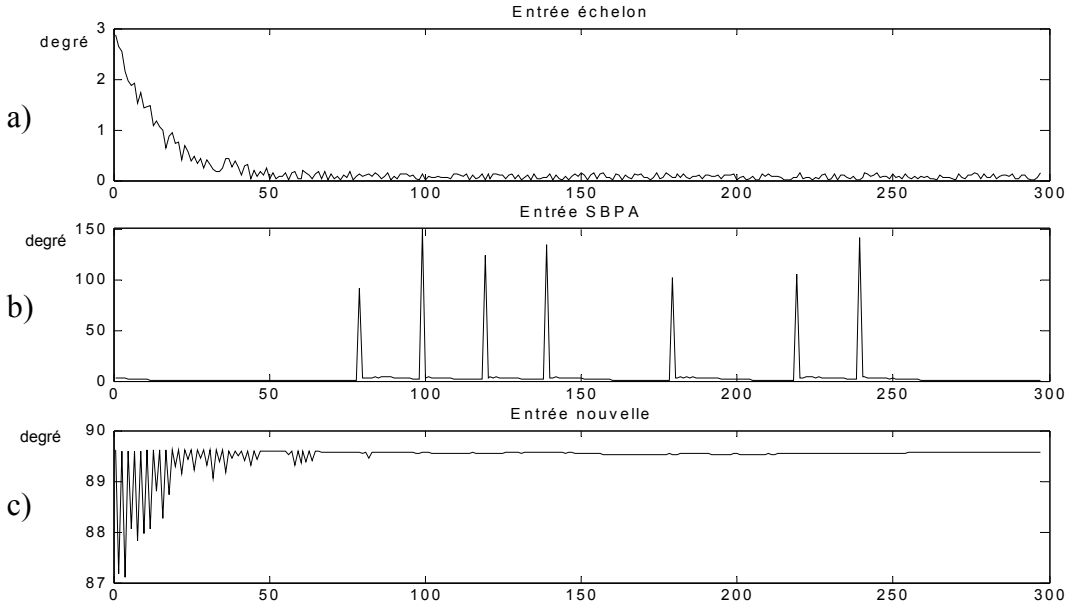


Figure 4.6 : Angle entre 2 régresseurs successifs pour les 3 fichiers de données.

L'algorithme utilisé ici est celui de Fogel et Huang, avec le critère du déterminant. La borne sur le bruit est choisie égale à $\gamma = \gamma^* = 0.003$.

On constate qu'avec une entrée en échelon, l'angle θ_{dd_+} reste faible (quelques degrés au maximum). Cela signifie que les bandes $\mathbf{B}_k$ sont quasiment toutes orientées de la même manière ce qui produit un ellipsoïde allongé (cf. figure 4.7 sur laquelle, pour un problème d'échelle, cet ellipsoïde n'est pas complètement tracé). Dans le cas d'une séquence SBPA, aux instants où u_k est modifié, deux bandes successives présentent un angle θ_{dd_+} important (au moins 100°) ce qui conduit à un ellipsoïde « plus rond » (figure 4.7). Enfin, avec le jeu de

données créé par la procédure 4.2 (saturation de l'entrée à 1 comprise), les régresseurs sont bien approximativement orthogonaux (cf. figure 4.6.c). Le tableau 4.1 présente un résumé des résultats obtenus pour les trois fichiers de données. c_k correspond au centre de l'ellipsoïde à la fin du processus d'identification (sans recirculation des données) et 2Δ est la largeur de la projection orthogonale de l'ellipsoïde sur les axes paramétriques.

L'ellipsoïde obtenu avec l'entrée proposée dans la procédure 4.2 est plus consistant dans le sens où il est plus « rond » et de taille plus petite que les autres ellipsoïdes. En outre, la projection sur les axes paramétriques de cet ellipsoïde donne un intervalle sur les paramètres plus petit qu'avec les deux autres entrées.

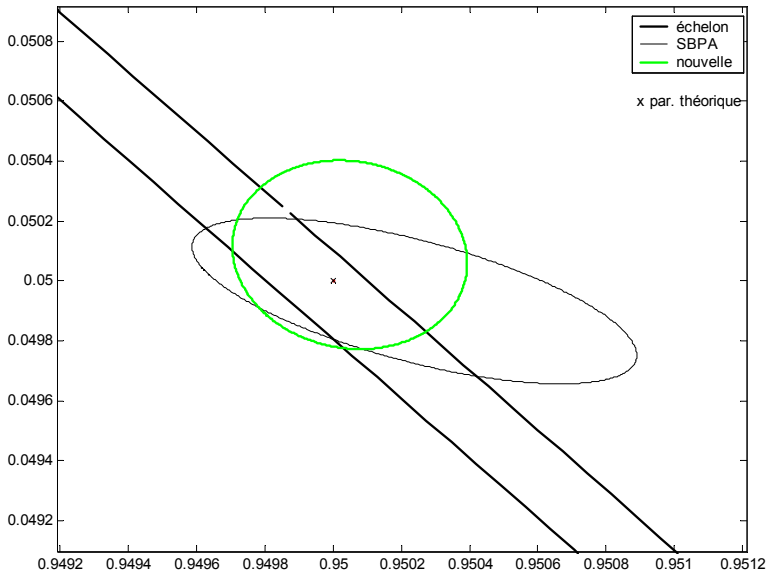


Figure 4.7 : Ellipsoïdes trouvés avec les 3 fichiers de données simulées.

Entrée	***param.***	c_k	Δ	μ_V
échelon	*a*	0.9498	$7.3\ 10^{-3}$	$1.15\ 10^{-12}$
	b	0.0502	$7.2\ 10^{-3}$	
SBPA	*a*	0.9502	$6.5\ 10^{-4}$	$1.85\ 10^{-14}$
	b	0.0499	$2.8\ 10^{-4}$	
Nouvelle	*a*	0.9501	$3.4\ 10^{-4}$	$1.16\ 10^{-14}$
	b	0.0501	$3.1\ 10^{-4}$	

Tableau 4.1 : Résultats d'identification sur les 3 fichiers de données simulées.

4.4.2 Influence du choix de la borne : données simulées

Pour évaluer les méthodes de choix de la borne présentées dans le paragraphe 4.3, on réalise l'identification sur les données simulées décrites au chapitre 2, paragraphe 2.1.1 et rappelées dans le paragraphe ci-avant. Rappelons que le vecteur de paramètres théoriques est $x^* = [a,b] = [0.95,\ 0.05]$ et que l'entrée est une SBPA. Deux fichiers de données sont utilisés :

- l'un pour lequel l'erreur (i.e. le bruit) introduite suit une distribution uniforme $[-\gamma^*, +\gamma^*]$ avec pour borne théorique $\gamma^* = 0.003$, normé SIMU-ARX-i (présenté dans le chapitre 2, section 2.1.1) ;
- l'autre pour lequel l'erreur introduite e_k suit une distribution gaussienne $N(0, 0.002)$, normé SIMU-ARX-g. L'analyse de $\{e_k\}$ montre que $\max(|e_k|) =$ 6.11 10^{-3}.

Dans les deux cas, l'erreur e_k ajoutée conduit à un rapport Signal sur Bruit $RSB \approx$ 50dB (valeur identique à celle choisie dans le chapitre 2).

Les résultats de l'identification ensembliste ellipsoïdale obtenus pour le fichier SIMU-ARX-i après 10 circulations des données sont présentés dans le tableau 4.2.

Pour la méthode « classique », nous avons utilisé une technique de moindres carrées pour calculer le vecteur de paramètres pré-estimé $\hat{x}$. La borne trouvée (et retenue) dans ce cas γ_{clas} = 0.0031 est approximativement égale à la borne de l'erreur (ou bruit) introduite dans la simulation ($\gamma^* = 0.003$). En fait, cela est logique puisque l'on a $\hat{x} = [9.5015\ 10^{-1},\ 5.0005\ 10^{-2}] \approx x^* = [9.5\ 10^{-1},\ 5\ 10^{-2}]$. Il en résulte que l'ellipsoïde obtenu et les intervalles Δ calculés par projection de l'ellipsoïde sur les axes paramétriques, sont très petits. Notons également que la condition PE (4.3) est bien satisfaite.

Méthode	***param.***	c_k	Δ	μ_V	γ_0	γ_∞
classique	*a*	0.9501	6.8 10^{-4}	3.69 10^{-14}	3.1 10^{-3}	3.1 10^{-3}
	b	0.0499	3.9 10^{-4}			
OBE-ABE	*a*	0.9501	3.2 10^{-3}	5.51 10^{-11}	6 10^{-2}	4.3 10^{-3}
	b	0.0500	2.4 10^{-3}			

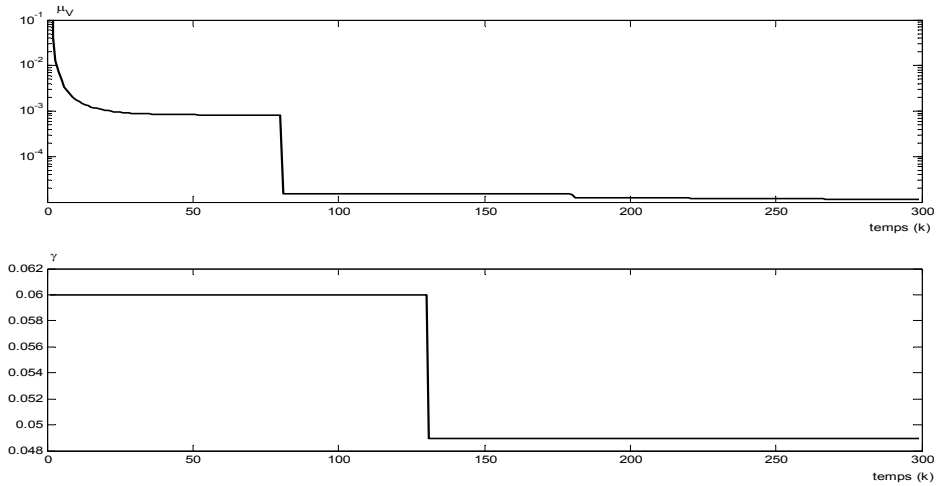
Tableau 4.2: Résultat de l'identification sur le fichier de données SIMU-ARX-i après 20 recirculations.

L'algorithme OBE-ABE est initialisé par $\gamma_0 = 20\gamma^* = 0.06$, $\varsigma = 10\%\gamma_k$ et $l_0 = 50$ afin d'avoir $\wp_\delta > 92\%$ (4.15). En suivant l'évolution de la borne (voir figure 4.8), on constate qu'elle converge vers une valeur « acceptable » (γ = 0.0043) puisque les conditions de convergence d'OBE-ABE sont vérifiées. On remarque (tableau 4.2) que les intervalles Δ trouvés avec l'algorithme OBE-ABE restent grands en comparaison ceux obtenus lorsque le réglage de la borne est réalisé par la méthode classique. Cela est dû au fait que la dernière borne utilisée γ = 0.0043 est surestimée par rapport à la borne théorique. La borne γ tend vers sa valeur théorique γ^* lorsque la longueur l du fichier de données tend vers l'infini et lorsque l_0 et ς sont « correctement » choisis.

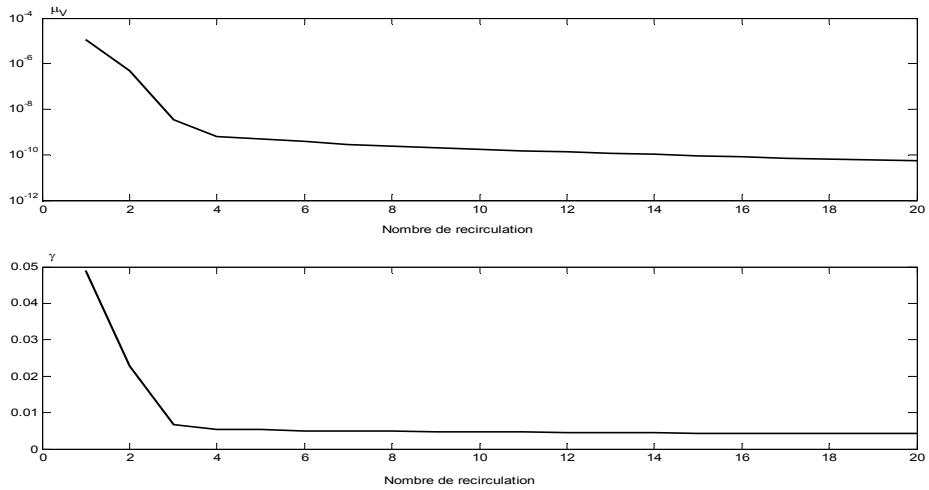
Pour la méthode basée sur le calcul du Kurtosis de la séquence d'erreur présentée au paragraphe 4.3.3, en appliquant la procédure 4.4, on obtient $kur\left(\{\hat{e}_k\}\right) = 1.78 < 2.5$. Cette valeur indique que l'histogramme de $\{\hat{e}_k\}$ est « plat » ce qui semble exact puisque le bruit ajouté dans ce fichier de données suit une distribution uniforme. Il n'est donc pas nécessaire de tronquer la séquence d'erreur. Les méthodes classique et basée sur le Kurtosis dans ce cas, fournissent les mêmes résultats d'identification.

On s'intéresse maintenant à l'identification ensembliste ellipsoïdale pour le fichier de données SIMU-ARX-g (bruit gaussien). Il semble évident que dans ce cas, la probabilité pour une erreur e_k de visiter la borne $\max\left(|e_k|\right)$ est faible ce qui signifie que l'une des conditions permettant de garantir la convergence de l'algorithme OBE-ABE n'est pas vérifiée. Dans un souci de comparaison, OBE-ABE est initialisé avec les mêmes paramètres que précédemment, à savoir $\gamma_0 = 20\gamma^*$, $\varsigma = 10\%\gamma_k$ et $l_0 = 50$. Le suivi de la réduction automatique de la borne (voir figure 4.9) montre qu'elle diminue très (trop) rapidement pour atteindre $\gamma = 5.75\ 10^{-3} < \gamma^*$. L'apparition d'intersections vides entre la bande $\mathbf{B}_k$ et l'ellipsoïde $\mathbf{E}_{k\text{-}1}$ conduit l'algorithme OBE-ABE à s'arrêter prématurément.

Les résultats obtenus avec les méthodes « classique » et « Kurtosis » (pour le choix de la borne), sont présentés dans le tableau 4.3. La méthode Kurtosis est appliquée avec $\alpha \approx 2$, ce qui conduit à $\gamma_{\text{kur}} = \gamma_{\text{s}} = 0.0041$ (voir théorème 4.9).



a) Evolution du volume et de la borne durant la 1ère circulation des données



b) Evolution de volume et de la borne après 20 recirculations

Figure 4.8

On constate que le centre de l'ellipsoïde est correctement estimé. L'intervalle Δ sur les paramètres est toujours plus petit avec la borne γ_{kur} puisque $\gamma_{\text{kur}} = 0.0041 < \gamma_{\text{clas}} = \max\left(\left|e_k\right|\right) = 0.0059$. Notons que l'on a rejeté 4% des données avec la « méthode Kurtosis », ce qui peut faire perdre de « l'information » pertinente. En contrepartie, on obtient un ellipsoïde englobant de plus petite dimension, sans être confronté à des cas d'intersection vide entre la bande et l'ellipsoïde. Ainsi, la méthode Kurtosis permet d'éliminer rapidement les données réputées aberrantes puisqu'elles présentent une « grande erreur » e_k et de régler la borne pour l'identification.

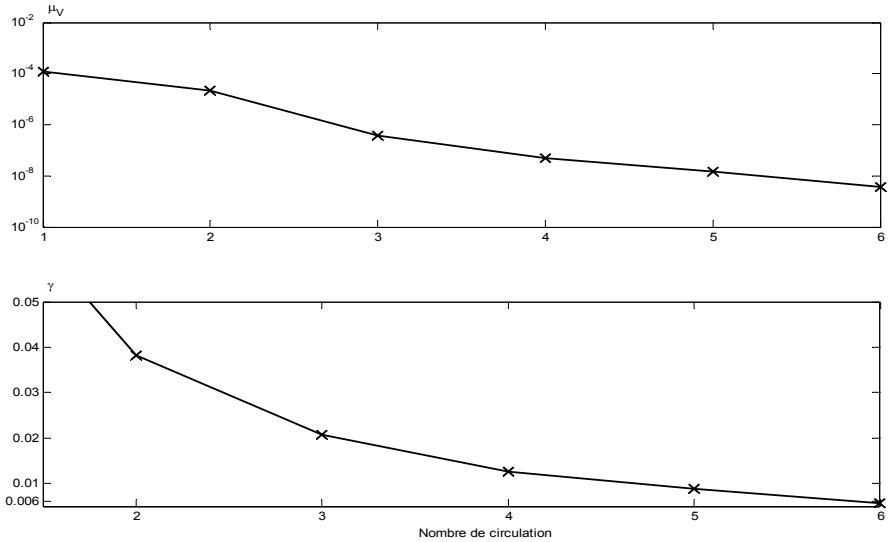


Figure 4.9 : Evolution du volume et de la borne dans l'algorithme OBE-ABE (bruit gaussien).

Méthode	**param.**	c_k	Δ	μ_V	γ	$R_e(\%)$
classique	*a*	0.9496	$3.0\ 10^{-3}$	$7.87\ 10^{-12}$	$5.9\ 10^{-3}$	0
	b	0.0495	$1.6\ 10^{-3}$			
Kurtosis	*a*	0.9497	$2.9\ 10^{-4}$	$8.79\ 10^{-16}$	$4.1\ 10^{-3}$	4%
	b	0.0499	$1.9\ 10^{-4}$			

Tableau 4.3 : Résultat d'identification sur le fichier de données SIMU-ARX-g.

4.4.3 Influence de l'excitation : données IFP

En examinant le fichier de données réelles de l'IFP (voir chapitre 2), on constate que la direction des bandes est quasiment inchangée pendant le processus d'identification (voir Figure 4.10). Les grands angles au début dans figure 4.10 sont en fait causés par le bruit (l'excitation reste nulle à ces instants, voir Figure 2.7). Ceci explique pourquoi l'ellipsoïde obtenu est allongé même si le critère de la trace est utilisé alors que ce dernier est réputé fournir un ellipsoïde « plus rond » (cf. chapitre 1). Autrement dit, ce jeu de données ne convient pas véritablement pour une méthode d'estimation ensembliste ellipsoïdale.

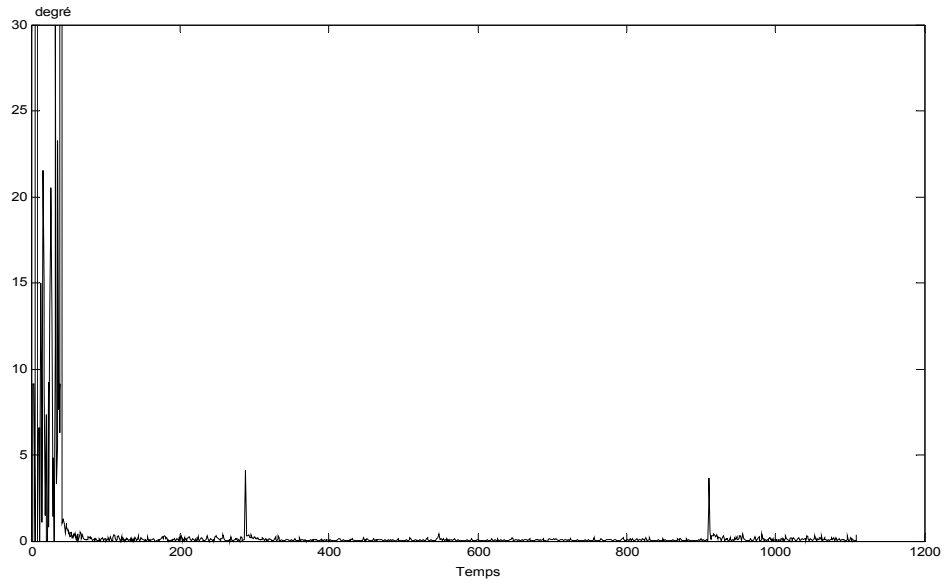


Figure 4.10 : L'angle entre 2 régresseurs successifs des données IFP.

4.4.4 Influence de l'excitation : données du robot parallèle

Pour les données du robot parallèle, l'existence de colonnes nulles dans chaque bloc (par exemple, les colonnes 2, 3, 4, 8, 9, 10, 12, 13, 14 restent à zéros dans le bloc 1, voir chapitre 2) conduit à une matrice des régresseurs (par bloc) qui n'est pas de rang plein. En conséquence, la condition de persistance de l'excitation PE (4.3) **pour chaque bloc** n'est pas vérifiée puisque l'on a $cond\left(bloc\ i\right) = Inf$, $(i = 1:4)$. Notons par contre que PE est satisfaire si l'on s'intéresse à tout le fichier de données puisque $cond\left(fichier\right) = 70.47\ 10^3$. On constate en outre, que l'angle entre deux régresseurs successifs θ_{dd_+} est faible (cf. figure 4.11a). En fait, on peut améliorer l'évolution de θ_{dd_+} en réorganisant les données (voir figure 4.11.b). Un nouveau fichier de données est créé dans lequel les régresseurs correspondant à chacun des 4 moteurs sont intercalés. La condition PE est maintenant satisfaite pour tout paquet de données pris au hasard dans le fichier de données.

Le Tableau 4.4 compare les intervalles des paramètres obtenus avec les deux fichiers de données du robot (initial et réarrangé). On utilise l'algorithme de Fogel et Huang avec le critère du déterminant. La borne retenue pour l'identification est $\gamma = 6$. On peut voir qu'après la 1ère circulation des données, l'ellipsoïde (et les intervalles sur les paramètres correspondants) trouvé à partir du nouveau fichier sont beaucoup plus faibles que ceux obtenus avec le fichier initial. Dès la 1ère circulation sur des données réarrangées, le centre de l'ellipsoïde conduit à des paramètres physiquement cohérents (pas de valeurs négatives).

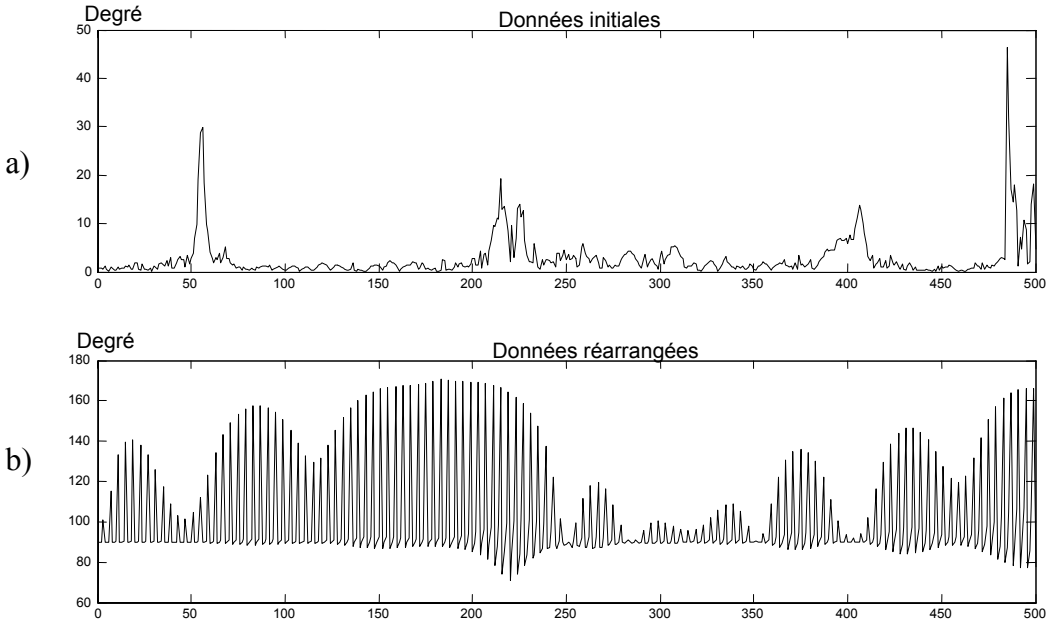


Figure 4.11 : Angle entre deux régresseurs successifs pour le robot parallèle.

Param.	**1ère circulation**				**60ème circulation**			
	Données initiales		Données réarrangées		Données initiales		Données réarrangées	
	centre	*Δ*	*centre*	*Δ*	*centre*	*Δ*	*centre*	*Δ*
I_{mot1}	$4.49\ 10^{-2}$	1.64	$1.39\ 10^{-2}$	$1.50\ 10^{-1}$	$4.54\ 10^{-2}$	$7.04\ 10^{-2}$	$4.54\ 10^{-2}$	$6.94\ 10^{-3}$
I_{mot2}	$4.93\ 10^{-2}$	$5.33\ 10^{-1}$	$1.22\ 10^{-2}$	$1.58\ 10^{-1}$	$4.99\ 10^{-2}$	$8.12\ 10^{-3}$	$4.99\ 10^{-2}$	$7.98\ 10^{-3}$
I_{mot3}	$3.40\ 10^{-2}$	$2.19\ 10^{-1}$	$1.85\ 10^{-2}$	$9.47\ 10^{-2}$	$2.47\ 10^{-2}$	$7.52\ 10^{-2}$	$2.49\ 10^{-2}$	$7.41\ 10^{-2}$
I_{mot4}	$3.60\ 10^{-2}$	$9.64\ 10^{-2}$	$2.26\ 10^{-2}$	$8.99\ 10^{-2}$	$2.83\ 10^{-2}$	$7.53\ 10^{-2}$	$2.84\ 10^{-2}$	$7.58\ 10^{-2}$
M_{nac}	$9.28\ 10^{-2}$	3.53	$7.57\ 10^{-1}$	3.19	$2.14\ 10^{-1}$	$7.49\ 10^{-2}$	$2.14\ 10^{-1}$	$7.34\ 10^{-2}$
I_{nac}	$1.83\ 10^{-3}$	$4.07\ 10^{-2}$	$5.74\ 10^{-4}$	$5.75\ 10^{-2}$	$-4.47\ 10^{-4}$	$3.57\ 10^{-2}$	$-6.48\ 10^{-4}$	$3.67\ 10^{-2}$
F_{v1}	$6.13\ 10^{-1}$	$7.29\ 10^{1}$	$2.97\ 10^{-1}$	5.58	$4.09\ 10^{-1}$	3.38	$4.15\ 10^{-1}$	3.33
F_{v2}	$3.18\ 10^{-1}$	$2.70\ 10^{1}$	$1.99\ 10^{-1}$	5.34	$4.99\ 10^{-1}$	$7.68\ 10^{-1}$	$4.99\ 10^{-1}$	$7.52\ 10^{-1}$
F_{v3}	$-2.12\ 10^{-2}$	7.12	$1.93\ 10^{-1}$	3.29	$4.91\ 10^{-2}$	2.86	$5.37\ 10^{-2}$	2.81
F_{v4}	$-6.56\ 10^{-2}$	2.69	$2.04\ 10^{-1}$	2.97	$1.74\ 10^{-2}$	2.68	$1.61\ 10^{-2}$	2.69
F_{s1}	$7.22\ 10^{-1}$	$1.92\ 10^{2}$	$3.50\ 10^{-1}$	$2.00\ 10^{1}$	$8.89\ 10^{-1}$	8.96	$8.81\ 10^{-1}$	8.80
F_{s2}	$2.38\ 10^{-1}$	$8.34\ 10^{1}$	$8.19\ 10^{-1}$	$1.86\ 10^{1}$	$1.01\ 10^{-1}$	1.35	$1.01\ 10^{-1}$	1.33
F_{s3}	$8.73\ 10^{-1}$	$2.63\ 10^{1}$	$1.21\ 10^{-1}$	$1.77\ 10^{1}$	$3.28\ 10^{-1}$	$1.65\ 10^{1}$	$2.38\ 10^{-1}$	$1.65\ 10^{1}$

Param.	**1ère circulation**				**60ème circulation**			
	Données initiales		Données réarrangées		Données initiales		Données réarrangées	
	centre	*Δ*	*centre*	*Δ*	*centre*	*Δ*	*centre*	*Δ*
F_{s4}	1.13	$1.12\ 10^{1}$	$3.30\ 10^{-1}$	$1.59\ 10^{1}$	$4.77\ 10^{-1}$	$1.49\ 10^{1}$	$4.82\ 10^{-1}$	$1.49\ 10^{1}$
μ_V	**$7.07\ 10^{14}$**		**$6.27\ 10^{3}$**		**$3.13\ 10^{-9}$**		**$2.54\ 10^{-9}$**	

Tableau 4.4 : Résultat d'identification avec données originelles et données réarrangées.

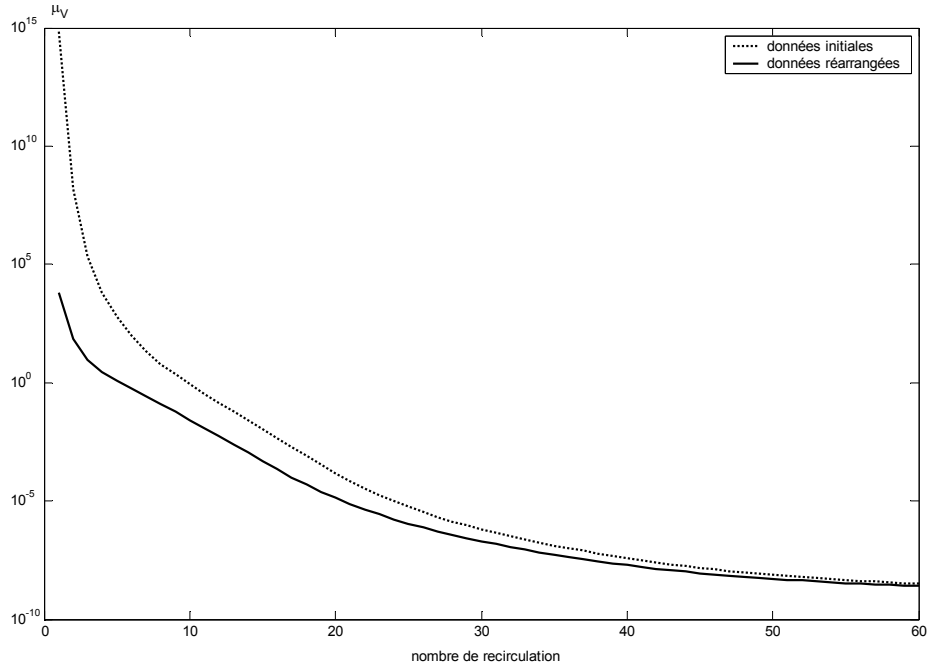


Figure 4.12 : Evolution de volume de l'ellipsoïde sur les données du robot parallèle.

Après 60 circulations des données, la taille des ellipsoïdes trouvés avec ces deux fichiers de données et leurs projections respectives sur les axes paramétriques sont quasiment identiques. Néanmoins, la convergence de l'algorithme en utilisant le fichier réarrangé est plus rapide (voir figure 4.12), ce qui implique que l'on peut obtenir une même « qualité » de l'estimée pour un coût calcul moindre lorsque les régresseurs successifs conduisent les bandes successives **B** à parcourir toutes les directions de l'espace paramétrique.

On constate donc que la persistance de l'excitation a une véritable influence sur la convergence de l'algorithme OBE. Ainsi, en réarrangeant raisonnablement (et lorsque cela est possible !) les régresseurs de manière à maximiser l'angle entre deux bandes successives (l'optimum étant d'obtenir deux bandes orthogonales), on peut considérablement améliorer la

vitesse de convergence de l'algorithme, comme le montre les résultats obtenus avec les données du robot parallèle du LIRMM.

4.4.5 Influence du choix de la borne : données du robot parallèle

On applique maintenant les méthodes de choix de la borne aux données réelles du robot parallèle. L'histogramme de la séquence d'erreur estimée $\{\hat{e}_k\}$ est présenté dans la figure 4.13.a. Le vecteur $\hat{x}$ de paramètres pré-estimé est obtenu par une méthode de moindres carrés. On a :

$$\hat{x} = [1.67\ 10^{-2},\ 1.64\ 10^{-2},\ 1.76\ 10^{-2},\ 2.34\ 10^{-2},\ 9.85\ 10^{-1},\ 2.92\ 10^{-3},\ 2.11\ 10^{-1},\ 1.24\ 10^{-1}$$
$$1.27\ 10^{-1},\ 1.13\ 10^{-1},\ 1.22,\ 1.03,\ 7.90\ 10^{-1},\ 1.04]$$

Bien que $\{\hat{e}_k\}$ ne soit pas égale à la séquence d'erreur « vraie » $\{e_k\}$ (non calculable ici !), on peut néanmoins supposer que son histogramme est assez proche de celui de $\{e_k\}$. La figure 4.13.a montre que $\{\hat{e}_k\}$ contient quelques « grandes erreurs » dues à des données aberrantes (ainsi qu'on l'a déjà souligné dans le chapitre 2). Le Kurtosis calculé à partir de $\{\hat{e}_k\}$ vaut 96.6.

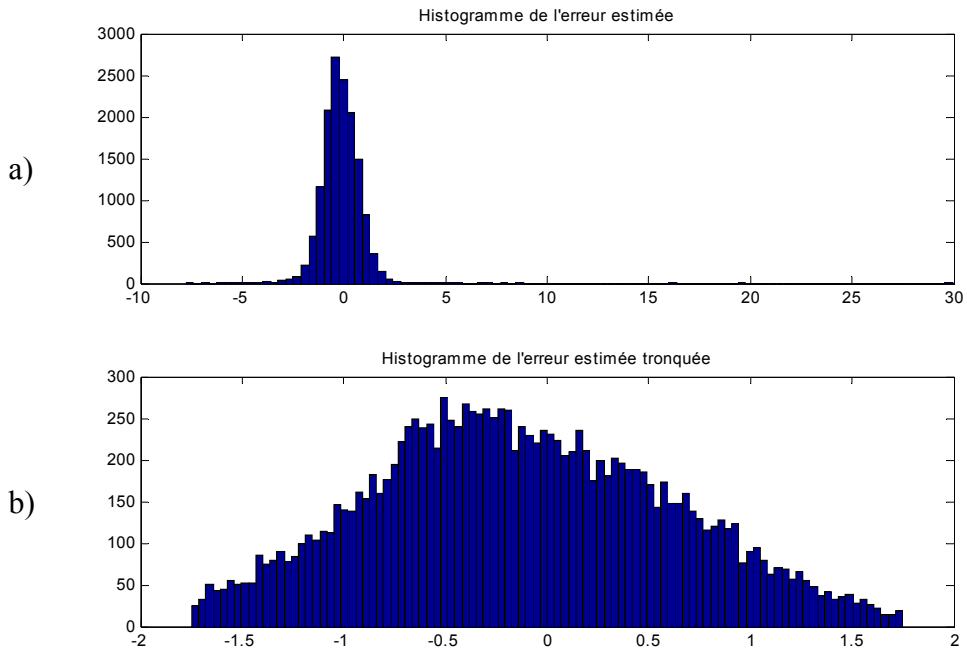


Figure 4.13 : Histogramme des séquences a) $\{\hat{e}_k\}$ et b) $\{\hat{e}_k^{tr}\}$

L'algorithme OBE-ABE ne peut être appliqué pour ce type de données (en fait son utilisation conduit à des intersections vides et à une diminution trop importante de la borne) puisque les conditions sur $\{e_k\}$ ne sont pas vérifiées.

L'utilisation de la procédure 4.3 donnée pour la méthode « classique » conduit à une très grande valeur de borne $\gamma_{\text{clas1}} = 29.9\text{Nm}$. En fait, on n'a pas réalisé une inspection préliminaire du fichier de données qui aurait pu permettre de retirer certaines mesures aberrantes et par la même, de réduire la valeur de γ_{clas1}. Le choix « manuel » utilisant des connaissances a priori et présenté dans le chapitre 2 conduit à $\gamma_{\text{clas2}} = 6\text{Nm}$. Néanmoins, cette valeur de la borne reste surestimée pour la plupart des données. Ainsi, quelle que soit la valeur retenue (γ_{clas1} ou γ_{clas2} respectivement), une élimination stricte des données réputées aberrantes est indispensable pour ne pas trop surestimer la borne.

Avec la procédure 4.4 de la méthode « Kurtosis », on élimine 4.4% des données afin d'obtenir $\text{kur}\left(\left\{\hat{e}_k^{tr}\right\}\right) \approx 2.5$. L'histogramme de la séquence d'erreurs tronquée $\left\{\hat{e}_k^{tr}\right\}$ est présenté dans la figure 4.13.b. Le seuil utilisé pour tronquer la séquence d'erreur devient la borne utilisée dans le processus d'identification ensembliste ellipsoïdale $\gamma_{\text{kur}} = \gamma_s = 1.75\text{Nm}$.

Les résultats obtenus avec $\gamma_{\text{clas2}} = 6\text{Nm}$ et $\gamma_{\text{kur}} = 1.75\text{Nm}$ sont reportés dans le tableau 4.5 (après 60 circulations). L'algorithme utilisé est celui de Fogel et Huang avec le critère du déterminant. Le fichier de données exploité a été réarrangé (voir la paragraphe 4.4.4).

Param.	**Méthode classique**			**Méthode Kurtosis**		
	centre	*[min ; max]*	*Δ*	*centre*	*[min ; max]*	*Δ*
I_{mot1}	$4.54\ 10^{-2}$	[-0.025 ; 0.116]	$7.04\ 10^{-2}$	$1.66\ 10^{-2}$	[0.014 ; 0.019]	$2.40\ 10^{-3}$
I_{mot2}	$4.99\ 10^{-2}$	[0.042 ; 0.058]	$8.12\ 10^{-3}$	$1.64\ 10^{-2}$	[0.0117; 0.021]	$4.62\ 10^{-3}$
I_{mot3}	$2.49\ 10^{-2}$	[-0.050 ; 0.099]	$7.52\ 10^{-2}$	$1.75\ 10^{-2}$	[0.011 ; 0.024]	$6.11\ 10^{-3}$
I_{mot4}	$2.84\ 10^{-2}$	[-0.047 ; 0.104]	$7.53\ 10^{-2}$	$2.32\ 10^{-2}$	[0.019 ; 0.027]	$3.52\ 10^{-3}$
M_{nac}	$2.14\ 10^{-1}$	[0.139 ; 0.289]	$7.49\ 10^{-2}$	$9.87\ 10^{-1}$	[0.923 ;1.051]	$6.42\ 10^{-2}$
I_{nac}	$-6.48\ 10^{-4}$	[-0.036 ; 0.035]	$3.57\ 10^{-2}$	$2.94\ 10^{-3}$	[0.002 ; 0.004]	$1.19\ 10^{-3}$
F_{v1}	$4.15\ 10^{-1}$	[-2.972 ; 3.791]	3.38	$2.13\ 10^{-1}$	[0.145 ; 0.281]	$6.78\ 10^{-2}$
F_{v2}	$4.99\ 10^{-1}$	[-0.269 ; 1.267]	$7.68\ 10^{-1}$	$1.23\ 10^{-1}$	[0.067 ; 0.179]	$5.64\ 10^{-2}$
F_{v3}	$5.37\ 10^{-2}$	[-2.806 ; 2.905]	2.86	$1.29\ 10^{-1}$	[0.059 ; 0.199]	$7.04\ 10^{-2}$
F_{v4}	$1.61\ 10^{-2}$	[-2.667 ; 2.702]	2.68	$1.15\ 10^{-1}$	[0.011 ; 0.219]	$1.04\ 10^{-1}$
F_{s1}	$8.81\ 10^{-1}$	[-8.075 ; 9.855]	8.96	1.22	[1.059 ; 1.374]	$1.58\ 10^{-1}$
F_{s2}	$1.01\ 10^{-1}$	[-1.248 ; 1.449]	1.35	1.02	[0.807 ; 1.239]	$2.16\ 10^{-1}$

Param.	Méthode classique			Méthode Kurtosis		
	centre	*[min ; max]*	*Δ*	*centre*	*[min ; max]*	*Δ*
F_{s3}	$2.38\ 10^{-1}$	[-16.21 ; 16.87]	$1.65\ 10^{1}$	$7.68\ 10^{-1}$	[0.378 ; 1.159]	$3.90\ 10^{-1}$
F_{s4}	$4.82\ 10^{-1}$	[-14.44 ; 15.39]	$1.49\ 10^{1}$	1.03	[0.840 ; 1.219]	$1.89\ 10^{-1}$
μ_V	**$3.1312\ 10^{-9}$**			**$1.5719\ 10^{-45}$**		

Tableau 4.5 : Résultat d'identification sur les données du robot parallèle (après 60 recirculations)

On voit qu'avec la méthode classique de choix de la borne sur l'erreur, *Δ* conduit à des intervalles surdimensionnés. En outre, ces intervalles peuvent contenir des valeurs numériques négatives qui n'ont pas de sens physique. La méthode Kurtosis produit quant à elle des résultats tout à fait satisfaisant pour ce qui est de la valeur du centre de l'ellipsoïde (compte tenu des connaissances *a priori* que l'on a sur le procédé, voir chapitre 2, paragraphe 2.3) ou de la projection de l'ellipsoïde sur les axes paramétriques. La valeur du Kurtosis de la séquence tronquée à atteindre kur_0 et la limite de données éliminées R_{emax} (voir la procédure 4.4, paragraphe 4.3.3.2) peuvent être réglés à partir d'une analyse préliminaire des données. Par exemple, s'il y a des mesures réellement aberrantes (repérées par une inspection préliminaire du fichier), on peut augmenter R_{emax} pour obtenir une borne moins « lâche ». On peut également les éliminer manuellement ce qui bien entendu s'avère plus fastidieux que d'appliquer la démarche proposée.

Enfin, pour des données réelles, l'ellipsoïde obtenu lors de l'identification doit avoir un sens physique si les paramètres à identifier en ont un (par exemple, positivité d'un paramètre). Cela doit également permettre de guider le réajustement de kur_0 et/ou R_{emax} si besoin est.

4.5 Conclusion

Dans ce chapitre, on s'est intéressé au problème de convergence des algorithmes OBE. Le terme « convergence » regroupe en fait plusieurs situations, à savoir :

- la convergence du centre de l'ellipsoïde vers le voisinage du « vrai » vecteur de paramètres ;
- la convergence du volume de l'ellipsoïde, ce dernier en limite tendant vers un point.

Avec le critère du déterminant (qui consiste en fait à minimiser le volume au carré de l'ellipsoïde), l'algorithme SMSA (et tous les algorithmes du groupe 1 associés) présente des

propriétés intéressantes de convergence puisque l'on peut démontrer que sous certaines conditions, on a bien les deux situations citées ci-avant.

Les propriétés de convergence ne sont pas (à notre connaissance) démontrées pour le critère de la trace.

Bien que les algorithmes du groupe 2 aient été les premiers à s'intéresser aux problèmes de « convergence », ils ne proposent des résultats que pour la convergence du centre de l'ellipsoïde. La convergence du coefficient σ_k^2 (c'est l'objectif poursuivi par cette famille d'algorithmes) n'a pas d'incidence sur la convergence de la taille de l'ellipsoïde englobant.

Les conditions théoriques permettant d'assurer la convergence de l'algorithme OBE sont au nombre de 3 trois :

- le vecteur de « vrais » paramètres x^* doit appartenir à l'ellipsoïde initial ;
- l'excitation doit être persistante ;
- l'erreur doit visiter souvent sa borne.

Le premier point ne pose pas habituellement de problème puisque l'algorithme est initialisé par un ellipsoïde de grande dimension permettant de garantir cette condition.

En analysant la condition de persistance de l'excitation et à partir d'une interprétation géométrique, on a montré que l'angle entre deux bandes d'espace successives est un facteur fondamental de l'algorithme puisqu'il a une influence directe sur sa vitesse de convergence. Pour un fichier de données déjà enregistré et en procédant à un réarrangement de l'ordre des régresseurs (lorsque cela est possible), on peut améliorer la vitesse de convergence de l'algorithme.

On a également proposé une procédure permettant de créer une entrée qui maximise l'angle entre deux bandes d'espace successives, la valeur optimale de cet angle étant de 90°. En fait, l'objectif est que la bande parcourt toutes les directions de l'espace de paramètres.

Différentes méthodes de choix de la borne ont été présentées dans ce chapitre. Après avoir rappelé les techniques classiques que l'on rencontre chez les utilisateurs des techniques ensemblistes ellipsoïdales, on a présenté l'algorithme OBE-ABE. Ce dernier permet de réduire automatiquement la borne durant le processus d'identification. C'est une solution idéale si les conditions de convergence sont vérifiées. Néanmoins, l'application de cet algorithme à des données réelles semble instable et dépend des paramètres l_0, ς qui s'avèrent être délicats à régler.

Une méthode de choix de la borne basée sur une analyse du Kurtosis de la séquence d'erreurs (réelle ou plus souvent estimée) a été proposée. Le principe de cette méthode est, à partir de l'histogramme de la séquence d'erreurs, d'éliminer les données contenant de « grandes » erreurs qui incluent (a priori) les mesures aberrantes. Plus précisément, cela est réalisé en tronquant les queues de l'histogramme de la séquence d'erreur afin d'obtenir une valeur du Kurtosis préalablement choisie, et en éliminant un taux de données maximum préalablement fixé. On garantit ainsi que la séquence d'erreurs tronquées visite « souvent » sa borne, ce qui est la troisième condition de convergence donnée ci-avant.

Notons enfin que cette démarche à base de Kurtosis semble attrayante car elle ne nécessite pas un réglage heuristique et/ou manuel, souvent non scientifiquement justifiable ou justifié, de la borne.

Chapitre 5

Traitement des données par paquet

Comme on l'a déjà vu dans les chapitres précédents, l'algorithme OBE traite séquentiellement les données. Ce traitement séquentiel conduit à des formules extrêmement simples pour le calcul de l'ellipsoïde englobant. Il permet en outre à l'algorithme OBE de poursuivre les variations des variables (dans notre contexte, des paramètres) en temps réel, à chaque nouvelle acquisition d'une donnée. Cependant, d'un point de vue optimisation, le traitement séquentiel ne permet d'atteindre qu'une solution optimale locale, l'ordre de traitement des données ayant (très souvent) une influence sur le résultat obtenu.

Dans ce chapitre, on s'intéresse à construire un algorithme de la famille OBE (plus spécifiquement, du groupe 1, c'est-à-dire mettant en œuvre un critère sur la taille de l'ellipsoïde obtenu) qui permette de traiter simultanément toutes les observations d'un paquet de données et à calculer l'ellipsoïde englobant l'intersection de ce paquet de bandes avec un ellipsoïde. Cet algorithme sera dénommé dans la suite **OBE-paquet**. Théoriquement, OBE-paquet devrait améliorer le résultat obtenu par rapport à celui fourni par un algorithme OBE standard. En effet, les données du paquet étant traitées simultanément, l'ordre de traitement n'a plus d'incidence et la solution obtenue pour ce paquet de données est optimale au sens où l'on obtient l'ellipsoïde de plus petite dimension englobant l'intersection de toutes les bandes définies par les données du paquet.

Néanmoins, il faut soulever une difficulté de taille : le problème d'optimisation sous-jacent qui avait une solution explicite dans le cas d'une bande et d'un ellipsoïde (ou plus généralement de deux ellipsoïdes) n'a plus maintenant de solution explicite.

Des procédures de prétraitement des données avant l'appel à l'algorithme OBE-paquet sont proposées. La forme factorisée d'OBE-paquet est également explicitée afin de garantir sa stabilité numérique.

L'évaluation des performances de l'algorithme OBE-paquet sera réalisée via les résultats d'identification pour des données simulées et réelles.

5.1 Position du problème

Rappelons que pour l'algorithme OBE, on calcule à chaque instant un ellipsoïde $\mathbf{E}_k$ qui englobe l'intersection de l'ellipsoïde précédent $\mathbf{E}_{k\text{-}1}$ avec une bande dans l'espace des paramètres $\mathbf{B}_k$. Supposons que l'ellipsoïde et la bande sont exprimés sous leur forme normalisée par :

$$\mathbf{E}_k = \left\{ x \in \mathbb{R}^p : \left(x - c_k\right)^T M_k \left(x - c_k\right) \leq 1,\ M_k = P_k^{-1} > 0 \right\} \tag{5.1}$$

$$\mathbf{B}_k = \left\{ x \in \mathbb{R}^p : \left(y_k - d_k^T x\right)^2 \leq 1 \right\} \tag{5.2}$$

Comme on l'a déjà indiqué, l'intersection entre la bande $\mathbf{B}_k$ et l'ellipsoïde $\mathbf{E}_{k\text{-}1}$ est testée avant tout calcul. Si besoin, la bande est bien entendue « réduite » avant l'appel à l'algorithme OBE. Les données sont traitées séquentiellement, ce qui a fait apparaître les inconvénients suivants :

- lorsqu'une mesure « aberrante » est exploitée à l'instant k, elle peut engendrer un « déplacement » irrémédiable des ellipsoïdes calculés ultérieurement. Cela signifie qu'on observe un décalage de l'ellipsoïde obtenu à l'instant k et que les ellipsoïdes suivant ne reviennent pas vers la zone de l'espace paramétrique où ils devraient se situer. Pire, le vecteur des « vrais » paramètres peut ne plus appartenir à ces ellipsoïdes ;
- lorsque l'ellipsoïde est petit par rapport à la largeur de la bande, on peut observer des situations pour lesquelles le traitement séquentiel des bandes ne réduit pas la taille de l'ellipsoïde englobant. Cependant, un traitement de plusieurs bandes en même temps permet de calculer un ellipsoïde englobant ayant une dimension qui diminue. Ceci est illustré dans la figure 5.1 dans laquelle on a $\mathbf{E}_0 \cap \mathbf{B}_i = \mathbf{E}_0$ $(i = 1,2,3)$ (ceci d'après le théorème 3.1, cas particulier de l'algorithme OBE) mais $\mathbf{E}_0 \bigcap_{i=1}^{3} \mathbf{B}_i = \mathbf{E}$ avec une taille de l'ellipsoïde $\mathbf{E}$ qui a diminué ;
- l'algorithme OBE avec un traitement séquentiel des données ne peut fournir qu'une solution sous optimale.

Pour obtenir la solution optimale globale, on doit résoudre le problème d'optimisation en considérant simultanément toutes les observations du fichier de données. Cela revient à rechercher l'ellipsoïde (de taille minimale) englobant l'intersection de toutes les bandes définies dans l'espace paramétrique. Le problème d'optimisation est non explicite et difficile à résoudre car la fonction critère contient un grand nombre de variables à optimiser

simultanément. La solution obtenue par une routine d'optimisation (par exemple « fmincon » dans Matlab) n'est pas, généralement, la solution optimale car il faut tenir compte de :

- l'influence de la précision demandée pour le calcul de cet optimum;
- des capacités de la routine d'optimisation utilisée pour calculer l'optimum recherché.

Le temps de calcul pour résoudre ce problème d'optimisation est également un frein certain au traitement simultané de toutes les données. Pronzato et Walter [PRO 94] proposent un algorithme de recherche de l'ellipsoïde optimal global dont la mise en œuvre est difficile. On va envisager cet algorithme dans le paragraphe qui suit.

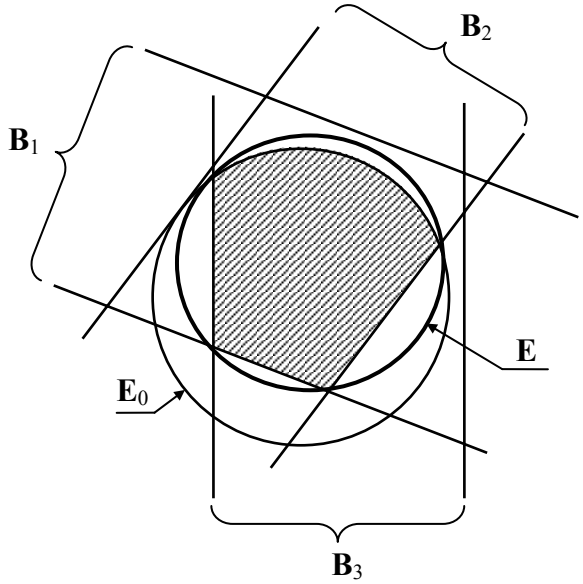


Figure 5.1 : Illustration de l'intersection d'un ellipsoïde avec 3 bandes.

Nous proposons une démarche dont l'objectif est d'améliorer l'ellipsoïde obtenu en traitant les données non pas toutes simultanément, mais par paquet. Ainsi, le fichier de données est vu comme une suite de paquets. Supposons que dans chaque paquet, les données sont décrites par :

$$y = Cx + e \tag{5.3}$$

où :

- $y \in \mathbb{R}^m$ est le vecteur contenant les sorties mesurées dans le paquer considéré et m est la longueur du paquet ;
- $x \in \mathbb{R}^p$ est le vecteur des paramètres à estimer ;
- $C \in \mathbb{R}^{mxp}$ est la matrice des régresseurs ;

- $e \in \mathbb{R}^m$ est le vecteur des erreurs affectant la sortie. Chaque élément est supposé borné (et normalisé) :

$$|e_i| \leq 1, \ \forall i, \ i = 1:m \tag{5.4}$$

La $i^{\text{ème}}$ bande du paquet est donc définie par :

$$\mathbf{B}_i = \left\{ x \in \mathbb{R}^p : y_i - 1 \leq C_i x \leq y_i + 1 \right\} \tag{5.5}$$

où C_i est la $i^{\text{ème}}$ ligne de la matrice C. Dans la démarche proposée, on cherche l'ellipsoïde (de taille minimale) englobant l'intersection de l'ellipsoïde initial et des m bandes traitées simultanément (d'où la notion de paquet). L'ellipsoïde ainsi trouvé est ensuite utilisé comme ellipsoïde initial pour le paquet suivant. Cette démarche possède les propriétés suivantes :

- pour chaque paquet, l'ellipsoïde englobant obtenu est optimal. La résolution du problème d'optimisation sous-jacent dans ce cas ne sera pas difficile si le nombre de bandes m (choisi par l'utilisateur) n'est pas « trop » grand ;
- en traitant des paquets séquentiellement, cette démarche hérite des propriétés et avantages de l'algorithme OBE.

En conséquence, on peut penser que le traitement par paquet conduira à un résultat moins pessimiste (i.e. moins surenglobant), ou pour le moins convergent plus rapidement que celui obtenu par un traitement séquentiel de toutes les données, l'optimum étant atteint si toutes les données sont traitées simultanément

Avant de décrire l'algorithme proposé, on présente maintenant l'algorithme qui cherche l'ellipsoïde optimal global. Les algorithmes rencontrés dans la littérature et qui s'intéressent à traiter un paquer de données seront donnés dans la suite. Sans perte de généralité, on va noter l'ellipsoïde initial $\mathbf{E}_0$ et les m bandes associées au paquet de données $\mathbf{B}_i$ avec $i = 1:m$.

5.2 Ellipsoïde optimal globalement

On envisage tout d'abord le problème de chercher l'ellipsoïde optimal globalement proposé par Pronzato et Walter dans [PRO 94]. Supposons que $\mathbf{S}_l$ est l'intersection de toutes les bandes $\mathbf{B}_k$ $(k = 1:l)$ de paramètres correspondant à toutes les observations dans le fichier de données. $\mathbf{S}_l$ est en fait un polyèdre avec un nombre fini de faces ou de sommets. On dénote $\mathbf{\Theta}$ l'ensemble de tous les sommets de $\mathbf{S}_l$, i.e. :

$$\mathbf{\Theta} = \left[x_{M_i} \in \mathbb{R}^p, \ i = 1:n, \ n > p \right] \tag{5.6}$$

où x_{M_i} sont les coordonnées du sommet M_i de $\mathbf{S}_l$. Pour éviter une situation dégénérée, on suppose que l'ensemble convexe fermé englobant $\mathbf{\Theta}$, dénoté $\mathbf{\Theta}^c$, a un intérieur non-vide. Pronzato et Walter dans [PRO 94] proposent un algorithme pour chercher l'ellipsoïde de volume minimal contenant $\mathbf{\Theta}^c$ dans le théorème suivant.

Théorème 5.1

L'ellipsoïde de volume minimal contenant $\mathbf{\Theta}$ *(et ainsi* $\mathbf{\Theta}^c$*) est unique. Il est exprimé par :*

$$\mathbf{E}\left(c(\lambda^*), P(\lambda^*)\right) = \left\{x \in \mathbb{R}^p : (x-c)^T P^{-1}(x-c) \leq 1\right\} \tag{5.7}$$

où

$$c(\lambda) = \sum_{i=1}^{n} \lambda_i x_{M_i} \Big/ \sum_{i=1}^{n} \lambda_i$$

$$P(\lambda) = \sum_{i=1}^{n} x_{M_i} x_{M_i}^T - \left(\sum_{i=1}^{n} \lambda_i\right) cc^T \tag{5.8}$$

$$\lambda^* = \arg\max_{\lambda \geq 0} \ln\left[\det P\right]$$

avec pour contrainte :

$$\sum_{i=1}^{n} \lambda_i = p \tag{5.9}$$

La démonstration peut être consultée dans [PRO 94].

Remarques :

- le calcul de l'ellipsoïde englobant nécessite la connaissance de tous les sommets de l'ensemble $\mathbf{S}_l$. En pratique, le nombre de sommets de $\mathbf{S}_l$ est souvent très grand, donc la détermination de tous ces sommets est très coûteuse en temps calcul ;
- le vecteur λ^* des coefficients de pondération est la solution d'un problème d'optimisation de n variables où n est le nombre de sommets. Il n'y a pas de solution explicite pour calculer λ^*. Il faut donc faire appel à une routine d'optimisation (par exemple « fmincon » de Matlab) pour chercher λ^*. Le problème d'optimisation dans ce cas est extrêmement lourd.

L'utilisation directe du théorème 5.1 pour calculer l'ellipsoïde englobant $\mathbf{E}\left(c(\lambda^*), P(\lambda^*)\right)$ est très difficile et semble irréalisable. Pronzato et Walter [PRO 94] proposent alors une autre façon de chercher $\mathbf{E}(c,P)$ sans avoir besoin des sommets de $\mathbf{S}_l$.

Définissons une distribution normalisée continue sur $\mathbf{\Theta}^c$ telle que :

$$\int_{\Theta^c} \xi dx = 1 \tag{5.10}$$

L'ellipsoïde englobant $\mathbf{E}\left(c(\xi^*), P(\xi^*)\right)$ est calculé par :

$$\begin{aligned} c(\xi) &= \int_{\Theta^c} x\xi dx \\ P(\xi) &= p\left[M(\xi) - c(\xi)c^T(\xi)\right] \end{aligned} \tag{5.11}$$

où

$$\begin{aligned} M(\xi) &= \int_{\Theta^c} xx^T \xi dx \\ \xi^* &= \arg\max_{\xi \in \Xi} \ln\left[\det P(\xi)\right] \end{aligned} \tag{5.12}$$

et $\mathbf{\Xi}$ est l'ensemble de toutes les distributions normalisées sur $\mathbf{\Theta}^c$.

Cette proposition ne fait plus appel aux sommets de $\mathbf{S}_l$. En revanche, on doit chercher une distribution normalisée vérifiant (5.10). Une procédure permettant de trouver une distribution discrète a été présentée dans [PRO 94]. Cependant cette procédure est encore compliquée. De plus, pour récupérer le centre c et la matrice caractéristique P de l'ellipsoïde, on doit réaliser un calcul d'intégration sur l'ensemble $\mathbf{\Theta}^c$ dans (5.11) qui est souvent très difficile.

En bref, l'objectif qui consiste à chercher l'ellipsoïde optimal globalement semble irréalisable car on doit résoudre un problème d'optimisation de trop grande dimension.

5.3 Méthodes classiques de traitement d'un paquet de données

5.3.1 Une procédure améliorée pour un traitement séquentiel [MAK 96]

Lors du traitement séquentiel des données, avant tout calcul, on teste la validité de l'intersection de l'ellipsoïde $\mathbf{E}_{i\text{-}1}$ (défini par sa matrice caractéristique $P_{i\text{-}1}$ et son centre $c_{i\text{-}1}$) et de la bande $\mathbf{B}_i$ (test dit « de cohérence » dans les chapitres précédents). Si besoin, on réajuste la bande. Ce réajustement est effectué en considérant la position des deux hyperplans « frontières » de $\mathbf{B}_i$ par rapport à $\mathbf{E}_{i\text{-}1}$. Rappelons à cet effet (1.43) - (1.44), que la bande $\mathbf{B}_i$ est remplacée par $\mathbf{B}_i^{'}$ où (notations normalisées de l'ellipsoïde) :

$$\mathbf{B}_i^{'} = \left\{x \in \mathbb{R}^p : b_i^{'-} \le C_i x \le b_i^{'+}\right\} \tag{5.13}$$

avec
$$\begin{cases} b_i^{'+} = \min\left\{ y_i + 1; C_i c_{i-1} + \sqrt{C_i P_{i-1} C_i^T} \right\} \\ b_i^{'-} = \max\left\{ y_i - 1; C_i c_{i-1} - \sqrt{C_i P_{i-1} C_i^T} \right\} \end{cases} \tag{5.14}$$

où C_i est est la $i^{\text{ème}}$ ligne de la matrice C.

La Figure 5.2 illustre cette réduction de bande ($\mathbf{B}_2$ devient $\mathbf{B}_2^{'}$) pour p = 2 paramètres, lorsque l'on considère un traitement séquentiel des bandes du paquet considéré.

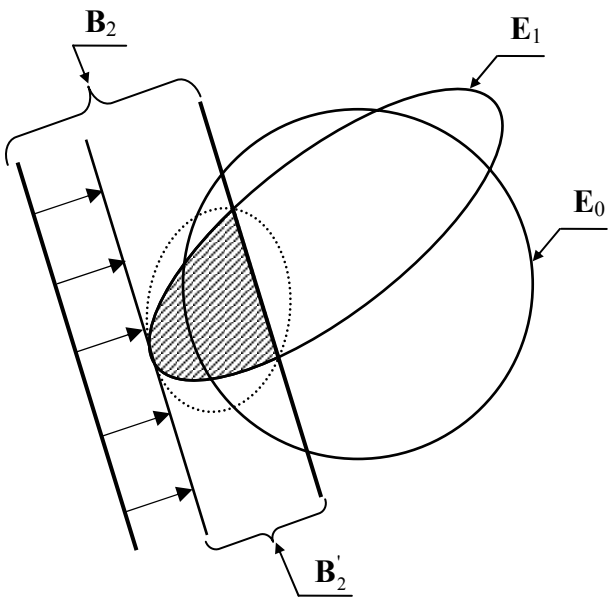


Figure 5.2: Procédure standard de modification de la bande $\mathbf{B}_2$.

En fait, tous les ellipsoïdes calculés avec les bandes précédentes (dans le paquet considéré) contiennent l'espace admissible de paramètres. Pour réduire la bande $\mathbf{B}_i$, Maksarov et Norton [MAK 96] proposent de modifier $\mathbf{B}_i$ en $\mathbf{B}_i^{''}$ en considérant la position des deux hyperplans frontière de $\mathbf{B}_i$ par rapport à **tous les ellipsoïdes antérieurs** (à savoir $\mathbf{E}_0$ à $\mathbf{E}_{i-1}$). Ainsi, $\mathbf{B}_i^{''}$ est déterminé par [MAK 96] :

$$\mathbf{B}_i^{''} = \left\{ x \in \mathbb{R}^p : b_i^{''-} \le C_i x \le b_i^{''+} \right\} \tag{5.15}$$

avec
$$\begin{cases} b_i^{''+} = \min_j \left\{ b_{ij}^{'+} \right\} \\ b_i^{''-} = \max_j \left\{ b_{ij}^{'-} \right\} \end{cases} \quad j = 0 : (i\text{-}1) \tag{5.16}$$

où $b_{ij}^{'+}$, $b_{ij}^{'-}$ représentent le réajustement de l'hyperplan frontière de la bande courante $\mathbf{B}_i$ par rapport au $j^{\text{ème}}$ ellipsoïde :

$$\begin{cases} b_{ij}^{'+} = \min\left\{ y_i + 1; C_i c_j + \sqrt{C_i P_j C_i^T} \right\} \\ b_{ij}^{'-} = \max\left\{ y_i - 1; C_i c_j - \sqrt{C_i P_j C_i^T} \right\} \end{cases} \tag{5.17}$$

Cette procédure de réduction de la borne est illustrée dans la figure 5.3 pour la 2ème bande du paquet de données. On voit qu'ainsi, l'hyperplan $\mathbf{H}_2^-$ est déplacé jusqu'à être tangent à l'ellipsoïde $\mathbf{E}_0$. $\mathbf{H}_2^+$ coupe les deux ellipsoïdes $\mathbf{E}_0$ et $\mathbf{E}_1$: il reste donc inchangé. La bande trouvée dans ce cas $\mathbf{B}_2^{''}$ est plus petite que celle obtenue par une réduction standard (i.e. en ne tenant compte que de l'ellipsoïde précédemment calculé $\mathbf{E}_1$). $\mathbf{B}_2^{''}$ est ensuite utilisée pour le traitement séquentiel avec $\mathbf{E}_1$, l'ellipsoïde calculé $\mathbf{E}_2$ étant de plus petite dimension. Notons néanmoins que cette procédure nécessite la mise en mémoire de tous les ellipsoïdes intermédiaires $\mathbf{E}_0$ à $\mathbf{E}_{m\text{-}1}$.

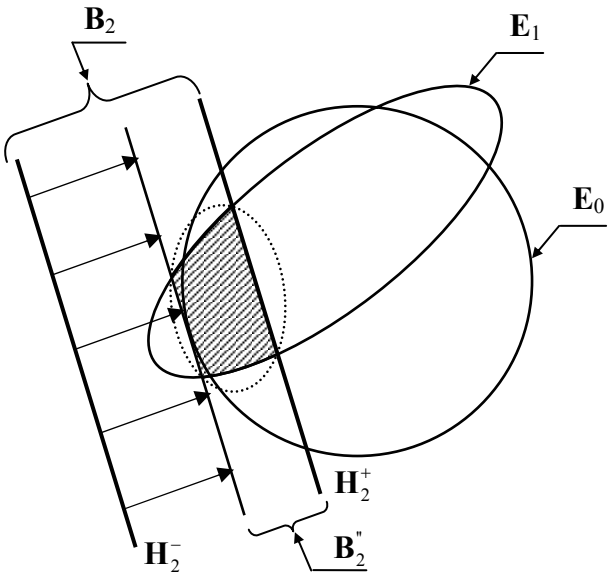


Figure 5.3 : Procédure améliorée de modification de la bande $\mathbf{B}_2$.

5.3.2 Un ellipsoïde approximant l'intersection de deux ellipsoïdes

Cette méthode est proposée dans [SCH 68] et [MAK 02] où le vecteur d'erreur e est supposé borné tel que :

$$e^T R^{-1} e \leq 1 \tag{5.18}$$

avec R une matrice symétrique définie positive. En fait, à partir de l'hypothèse sur la borne de chaque élément de l'erreur (5.4), on peut toujours se ramener à la condition (5.18). Par exemple, si la contribution de chaque bande du paquet est identique, on peut choisir

$R = \frac{1}{m} I_m$. En utilisant l'équation (5.3) et avec la condition (5.18), [MAK 02] montre que tous les paramètres admissibles appartiendront à un ellipsoïde $\mathbf{E_M}$ défini par :

$$\mathbf{E_M} = \left\{ x \in \mathbb{R}^p : (y - Cx)^T R^{-1} (y - Cx) \leq 1 \right\} \tag{5.19}$$

L'ellipsoïde $\mathbf{E_M}$ englobe bien maintenant l'intersection des m bandes du paquet. Le calcul de l'ellipsoïde $\mathbf{E}$ englobant $\mathbf{E}_0 \bigcap_{i=1}^{m} \mathbf{B}_i$ devient un problème de calcul de l'intersection de deux ellipsoïdes, i.e. $\mathbf{E} \supset \mathbf{E}_0 \cap \mathbf{E_M}$. D'après [MAK 02], cette intersection existe si :

$$p\left(1 - e^T R^{-1} e\right) - tr\left(P_0 C^T R^{-1} C\right) < 0 \tag{5.20}$$

où p est le nombre de paramètres, P_0 est la matrice caractéristique de l'ellipsoïde $\mathbf{E}(c_0, P_0)$ de centre c_0 et C est la matrice des m régresseurs. L'ellipsoïde englobant est calculé par :

$$\mathbf{E} = \left\{ x \in \mathbb{R}^p : (x - c)^T P^{-1} (x - c) \leq 1 \right\} \tag{5.21}$$

avec
$$\begin{cases} c = c_0 + L\nu \\ P = \left[1 + \lambda - \lambda \nu^T \left(\lambda C P_0 C^T + R \right)^{-1} \nu \right] (I - LC) P_0 \\ \nu = y - C c_0 \\ L = \lambda P_0 C^T \left(\lambda C P_0 C^T + R \right)^{-1} \end{cases} \tag{5.22}$$

Comme dans l'algorithme OBE, l'ellipsoïde englobant dépend du coefficient $\lambda \geq 0$ calculé selon un critère choisi par l'utilisateur. Lorsque l'on minimise le volume de l'ellipsoïde englobant, λ est la solution réelle positive de l'équation [MAK 02] :

$$\det\left(C^T R^{-1} C - \lambda P_0^{-1}\right) = \det\left(P_0 C^T R^{-1} C - \lambda I\right) = 0 \tag{5.23}$$

Notons que (5.23) ne donne pas (en général) une solution explicite pour λ. Numériquement, on est devant un problème de calcul des valeurs propres généralisées de la paire $\left(C^T R^{-1} C, P_0^{-1}\right)$ ou classique de la matrice $\left(P_0 C^T R^{-1} C\right)$.

5.3.3 Synthèse

En résumé, les deux méthodes présentées ci-avant ont abordé le problème de traitement d'un paquet de données avec deux stratégies différentes :

- la première méthode réalise encore un traitement séquentiel des données. La procédure de modification de la bande $\mathbf{B}_i$ (qui appartient au paquet de m bandes), est simpliste mais nécessite la mise en mémoire des ellipsoïdes précédemment calculés pour les bandes déjà traitées. Néanmoins, cette procédure (toujours séquentielle) améliore l'ellipsoïde obtenu

au sens où sa taille est diminuée en comparaison à un traitement standard séquentiel du paquet de bandes) ;

- la deuxième méthode calcule un ellipsoïde $\mathbf{E_M}$ (5.19) englobant l'intersection des m bandes du paquet. On doit ensuite calculer l'ellipsoïde englobant l'intersection de deux ellipsoïdes, $\mathbf{E} \supset \mathbf{E}_0 \cap \mathbf{E_M}$ sous l'hypothèse (5.20).

5.4 Algorithme OBE-paquet

On va construire maintenant un algorithme de type OBE qui effectue un traitement simultané des m données d'un paquet. Géométriquement, on recherche un ellipsoïde qui englobe l'intersection de l'ellipsoïde $\mathbf{E}_0$ avec les m bandes, et ceci, en une seule étape. La contribution de chaque bande à l'ellipsoïde englobant global est analysée pour calculer l'ellipsoïde de taille minimum.

5.4.1 Formulation de l'algorithme

La formulation de l'algorithme OBE-paquet est dérivée des algorithmes qui permettent de calculer l'intersection simultanée de plusieurs bandes [DUR 96] et de plusieurs ellipsoïdes [DUR 01b]. On a adapté ces algorithmes au cas particulier de l'intersection entre l'ellipsoïde $\mathbf{E}_0$ avec les m bandes. L'algorithme OBE-paquet est donné dans le théorème 5.2.

Théorème 5.2

Supposons que l'ellipsoïde $\mathbf{E}_0$ *et les m bandes* $\mathbf{B}_i$ $(i = 1:m)$ *soient définies par leur forme normalisée (5.1) et (5.5) respectivement. L'ellipsoïde* $\mathbf{E}$ *englobant l'intersection* $\mathbf{E}_0 \bigcap_{i=1}^{m} \mathbf{B}_i$ *est calculé par :*

$$\mathbf{E} = \left\{ x \in \mathbb{R}^p : (x-c)^T M (x-c) \leq 1,\ M = P^{-1} \right\} \tag{5.24}$$

où

$$\begin{cases} c = L^{-1}\left(\alpha_0 M_0 c_0 + \sum_{i=1}^{m} \alpha_i C_i^T y_i \right) \\ M = \left(1 - \delta_\alpha\right)^{-1} L \\ L = \alpha_0 M_0 + \sum_{i=1}^{m} \alpha_i C_i^T C_i \\ \delta_\alpha = \alpha_0 c_0^T M_0 c_0 + \sum_{i=1}^{m} \alpha_i y_i^2 - c^T L c \end{cases} \tag{5.25}$$

avec pour contraintes :

$$0 < \alpha_0 \leq 1; \qquad 0 \leq \alpha_i < 1 \ \textit{pour} \ i = 1:m \qquad \textit{et} \qquad \sum_{i_+=0}^{m} \alpha_{i_+} = 1 \tag{5.26}$$

Remarque : on note ici $i = 1:m$ et $i_+ = 0:m$.

Démonstration :

D'après [DUR 96] et [DUR 01b], tous les ellipsoïdes englobant l'intersection $\mathbf{E}_0 \bigcap_{i=1}^{m} \mathbf{B}_i$ satisfont l'inéquation :

$$\alpha_0 \left(x - c_0\right)^T M_0 \left(x - c_0\right) + \sum_{i=1}^{m} \alpha_i \left(y - C_i x\right)^T \left(y - C_i x\right) \leq \sum_{i_+=0}^{m} \alpha_{i_+} = 1 \tag{5.27}$$

Le développement de (5.27) conduit à :

$$\left(x - c\right)^T L \left(x - c\right) \leq 1 - \delta_\alpha \tag{5.28}$$

Les expressions de C, L et δ_α sont celles données dans l'équation (5.25). (5.28) peut être réécrite en :

$$\left(x - c\right)^T \left(1 - \delta_\alpha\right)^{-1} L \left(x - c\right) \leq 1$$

En comparant avec (5.24), on trouve $M = P^{-1} = (1 - \delta)^{-1} L$. □

L'ellipsoïde englobant **E** dépend des coefficients α_{i_+} $\left(i_+ = 0:m\right)$ sous les contraintes (5.26). Ainsi, l'influence de chaque bande $\mathbf{B}_i$ sur l'ellipsoïde **E** est représentée via le coefficient (le poids) α_i. Le calcul du vecteur $\alpha = \left[\alpha_0 \quad \alpha_1 \quad \ldots \quad \alpha_m\right]^T$ est basé sur les deux critères suivants :

- *Critère du logarithme du déterminant :*

$$\alpha^* = \arg\min_{\alpha} \mu_{lV} \tag{5.29}$$

avec $$\mu_{lV} = \log\det P = \log\det M^{-1} \tag{5.30}$$

- *Critère de la trace :*

$$\alpha^* = \arg\min_{\alpha} \mu_T \tag{5.31}$$

avec $$\mu_T = \mathrm{tr} P = \mathrm{tr}\left(M^{-1}\right) \tag{5.32}$$

Il faut noter que l'on préfère utiliser pour critère *le logarithme du déterminant* à la place du *déterminant* lui-même ce qui évite d'avoir un minimum plat en échelle linéaire. Le vecteur des coefficients optimaux α^* est la solution d'un problème d'optimisation à plusieurs variables, sans solution explicite α^*. Il faut donc utiliser une routine d'optimisation pour calculer α^*, routine qui ne fonctionnera véritablement que si le gradient analytique est connu. On a calculé l'expression des gradients de $\mu_{IV} = \log\det P$ (5.30) et de $\mu_T = \mathrm{tr}P$ (5.32).

a) Calcul du gradient de $\mu_{IV} = \log\det P$

On a : $\mu_{IV} = \log\det P = \log\det\left(1-\delta_\alpha\right)L^{-1} = p\log\left(1-\delta_\alpha\right) + \log\det L^{-1}$

Donc
$$\frac{\partial\mu_{IV}}{\partial\alpha_{i_+}} = -p\frac{\partial\delta_\alpha}{\partial\alpha_{i_+}}\frac{1}{1-\delta_\alpha} + \frac{\partial\left(\log\det L^{-1}\right)}{\partial\alpha_{i_+}} \tag{5.33}$$

- Calcul de $\dfrac{\partial\delta_\alpha}{\partial\alpha_{i_+}}$:

Posons $w = \alpha_0 M_0 c_0 + \sum_{i=1}^{m}\alpha_i C_i^T y_i$. D'après (5.25) :

$$\delta_\alpha = \alpha_0 c_0^T M_0 c_0 + \sum_{i=1}^{m}\alpha_i y_i^2 - w^T L w$$

alors
$$\frac{\partial\delta_\alpha}{\partial\alpha_0} = c_0^T M_0 c_0 - \frac{\partial\left(w^T L^{-1} w\right)}{\partial\alpha_0};\quad \frac{\partial\delta_\alpha}{\partial\alpha_i} = y_i^2 - \frac{\partial\left(w^T L^{-1} w\right)}{\partial\alpha_i}$$

On a
$$\begin{cases} \dfrac{\partial\left(w^T L^{-1} w\right)}{\partial\alpha_{i_+}} = 2\dfrac{\partial w^T}{\partial\alpha_{i_+}}L^{-1}w + w^T\dfrac{\partial L^{-1}}{\partial\alpha_{i_+}}w \\ \dfrac{\partial L^{-1}}{\partial\alpha_{i_+}} = -L^{-1}\dfrac{\partial L}{\partial\alpha_{i_+}}L^{-1} \end{cases}$$

En dérivant $\dfrac{\partial w}{\partial\alpha_{i_+}}$, $\dfrac{\partial L}{\partial\alpha_{i_+}}$ et en les remplaçant dans la formule ci-avant, on obtient :

$$\begin{cases} \dfrac{\partial\delta_\alpha}{\partial\alpha_0} = c_0^T M_0 c_0 - 2c_0^T M_0 L^{-1} w + w^T L^{-1} M_0 L^{-1} w \\ \dfrac{\partial\delta_\alpha}{\partial\alpha_i} = y_i^2 - 2y_i C_i L^{-1} w + w^T L^{-1} C_i^T C_i L^{-1} w \end{cases} \tag{5.34}$$

- Calcul de $\dfrac{\partial\left(\log\det L^{-1}\right)}{\partial\alpha_{i_+}}$

D'après [DUR 01b], cette dérivée est donnée par :

$$\frac{\partial\left(\log\det L^{-1}\right)}{\partial\alpha_0}=-\mathrm{tr}\left(L^{-1}M_0\right)\ ;\ \frac{\partial\left(\log\det L^{-1}\right)}{\partial\alpha_i}=-\mathrm{tr}\left(L^{-1}C_i^TC_i\right) \tag{5.35}$$

En remplaçant (5.34), (5.35) dans (5.33), on obtient finalement :

$$\begin{cases}\dfrac{\partial\mu_{lV}}{\partial\alpha_0}=-p\dfrac{c_0^TM_0c_0-2c_0^TM_0L^{-1}w+w^TL^{-1}M_0L^{-1}w}{1-\delta_\alpha}-\mathrm{tr}\left(L^{-1}M_0\right)\\ \dfrac{\partial\mu_{lV}}{\partial\alpha_i}=-p\dfrac{y_i^2-2y_iC_iL^{-1}w+w^TL^{-1}C_i^TC_iL^{-1}w}{1-\delta_\alpha}-\mathrm{tr}\left(L^{-1}C_i^TC_i\right)\end{cases} \tag{5.36}$$

b) Calcul du gradient de $\mu_T=\mathrm{tr}P$

$$\mu_T=\mathrm{tr}P=\left(1-\delta_\alpha\right)\mathrm{tr}L^{-1}$$

$$\Rightarrow\quad \frac{\partial\mu_T}{\partial\alpha_{i_+}}=-\frac{\partial\delta}{\partial\alpha_{i_+}}\mathrm{tr}L^{-1}+\left(1-\delta_\alpha\right)\frac{\partial\mathrm{tr}L^{-1}}{\partial\alpha_{i_+}} \tag{5.37}$$

D'après [DUR 01b] :

$$\frac{\partial\mathrm{tr}L^{-1}}{\partial\alpha_0}=-\mathrm{tr}\left(L^{-1}M_0L^{-1}\right);\ \frac{\partial\mathrm{tr}L^{-1}}{\partial\alpha_i}=-\mathrm{tr}\left(L^{-1}C_i^TC_iL^{-1}\right) \tag{5.38}$$

Remplaçons (5.34), (5.38) dans (5.37), on obtient finalement :

$$\begin{cases}\dfrac{\partial\mu_T}{\partial\alpha_0}=-\left(c_0^TM_0c_0-2c_0^TM_0L^{-1}w+w^TL^{-1}M_0L^{-1}w\right)\mathrm{tr}L^{-1}-\left(1-\delta_\alpha\right)\mathrm{tr}\left(L^{-1}M_0L^{-1}\right)\\ \dfrac{\partial\mu_T}{\partial\alpha_i}=-\left(y_i^2-2y_iC_iL^{-1}w+w^TL^{-1}C_i^TC_iL^{-1}w\right)\mathrm{tr}L^{-1}-\left(1-\delta_\alpha\right)\mathrm{tr}\left(L^{-1}C_i^TC_iL^{-1}\right)\end{cases} \tag{5.39}$$

5.4.2 Prétraitement des données

Comme dans l'algorithme OBE standard, l'utilisation directe des données requises pour calculer l'ellipsoïde englobant peut conduire à une solution sous-optimale. De plus, l'algorithme OBE-paquet doit utiliser une fonction d'optimisation pour rechercher α^*. Il en résulte qu'ici encore, l'ellipsoïde obtenu dépend largement des données utilisées. Il semble donc indispensable d'appliquer un prétraitement aux données avant d'appeler l'algorithme OBE-paquet. Ce prétraitement comprend 2 phases :

- **Phase 1 :** réajustement des bandes en relation avec la région $\mathbf{S}_m$ définissant l'intersection de ces bandes elles-mêmes (voir figure 5.4) ;
- **Phase 2 :** Réajustement des bandes en relation avec l'ellipsoïde initial $\mathbf{E}_0$.

Ces deux phases sont indépendantes ; on peut donc réaliser la phase 1 puis la phase 2 ou inversement.

5.4.2.1 Réajustement des bandes en relation avec la région $\mathbf{S_m}$

La figure 5.4 illustre l'intersection de $m = 4$ bandes dans un espace paramétrique de dimension $p = 2$. Considérons la région $\mathbf{S}_m$ générée par l'intersection des m bandes dans l'espace des paramètres. Deux cas se présentent :

- la région $\mathbf{S}_m$ est complètement contenue dans une bande, par exemple la bande $\mathbf{B}_4$ de la figure 5.4. Ainsi la présence de cette bande n'a pas d'influence sur $\mathbf{S}_m$. On peut donc éliminer cette bande inutile pour la définition de $\mathbf{S}_m$. L'observation correspondant à cette bande est également supprimée dans le paquet de données considéré ;
- un des deux hyperplans « frontière » d'une bande n'est pas tangent à $\mathbf{S}_m$, par exemple $\mathbf{H}_2^+$, $\mathbf{H}_3^-$ dans la figure 5.4. Il nous faut alors ramener ces hyperplans de manière à ce qu'ils soient tangents à $\mathbf{S}_m$.

Pour éliminer les bandes inutiles et déplacer les hyperplans comme on vient de l'indiquer, on doit tout d'abord détecter les sommets de la région $\mathbf{S}_m$. Dans l'espace de dimension p, un point (ou sommet) est créé par l'intersection de p hyperplans. En conséquence, avec un paquet de m données, l'existence des sommets est assurée par $m \geq p$. Les points peuvent être classés suivant :

- qu'ils se situent sur la frontière de $\mathbf{S}_m$, par exemple M_1, M_2, M_3, M_4 sur la figure 5.4. Ces points sont les sommets de $\mathbf{S}_m$;
- qu'ils n'appartiennent pas à $\mathbf{S}_m$, par exemple N_1, N_2, N_3, …

Considérons maintenant un sommet M où x_M sont les coordonnées de M dans l'espace des paramètres. x_M possède les deux propriétés suivantes :

- x_M est la solution d'un système de p équations définissant les hyperplans qui génèrent le sommet M considéré ;
- x_M satisfait $(2m - p)$ inéquations restantes.

La détermination du sommet M vérifiant ces deux propriétés peut être effectuée grâce à la méthode du simplex (voir annexe A). Dans cette méthode, chaque sommet M de régions $\mathbf{S}_m$ correspondra à un point *bfp* (basic feasible point). A partir des inéquations de contraintes de la bande (5.5), la méthode du simplex construit le système d'équations linéaires :

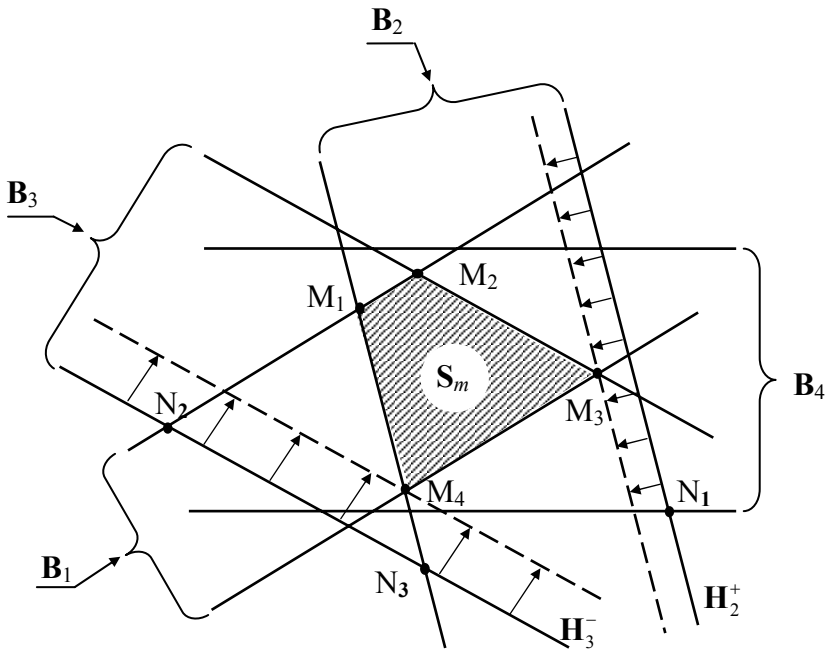


Figure 5.4 : Intersection de $m = 4$ bandes dans un espace à $p = 2$ paramètres.

$$\begin{cases} C_{11}x_1 + C_{12}x_2 + \ldots + C_{1p}x_p + t_1 = y_1 + 1 \\ \ldots \\ C_{m1}x_1 + C_{m2}x_2 + \ldots + C_{mp}x_p + t_m = y_m + 1 \\ -\left(C_{11}x_1 + C_{12}x_2 + \ldots + C_{1p}x_p\right) + t_{m+1} = -\left(y_1 - 1\right) \\ \ldots \\ -\left(C_{m1}x_1 + C_{m2}x_2 + \ldots + C_{mp}x_p\right) + t_{ém} = -\left(y_m - 1\right) \end{cases} \tag{5.40}$$

où $t \in \mathbb{R}^{2m}$ est le vecteur des variables supplémentaires [ROT 79] satisfaisant :

$$t_j \geq 0,\ \forall j = 1 : 2m \tag{5.41}$$

Posons :

$$\begin{aligned} \mathcal{A} &= \left[\begin{bmatrix} C \\ -C \end{bmatrix} \quad \left[I_{2m}\right]\right],\ \mathcal{A} \in \mathbb{R}^{(2m)\times(2m+p)} \\ \mathcal{b} &= \left[(y_1+1) \quad \ldots \quad (y_m+1) \quad -(y_1-1) \quad \ldots \quad -(y_m-1)\right]^T \\ \mathcal{X} &= \begin{bmatrix} x \\ t \end{bmatrix},\ \mathcal{X} \in \mathbb{R}^{2m+p} \end{aligned} \tag{5.42}$$

Le système des équations dans (5.40) peut être réécrit par :

$$\mathcal{A}\mathcal{X} = \mathcal{b} \tag{5.43}$$

Ce système de $2m$ équations linéaires possède p variables x_i et $2m$ variables t_j. Par application de la méthode du simplex, un point *bfp*, et donc un sommet M, peut être déterminé par la procédure 5.1 donnée ci-après.

Procédure 5.1

- *Choisir une combinaison* Ω *de p hyperplans dont on cherche le point d'intersection, par exemple* $\Omega = \{1, 2, ..., p\}$.
- *Eliminer les p variables du vecteur t correspondant aux indices dans* Ω. *On obtient les vecteur* $\bar{t}$ *et* $\bar{X} = \begin{bmatrix} x \\ \bar{t} \end{bmatrix}$.
- *Eliminer les p colonnes de matrice* I_{2m} *dans (5.42) correspondant aux indices dans* Ω. *On obtient la matrice* $\bar{\mathcal{A}}$ *avec* $\bar{\mathcal{A}} \in \mathbb{R}^{2m \times 2m}$.
- *Résoudre le système de 2m équations linaires* $\bar{\mathcal{A}}\bar{X} = b$ *:*
  - *Si* $\bar{\mathcal{A}}$ *est de rang plein, on pose* $\bar{X}_\Omega = \begin{bmatrix} x_\Omega \\ \bar{t}_\Omega \end{bmatrix}$
    - *Si tous les éléments du vecteur* $\bar{t}_\Omega$ *sont non négatifs, alors* x_Ω *est un sommet de* $\mathbf{S}_m$ *(* $x_\Omega \equiv x_M$ *);*
    - *Sinon* $x_\Omega \neq x_M$ *;*
  - *Sinon, les p hyperplans sélectionnés dans* Ω *n'ont pas de point d'intersection.*

Remarques

- Dans la combinaison Ω, si l'on choisit deux hyperplans parallèles (par exemple deux hyperplans « frontières » d'une bande), la matrice $\bar{\mathcal{A}}$ créée dans la procédure 5.1 ne sera pas de rang plein et il est aisé de vérifier que les hyperplans sélectionnées dans Ω n'ont pas de point d'intersection.
- Pour un paquet de longueur *m*, dans un espace de paramètres de dimension *p*, le nombre des combinaisons Ω est donné par :

$$\Sigma = \frac{(2m-2)!}{(p-1)!(2m-p-1)!} \tag{5.44}$$

  Dès que *p* dépasse quelques unités (par exemple $p \geq 5$), compte tenu de la condition $m \geq p$ garantissant l'existence d'un sommet, Σ devient très grand (phénomène d'explosion combinatoire).
- Si l'on ne trouve aucun sommet pour la région $\mathbf{S}_m$, deux cas peuvent apparaître, à savoir, soit $\mathbf{S}_m$ n'est pas borné (par exemple, $\mathbf{S}_m$ est l'intersection de 2 bandes dans l'espace à 2 paramètres où tous les hyperplans de contrainte des bandes sont parallèles) ; soit $\mathbf{S}_m$ est vide.

Après avoir déterminé tous les sommets de la région $\mathbf{S}_m$, on peut rapidement détecter les bandes inutiles pour lesquelles les deux hyperplans de contraintes ne contiennent aucun sommet M. Lorsque l'un des deux hyperplans de contraintes de la bande $\mathbf{B}_i$ est en dehors de la région $\mathbf{S}_m$ (il ne contient aucun sommet), on « ramène » cet hyperplan vers le sommet le plus proche. La bande $\mathbf{B}_i$ correspondante est ainsi modifiée. Son réajustement et sa normalisation en $\mathbf{B}_i^{'}$ sont présentés dans la procédure 5.2.

Procédure 5.2

- *Détecter l'hyperplan à déplacer, par exemple* $\mathbf{H}_i$ *où* $\mathbf{H}_i = \left\{ x \in \mathbb{R}^p : C_i x = b_i \right\}$.
  - *Chercher les coordonnées* $x_{M_{\min}}$ *du sommet le plus proche de* $\mathbf{H}_i$ *en résolvant:*

$$x_{M_{\min}} = \arg\min_{x_M} \left| C_i x_M - b_i \right| \tag{5.45}$$

- *Ramener* $\mathbf{H}_i$ *vers* $\mathbf{H}_i^{'}$ *tel que* $x_{M_{\min}} \in \mathbf{H}_i^{'}$. *On obtient ainsi* $\mathbf{B}_i^{'}$ *(non normalisée).*
- *Normaliser* $\mathbf{B}_i^{'}$ *telle que* $\mathbf{B}_i^{'} = \left\{ x \in \mathbb{R}^p : y_i^{'} - 1 \le C_i^{'} x \le y_i^{'} + 1 \right\}$ *où :*

$$y_i^{'} = \frac{b_i^{'+} + b_i^{'-}}{b_i^{'+} - b_i^{'-}};\ \mathrm{C}_i^{'} = \frac{2C_i}{b_i^{'+} - b_i^{'-}} \tag{5.46}$$

  - *Si* $\mathbf{H}_i \equiv \mathbf{H}_i^{-}$: $b_i^{'-} = C_i x_{M_{\min}}; b_i^{'+} = y_i + 1$
  - *Si* $\mathbf{H}_i \equiv \mathbf{H}_i^{+}$: $b_i^{'-} = y_i - 1; b_i^{'+} = C_i x_{M_{\min}}$

5.4.2.2 Réajustement des bandes en relation avec l'ellipsoïde $\mathbf{E}_0$

Dans cette phase, chaque bande est testée et modifiée (si besoin) indépendamment selon la procédure présentée dans la section 1.2.4. Rappelons que l'on peut rencontrer les trois cas suivants :

- si $\mathbf{B}_i \cap \mathbf{E}_0 = \varnothing$, $\mathbf{B}_i$ n'est pas « cohérente » avec $\mathbf{E}_0$. Il en résulte que $\mathbf{S}_m \cap \mathbf{E}_0 = \varnothing$;
- si $\mathbf{B}_i \supset \mathbf{E}_0$: on enlève du paquet de données l'observation correspondante à la bande $\mathbf{B}_i$;
- si l'un des deux hyperplans de contrainte de $\mathbf{B}_i$ ne coupe pas $\mathbf{E}_0$, on déplace cet hyperplan tel qu'il tangente l'ellipsoïde $\mathbf{E}_0$.

5.4.2.3 Synthèse sur le réajustement des bandes

En résumé, les deux phases de prétraitement des données proposées aux deux paragraphes précédents permettent :

- d'éliminer les bandes inutiles (i.e. qui n'apportent pas d'information pour la construction du nouvel ellipsoïde englobant). Ainsi, la longueur de paquet est réduite et le nombre de variable lors de l'appel à la fonction d'optimisation est diminué ;
- grâce au réajustement des bandes, les bandes utilisées pour le problème d'optimisation sont « complètement utiles » dans le sens où elles contribuent pleinement au calcul du nouvel ellipsoïde. Ainsi, le temps de calcul dans la partie optimisation de l'algorithme OBE-paquet sera diminué.

Enfin, avant tout appel de l'algorithme OBE paquet, il est nécessaire de vérifier que l'intersection $\mathbf{S}_m \cap \mathbf{E}_0$ existe. Notons qu'il n'existe pas de formule explicite pour ce test. Cependant, en utilisant la propriété du théorème 1.1, cette intersection est non vide si :

$$\max_{\alpha} \delta_\alpha < 1 \tag{5.47}$$

où δ_α est calculé à partir de (4.25), les éléments du vecteur α satisfaisant les contraintes de (5.27).

5.4.3 Forme factorisée de l'algorithme OBE-paquet

On construit maintenant la forme factorisée de l'algorithme OBE-paquet afin de garantir sa stabilité numérique. En posant $M_0 = X_0^T X_0$ où X_0 est la factorisation de Cholesky de M_0, on peut réécrire l'équation de l'ellipsoïde englobant (5.27) par :

$$\begin{aligned} f(x) = &\left(\sqrt{\alpha_0} X_0 x - \sqrt{\alpha_0} X_0 c_0\right)^T \left(\sqrt{\alpha_0} X_0 x - \sqrt{\alpha_0} X_0 c_0\right) \\ &+ \sum_{i=1}^{m} \left(\sqrt{\alpha_i} C_i x - \sqrt{\alpha_i} y_i\right)^T \left(\sqrt{\alpha_i} C_i x - \sqrt{\alpha_i} y_i\right) \le 1 \end{aligned}$$

$$\Leftrightarrow \quad f(x) = \left\| \sqrt{\alpha_0} X_0 x - \sqrt{\alpha_0} X_0 c_0 \right\|_2^2 + \sum_{i=1}^{m} \left\| \sqrt{\alpha_i} C_i x - \sqrt{\alpha_i} y_i \right\|_2^2 \le 1$$

$$\Leftrightarrow \quad f(x) = \left\| \begin{bmatrix} \sqrt{\alpha_0} X_0 \\ \sqrt{\alpha_1} C_1 \\ \dots \\ \sqrt{\alpha_m} C_m \end{bmatrix} x - \begin{bmatrix} \sqrt{\alpha_0} X_0 c_0 \\ \sqrt{\alpha_1} y_1 \\ \dots \\ \sqrt{\alpha_m} y_m \end{bmatrix} \right\|_2^2 = \left\| Ax - b \right\|_2^2 \le 1 \tag{5.48}$$

avec $$A = \begin{bmatrix} \sqrt{\alpha_0} X_0 \\ \sqrt{\alpha_1} C_1 \\ \dots \\ \sqrt{\alpha_m} C_m \end{bmatrix} \text{ et } b = \begin{bmatrix} \sqrt{\alpha_0} X_0 c_0 \\ \sqrt{\alpha_1} y_1 \\ \dots \\ \sqrt{\alpha_m} y_m \end{bmatrix} \tag{5.49}$$

On remarque qu'ici encore, $f(x)$ s'écrit sous la forme d'un problème de moindres carrés. En suivant les démarches présentées au chapitre 2, on construit la forme factorisée de l'algorithme OBE-paquet. Il nécessite de construire tout d'abord la matrice $\Pi = \begin{bmatrix} A & b \end{bmatrix}$ et de la « triangulariser » telle que :

$$Q \perp: \quad Q\Pi = Q\begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} U & r \\ 0 & s \end{bmatrix} = \tilde{\Pi} \tag{5.50}$$

où Q est une matrice orthogonale, $Q^T = Q^{-1}$.

Théorème 5.3 : ellipsoïde englobant de l'algorithme OBE-paquet

Les paramètres de l'ellipsoïde englobant l'intersection de E_0 et des m bandes de l'espace paramétrique peuvent être calculés à partir des termes de la matrice $\tilde{\Pi}$ (5.50) :

$$\begin{cases} c = U^{-1} r \\ L = U^T U ; \delta = s^T s \\ M = X^T X \text{ avec } X = \dfrac{U}{\sqrt{1 - s^T s}} \end{cases} \tag{5.51}$$

Démonstration

D'après [STE 73], la solution du problème d'optimisation de type moindres carrés (5.48) est :

$$\hat{x} = \left(A^T A\right)^{-1} A^T b$$

avec A, b donnés par (5.49). On obtient donc :

$$\begin{aligned} \hat{x} &= \left(\alpha_0 X_0^T X_0 + \alpha_1 C_1^T C_1 + \ldots + \alpha_m C_m^T C_m\right)^{-1} \left(\alpha_0 X_0 c_0 + \alpha_1 C_1^T y_1 + \ldots + \alpha_m C_m^T y_m\right) \\ &= L^{-1}\left(\alpha_0 X_0 c_0 + \sum_{i=1}^{m} \alpha_i C_i^T y_i\right) = c \end{aligned}$$

En outre, $\hat{x}$ peut être calculé via une factorisation orthogonale, i.e. $\hat{x} = U^{-1} r$. Il vient :

$$c = \hat{x} = U^{-1} u$$

et on a :

$$\tilde{\Pi}^T \Pi = \begin{bmatrix} U^T U & U^T r \\ = r^T U & r^T r + s^T s \end{bmatrix} \tag{5.52}$$

$$\Pi^T \Pi = \begin{bmatrix} \alpha_0 X_0^T X_0 + \sum_{i=1}^{m} \alpha_i C_i^T C_i & \alpha_0 X_0^T X_0 c_0 + \sum_{i=1}^{m} \alpha_i C_i^T y_i \\ \alpha_0 c_0^T X_0^T X_0 + \sum_{i=1}^{m} \alpha_i C_i y_i & \alpha_0 c_0^T X_0^T X_0 c_0 + \sum_{i=1}^{m} \alpha_i y_i^2 \end{bmatrix} \tag{5.53}$$

Q étant une matrice orthogonale, $\tilde{\Pi}^T\Pi = \Pi^T Q^T Q\Pi = \Pi^T\Pi$. En identifiant (5.52) et (5.53), on obtient :

$$U^T U = \alpha_0 M_0 + \sum_{i=1}^{m} C_i^T C_i = L$$

De plus :

$$\begin{cases} Uc = r \quad \rightarrow \quad \mathrm{r}^T r = c^T U^T U c = c^T L c \\ r^T r + s^T s = \alpha_0 c_0^T M_0 c_0 + \sum_{i=1}^{m} \alpha_i y_i^2 \end{cases}$$

$$\Rightarrow \quad \mathrm{s}^T s = \alpha_0 c_0^T M_0 c_0 + \sum_{i=1}^{m} \alpha_i y_i^2 - c^T L c = \delta_\alpha$$

La matrice M s'exprime maintenant par

$$M = \left(1-\delta_\alpha\right)^{-1} L = \frac{U^T U}{\left(1-s^T s\right)} = X^T X$$

ce qui donne

$$X = U \Big/ \sqrt{1-s^T s}\,. \qquad \square$$

L'utilisation de la forme factorisée de l'algorithme OBE-paquet pour calculer δ_α, puis max(δ_α), nous permet de donner une conclusion numériquement exacte pour le test de cohérence (5.47). Par exemple, lors de la procédure d'identification avec les données simulées du fichier SIMU-ARX-i, et pour $m = 2$, $\gamma = \gamma^*$, à la 27ème observation, on trouve par la formule standard $\delta_{\max}$ = 115.5. Ce qui conduit d'après (5.47) à conclure que l'intersection de l'ellipsoïde précédent avec 2 bandes dans ce paquet est vide. L'intersection effective est non vide (vérification par voie géométrique) et clairement, dans ce cas, l'utilisation de la forme standard de l'algorithme OBE-paquet a conduit à une fausse conclusion. Le calcul par la formule factorisée donne $\delta_{\max}$ = 5.09 10^{-3}, et on retrouve alors la conclusion correcte sur le test de cohérence.

5.4.4 Algorithme OBE-paquet pour un modèle à erreur de sortie

Comme on l'a déjà souligné dans la section 1.3, l'espace admissible des paramètres correspondant à la $i^{\text{ème}}$ observation se restreint à un espace compris entre deux hyperplans non parallèles, i.e. $\mathbf{B}_i = \left[\mathbf{H}_{iM}^-, \mathbf{H}_{im}^+\right]$ avec :

$$\begin{cases} \mathbf{H}_{iM}^{-} : \ \mathrm{C}_{iM}^{T} x = y_i - 1 \\ \mathbf{H}_{im}^{+} : \ \mathrm{C}_{im}^{T} x = y_i + 1 \end{cases} \tag{5.54}$$

où C_{im}, C_{iM} sont calculés à partir de C_i en tenant compte des bornes sur l'erreur de sortie données dans (1.60).

On sait que pour pouvoir utiliser l'algorithme OBE standard, on doit se ramener pour la bande $\mathbf{B}_i$ à une forme quadratique dans le sens où les deux hyperplans « frontières » sont parallèles. Cette démarche a été présentée dans la section 1.3.4 avec l'algorithme FHCL et dans la section 1.3.5 avec l'algorithme de Ferreres de M'Saad. Cependant, dans ces deux cas, la bande ainsi obtenue (et qui remplace la bande originale) a été élargie. L'ellipsoïde englobant est alors forcément pessimiste puisque de dimension supérieure à celui qu'on aurait obtenu en considérant la « bande » originale.

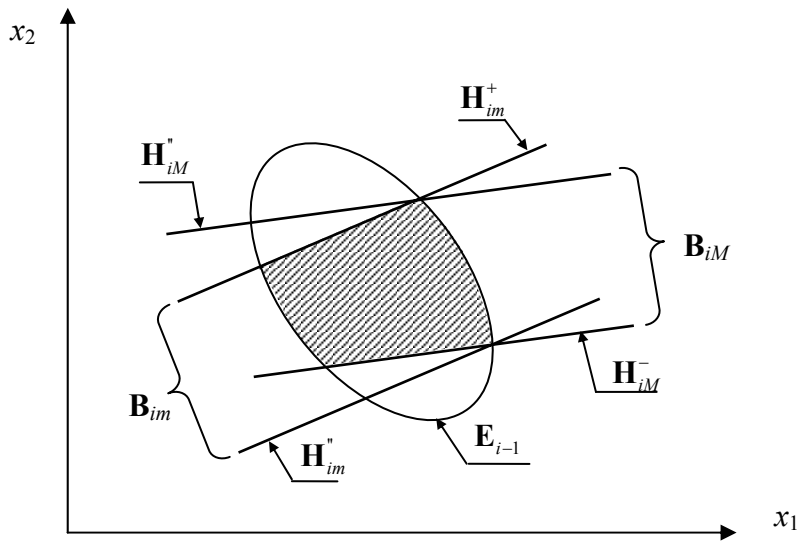


Figure 5.5 : Espace des paramètres pour un modèle à erreur de sortie.

Supposons que les deux hyperplans $\mathbf{H}_{iM}^{-}$, $\mathbf{H}_{im}^{+}$ coupent l'ellipsoïde $\mathbf{E}_{i\text{-}1}$. En utilisant les théorèmes 1.4 et 1.5 du chapitre 1, on peut toujours déterminer les hyperplans $\mathbf{H}_{iM}^{''}$, $\mathbf{H}_{im}^{''}$ (cf. figure 5.5). Ainsi, on peut remplacer la bande $\mathbf{B}_i$ définie dans l'espace paramétrique par les deux hyperplans non parallèles par une paire de 2 bandes $\mathbf{B}_{im}$, $\mathbf{B}_{iM}$ données sous la forme quadratique standard avec :

$$\begin{cases} \mathbf{B}_{im} = \left\{ x \in \mathbb{R}^p : f_m^{''} \le C_{im} x \le y_i + 1 \right\} \\ \mathbf{B}_{iM} = \left\{ x \in \mathbb{R}^p : y_i - 1 \le C_{iM} x \le f_M^{''} \right\} \end{cases} \tag{5.55}$$

où $f_m^{''}$, $f_M^{''}$ sont calculés par (1.72) et (1.78) respectivement. Ces deux nouvelles bandes peuvent être réécrites sous la forme normalisée :

$$\begin{cases} \mathbf{B}_{im} = \left\{ x \in \mathbb{R}^p : y_{im}^{'} - 1 \le C_{im}^{'} x \le y_{im}^{'} + 1 \right\} \\ \mathbf{B}_{iM} = \left\{ x \in \mathbb{R}^p : y_M^{'} - 1 \le C_{iM}^{'} x \le y_{iM}^{'} + 1 \right\} \end{cases} \tag{5.56}$$

où

$$\begin{aligned} C_{im}^{'} &= \frac{2C_{im}}{(y_i + 1) - f_m^{''}}; \quad y_{im}^{'} = \frac{(y_i + 1) + f_m^{''}}{(y_i + 1) - f_m^{''}} \\ C_{iM}^{'} &= \frac{2C_{iM}}{f_M^{''} - (y_i - 1)}; \quad y_{iM}^{'} = \frac{f_M^{''} + (y_i - 1)}{f_M^{''} - (y_i - 1)} \end{aligned} \tag{5.57}$$

A partir de la paire de bandes ainsi obtenues, on recherche l'ellipsoïde englobant l'intersection de l'ellipsoïde initial avec ces deux bandes **en même temps**. Cette étape est facilement mise en œuvre grâce à l'algorithme OBE-paquet où $m = 2$ bandes. Théoriquement, l'ellipsoïde ainsi obtenu sera « meilleur » (au sens où sa dimension sera plus petite) que celui obtenu via les démarches indirectes des algorithmes FHCL ou de Ferreres de M'Saad. Dans le cas où l'un des deux hyperplans de $\mathbf{B}_i$ ne coupe pas $\mathbf{E}_{i\text{-}1}$, on le remplace par l'hyperplan tangent à $\mathbf{E}_{i\text{-}1}$ tel que la nouvelle bande $\mathbf{B}_i^{'}$ contienne l'espace admissible des paramètres. On retrouve alors la situation de l'algorithme OBE standard.

On peut donc appliquer l'algorithme OBE-paquet pour traiter un paquet de données dans le cas d'un modèle à erreur de sortie. Dans ce cas, chaque bande $\mathbf{B}_i$ est remplacée par une paire de 2 bandes $\mathbf{B}_{im}$, $\mathbf{B}_{iM}$ si les deux hyperplans $\mathbf{H}_{iM}^{-}$, $\mathbf{H}_{im}^{+}$ coupent $\mathbf{E}_0$, ou par une bande $\mathbf{B}_i^{'}$ si l'un des deux hyperplans $\mathbf{H}_{iM}^{-}$, $\mathbf{H}_{im}^{+}$ est non sécant à $\mathbf{E}_0$. On obtient alors un nouveau paquet contenant au maximum $2m$ bandes. L'ellipsoïde englobant l'intersection entre l'ellipsoïde $\mathbf{E}_0$ et les bandes de ce nouveau paquet est maintenant calculé par l'algorithme OBE-paquet présenté ci-avant.

5.5 Résultats expérimentaux avec l'algorithme OBE-paquet

Dans cette partie, l'algorithme OBE-paquet est appliqué à l'estimation de paramètres d'un modèle pour les données simulées (décrites au chapitre 2) et les données réelles de l'IFP et du robot parallèle. A chaque appel de cet algorithme, on utilise la fonction d'optimisation

« fmincon » dans l'environnement Matlab afin de déterminer la valeur de $\delta_{\max}$ et le vecteur α^*. Les paramètres intrinsèques de la fonction « fmincon » sont pris selon les valeurs assignées par défaut dans Matlab. On présente maintenant les résultats obtenus pour l'identification en utilisant le critère du logarithme de déterminant.

5.5.1 Résultat sur les données simulées

Le fichier de données SIMU-ARX-i est exploité lors de la procédure d'identification ensembliste ellipsoïdale. Le tableau 5.1 compare les volumes au carré de l'ellipsoïde obtenu et le coût calcul (ordinateur PC Pentium IV à 2.4Ghz) en utilisant l'algorithme OBE (m = 1) et l'algorithme OBE-paquet pour m variant de 2 à 20. C'est le modèle à erreur d'équation qui est utilisé ici.

On utilise le processus de recirculation des données (20 circulations) afin de comparer les résultats obtenus avec les approches OBE standard et OBE-paquet lorsque cette « réutilisation » des données est mise en œuvre. Il faut noter qu'après 20 circulations, l'ellipsoïde semble avoir convergé puisque sa taille n'évolue plus. La borne utilisée correspond à la borne réelle du bruit introduit dans la simulation ($\gamma = \gamma^* = 0.003$).

Algorithme	***m***	**1 recirculation**		**20 recirculations**	
		det$P$	Δt(s)	det$P$	Δt(s)
OBE	1	$1.85\ 10^{-14}$	2	$2.43\ 10^{-16}$	17
OBE paquet	2	$2.02\ 10^{-14}$	12	$2.43\ 10^{-16}$	50
	4	$1.39\ 10^{-14}$	15	$2.43\ 10^{-16}$	65
	10	$4.07\ 10^{-15}$	25	$2.38\ 10^{-16}$	187
	20	$3.89\ 10^{-15}$	121	$2.38\ 10^{-16}$	1347

Tableau 5.1 : Résultat de l'identification avec les algorithmes OBE et OBE-paquet ($\gamma = \gamma^*$).

Au vu des résultats du tableau 5.1, quelques remarques s'imposent :

- après la 1ère recirculation, la taille de l'ellipsoïde obtenu réduit fortement lorsque la longueur du paquet m augmente. Ceci vérifie l'objectif visé pour la construction de cet algorithme. Cependant, après 20 recirculations du fichier de données, les ellipsoïdes trouvés par les algorithmes OBE standard et OBE-paquet ont une dimension comparable. Cela signifie que lorsque la borne retenue pour l'identification est « juste » (i.e. ni

surestimée, ni sous-estimée), l'algorithme OBE-paquet ne présente pas d'avantage par rapport à l'algorithme OBE standard ;

- le coût calcul augmente rapidement en fonction de m, ce qui était prévisible puisque l'on se trouve en présence d'un phénomène d'explosion combinatoire. Avec m « grand », les résultats de l'identification ne sont pas significativement améliorés (det(P) est de l'ordre de 10^{-15} pour m = 10 et m = 20) mais le coût calcul est fortement augmenté. Il faut donc choisir une valeur m « raisonnable » au sens où le coût calcul reste faible tout en assurant une diminution significative de la taille de l'ellipsoïde (m variant de 4 à 10). Notons que le choix de m dépend avant tout du nombre p de paramètres. En outre, le prétraitement des données permet de réduire significativement le temps calcul ;
- le vecteur α^* déterminé par la routine d'optimisation peut ne pas avoir atteint l'optimum théorique pour des raisons de complexité de l'hypersurface. Cette situation a été rencontrée avec m = 2 pour laquelle on constate que le volume de l'ellipsoïde résultat est supérieur à celui obtenu par l'algorithme OBE standard.

Algorithme	***m***	**1 recirculation**		**20 recirculations**	
		det(P)	Δt(s)	det(P)	Δt(s)
OBE	1	$1.09\ 10^{-9}$	1.1	$7.52\ 10^{-10}$	20
OBE paquet	2	$1.16\ 10^{-9}$	34	$7.53\ 10^{-10}$	590
	4	$5.48\ 10^{-10}$	23	$3.29\ 10^{-10}$	321
	10	$4.00\ 10^{-10}$	33	$2.34\ 10^{-10}$	526
	20	$3.48\ 10^{-10}$	124	$2.32\ 10^{-10}$	2409

Tableau 5.2 : Résultat de l'identification par les algorithmes OBE et OBE-paquet avec une borne surestimée ($\gamma = 2\gamma^*$).

Le tableau 5.2 présente les résultats de l'identification sur ce même fichier de données simulées avec une borne surestimée $\gamma = 2\gamma^*$. On peut voir que les ellipsoïdes obtenus par l'algorithme OBE-paquet ont été améliorés lorsque m augmente. Dans le cas d'une borne surestimée, l'algorithme OBE standard arrête rapidement la mise à jour de l'ellipsoïde tandis que l'algorithme OBE-paquet a systématiquement utilisé toutes les données du paquet pour réduire la taille de l'ellipsoïde, compte tenu des techniques de réduction de bandes présentées ci-avant. Le tableau 5.2 montre qu'avec m = 2, l'algorithme OBE-paquet n'apporte pas une amélioration significative à de la taille de l'ellipsoïde tout en induisant un coût calcul

fortement augmenté par rapport à l'algorithme OBE standard. La raison est que peu de bandes réputées « inutiles » sont détectées et éliminées par les procédures de prétraitement des données. Autrement dit, les procédures de prétraitement des données ne permettent pas d'améliorer le temps calcul lorsque la valeur de m est inappropriée.

En pratique, on constate que la fonction critère basée sur la trace de la matrice P est souvent plus « plate » que celle du déterminant (ou plus exactement du logarithme du déterminant). Il en résulte que la recherche de α^* à l'aide d'une routine d'optimisation (« fmincon » dans notre cas) pour le critère de la trace est souvent plus difficile que pour le critère du déterminant, ou son logarithme. De ce fait, l'algorithme OBE-paquet donne de « meilleurs » résultats (au sens où la taille de l'ellipsoïde obtenu est moindre) lorsque le critère du déterminant est exploité.

5.5.2 Résultats pour les données de l'IFP

On sait que pour des données réelles/industrielles, la borne utilisée pour la procédure d'identification ensembliste ellipsoïdale est souvent surestimée. Ainsi, en utilisant la combinaison de plusieurs données (et donc de plusieurs bandes), l'algorithme OBE-paquet est susceptible de fournir de meilleurs résultats que l'algorithme OBE standard.

On applique d'abord l'algorithme OBE-paquet aux données de l'IFP avec un modèle à erreur d'équation. La borne choisie pour l'identification est la valeur maximale de l'erreur estimée, comme on l'a déjà fait au chapitre 2, soit $\gamma = 0.0022$. L'évolution du volume de l'ellipsoïde englobant est présenté dans la figure 5.6. On constate que l'algorithme OBE-paquet permet de réduire la taille de l'ellipsoïde qui converge vers une valeur optimale. Avec $4 \leq m \leq 10$, la taille de l'ellipsoïde est fortement diminuée sans pour autant augmenter trop fortement le temps calcul.

L'application de l'algorithme OBE-paquet avec un modèle à erreur de sortie est présentée dans les figures 5.7 et 5.8. On constate que dès la 2ème recirculation, le volume de l'ellipsoïde trouvé par l'algorithme OBE-paquet a approximativement convergé. En outre, le temps calcul pour deux recirculations reste raisonnable. Il en résulte qu'un nombre restreint de recirculations des données permet d'améliorer le résultat sans augmenter de manière inconsidérée le coût calcul. L'ellipsoïde trouvé avec $m = 8$, présenté dans la figure 5.8, est visiblement amélioré. La projection de cet ellipsoïde sur le repère paramétrique conduit à des intervalles consistants avec les connaissances physiques sur le procédé, contrairement à ce que l'on obtient avec les algorithmes FHCL ou FM (pour la même borne).

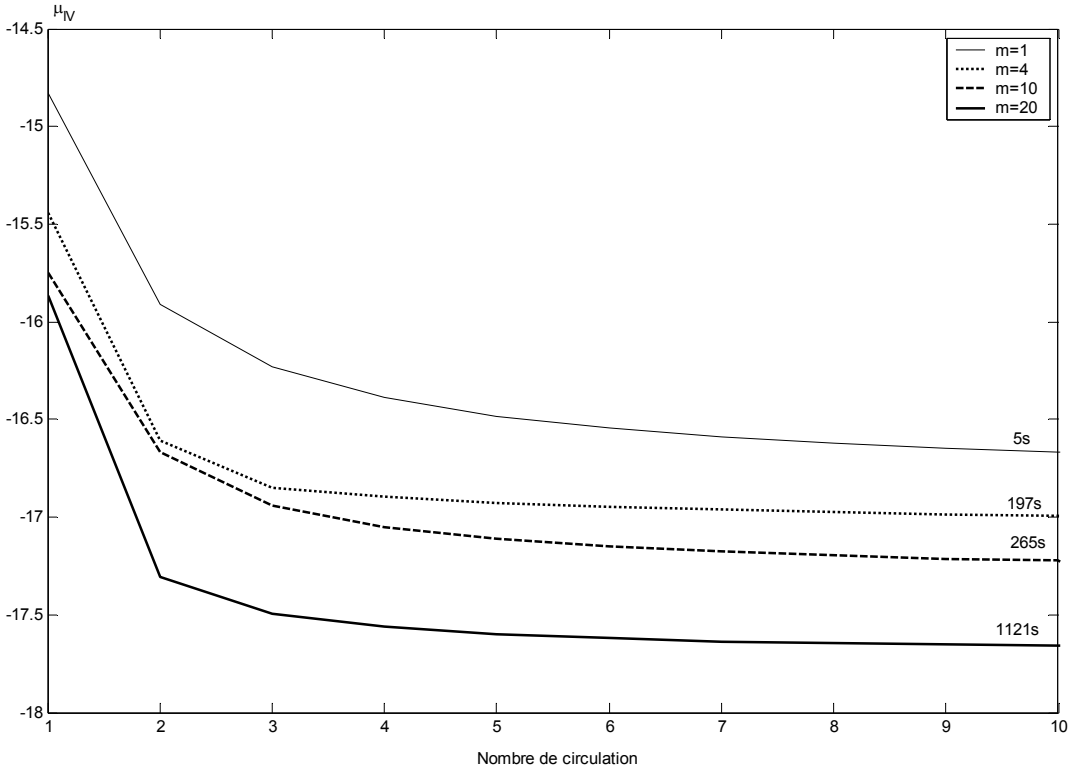


Figure 5.6 : Evolution du volume de l'ellipsoïde (modèle à erreur d'équation).

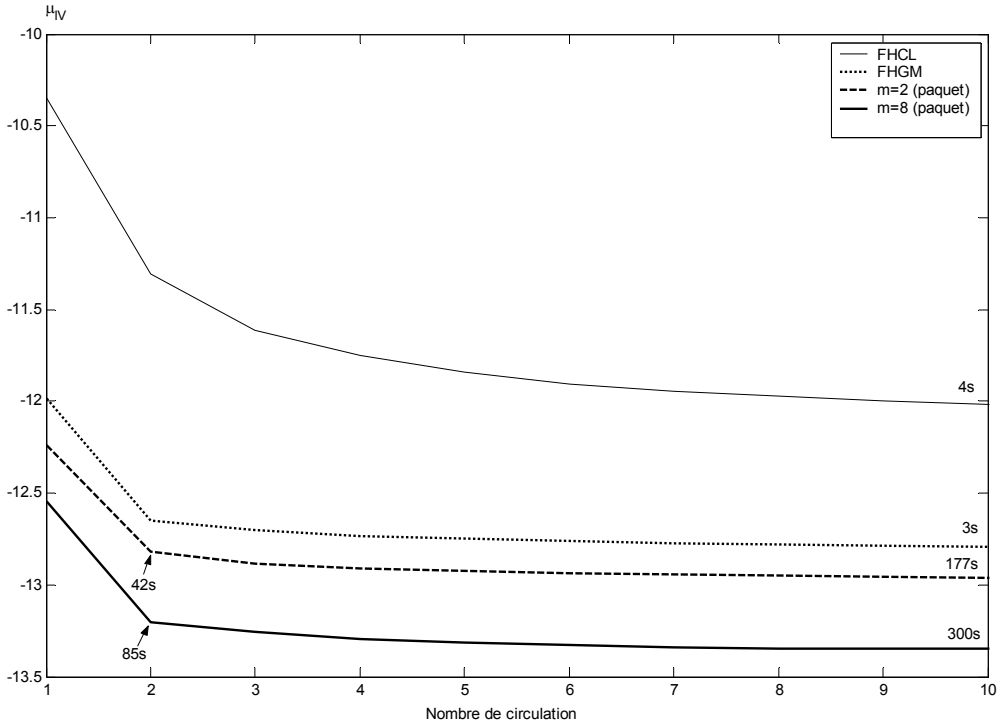


Figure 5.7 : Evolution du volume de l'ellipsoïde (modèle « erreur de sortie »).

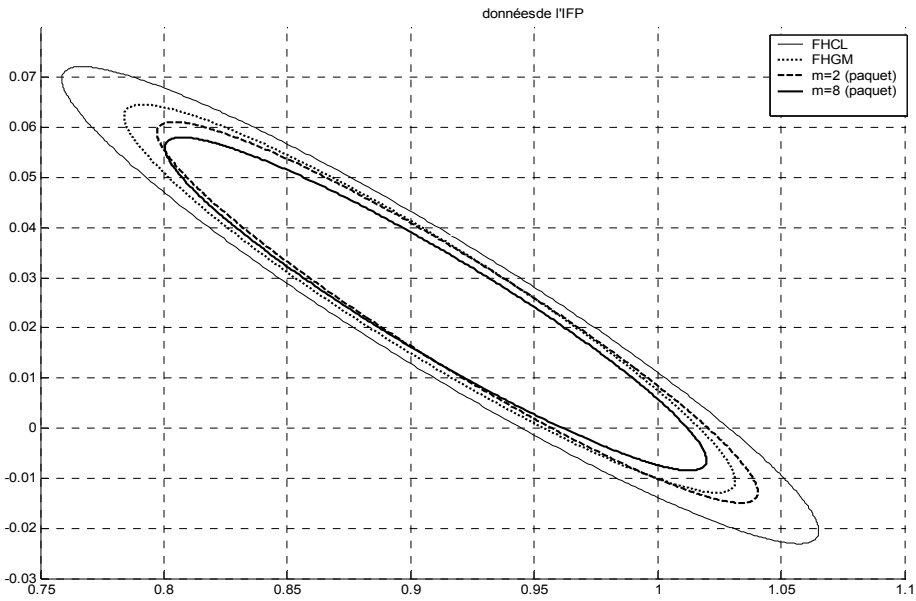


Figure 5.8 : Ellipsoïdes obtenus par les algorithme OBE et OBE-paquet.

5.5.3 Résultats pour les données du robot parallèle

On applique maintenant l'algorithme OBE-paquet à l'estimation des paramètres du robot parallèle du LIRMM. La borne utilisée en identification est choisie de manière heuristique comme décrit dans le chapitre 2, soit $\gamma = 6\text{Nm}$. L'évolution du volume de l'ellipsoïde obtenu est présentée dans la figure 5.9.

Le nombre de paramètres à estimer peut être considéré comme étant « grand » (p = 14) pour une approche ensembliste. La procédure de prétraitement des données telle qu'introduite dans ce chapitre ne peut être utilisée car on est confronté à une explosion combinatoire dans la formule (5.44), ce qui induit deux conséquences :

- le coût calcul est prohibitif, par exemple après 70 recirculations, lorsque la taille de l'ellipsoïde a convergé, on a $t_\text{CPU}_{\text{OBE}} = 9\text{mn}$, $t_\text{CPU}_{\text{OBEpaquet}(m=4)} = 7\text{h}25\text{mn}$ et $t_\text{CPU}_{\text{OBEpaquet}(m=14)} = 4\text{h}20\text{mn}$ pour un ordinateur PC Pentium IV à 2.4Ghz ;
- le vecteur α^* trouvé à l'aide de la routine d'optimisation n'a pas atteint l'optimum théorique.

La figure 5.9 montre que le volume de l'ellipsoïde après 70 recirculations a convergé. L'utilisation d'un paquet de données (m = 4 ou m = 14) permet de le réduire mais de manière peu significative par rapport au cas m = 1 (algorithme OBE standard). Il semble donc que pour un modèle avec « beaucoup » de paramètres, l'utilisation conjointe de l'algorithme OBE-paquet et du processus de recirculation, associée à un choix cohérent de la borne

n'apporte pas d'amélioration flagrante de la taille de l'ellipsoïde tout en induisant un surcoût calcul très important.

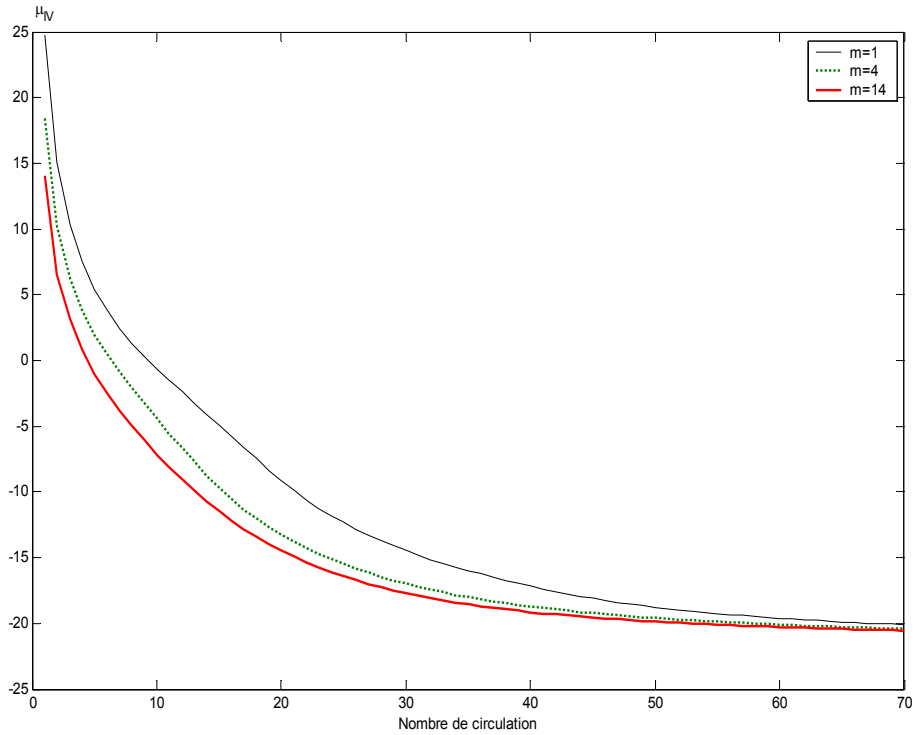


Figure 5.9 : Evolution du volume de l'ellipsoïde (robot parallèle).

5.6 Conclusion

Dans ce chapitre, l'algorithme OBE-paquet permettant de calculer un ellipsoïde englobant (compte tenu d'un critère fixé) l'intersection de l'ellipsoïde initial avec plusieurs bandes simultanément a été présenté. OBE-paquet combine un traitement global de chaque paquet et un traitement séquentiel des paquets du fichier de données. A chaque appel de l'algorithme, on doit utiliser une fonction d'optimisation avec contraintes (par exemple « fmincon » dans l'environnement Matlab) pour rechercher le vecteur optimum α^* des coefficients α_i.

Deux phases de prétraitement des données sont indispensables pour éliminer les bandes réputées inutiles pour le calcul de l'ellipsoïde englobant et pour réajuster les bandes si besoin est. L'algorithme proposé est donné sous sa forme factorisée pour garantir sa stabilité numérique. Le développement de cet algorithme pour un modèle à erreur de sortie est également présenté.

Les résultats obtenus pour l'identification de paramètres à partir de données simulées et réelles montrent que, pour des conditions opératoires identiques (même nombre de

recirculations, même borne), la taille de l'ellipsoïde obtenu via l'algorithme OBE-paquet est plus faible que celle obtenue via l'algorithme OBE standard. Néanmoins, le coût calcul d'OBE-paquet est supérieur à celui de l'algorithme OBE standard puisque l'on est confronté à un problème d'explosion combinatoire. Notons que le choix de la longueur m du paquet a une influence importante sur la taille de l'ellipsoïde et sur le temps calcul. Pour les problèmes abordés dans ce travail, cet algorithme s'est avéré être particulièrement intéressant pour de petites dimensions ($p \leq 5$) puisqu'il permet alors de diminuer fortement le nombre de recirculation nécessaire pour que la taille de l'ellipsoïde converge sans trop augmenter le coût calcul dû aux phénomènes combinatoires induits par la technique mise en œuvre.

Conclusion et perspectives

Notre travail a été dédié à l'identification des systèmes dynamiques linéaires, ou linéaires en les paramètres, par l'approche ensembliste ellipsoïdale. A la différence des méthodes « traditionnelles » qui requièrent des hypothèses statistiques sur les erreurs attachées à la sortie du système, l'approche ensembliste nécessite seulement la connaissance d'une borne de ces erreurs. En conséquence, le résultat de l'identification est non plus un vecteur ponctuel de paramètres mais un ensemble englobant de manière sûre l'espace admissible des paramètres (c'est-à-dire compatible avec l'incertitude sur les mesures), ici un ellipsoïde. L'intervalle de paramètres est obtenu, si besoin, par la projection de l'ellipsoïde sur les axes paramétriques.

Dans le premier chapitre de cette thèse, nous avons présenté la formulation des algorithmes OBE et les avons classés en deux groupes. Le premier comprend des algorithmes qui cherchent à réduire la taille de l'ellipsoïde englobant. Cette taille est évaluée via le produit de la longueur au carré des demi axes de l'ellipsoïde (critère du déterminant) ou leur somme (critère de la trace). On a prouvé que les ellipsoïdes obtenus par les différents algorithmes du groupe 1 sont identiques (avec le même critère utilisé). Les algorithmes du groupe 2, en revanche, se basent sur la convergence d'un facteur d'échelle, noté usuellement σ_k^2.

Le développement de ces techniques a été systématiquement adapté au modèle à erreur de sortie. La bande d'espace des paramètres s'en trouve élargie et délimitée par deux hyperplans non parallèles. On a comparé deux algorithmes pour ce contexte : l'algorithme de Clément (FHCl) qui construit l'ellipsoïde en deux étapes, et l'algorithme de Ferreres et M'Saad (FM) qui détermine l'ellipsoïde englobant par une seule coupe.

Dans tous les cas, le test de cohérence (existence d'une intersection non vide) et la réduction si besoin de la bande avant tout appel à un algorithme de type OBE s'averre indispensable pour mieux prendre en compte les informations contenues dans les mesures.

L'évaluation et la comparaison des algorithmes OBE ont été réalisées dans le chapitre 2. Celles-ci ont d'abord été faites sur données simulées. On a utilisé un modèle du 1er ordre avec un bruit uniforme ou un bruit gaussien ajouté à la sortie. Les résultats obtenus montrent clairement les propriétés de chaque algorithme. On a ensuite appliqué ces techniques à des données réelles fournies par l'IFP, qui conduisent à un ellipsoïde allongé quel que soit le

critère utilisé (trace ou déterminant) en raison de la présence d'une entrée au spectre très pauvre. Puis on a traité des données issues d'un robot parallèle de LIRMM, pour lequel on constate visiblement des données aberrantes dans les régresseurs. On a montré que la borne utilisée en identification tenant compte de ces données aberrantes conduit à un ellipsoïde surdimensionné. En analysant les propriétés des données, nous avons proposé de traiter les données du robot en 4 blocs. L'identification par bloc a montré plusieurs avantages comme par exemple, une réduction du temps de calcul, un libre choix de la borne pour chaque bloc de données, etc.

A la différence de la quasi-totalité des publications faites dans ce domaine, nous avons présenté au chapitre 3 une implantation numériquement stable des algorithmes OBE. Au préalable, les points faibles des algorithmes originaux lors des applications numériques ont été analysés. Le risque d'instabilité numérique provient du calcul des termes quadratiques qui peuvent devenir négatif si l'on exploite directement les formules données dans l'algorithme OBE. En reformulant l'algorithme OBE sous la forme d'un problème d'optimisation quadratique, on a établi une forme factorisée en exploitant la technique de factorisation orthogonale du type QR. On manipule alors une factorisation de Cholesky de la matrice M, i.e. $M = X^T X$ où X est une matrice triangulaire. Cette démarche s'applique également à la formulation avec la matrice P, i.e. $P = \overline{X}^T \overline{X}$. Les algorithmes OBE factorisés exhibent deux avantages : les propriétés théoriques de certains coefficients et paramètres sont satisfaites ; les calculs des différents paramètres deviennent (dans certains cas) indépendants.

Le chapitre 4 a abordé les problèmes de convergence des algorithmes OBE. Avec le critère du déterminant, l'algorithme SMSA (et tous les algorithmes du groupe 1 associés) présente deux propriétés : la convergence du centre de l'ellipsoïde vers le voisinage du « vrai » vecteur de paramètres et la convergence du volume de l'ellipsoïde vers un point. Pour les algorithmes du groupe 2 on ne peut démontrer que la convergence du centre de l'ellipsoïde. Trois conditions théoriques permettent d'assurer la convergence des algorithmes OBE : le vecteur de vrais paramètres x^* est contenu dans l'ellipsoïde initial ; la persistance de l'excitation et la visite régulière des erreurs sur la borne γ utilisée lors de l'identification.

A partir d'une interprétation géométrique, on a montré que l'angle θ_{dd_+} entre deux bandes d'espace successives est un facteur fondamental influant sur la vitesse de convergence. Une procédure permettant de créer une entrée qui rende l'angle entre deux bandes d'espace

successives quasiment égal à $\pi/2$ a été proposée. En fait, l'objectif souhaité est que la bande parcourre toutes les directions de l'espace de paramètres.

Les difficultés associées au choix de la borne ont été analysées, en particulier avec des données réelles qui contiennent parfois des mesures aberrantes. L'algorithme OBE-ABE qui permet de réduire automatiquement la borne durant le processus d'identification, a été présenté. Néanmoins, l'application de cet algorithme à des données réelles semble instable et dépend de quelques paramètres délicats à régler. Nous avons proposé une méthode de choix de la borne basée sur une analyse du Kurtosis de la séquence d'erreurs (réelle ou plus souvent estimée). Dans cette méthode, les données contenant de « grandes » erreurs (incluant les mesures aberrantes) sont éliminées afin d'obtenir une séquence d'erreur tronquée ayant une valeur du Kurtosis préalablement choisie. Cette démarche à base de Kurtosis semble très constructive, particulièrement avec des données réelles, car elle ne nécessite pas un réglage heuristique et/ou manuel, souvent non scientifiquement justifié, de la borne.

Le problème de recherche de l'ellipsoïde optimal global a été abordé dans le chapitre 5. Exprimé comme un problème d'optimisation, l'ellipsoïde obtenu par les algorithmes OBE séquentiels est en fait une solution sous-optimale du problème initial. Rechercher un ellipsoïde au sens de l'optimisation globale est très difficile et pratiquement irréalisable sur le plan calcul. Pour concilier ces deux situations antagonistes, on a proposé l'algorithme OBE « paquet » qui combine une optimisation globale dans chaque paquet avec un traitement séquentiel des paquets du fichier de données. D'un point de vue géométrique, dans chaque paquet de m bandes, on cherche l'ellipsoïde qui englobe l'intersection de l'ellipsoïde initial et des m bandes d'espace simultanément. L'ellipsoïde trouvé lors du traitement du paquet précédent devient l'ellipsoïde initial pour le paquet suivant. Des procédures de prétraitement des données ont été introduites pour diminuer le pessimisme de l'optimum obtenu et réduire le temps calcul. Des résultats sur données simulées et données réelles ont montré les avantages théoriques de cet algorithme. Cependant, le temps de calcul reste élevé et induit un choix approprié de la longueur des paquets pour réaliser le meilleur compromis précision/temps.

Le domaine exploré est très vaste et il va de soi que ces techniques sont encore perfectibles. Parmi les questions qui pourraient être abordées en voici quelques unes qui nous ont paru significatives :

- l'utilisation de la factorisation LDL^T, qui ne nécessite pas de racines carrées, pourrait être exploitée pour des implémentations sur processeurs dédiés ;
- pour le problème du choix de la borne, dans la méthode basée sur une analyse du Kurtosis, on a rejeté les données contenant de « grandes » erreurs. On pourrait analyser ces données éliminées afin de vérifier *a posteriori* qu'elles n'apportaient pas d'information. Cette action aurait une incidence qu'il faudrait analyser sur le choix de la borne ;
- dans l'algorithme OBE « paquet », le temps calcul peut être très élevé (appel à une routine d'optimisation non linéaire avec contraintes). De plus, cette fonction est appelée deux fois dans chaque traitement d'un paquet de données : une fois pour le test de cohérence et l'autre pour la recherche du vecteur optimal α^*. On pourrait envisager une autre approche pour le test de cohérence. Une manière possible est un changement de coordonnées telles que l'ellipsoïde initial $\mathbf{E}_0$ devienne une sphère unité, centrée en zéro. Cette fois, en calculant la distance de l'origine aux hyperplans délimitant la région $\mathbf{S}_m$ dans le nouveau repère de coordonnées, on pourrait obtenir rapidement la conclusion sur le test de cohérence ;
- Enfin, à partir de l'ellipsoïde obtenu après identification, on pourrait construire l'intervalle de confiance sur la sortie. Cet intervalle pourrait servir à un diagnostic pour détecter les défauts dans le système considéré. On a ébauché dans la dernière annexe quelques pistes sur le thème « diagnostic » qui pourrait faire l'objet d'un sujet de recherche à lui seul.

Annexe A

Le problème de moindres carrés linéaire

A.1 Solution du problème de moindres carrées

La résolution d'un problème de moindres carrés consiste à rechercher un vecteur $\hat{x}$ de dimension p satisfaisant :

$$\hat{x} = \arg\min_{x} \left\{ \varrho^2(x) = \|b - Ax\|_2^2 \right\} \tag{A.1}$$

où $A \in \mathbb{R}^{m \times p}$ est une matrice fixée

$b \in \mathbb{R}^m$ est un vecteur

Posons :

$$r = b - Ax \tag{A.2}$$

le vecteur résidu, linéaire en x.

Lorsque x varie dans $\mathbb{R}^p$, le vecteur $y \triangleq Ax$ varie sur $\Re(A)$ où $\Re(A)$ est un espace engendré par les colonnes de A. Le problème (A.1) se ramène maintenant à la recherche de $y_{\min}$ dans l'espace $\Re(A)$ qui minimise $\|b - y\|_2^2$. La Figure A.1 illustre ce problème avec $m = 3$ et rang(A) = 2. Un point de vue géométrique montre qu'il est évident que $r_{\min}$ est atteint si (b - y) est perpendiculaire au plan $\Re(A)$.

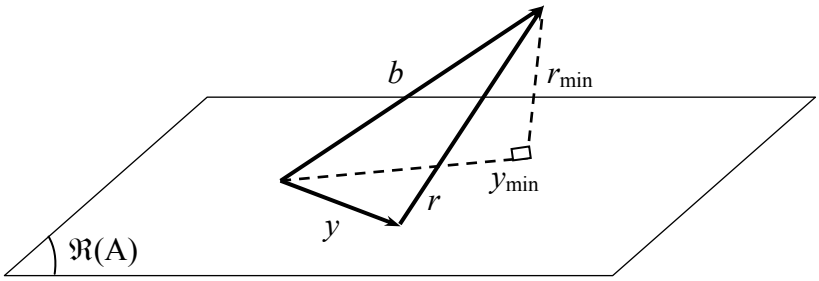


Figure A.1

Si l'on écrit :

$$b = u + v \tag{A.3}$$

où $u \in \Re(A)$, v étant dans le complément orthogonal de $\Re$(A), la fonction $\varrho^2(x)$ dans (A.1) devient :

$$\varrho^2(x) = \|r\|_2^2 = \|b - Ax\|_2^2 = \|u - Ax\|_2^2 + \|v\|_2^2 \tag{A.4}$$

On peut voir que $\varrho^2(x)$ sera minimum si $\|u - Ax\|_2^2$ est minimum. Comme $u \in \Re(A)$, la solution $\hat{x}$ pour le problème (A.1) satisfait :

$$A\hat{x} = u \tag{A.5}$$

On sait aussi que le vecteur résidu *r* vérifie :

$$A^T r = 0 \tag{A.6}$$

$$\Leftrightarrow \quad A^T(b - Ax) = 0 \quad \Leftrightarrow \quad A^T Ax = A^T b$$

Si $A^T A$ est non singulière, la solution (d'un point de vue théorique, on n'utilisera pas cette formulation pour la résolution numérique !) du problème (A.1) est alors donnée par :

$$\hat{x} = \left(A^T A\right)^{-1} A^T b \tag{A.7}$$

A.2 Triangularisation orthogonale pour le problème de moindres carrées

En utilisant la technique de factorisation QR, on peut toujours triangulariser la matrice *A* dans (A.1) via une matrice *Q* orthogonale ($Q^T = Q^{-1}$) telle que :

$$QA = \begin{pmatrix} U \\ 0 \end{pmatrix} \tag{A.8}$$

où $U \in \mathbb{R}^{p \times p}$ est une matrice carrée, triangulaire supérieure. Si l'on pose :

$$Qb = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{A.9}$$

en multipliant le vecteur résidu *r* par *Q*, on obtient :

$$Qr = Q\left(b - Ax\right) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} U \\ 0 \end{pmatrix} x = \begin{pmatrix} b_1 - Ux \\ b_2 \end{pmatrix} \tag{A.10}$$

Q étant une matrice orthogonale, on a :

$$\|r\|_2^2 = \|Qr\|_2^2 = \|b_1 - Ux\|_2^2 + \|b_2\|_2^2 \tag{A.11}$$

Il est clair que $\varrho^2(x) = \|r\|_2^2$ est maintenant minimisé si le vecteur $\hat{x}$ vérifie :

$$b_1 - U\hat{x} = 0 \tag{A.12}$$

De plus : $\min \varrho^2(x) = \|r_{\min}\|_2^2 = \|b_2\|_2^2$ (A.13)

Alors
$$\begin{cases} \hat{x} = U^{-1} b_1 \\ r_{\min} = Q^T \begin{pmatrix} 0 \\ b_2 \end{pmatrix} \end{cases} \tag{A.14}$$

avec U (donc A) de rang plein, ce qui garantit l'existence de son inverse.

En résumé, on peut résoudre le problème de moindres carrées (A.1) en utilisant une factorisation QR et la démarche suivante :

1. Construire la matrice $\Pi = \begin{bmatrix} A & b \end{bmatrix}$
2. Triangulariser Π par en utilisant une factorisation QR :

$$Q\Pi = \begin{bmatrix} U & b_1 \\ 0 & b_2 \end{bmatrix}$$

3. En déduire $\hat{x} = U^{-1} b_1$
4. Calculer $r_{\min} = Q^T \begin{pmatrix} 0 \\ b_2 \end{pmatrix}$, si besoin
5. $\|r_{\min}\|_2^2 = b_2^2$

Annexe B

La méthode du Simplex

L'algorithme du Simplex est destiné à la résolution du problème de programmation linéaire suivant :

$$\textit{maximiser} \quad u = c_1x_1 + c_2x_2 + \ldots + c_px_p, \quad x \in \mathbb{R}^p \tag{B.1}$$

où u est la fonction objectif, avec m contraintes inégalités :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p \le b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mp}x_p \le b_p \end{cases} \tag{B.2}$$

et p contraintes de non négativité sur x_1 à x_p, i.e. :

$$x_i \ge 0, \ \forall i = 1 : p \tag{B.3}$$

D'un point de vue géométrique, l'ensemble des points x vérifiant toutes les contraintes (B.2) définit dans $\mathbb{R}^p$ un polyèdre $\mathbf{S}_m$ dont chaque équation de (B.2) est une face.

On peut transformer les m inégalités de (B.2) en m équations par introduction de variables supplémentaires. Par exemple, pour la première inéquation de (B.2), on peut réécrire :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p - b_1 = -t_1 \\ t_1 \ge 0 \end{cases} \tag{B.4}$$

Ainsi, en utilisant les variables supplémentaires, le problème de programmation linéaire peut être maintenant représenté sous la forme :

$$\textit{maximiser} \quad u = c_1x_1 + c_2x_2 + \ldots + c_px_p, \quad x \in \mathbb{R}^p \tag{B.1}$$

sous les m contraintes égalité :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p - b_1 = -t_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p - b_2 = -t_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mp}x_p - b_p = -t_p \end{cases} \tag{B.5}$$

et $m + p$ contraintes de non négativité sur x et t :

$$\begin{cases} x_i \geq 0, \ \forall i = 1:p \\ t_j \geq 0, \ \forall j = 1:m \end{cases} \tag{B.6}$$

Posons :

$$\mathcal{X} = \left[x_1, x_2, ..., x_p, t_1, t_2, ..., t_m \right]^T, \ \mathcal{X} \in \mathbb{R}^{m+p} \tag{B.7}$$

Le vecteur $\mathcal{X}$ est également appelé « point ». L'ensemble des vecteurs (ou points) $\mathcal{X}$ satisfaisant toutes les équations (B.5) et inégalités (B.6) constitue un ensemble de contraintes $\mathbf{S}_c$.

Un *point de base* est une solution du système d'équations (B.5). Cette solution est obtenue en mettant p variables dans $\mathcal{X}$ (B.7) à zéro et en résolvant pour les m variables restantes. Notons que l'on peut avoir au maximum $C_m^p = \dfrac{m!}{p!(m-p)!}$ points de base.

Un *point de base admissible*, noté *bfp* (basic feasible point) est un point de base dans lequel toutes les m variables restantes sont supérieures ou égales à zéros. Les points *bfp* correspondent aux sommets du polyèdre $\mathbf{S}_m$ des contraintes. La méthode du simplex consiste à observer l'évolution de la fonction objectif en parcourant les arrêtes du polyèdre d'un sommet à l'autre, jusqu'au maximum. On montre que s'il existe une solution bornée au problème (B.1), elle correspond à l'un des sommets du polyèdre (sauf cas de dégénérescence).

Comme l'algorithme du Simplex [ROT 79] est construit de façon à ne jamais explorer deux fois le même sommet, on aboutit à la solution en un nombre fini d'itérations.

Remarques

- Si le problème de programmation linéaire considéré est la minimisation d'une fonction objectif, i.e.

$$minimiser \ \overline{u} = d_1 x_1 + d_2 x_2 + ... + d_p x_p, \ \ x \in \mathbb{R}^p \tag{B.8}$$

  on peut se ramener à la formulation standard (B.1) par :

$$maximiser \ u = -\overline{u} = -d_1 x_1 - d_2 x_2 - ... - d_p x_p, \ \ x \in \mathbb{R}^p \tag{B.9}$$

- Si une ou plusieurs inéquations de (B.2) changent de signe, par exemple :

$$a_{11} x_1 + a_{12} x_2 + ... + a_{1p} x_p \geq b_1 \tag{B.10}$$

  pour pouvoir écrire des contraintes de non négativité sur x et t comme dans (B.6), on introduit cette fois la variable supplémentaire suivante :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p - b_1 = t_1 \\ t_1 \geq 0 \end{cases} \tag{B.11}$$

- Dans le cas général, si l'on a des contraintes décrites par :

$$b_{\min} \leq Ax \leq b_{\max} \tag{B.12}$$

où $A \in \mathbb{R}^{mxp}$; $b_{\min}, b_{\max} \in \mathbb{R}^m$

on peut se ramener à la forme standard de (B.2) par :

$$\begin{bmatrix} A \\ -A \end{bmatrix} x \leq \begin{bmatrix} b_{\max} \\ -b_{\min} \end{bmatrix} \tag{B.13}$$

- La condition $x \geq 0$ est une hypothèse standard de programmation linéaire, dont les domaines d'application font intervenir des variables avec une signification physique. Si l'on rencontre en pratique un problème où les variables x_i sont de signe arbitraire, on pourra encore se ramener à la formulation standard en posant :

$x = u - u'$ avec $u, u' \geq 0$.

Annexe C

Démonstration du théorème 1.1 dans le cas de l'intersection d'un ellipsoïde et d'une bande

Rappel : théorème 1.1

La condition nécessaire et suffisante pour que l'intersection $\mathbf{E}_1 \cap \mathbf{E}_2$ *soit non vide est*

$$\max(\delta_\alpha) \leq 1 \tag{C.1}$$

où $\mathbf{E}_1$ et $\mathbf{E}_2$ sont écrits sous forme normalisée.

Dans le cas de l'intersection entre un ellipsoïde et une bande (tous deux définis sous leur forme normalisée), δ_α est calculé par la formule (1.25) :

$$\delta_\alpha = \alpha_k c_{k-1}^T P_{k-1}^{-1} c_{k-1} + (1-\alpha_k) y_k^2 - c_k^T L_\alpha c_k \tag{C.2}$$

avec
$$\begin{cases} L_\alpha = \alpha_k P_{k-1}^{-1} + (1-\alpha_k) d_k d_k^T \\ c_k = L_\alpha^{-1}\left[\alpha_k P_{k-1}^{-1} c_{k-1} + (1-\alpha_k) d_k y_k\right] \end{cases} \tag{C.3}$$

et $\quad 0 < \alpha_k \leq 1$

Remplaçons (C.3) dans (C.2). La fonction δ_α peut se réécrire sous la forme :

$$\delta_\alpha = \alpha_k c_{k-1}^T P_{k-1}^{-1} c_{k-1} + (1-\alpha_k) y_k^2 - w^T L_\alpha^{-1} w \tag{C.4}$$

avec
$$w = \alpha_k P_{k-1}^{-1} c_{k-1} + (1-\alpha_k) d_k y_k \tag{C.5}$$

On cherche maintenant $\max(\delta_\alpha)$. Calculons d'abord $\dfrac{\partial \delta_\alpha}{\partial \alpha}$:

$$\begin{aligned} \frac{\partial \delta_\alpha}{\partial \alpha} &= c_{k-1}^T P_{k-1}^{-1} c_{k-1} - y_k^2 - \left(2\frac{\partial w^T}{\partial \alpha} L_\alpha^{-1} w + w^T \frac{\partial L_\alpha^{-1}}{\partial \alpha} w\right) \\ &= c_{k-1}^T P_{k-1}^{-1} c_{k-1} - y_k^2 - \left(2\frac{\partial w^T}{\partial \alpha} L_\alpha^{-1} w - w^T L_\alpha^{-1} \frac{\partial L_\alpha}{\partial \alpha} L_\alpha^{-1} w\right) \end{aligned} \tag{C.6}$$

(C.6) fait apparaître les termes $\dfrac{\partial w^T}{\partial \alpha} L_\alpha^{-1} w$ et $w^T L_\alpha^{-1} \dfrac{\partial L_\alpha}{\partial \alpha} L_\alpha^{-1} w$ qui s'obtiennent en plusieurs étapes. En utilisant le lemme d'inversion matricielle, on développe L_α^{-1} via :

$$L_\alpha^{-1} = \left[\alpha P_{k-1}^{-1} + (1-\alpha) d_k d_k^T\right]^{-1} = \frac{I}{\alpha} - \frac{1-\alpha}{\alpha\left[1+(1-\alpha) g_k\right]} P_{k-1} d_k d_k^T P_{k-1} \tag{C.7}$$

avec $g_k = d_k^T P_{k-1} d_k$

Il vient :

$$\frac{\partial w^T}{\partial \alpha} L_\alpha^{-1} w = c_{k-1}^T P_{k-1}^{-1} c_{k-1} - d_k^T c_{k-1} y_k + \frac{1-\alpha_k}{\alpha_k + (1-\alpha_k) g_k} \nu_k \left(d_k^T c_k - g_k y_k \right) \tag{C.8}$$

On détermine ensuite $\frac{\partial L_\alpha}{\partial \alpha} = P_{k-1}^{-1} - d_k d_k^T$. En utilisant (C.7), on a :

$$\begin{aligned} w^T L_\alpha^{-1} \frac{\partial L_\alpha}{\partial \alpha} L_\alpha^{-1} w = c_{k-1}^T P_{k-1}^{-1} c_{k-1} - \left(d_k^T c_{k-1} \right)^2 + 2 \frac{1-\alpha_k}{\alpha_k + (1-\alpha_k) g_k} d_k^T c_{k-1} \nu_k (1-g_k) \\ + \left(\frac{1-\alpha_k}{\alpha_k + (1-\alpha_k) g_k} \right)^2 (1-g_k) g_k \nu_k^2 \end{aligned} \tag{C.9}$$

Remplaçons (C.8) et (C.9) dans (C.6) et après quelques manipulations, on obtient :

$$\frac{\partial \delta_\alpha}{\partial \alpha_k} = -\frac{(1-g_k)\alpha_k^2 + 2 g_k \alpha_k - g_k}{\left[\alpha_k + (1-\alpha_k) g_k \right]^2} \nu_k^2$$

$$\Leftrightarrow \frac{\partial \delta_\alpha}{\partial \alpha_k} = 0 \Leftrightarrow \alpha_k^2 - (1-\alpha_k)^2 g_k = 0 \tag{C.10}$$

Avec la condition $0 < \alpha_k \leq 1$, $g_k > 0$, la solution de (C.10) est :

$$\alpha_k^* = \frac{\sqrt{g_k}}{1+\sqrt{g_k}} \tag{C.11}$$

De plus $$\frac{\partial \delta_\alpha^2}{\partial \alpha_k^2} = -\frac{2 g_k \nu_k^2}{\left[\alpha_k + (1-\alpha_k) g_k \right]^3} < 0, \ \forall \alpha_k \in]0,1] \tag{C.12}$$

Alors $$\max(\delta_\alpha) = \delta_\alpha \big|_{\alpha_k = \alpha_k^*} = \nu_k^2 \Big/ \left(1 + \sqrt{g_k} \right)^2 \tag{C.13}$$

D'après le théorème 1.1, pour avoir l'intersection $\mathbf{E}_{k-1} \cap \mathbf{B}_k \neq \varnothing$, on doit vérifier :

$$\max(\delta_\alpha) < 1$$

$$\Leftrightarrow \frac{\nu_k^2}{\left(1 + \sqrt{g_k} \right)^2} < 1 \quad \Leftrightarrow \quad -1 - \sqrt{g_k} \leq \nu_k \leq 1 + \sqrt{g_k}$$

$$\Leftrightarrow \begin{cases} y_k - 1 < d_k^T c_{k-1} + \sqrt{d_k^T P_{k-1} d_k} \\ y_k + 1 > d_k^T c_{k-1} - \sqrt{d_k^T P_{k-1} d_k} \end{cases} \tag{C.14}$$

Les relations de (C.14) correspondent au test de cohérence présenté dans le paragraphe 1.2.4.

Annexe D

Démonstration du théorème 4.9

D.1 Taux de données éliminées

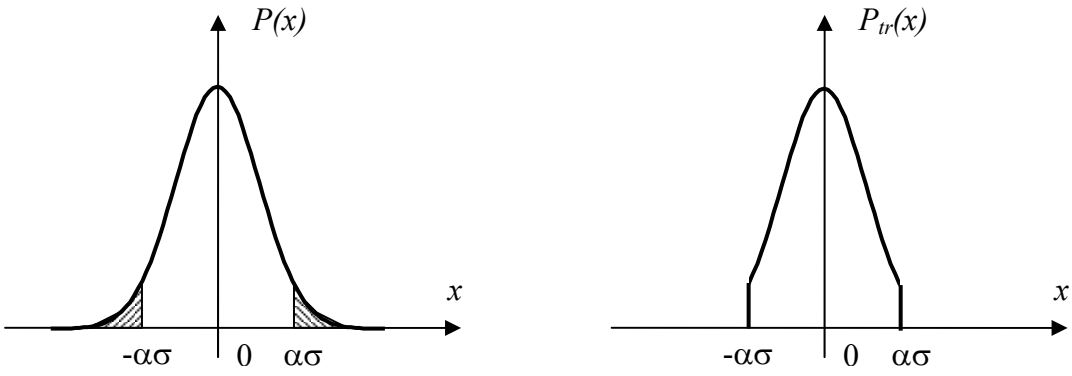
Considérons une distribution gaussienne, sa densité de probabilité est :

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{D.1}$$

où μ est la valeur moyenne et σ^2 la variance.

Rappelons que $P(x)$ satisfait :

$$\int_{-\infty}^{\infty} P(x)dx = 1 \tag{D.2}$$



a) Distribution gaussienne b) Distribution gaussienne tronquée

Figure D.1

Supposons que l'on tronque les deux queues de la distribution à $\mu - \alpha\sigma$ et $\mu + \alpha\sigma$. Alors le cumul de la densité de probabilité de la variable aléatoire ainsi obtenue est :

$$I = \int_{\mu-\alpha\sigma}^{\mu+\alpha\sigma} P(x)dx = \frac{1}{\sigma\sqrt{2\pi}} \int_{\mu-\alpha\sigma}^{\mu+\alpha\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \frac{2}{\sigma\sqrt{2\pi}} \int_{0}^{\mu+\alpha} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \tag{D.3}$$

Posons $t = \frac{x-\mu}{\sigma}$, la formule (D.3) devient :

$$I = \frac{2}{\sqrt{2\pi}} \int_0^{\alpha} e^{-\frac{t^2}{2}} dt \tag{D.4}$$

Définissons :

$$erf\left(x\right) \triangleq \frac{2}{\sqrt{\pi}} \int_0^{x} e^{-t^2} dt \tag{D.5}$$

erf est appelée fonction erreur. De (D.4) et (D.5), on déduit :

$$I = erf\left(\frac{\alpha}{\sqrt{2}}\right) \tag{D.6}$$

Le taux des variables éliminées en tronquant les deux queues d'une distribution gaussienne est :

$$R_e(\%) = 100(1-I) = 100\left(1-\frac{1}{q}\right) \tag{D.7}$$

avec $$q = \frac{1}{erf\left(\alpha/\sqrt{2}\right)} \tag{D.8}$$

D.2 Kurtosis d'une distribution gaussienne tronquée

Par définition, le Kurtosis d'une distribution est calculé par :

$$kurtosis(x) = \frac{\mathrm{E}\left[\left(x-\mu\right)^4\right]}{\sigma^4} \tag{D.9}$$

où E(.) désigne l'opérateur espérance mathématique.

Sans perte de généralité, on suppose que la moyenne est nulle, μ = 0. Alors pour la distribution gaussienne tronquée définie ci-avant, on calcule :

$$kurtosis(x_{tr}) = \frac{\mathrm{E}\left[\left(x_{tr}\right)^4\right]}{\sigma_{tr}^4} \tag{D.10}$$

où l'indice *tr* dénote la distribution tronquée.

Supposons que $P_{tr}(x)$ est la densité de probabilité d'une distribution tronquée. Elle doit satisfaire la propriété suivante :

$$\int_{-\infty}^{\infty} P_{tr}(x)dx = 1$$

$$\Rightarrow \int_{-\alpha\sigma}^{\alpha\sigma} P_{tr}(x)dx = 1 \tag{D.11}$$

D'après (D.3) et (D.6), on a :

$$\int_{-\alpha\sigma}^{\sigma} P(x)dx = erf\left(\frac{\alpha}{\sqrt{2}}\right) = \frac{1}{q} \tag{D.12}$$

Dans l'intervalle $\left[-\alpha\sigma, \alpha\sigma\right]$, $P(x)$ et $P_{tr}(x)$ sont « semblables » (à un facteur de normalisation près). Alors, de (D.11) et (D.12), on déduit :

$$P_{tr}(x) = qP(x), \quad x \in \left[-\alpha\sigma, \alpha\sigma\right] \tag{D.13}$$

La variance de la distribution gaussienne tronquée est calculée par :

$$\sigma_{tr}^2 = \int_{-\infty}^{\infty} P_{tr}(x)x^2 dx = \int_{-\alpha\sigma}^{\alpha\sigma} qP(x)x^2 dx = \frac{p}{\sigma\sqrt{2\pi}} \int_{-\alpha\sigma}^{\alpha\sigma} e^{-\frac{x^2}{2\sigma^2}} x^2 dx$$

Après quelques manipulations classiques, on obtient :

$$\sigma_{tr}^2 = \sigma^2 \left(1 - \sqrt{\frac{2}{\pi}} q\alpha e^{-\alpha^2/2} \right) \tag{D.14}$$

où σ^2 est la variance de la distribution gaussienne initiale.

Calculons maintenant le moment d'ordre 4 de la distribution tronquée :

$$\mathrm{E}\left(x_{tr}^4\right) = \int_{-\infty}^{\infty} P_{tr}(x)x^4 dx = \int_{-\alpha\sigma}^{\alpha\sigma} qP(x)x^4 dx = \frac{q}{\sigma\sqrt{2\pi}} \int_{-\alpha\sigma}^{\alpha\sigma} e^{-\frac{x^2}{2\sigma^2}} x^4 dx$$

Après calcul de l'intégrale, on obtient :

$$\mathrm{E}\left(x_{tr}^4\right) = \sigma^4 \left[3 - \sqrt{\frac{2}{\pi}} \alpha q e^{-\alpha^2/2} \left(\alpha^2 + 3\right) \right] \tag{D.15}$$

En remplaçant (D.14) et (D.15) dans (D.10), on déduit :

$$kurtosis(x_{tr}) = \frac{3 - \sqrt{\frac{2}{\pi}} \alpha q e^{-\alpha^2/2} \left(\alpha^2 + 3\right)}{\left(1 - \sqrt{\frac{2}{\pi}} \alpha q e^{-\alpha^2/2} \right)^2} \tag{D.16}$$

Annexe E

Application : approche ensembliste ellipsoïdale et détection des défauts

Théoriquement, l'ellipsoïde trouvé par l'approche ensembliste ellipsoïdale contient de manière sûre l'espace admissible de paramètres. Ainsi on peut utiliser cette propriété pour détecter les défauts dans le système. Supposons que ces défauts apparaissent après le processus d'identification (i.e. données acquises sans présence de défaut). On propose dans la suite deux techniques de détection. Cette démarche sera vérifiée sur les données simulées.

E.1 Technique de vérification de l'intersection

Dans cette technique, on réalise un test de cohérence entre la bande $\mathbf{B}_k$ obtenue à l'instant k (nouvelle donnée) avec l'ellipsoïde $\mathbf{E}(c,P)$ obtenu lors de l'identification. Naturellement, une intersection vide entre eux indique qu'il y a un défaut dans le système, à condition que la bande ne soit pas générée par une donnée aberrante.

Utilisons les tests de cohérence présentés dans les paragraphes 1.2.4 et 1.3.3. L'intersection $\mathbf{B}_k \cap \mathbf{E}$ est vide si :

- pour modèle à erreur d'équation

$$y_k < d_k^T c - \sqrt{d_k^T P d_k} - \gamma \text{ ou } y_k > d_k^T c + \sqrt{d_k^T P d_k} + \gamma \tag{E.1}$$

- pour modèle à erreur de sortie

$$y_k < d_m^T(k)c - \sqrt{d_m^T(k) P d_m(k)} - \eta^M \text{ ou } y_k > d_M^T(k)c + \sqrt{d_M^T(k) P d_M(k)} + \eta^M \tag{E.2}$$

où les paramètres avec l'indice k correspondent aux nouvelle données, γ et η^M sont les bornes utilisées dans l'identification (pour le modèle à erreur d'équation et le modèle à erreur de sortie respectivement).

E.2 Technique de calcul de l'intervalle de sortie

La deuxième technique utilise l'ellipsoïde $\mathbf{E}(c,P)$ et les nouvelles données pour calculer l'intervalle prédictif sur la sortie à chaque instant, noté $[\hat{y}_k] = [\hat{y}_m(k), \hat{y}_M(k)]$ où $\hat{y}_m(k)$, $\hat{y}_M(k)$ sont les valeurs prédites minimum et maximum. Un défaut est détecté si la

nouvelle mesure n'appartient pas à cet intervalle, i.e. $y_k \notin [\hat{y}_k]$. 5 méthodes de calcul $[\hat{y}_k]$ sont proposées.

E.2.1 Pour un modèle à erreur d'équation

a) Méthode 1

On considère le centre c de l'ellipsoïde obtenu $\mathbf{E}(c,P)$ comme le vecteur des paramètres estimés. L'intervalle prédictif sur la sortie est calculé par :

$$\begin{cases} \hat{y}_m(k) = d_k^T c - \gamma \\ \hat{y}_M(k) = d_k^T c + \gamma \end{cases} \tag{E.3}$$

où le vecteur d_k contient les nouvelles données et γ est la borne utilisée lors de la procédure d'identification.

Remarque

Cet intervalle prédictif dépend uniquement du centre de l'ellipsoïde mais pas de sa taille. En fait, le vecteur des « vrais » paramètres peut se situer n'importe où dans l'ellipsoïde **E**. En conséquence, cet intervalle ne peut être celui qui contient toutes les sorties possibles du système.

b) Méthode 2

Cette méthode tient compte de la taille de l'ellipsoïde obtenu pour calculer l'intervalle prédictif. A partir du vecteur directeur d_k de la bande $\mathbf{B}_k$, on cherche deux hyperplans tangents à l'ellipsoïde **E** et on trouve $[\hat{y}_k]$.

$$\begin{cases} \hat{y}_m(k) = d_k^T c - \sqrt{d_k^T P d_k} \\ \hat{y}_M(k) = d_k^T c + \sqrt{d_k^T P d_k} \end{cases} \tag{E.4}$$

La figure E.1 illustre cette technique de calcul de l'intervalle prédictif pour la sortie. Le vecteur d_k est normalisé sur x_2.

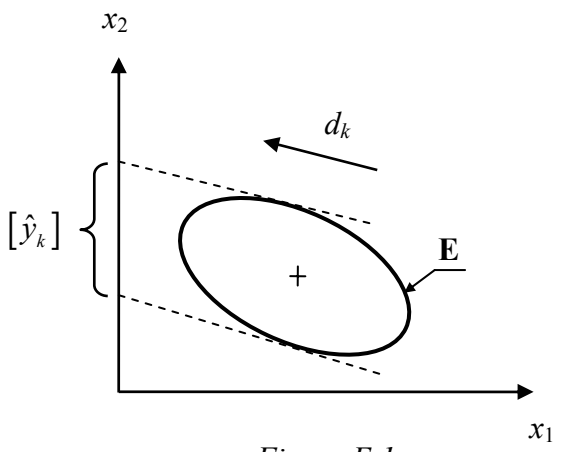


Figure E.1

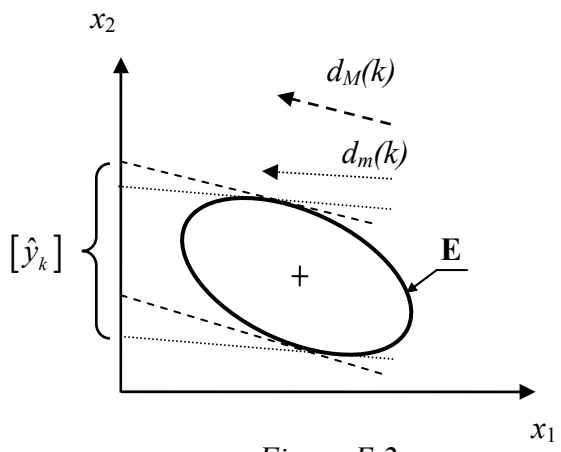


Figure E.2

c) Méthode 3

Cette méthode élargit l'intervalle de sortie trouvé dans la méthode 2 en y ajoutant la borne du bruit :

$$\begin{cases} \hat{y}_m(k) = d_k^T c - \sqrt{d_k^T P d_k} - \gamma \\ \hat{y}_M(k) = d_k^T c + \sqrt{d_k^T P d_k} + \gamma \end{cases} \tag{E.5}$$

Alors, l'intervalle trouvé par cette méthode est le plus grand (pour le modèle à erreur d'équation). Il possède la propriété énoncée ci-après.

Propriété

Si $x^* \in \mathbf{E}$ et $|e_k| < \gamma$, *alors* $y_k \in [\hat{y}_k]$, $\forall k = 1:l$.

où x^* *est le vecteur des « vrais » paramètres du système,* γ *est la borne utilisée en identification et l est la longueur des données.*

Démonstration

D'après (1.1), la sortie d'un système linéaire en les paramètres s'écrit :

$$y_k = d_k^T x_k + e_k$$

Avec l'hypothèse que $|e_k| \le \gamma_k \le \gamma \ \forall k$ (1.2), y_k satisfait :

$$\min\left\{d_k^T x^*\right\} - \gamma \le y_k \le \max\left\{d_k^T x^*\right\} + \gamma$$

D'un point de vue géométrique, on sait que pour tous les points $x \in \mathbf{E}$, pour un vecteur direction d_k donné, on a :

$$d_k^T x_k^{'-} \le d_k^T x \le d_k^T x_k^{'+}$$

où $x_k^{'\pm} = c \pm \frac{Pd}{\sqrt{d^T Pd}}$ sont deux points de la surface de l'ellipsoïde **E**(c,P) dans la direction d_k.

Alors si $x^* \in \mathbf{E}$, on aura :

$$d_m^T(k)c - \sqrt{d_m^T(k)Pd_m(k)} - \gamma \le y_k \le d_M^T(k)c + \sqrt{d_M^T(k)Pd_M(k)} + \gamma$$

donc $\quad y_k \in [\hat{y}_k]$. □

A partir des propriétés précédentes, on en déduit le corollaire suivant :

Corollaire

Si $y_k \notin [\hat{y}_k]$, alors il y a défaut. Le défaut se situe soit dans les paramètres $x^* \notin \mathbf{E}$, soit concerne le capteur $|e_k| > \gamma$.

E.2.2 Pour un modèle à erreur de sortie

a) Méthode 4

Cette méthode correspond à la méthode 2 pour un modèle à erreur d'équation. La bande $\mathbf{B}_k$ se délimite maintenant entre deux hyperplans non parallèles. L'intervalle prédictif sur la sortie est alors calculé par :

$$\begin{cases} \hat{y}_m(k) = d_m^T(k)c - \sqrt{d_m^T(k)Pd_m(k)} \\ \hat{y}_M(k) = d_M^T(k)c + \sqrt{d_M^T(k)Pd_M(k)} \end{cases} \tag{E.6}$$

La figure E.2 illustre l'intervalle prédictif trouvé par cette méthode (en supposant que les vecteurs régresseurs sont normalisés sur x_2).

b) Méthode 5

C'est une application de la méthode 3 à ce nouveau contexte. On calcule alors l'intervalle le plus grand pour un modèle à erreur de sortie :

$$[\hat{y}_k] = \begin{cases} \hat{y}_m(k) = d_m^T(k)c - \sqrt{d_m^T(k)Pd_m(k)} - \gamma \\ \hat{y}_M(k) = d_M^T(k)c + \sqrt{d_M^T(k)Pd_M(k)} + \gamma \end{cases} \tag{E.7}$$

Avec cet intervalle prédictif, on obtient également la propriété E.1 pour le modèle à erreur de sortie.

Remarque

En comparant les équations (E.1) avec (E.5) (modèle à erreur d'équation) et (E.2) avec (E.7) (modèle à erreur de sortie), on constate qu'elles sont identiques. Ainsi, les deux techniques ci-avant apportent la même conclusion sur la détection de défauts. Néanmoins, la technique via l'intervalle de sortie semble plus facile à interpréter.

E.3 Simulation

Pour vérifier la validité de la technique de détection de défaut présentée ci-avant, on réalise la simulation sur le modèle du 1ère ordre donné dans (2.1) :

$$H(z) = \frac{bz^{-1}}{1-az^{-1}} = \frac{0.05z^{-1}}{1-0.95z^{-1}}$$

On utilise 3 jeux données simulées avec $\mathrm{RSB} \approx 70\mathrm{dB}$, 50dB et 30dB respectivement. Deux types de défauts sont considérés, à savoir, défaut capteur (un biais est introduit) et défaut paramétrique.

Pour simuler le défaut capteur, on crée des données par :

$$\begin{cases} y_k = 0.95y_{k-1} + 0.05u_k + e_k \\ y_{kf} = y_k + f_k \end{cases}$$

où e_k est un bruit uniforme, y_{fk} est la sortie avec défaut et f_k est le défaut ajouté à la sortie. Dans la simulation, un défaut abrupt (en échelon) se produit à partir de l'instant $k = 50$.

Le Tableau E.1 présente les résultats de détection pour lesquels la méthode 5 est utilisée pour calculer l'intervalle de sortie. On constate qu'avec un même niveau de défaut, plus le niveau de bruit est important, plus la détection du défaut est difficile, ce qui est logique. En effet, un défaut peut être détecté si le niveau du bruit de mesure est plus faible que l'impact du défaut (i.e. le défaut n'est pas noyé dans le bruit). La figure E.3 donne l'intervalle prédictif et la sortie avec un défaut en rampe dans le jeu de données numéro 2 ($\mathrm{RSB} \approx 50\mathrm{dB}$). On peut voir que l'intervalle prédictif sur la sortie est assez grand pour détecter ce défaut. Dans l'intervalle de temps (k = 50 : 80) sur la figure E.3, on ne détecte le défaut à l'instant k = 78) avec $y_k \notin [\hat{y}_k]$.

Données	**Taux de détection**		
	Jeu 1 (RSB ≈70dB)	Jeu 2 (RSB ≈50dB)	Jeu 3 (RSB ≈30dB)
Sans défaut	0/300	0/300	0/300
Défaut rampe **(k_r=0.001)**	299/300	158/300	0/300
Défaut échelon **(h_e=0.2)**	250/250	207/250	1/250

Tableau E.1 : Taux de détection des défauts dans un capteur

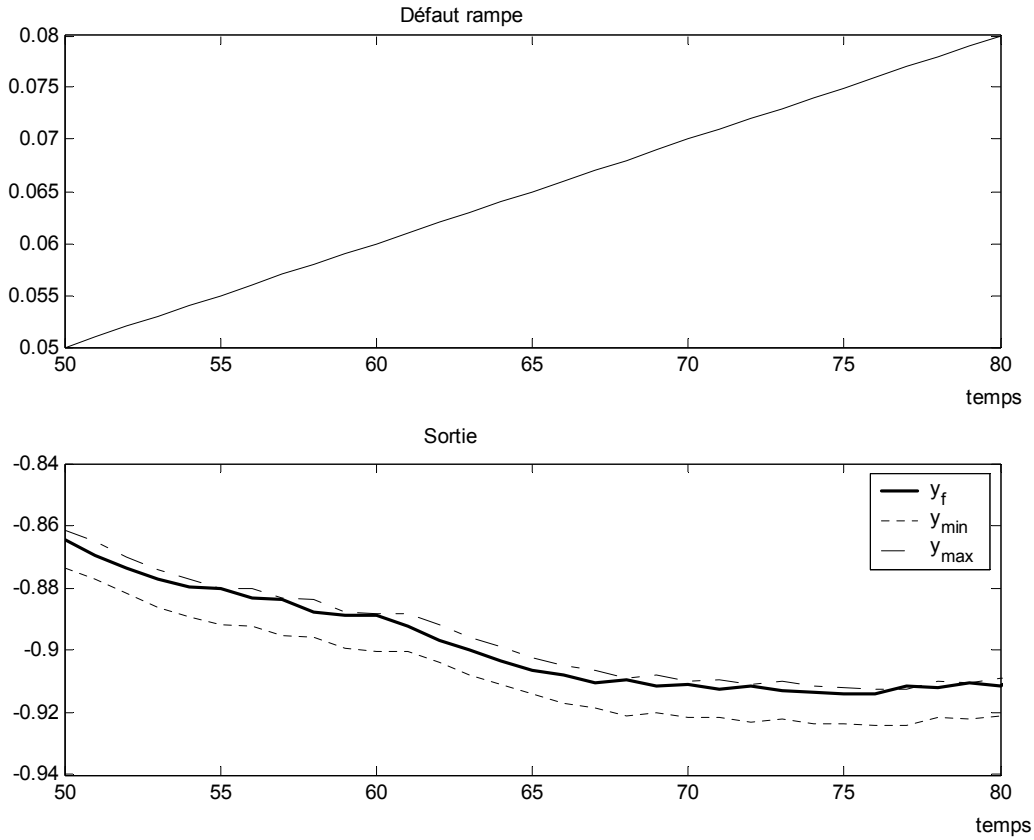


Figure E.3 : L'intervalle prédictif et la sortie avec un défaut un rampe.

Données	**Type de défaut**						
	2*gain	gain/2	2*t_M	t_M/2	2*gain & 2*t_M	-10%a	-10%b
Jeu 1(70dB)	299/300	299/300	299/300	242/300	293/300	296/300	299/300
Jeu 2 (50dB)	299/300	299/300	262/300	173/300	233/300	274/300	18/300
Jeu 3 (30dB)	0/300	0/300	0/300	2/300	0/300	0/300	0/300

Tableau E.2 : Taux de détection des défauts sur les paramètres.

Le Tableau E.2 présente les taux de détection en cas de défaut sur les paramètres (t_M étant le temps de montée du système). Comme dans le cas de défaut capteur, on peut détecter la plupart des défauts lorsque le bruit affectant la sortie est faible (jeu 1). Il semble impossible de détecter avec cette technique, les défauts dans un système présentant un bruit de mesure très important.

En conclusion, on a brièvement illustré ce que pourrait apporter la démarche ensembliste ellipsoïdale à la détection de défauts additifs et multiplicatifs. Deux techniques de détections sont prouvées identiques (intersection bande/ellipsoïde ou intervalle de prédiction pour la sortie du système). Les défauts importants ont été détectés en simulation. La réduction de l'intervalle prédictif sur la sortie nous aiderait à mieux détecter les défauts de faible amplitude.

Bibliographie

[AND 82] Anderson B.D.O., C.R. Johnson Jr., « Exponential Convergence of Adaptive Identification and Control Algorithms », *Automatica*, Vol. 18, No. 1, 1982, pp. 1-13.

[ARR 91] Arruda L. V. R., Favier G., Amaral W., « Reformulation of optimal bounding ellipsoid algorithms as robust identification algorithms with dead zone », *ECC 91 European Control Conference*, Grenoble, France, July 1991, pp. 1194-1199.

[BAI 95] Bai E. W., Tempo R., Cho H., « Membership Set Estimators : Size, Optimal Inputs, Complexity and Relations with Least Squares », *IEEE Transactions on Circuits and Systems –I : Fundamental Theory and Application*, Vol. 42, No. 5, May 1995, pp. 266-277.

[BAI 99] Bai E. W., Huang Y. F., « Convergent of optimal sequential outer bounding sets in bounded error parameter estimation », *Mathematics and Computers in Simulation*, Vol. 49, 1999, pp. 307-317.

[BEC 03] Becis-Aubry Y., « Contribution à l'estimation ensembliste des systèmes linéaires et non linéaires », Thèse de doctorat, Université Henri Poincaré-Nancy 1, 2003.

[BIT 84] Bitmead R.R., « Persistence of Excitation Conditions and the Convergence of Adaptive Schemes », *IEEE Transaction on Information Theory*, vol. IT-30, No. 2, March 1984, pp. 183-191.

[BJO 96] Björck A., « Numerical Methods for Least Squares problems », *SIAM,* Philadelphia, USA, 1996.

[BRA 02] Braems I., Méthodes ensemblistes garanties pour l'estimation de grandeurs physiques, thèse de doctorat Université Paris XI, L2S, Supélec, 2002.

[BRO 90] Broman V., Shensa M.J., « A compact algorithm for the intersection and approximation of N-dimensional polytopes », *Mathematics and Computers in Simulation* 32, 1990, pp. 469-480.

[CER 93] Cerone V., « Feasible Parameter Set for Linear Models with Bounded Errors in All Variables », *Automatica,* Vol. 29, No 6, 1993, pp. 1551-1555.

[CLE 87] Clement T., « Estimation d'incertitude paramétrique dans un contexte de bruit de sortie inconnu mais borné », Thèse de doctorat, INP Grenoble, 1987.

[CLE 90] Clement T., Gentil S., « Recursive membership set estimation for output error models », *Mathematics and Computers in Simulation*, Vol. 32, 1990, pp. 505-513.

[DAS 87] Dasgupta S., Huang Y. F., « Asymptotically Convergent Modified Recursive Least-square with Data-dependent Updating and Forgetting Factor for Systems with Bounded Noise », *IEEE Transaction on Information Theory*, Vol. IT 33, No. 3, May 1987, pp. 383-393.

[DEL 94] Deller J. R., Nayeri M., Liu M. S., « Unifying the Landmark Development in Optimal Bounding Ellipsoid Identification », *International Journal of Adaptive Control and Signal Processing*, Vol. 8, 1994, pp. 43-60.

[DUR 00] Durieu C., Walter E., Polyak B., « Set-membership estimation with the trace criterion made simpler than with the determinant criterion », IFAC System Identification, Santa Barbara, California, USA, 2000.

[DUR 01a] Durieu C., Walter E., « Estimation ellipsoïdale à erreur bornée », In *Identification des systèmes* sous la direction de Landau I.D. & Besançon-Voda A., Chapter 2, Hermes Science 2001, pp. 131-154.

[DUR 01b] Durieu C., Walter E., Polyak B., « Multi-input multi output ellipsoidal state bounding », *Journal of Optimization Theory and Application*, Vol. 111(2), 2001, pp. 273-303.

[DUR 96] Durieu C., Polyak B., Walter E., « Trace versus determinant in ellipsoidal outer bounding, with application to state estimation », 13th IFAC World Congress, Sanfrancisco, USA, Vol. I, 1996, pp. 9-12.

[EVA 94] Evans R. J., Zhang C., Soh Y. C., « Bounded Error Estimation Using Dead Zone and Bounding Ellipsoid », *International Journal of Adaptive Control and Signal Processing*, Vol. 8, 1994, pp. 31-42.

[FAV 96] Favier G., Arruda L.V.R., « Review and Comparison of Ellipsoidal

Algorithms », In *Bounding approaches to System Identification*, edited by M. Milanese et al., Plenium Press, New York 1996, pp. 43-68.

[FER 97] Ferreres G., M'Saad M., « Estimation of output error models in the presence of unknown but bounded disturbances », *International Journal of Adaptive Control and Signal Processing*, Vol. 11, 1997, pp. 115-140.

[FOG 82] Fogel E., Huang Y., « On the Value of Information in System Identification – Bounded noise case », *Automatica*, Vol. 18, 1982, pp. 229-238.

[GOL 82] Goldfarb D., Told M.J., « Modification and implementation of the ellipsoid algorithm for linear programming », *Mathematical Programming,* Vol. 23, 1982, pp. 1-19.

[GOL 98] Gollamudi S., Kapoor S., « Set-Membership Filtering and a Set-Membership Normalized LMS Algorithm with an Adaptive Step Size », *IEEE Signal Processing Letters*, Vol. 5, No. 5, May 1998, pp. 111-114.

[GRO 88] Grötschel M., Lovasz L., Schrijver A., *Geometric Algorithms and Combinatorial Optimization*, Springer, Berlin, 1988, Chap. 3.

[GUT 94] Gutman P. O., « On-line parameter interval estimation using recursive least square », *International Journal of Adaptive Control and Signal Processing*, Vol. 8, 1994, pp. 61-72.

[HEI 03] Heim B., « Approches ensembliste et par logique floue pour le diagnostic causal de procédés de raffinage. Application à un pilote de FCC », Thèse de doctorat, INP Grenoble, 2003.

[JOA 97] Joachim D., Deller J.R., Nayeri Jr. and M., « Practical consideration in the use of a new OBE algorithm that blindly estimates error bounds », *Proceeding of the 40th Midwest Symposium on Circuit and systems, Sacramento,* California., USA, Vol. 2, August 1997, pp 762-765.

[LAN 93] Landau I. D., *Identification et commande des systèmes*, Hermès, Paris, 1993.

[LAW 74] Lawson C.L., Hanson R.J., « Solving least squares problems », *Prentice Hall*, 1974.

[LES 02] Lesecq S., Barraud A., « Une approche factorisée plus simple et numériquement stable pour l'estimation ensembliste », *JESA*, Vol. 36(4), 2002,

pp. 505-518.

[LES 03] Lesecq S., Barraud A., Tran Dinh K.Q., « Numerical Accurate Computations for Ellipsoidal State Bounding », *MED'03*, Rhodos, Greece, June 2003.

[LIN 98] Lin T.M., Nayeri M., Deller J.R., « A consistently convergent OBE algorithm with automatic estimation of error bounds », *International Journal of Adaptive Control and Signal Processing*, Vol. 12, 1998, pp. 305-324.

[LIU 94] Liu M.S., Nayeri M., Deller J.R., « Do interpretable optimal bounding ellipsoid algorithms converge? Part II – OBE vs. RLS: clearing the smoke », *SYSID'94*, Copenhagen, Denmark, 1994, pp. 395-400.

[LJU 83] Ljung L., Söderström T., « Theory and Practice of Recursive Identification », MIT Press, Cambridge.

[LOZ 87] Lozano-Leal R., Ortega R., « Reformulation of the Parameter Identification Problem for Systems with Bounded Disturbances », *Automatica*, Vol. 23, No. 2, 1987, pp. 247-251.

[MAK 02] Maksarov D. G., Norton J. P., « Computationally efficient algorithms for state estimation with ellipsoidal approximation », *International Journal of Adaptive Control and Signal Processing*, Vol. 16, 2002, pp. 411-434.

[MAK 96] Maksarov D. G., Norton J.P., « Tuning of Noise Bounds in State Bounding », *Proceeding 10th IMACS Symposium on Modelling, Analysis and Simulation*, Lille, France, July 1996, pp. 837-842.

[MIL 82] Milanese M., Belforte G., « Estimation Theory and Uncertainty Intervals Evaluation in Presence of Unknown but Bounded Errors. Linear Families of Models and Estimators », *IEEE Transaction on Automatic Control*, Vol. AC 27, No. 2, Avril 1982.

[MO 90] Mo S. H., Norton J. P. « Fast and robust algorithm to compute exact polytope parameters bounds », *Mathematics and Computers in Simulation*, Vol. 32, 1990, pp. 481-493.

[NAG 97b] Nagaraj S., Gollamudi S., Deller J. R., Huang Y. F., Kapoor S., « Performance Studies on a "Quasi-OBE" Algorithm for Real-time Signal Processing », *Proceedings of the 40th Midwest Symposium on Circuit and Systems*, Vol. 2,

Sacramento, CA, USA, 1997, pp 770-772.

[NAG 97] Nagaraj S., Gollamudi S., Kapoor S., Huang Y. F., « Bounded Error Estimation : Set theoretic and Least-Square Formulations », *CISS*, The Johns Hopkin University, Baltimore, MD, March 1997.

[NAY 93] Nayeri M., Liu M. S, Deller J. R « An interpretable and converging set-membership algorithm », *Proceeding IEEE of International Conference on Acoustics, Speech and Signal Processing*, Vol IV, Minneapolis, New York, 1993, pp. 472-475.

[NAY 94] Nayeri M., Liu M.S., Deller J.R., « Do interpretable optimal bounding ellipsoid algorithms converge? Part I – The long awaited set-convergence proof », *SYSID'94*, Copenhagen, Denmark 1994, pp. 389-394.

[NOR 87] Norton J. P., « Identification and Application of Bounded-parameter Models », *Automatica*, Vol. 23, No. 4, 1987, pp. 497-507.

[POL 02] Polyak B. T., Nazin S. A., Durieu C., Walter E., « Ellipsoidal estimation under model uncertainty », In 15th IFAC Triennial World Congress, Barcelona, Spain, 2002.

[POL 98] Polyak B. T., « Convexity of Quadratic Transformations and Its Use in Control and Optimization », *Journal of Optimization Theory and Application*, Vol. 99, December 1998, pp. 553-583.

[PRO 94] Pronzato L., Walter E., « Minimal volume ellipsoids », *International Journal of Adaptive Control and Signal Processing*, Vol. 8, 1994, pp. 15-30.

[ROT 79] Rothenberg R. I., *Linear Programing*, Elsevier North Holland, 1979.

[SCH 68] Schweppe C. F., « Recursive State Estimation: Unknown but Bounded Errors and System Inputs », *IEEE Transactions on Automatic Control*, Vol. 13(1), 1968, pp. 22-28.

[SED 98] Sedda E., « Estimation en ligne de l'état et des paramètres d'une machine asynchrone par filtrage à erreur borne et par filtrage de Kalman », Thèse de doctorat, ENS Cachan, 1998.

[STE 73] Stewart G. W., *Introduction to Matrix Computation*, Academic Press, 1973.

[TAN 97] Tan G., Wen C., Soh Y. C., « Identification for System with Bounded Noise »,

IEEE Transaction on Automatic Control, Vol. 42, No. 7, July 1997, pp. 996-1001.

[TRA 03] Tran Dinh K.Q., Estimation ensembliste ellipsoïdale, analyse bibliographique, rapport interne LAG, INP Grenoble, 2003.

[VER 91] Verses S.M., Norton J.P., « Structure selection for bounded-parameter models: consistency conditions and selection criterion », *IEEE Transactions on Automatic Control,* Vol. 36(4), 1991, pp. 474-481.

[VIC 93] Vicino A., Zappa G., « Sequential approximation of parameter sets for identification with parametric and nonparametric uncertainty », *Proceeding IEEE CDC*, IEEE, New York, 1993, pp. 2044-2049.

[VIV 03] Vivas A.O., Poignet Ph., Marquet F., Pierrot F., Gauthier M., « Experimental Dynamic Identification of a Fully Parallel Robot », *Proceeding of the 2003 IEEE International Conference one Robotics and Automation*, Taïwan, September 14-19, 2003.

[WAL 88] Walter E., Piet-Lahanier H., « Estimation of parameter from bounded-error data : A survey », *Proceeding 12th IMACS World Congress*, Paris, France, 1998, pp. 467-472.

Résumé

Ce travail concerne les techniques d'identification ensembliste dites à erreur inconnue mais bornée. On s'est ici exclusivement intéressé à l'approche ensembliste ellipsoïdale. Dans ce contexte précis, nos contributions portent sur une évaluation réaliste de la borne, basée sur l'analyse du Kurtosis de la séquence d'erreur. On propose également une approche unifiée des algorithmes. Une formulation numériquement stable a également été proposée, qualité qui n'est pas assurée par la quasi-totalité des solutions publiées. L'analyse de la convergence de ce type de techniques fait intervenir la notion habituelle d'excitation persistante pour laquelle on donne une solution algorithmique permettant de qualifier l'entrée, et comment construire des entrées « optimales ». Enfin on s'est intéressé au compromis que l'on pouvait établir entre l'algorithme séquentiel sous optimal et une approche globale optimale mais numériquement inaccessible. L'ensemble de ces points est illustré tant en simulation qu'avec des données réelles provenant du monde industriel.

Abstract

This work deals with the membership set estimation techniques based upon an unknown but bounded error. Only the ellipsoidal approach has been considered here. In this particular context, a realistic evaluation of the bound based upon an analysis of the Kurtosis number of the error sequence has been proposed. A unified approach of the different algorithms has been given and their numerically stable implementation has been shown. Note that nearly all published algorithms do not possess this fundamental property. The convergence analysis of these estimation techniques requires the well-known persistent excitation property. An algorithmic evaluation of this persistency hypothesis of the input has been proposed and a way to built such an input in this particular context is exhibited. Lastly, the compromise between the suboptimal sequential algorithm and an optimal global one but numerically unrealistic has been studied. The different results are illustrated thanks to some simulated data but also with some real data from industrial applications.

Mots clés : approche ensembliste, ellipsoïde, algorithme OBE, factorisation orthogonale, convergence, persistance d'excitation, Kurtosis.

Key words: membership set estimation, ellipsoid, OBE algorithm, orthogonal factorization, convergence, excitation persistency, Kurtosis.