

M0009/997

Ecole Nationale Polytechnique



المدرسة الوطنية للتكنولوجيا
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

D.E.R de Génie Electrique et Informatique
Département d'Electronique

THESE DE MAGISTER

en
Electronique Appliquée
Option
Acquisition et traitement de l'information

présentée par

BENSAID Houcine
Ingénieur d'Etat en Electronique

Thème

***Synthèse des Filtrés Numériques
R.I.I. par la Méthode des
Moindres Carrés Vrais (MCV).***

Soutenue publiquement le : 22 juin 1999

Devant le jury composé de :

M. D. BERKANI

M. A. BELOUHRANI

M. M. TADJINE

Mme. L. HAMAMI

M. R. ZAKNOUNE

Professeur (ENP)

Docteur d'état (ENP)

Docteur d'état (ENP)

Chargé de Cours (ENP)

Docteur d'état (EMP)

Président

Rapporteur

Examinateur

Examinateur

Examinateur

ملخص

الهدف من هذا العمل هو تركيب مرشحات عددية ذات الاستجابة الدفعية الامتناهية باستعمال طريقة المربعات الصغرى الحقيقية. في هذا الإطار، تمنا بتحليل هذه الطريقة مرتكزين على الفكرة الأولية التي اقترحها "ستايغلتر" و "ماك برايد" لتشخيص الأنظمة الآلية. لقد تمنا أيضا بتطبيق هذه الطريقة لتركيب المرشحات العددية و تقليص رتبة النماذج. تظهر لنا فعالية هذه الطريقة من خلال النتائج المحصل عليها وكذلك مقارنتها بتلك التي تعطىها طريقة "بادي" و طريقة المربعات الصغرى المعدلة.

المفاتيح : المرشحات العددية ذات الإستجابة الدفعية الامتناهية، تركيب المرشحات، طريقة المربعات الصغرى المعدلة، طريقة المربعات الصغرى الحقيقية.

Résumé

L'objectif du présent travail est la synthèse de filtres numériques récurrents (filtres RII) par la méthode des moindres carrés vrais (MCV).

Dans ce cadre, nous avons développé la méthode (MCV), en nous basant sur une idée de Steiglitz et McBride proposée initialement pour l'identification des systèmes en automatique. Nous avons appliqué la méthode MCV pour la synthèse de filtres numériques RII et aussi pour la réduction de modèles.

L'efficacité de la méthode est observée à travers de nombreuses simulations et aussi par comparaison avec les méthodes de Padé et des moindres carrés modifiés (MCM).

Mots clés : Filtres numériques récurrents (filtres RII), Synthèse des filtres, Méthode des Moindres Carrés Modifiés (MCM), Méthode des Moindres Carrés Vrais (MCV).

Abstract

The aim of this work is the study of the True Least Squares method (TLS) and its application to the synthesis of recursive digital filters (IIR).

In this context, we have developed the TLS method, based on an original idea proposed by Steiglitz et McBride for system identification in control.

We have applied the TLS method to the synthesis of recursive digital filters and model reduction.

The efficiency of the TLS method is clearly shown by the simulation results on several examples and comparisons with Padé and modified least squares (MLS) methods.

Key words : Recursive digital filters (IIR filters), Filter synthesis, Modified Least Squares (MLS) Method, True Least Squares (TLS) Method.



Département d'Electronique

THESE DE MAGISTER

en

Electronique Appliquée

Option

Acquisition et traitement de l'information

Thème

*Synthèse des Filtrés Numériques
R.I.I. par la Méthode des
Moindres Carrés Vrais (MCV).*

Proposé par

M. Belkacem DERRAS

Dirigé par

M. Belkacem DERRAS & M. Adel BELOUCHRANI
Maître de conférences Docteur d'Etat

Etudié par

M Houcine BENSALD
Ingénieur d'état en Electronique

Jun 1999

Dédicaces



A toi, ma mère, je dédie ce modeste travail, fruit de mes humbles efforts et de tes énormes sacrifices.

A vous aussi, mes chers neveux LOUNIS et ROZA.

Houcine

Remerciements



Je tiens particulièrement à exprimer, à travers ces quelques lignes, toute ma gratitude et mon profond respect à mes Directeurs de thèse Mr. Belkacem DERRAS et Mr. Adel BELOUHRANI, pour leur suivi permanent et surtout l'immense patience dont ils ont toujours fait preuve à mon égard.

Je remercie également l'ensemble des membres du jury qui m'ont fait l'honneur de bien vouloir accepter d'examiner et d'évaluer mon travail.

Mes meilleurs sentiments vont aussi aux membres de ma famille, mes soeurs Mahdjouba et Nacera ainsi que mes deux frères Ramdane et Farid, dont la présence à mes côtés, et le soutien constant, surtout pendant les moments difficiles, m'a permis d'achever ce travail.

Enfin, je remercie tous ceux qui ont contribué à la finalisation de cette thèse, tout particulièrement Mr. Belkacem MEZHOUD.

Houcine

Table des matières

1	Introduction	1
1.1	<i>Introduction</i>	1
1.2	<i>Introduction au filtrage numérique</i>	2
1.3	<i>But et présentation du travail</i>	2
1.4	<i>Organisation de la thèse</i>	4
2	Le Filtrage Numérique	5
2.1	<i>Introduction</i>	5
2.2	<i>Représentation des filtres numériques</i>	6
2.3	<i>Analyse des filtres numériques</i>	6
2.3.1	<i>Analyse temporelle</i>	7
2.3.2	<i>Analyse fréquentielle</i>	8
2.4	<i>Stabilité des filtres numériques</i>	11
2.5	<i>Classification des filtres numériques</i>	12
2.5.1	<i>Les filtres RIF</i>	13
2.5.2	<i>Les filtres R.I.I.</i>	13
2.5.3	<i>Comparaison entre filtres RIF et RII</i>	14
2.6	<i>Synthèse des filtres numériques</i>	14
2.7	<i>Synthèse des filtres non-récurrents (R.I.F)</i>	15

2.7.1	Méthode d'échantillonnage fréquentiel	15
2.7.2	Méthode de fenêtrage	15
2.8	Synthèse des filtres récurrents (R.I.I)	17
2.8.1	Méthode d'invariance de la réponse impulsionnelle	17
2.8.2	Méthode de la transformation bilinéaire	17
2.8.3	Méthode d'optimisation	18
3	Synthèse par les Méthodes de Padé et MCM	19
3.1	Introduction	19
3.2	La méthode de Padé	21
3.2.1	Principe	21
3.2.2	Procédure de synthèse	23
3.3	Méthode des Moindres Carrés Modifiés	24
3.3.1	Principe	24
3.3.2	Procédure de synthèse	24
4	La Méthode des Moindres Carrés Vrais (MCV)	31
4.1	Introduction	31
4.2	Développement théorique du cas général MCP	31
4.2.1	Etude et résolution du problème	31
4.2.2	Calcul des matrices R_k	39
4.3	Etude de la méthode des moindres carrés vrais (MCV)	45
4.4	Synthèse des filtres numériques RII par la méthode des moindres carrés vrais (MCV)	48
5	Exemples de Synthèse	51
5.1	Introduction	51

5.2	<i>Application à la synthèse des filtres</i>	52
5.2.1	<i>Synthèse de filtres passe-bas</i>	52
5.2.2	<i>Synthèse de filtres passe-haut</i>	53
5.2.3	<i>Synthèse de filtres passe-bande</i>	53
5.3	<i>Application à la réduction de modèles</i>	54
5.4	<i>Commentaires</i>	54
	Conclusion Générale	77
A	Algorithme de Horner	83
B	Algorithme de Levinson-Durbin	84
C	Algorithme de Mullis-Roberts	85
D	Algorithme de Levinson à deux canaux (bicanal)	86
E	Algorithme de Sylvester	87
F	Algorithme de Jury	89

Chapitre 1

Introduction

1.1 Introduction

De par sa nature, l'homme a toujours besoin d'être en contact avec son environnement, pour cela il utilisa très tôt son génie créateur pour inventer différentes méthodes (la parole, l'écriture, ... etc) qui lui permirent de communiquer avec ses semblables et assouvir sa soif de savoir et de connaissances sans cesse grandissante sur le monde dans lequel il évolue.

Par la suite, la notion de filtrage c'est tout naturellement imposée, car souvent il fallait séparer, trier et recueillir au milieu d'une multitude d'informations, celle recherchée, qui présentait un intérêt à ce moment donné.

Cette notion de filtrage au sens large a été particulièrement adoptée en électricité et en électronique afin de séparer des signaux ayant des spectres de fréquences différents.

Ainsi, les premiers filtres analogiques passifs sont apparus dans le domaine de la radio-électricité et depuis, leur utilisation a pris un essor considérable, dû principalement au développement très rapide de la technologie en général et plus particulièrement de l'électronique.

Actuellement, le filtrage est présent dans un grand nombre de domaines tels que : La médecine, l'astrophysique, la géophysique ... etc.

1.2 Introduction au filtrage numérique

C'est certainement l'énorme développement des calculateurs et de l'informatique au cours de la décennie 1960-1970, qui a été à l'origine du développement des techniques numériques et a permis la mise en œuvre pratique de théories et concepts établis bien auparavant.

Le filtrage, fût l'un des premiers domaines à bénéficier de l'outil informatique, on l'utilisa d'abord pour simuler des filtres analogiques afin d'en étudier le comportement, améliorer les performances avant une éventuelle réalisation pratique.

Le progrès immense réalisé dans le domaine de la technologie des composants électroniques, l'avènement des techniques d'intégration à grande échelle (LSI, VLSI), des circuits intégrés à applications spécifiques (ASICs), la disponibilité et le faible coût des convertisseurs analogique-numérique (CAN) et numérique-analogique (CNA) ont constitués le point de départ effectif du filtrage numérique.

Le filtrage numérique est devenu dès lors un axe de recherche à part entière, enrichi au fur et à mesure par le travail de nombreux scientifiques qui développèrent diverses théories de synthèse des filtres numériques ainsi que des méthodes et schémas pour leurs réalisations matérielle.

1.3 But et présentation du travail

Le traitement numérique du signal en général, le filtrage numérique en particulier, sont à l'heure actuelle largement utilisés dans plusieurs domaines de la science et de la technologie modernes, on peut citer à titre d'exemple :

- l'analyse des signaux biomédicaux;
- le traitement d'image;
- la synthèse de la parole;
- l'analyse des signaux sismiques;
- la prospection pétrolière ... etc.

Cet intérêt s'explique par les nombreux avantages que présentent les filtres numériques par rapport à leurs homologues analogiques et qui sont :

- L'absence de dérives dues à l'échauffement ou au vieillissement des composants.

- Une plus grande fiabilité.
- Un encombrement de plus en plus réduit.
- La possibilité de travailler sur des filtres avec des bandes passantes très étroites.
- La possibilité de travailler dans les bandes des très basses fréquences.
- Une mise en œuvre pratique plus simple.
- Une plus grande souplesse d'utilisation, car on peut changer les caractéristiques du filtres simplement en modifiant ses coefficients sans modifier sa structure matérielle.
- Un prix de revient de plus en plus faible.
- La possibilité de synthèse de filtres adaptatifs d'une manière très simple.

Cependant l'inconvénient des filtres numériques reste certainement leur vitesse de traitement toujours inférieure à celle des filtres analogiques.

En effet, le temps de transit du signal à travers un filtre analogique n'est conditionnée que par le déplacement des électrons dans les conducteurs et ses composants, par contre, pour un filtre numérique s'ajoutent à cela les temps nécessaires pour effectuer les différentes opérations arithmétiques qui allongent de ce fait le temps de transit des signaux et réduisent par conséquent la vitesse de traitement.

Sachant en outre que le nombre d'opérations est directement lié à l'ordre du filtre donc au nombre de coefficients qui le caractérisent, nous permet d'expliquer en partie, le fait qu'une grande partie des travaux consacrés au filtrage numérique soient orientés vers le développement de méthodes de synthèse de filtres numériques définis par un nombre minimum de coefficients afin de réduire le nombre de ces opérations et en augmenter la vitesse de traitement.

L'objectif et le but de notre travail s'inscrivent dans cette optique. A cet effet, trois méthodes de synthèse de filtres numériques R.I.I seront étudiées et implémentées.

1. La méthode de Padé.
2. La méthode des moindres carrés modifiés (MCM).
3. La méthode des moindres carrés vrais (MCV).

Une attention toute particulière sera réservée à la méthode des moindres carrés vrais qui a été utilisée initialement par STEIGLITZ et McBRIDE pour l'identification des systèmes en automatique [42].

On examinera aussi les possibilités d'application de cette méthode pour la réduction de modèles et l'amélioration des résultats de synthèse des filtres obtenus par les deux premières méthodes.

1.4 Organisation de la thèse

L'objectif principal de la présente thèse est l'étude de la méthode des moindres carrés vrais et son application à la synthèse de filtres numériques à réponse impulsionnelle infinie (R.I.I) et à la réduction de modèles.

Pour mieux cerner cet objectif, il nous a semblé intéressant d'organiser notre travail comme suit :

Après ce premier chapitre destiné essentiellement à introduire, présenter et situer le but visé par notre travail, vient un deuxième chapitre consacré à la présentation de quelques outils et connaissances de base essentielles pour l'étude du filtrage numérique ainsi qu'un bref aperçu sur les différentes méthodes dites classiques, de synthèse des filtres numériques.

Deux méthodes particulières de synthèse des filtres R.I.I., la méthode de Padé et la méthode des moindres carrés modifiés (M.C.M) seront présentées et développées tout au long du troisième chapitre.

Le chapitre 4, sera entièrement consacré à l'étude de la méthode des moindres carrés vrais et son application à la synthèse des filtres numériques R.I.I.

Les différents résultats obtenus, donnés par les programmes développés pour la synthèse de filtres par les trois méthodes étudiées seront présentés et dans le cinquième chapitre.

Enfin, nous terminerons cette thèse par une conclusion générale.

Chapitre 2

Le Filtrage Numérique

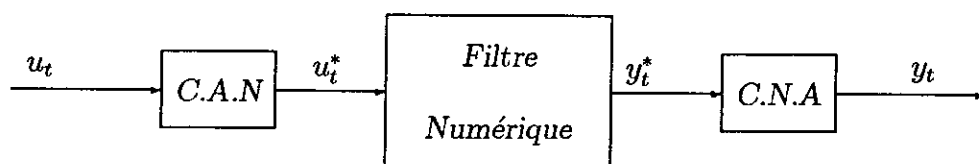
2.1 Introduction

Un filtre numérique est un système numérique linéaire, invariant dans le temps [23], utilisé pour réaliser le filtrage de signaux numériques représentés par des séquences de valeurs discrètes correspondant à des échantillons du signal analogique à des instants déterminés.

Le filtrage numérique est défini comme étant l'opération qui consiste à modifier, en utilisant des opérations arithmétiques (addition, multiplication, ...), la distribution des composantes fréquentielles du signal selon des spécifications données, pouvant être par exemple : l'amplification, l'atténuation ou aussi la rejection d'une bande de fréquences.

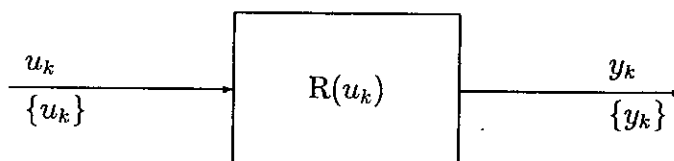
En pratique, la réalisation d'un filtre numérique s'effectue soit en utilisant les capacités de calcul d'un ordinateur (ou d'une carte DSP) par le développement de programmes (réalisation software) ou aussi en utilisant directement les composants de l'électronique digitale (réalisation hardware) [1, 22, 30, 44]. Cependant, ce dernier type de réalisation semble être abandonné, l'implémentation hardware se fait directement sur des ASICs (Application Specific Integrated Circuits) tels que les FPGA (Field Programmable Gate Array).

Dans tous ces cas de réalisation, le système pourra toujours être utilisé soit en temps réel pour le filtrage direct de signaux continus (figure 2.1) où en temps différé dans le cas de signaux préalablement enregistrés.

Figure 2.1: *Filtre numérique utilisé en temps réel*

2.2 Représentation des filtres numériques

Un système numérique en général et un filtre numérique en particulier est caractérisé par la relation $\mathfrak{R}(u_k)$ reliant les deux signaux discrets d'entrée u_k et de sortie y_k , représentés aussi par les séquences $\{u_k\}$ et $\{y_k\}$ (figure 2.2), où \mathfrak{R} est un opérateur linéaire, invariant représentant le filtre.

Figure 2.2: *Représentation d'un filtre numérique*

2.3 Analyse des filtres numériques

L'analyse des filtres numériques peut se faire dans le domaine temporel où le domaine fréquentiel [7]. Dans les deux cas, la connaissance du modèle mathématique [16] décrivant le système est nécessaire.

Pour le cas de l'analyse temporelle, selon le modèle mathématique adopté, trois méthodes sont généralement utilisées :

1. Les équations aux différences.
2. La réponse impulsionnelle.
3. Les variables d'état [23].

Chacune des trois méthodes nous renseigne sur certains aspects du comportement du filtre, elles sont de ce fait complémentaires et constituent ensemble un moyen complet d'analyse.

L'analyse fréquentielle du filtre s'effectue au moyen de transformations qui permettent le passage du domaine temporel au domaine fréquentiel. (transformée de Laplace, transformée en Z , voir [39, 28, 32, 40])

2.3.1 Analyse temporelle

Les équations aux différences

Un filtre numérique est entièrement défini par la connaissance d'une relation de récurrence du type :

$$y_k = \sum_{i=0}^m b_i u_{k-i} - \sum_{j=1}^n a_j y_{k-j} \quad (2.1)$$

dite équation aux différences où les $(1, a_1, \dots, a_n)$ et les (b_0, b_1, \dots, b_m) sont des coefficients de pondération qui caractérisent le filtre.

La résolution de cette équation aux différences permet de déterminer la séquence de sortie $\{y_k\}$ à partir de la connaissance des coefficients du filtre et des éléments de la séquence d'entrée $\{u_k\}$.

La réponse impulsionnelle

En utilisant les hypothèses de linéarité et d'invariance d'un filtre numérique [23], on peut écrire :

$$y_k = \sum_{i=-\infty}^{+\infty} h_i u_{k-i} = h_k * u_k \quad (2.2)$$

qui représente la convolution discrète [3, 39, 38] de la séquence d'entrée $\{u_k\}$ et de la séquence $\{h_k\}$ constituée par les coefficients (h_0, h_1, \dots, h_n) représentant la réponse impulsionnelle du filtre.

On peut citer deux propriétés importantes de la réponse impulsionnelle :

$$h_k = 0 \quad \text{pour} \quad k < 0 \quad (2.3)$$

Cette propriété exprime la condition de causalité du filtre numérique [3, 23].

$$\lim_{k \rightarrow \infty} h_k = 0 \quad (2.4)$$

Cette propriété exprime la condition de stabilité du filtre numérique [3, 23].

2.3.2 Analyse fréquentielle

L'étude du comportement fréquentiel d'un filtre numérique se fait grâce à l'utilisation de la transformée en Z [18, 39, 28], qui permet dans le cas des systèmes discrets, le passage du domaine temporel au domaine fréquentiel.

$$y_k \quad \underline{Z\{y_k\}} \quad Y(z)$$

où z représente une variable complexe appartenant à la région de convergence de la fonction considérée.

Fonction de transfert

La fonction de transfert $H(z)$ d'un filtre numérique représente la transformée en Z de sa réponse impulsionnelle h_k

$$H(z) = Z\{h_k\} \quad (2.5)$$

En utilisant la correspondance entre convolution dans le domaine temporel et fréquentiel, on aura :

$$y_k = h_k * u_k \quad (2.6)$$

$$Y(z) = H(z).U(z) \quad (2.7)$$

la fonction de transfert d'un filtre numérique peut donc s'écrire:

$$H(z) = \frac{Y(z)}{U(z)} \quad (2.8)$$

où $Y(z)$ et $U(z)$ représentent les transformées en Z des séquences de sortie y_k et d'entrée u_k , respectivement.

La fonction de transfert $H(z)$ se présente généralement sous la forme d'un rapport de deux polynômes en z , $A(z)$ et $B(z)$ d'ordres n et m , respectivement.

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1z^{-1} + \dots + b_mz^{-m}}{a_0 + a_1z^{-1} + \dots + a_nz^{-n}} \quad (2.9)$$

généralement, $a_0 = 1$, d'où :

$$H(z) = \frac{\sum_{i=0}^m b_i z^{-i}}{1 + \sum_{j=1}^n a_j z^{-j}} \quad (2.10)$$

Spectre d'amplitude

Un aspect important de l'étude du comportement fréquentiel d'un filtre numérique est l'étude de sa réponse fréquentielle donnée par l'évaluation du module de sa fonction de transfert $H(z)$ sur le cercle unité.

$$H(f) = H(z) \Big|_{z=e^{j2\pi(\frac{f}{f_e})}} \quad (2.11)$$

où f_e représente la fréquence d'échantillonnage du signal d'entrée.

Pour un filtre numérique, $H(f)$ sera une fonction périodique de période $T = \frac{1}{f_e}$. Le tracé des variations de cette fonction représentant la réponse fréquentielle du filtre se fera généralement sur une seule période (période principale) et toujours par rapport à des fréquences normalisées f_n (figures 2.3 et 2.4)

avec :

$$f_n = \frac{f}{f_e} \quad (2.12)$$

On notera que le spectre d'amplitude d'un filtre sur une période est caractérisé par trois types de bandes de fréquences :

- Les bandes passantes
- Les bandes de transition
- les bandes atténuées

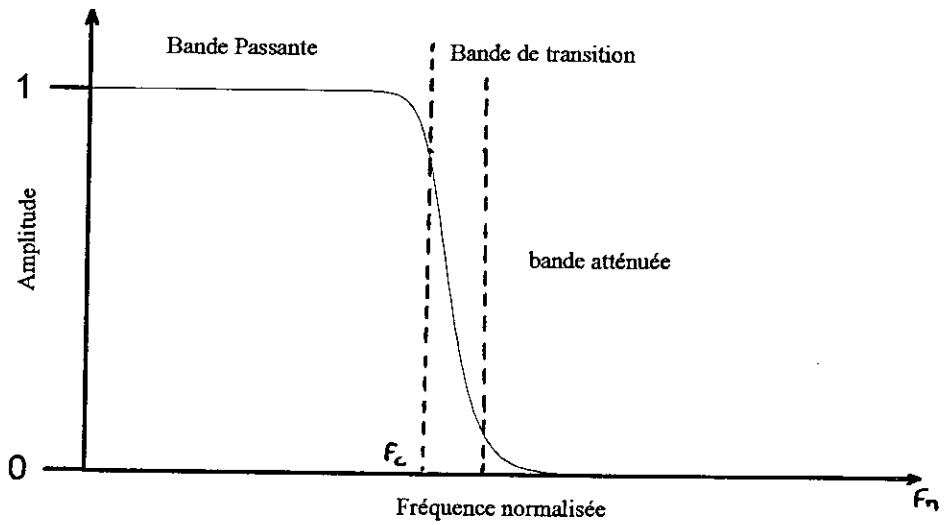


Figure 2.3: Réponse en fréquence d'un filtre numérique passe-bas

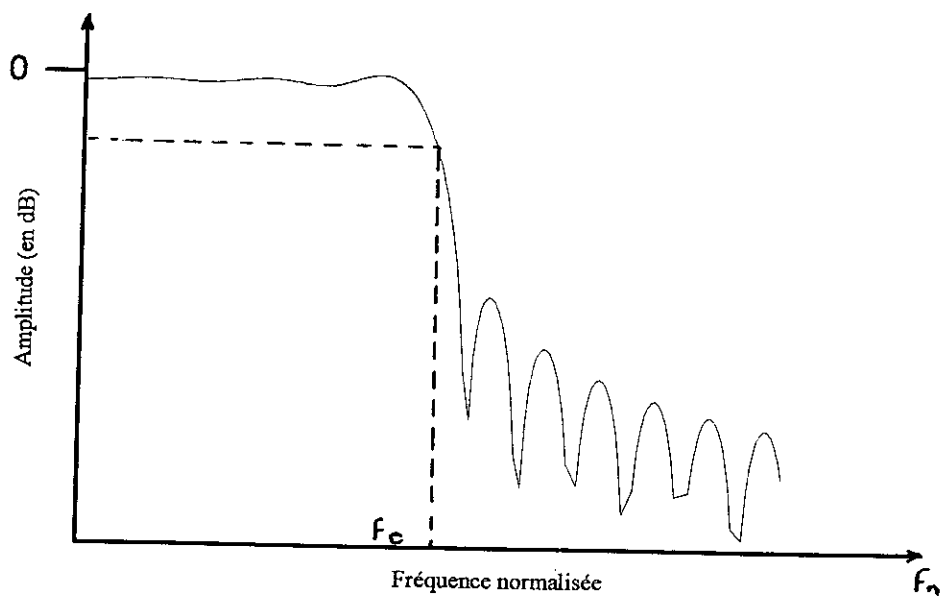


Figure 2.4: Réponse fréquentielle en dB (Spectre d'amplitude) d'un filtre numérique passe-bas

Spectre de phase

Un autre aspect important du comportement fréquentiel d'un filtre numérique est l'observation de sa réponse en phase. Celle-ci est obtenue par l'observation et le tracé de l'argument $\phi(f)$ de sa fonction de transfert évaluée sur le cercle unité.

$$\phi(f) = \text{Arg}(H(z)) \Big|_{z=e^{j2\pi f T_e}} = \text{Arg}(H(f)) \quad (2.13)$$

Une propriété importante recherchée lors de la synthèse de filtres numériques est la linéarité de la phase [8, 10, 14, 29]

$$\phi(\omega) = \alpha\omega + \omega_0 \quad (2.14)$$

où $\omega = 2\pi f$

Cette propriété est particulièrement souhaitée, surtout dans les applications où les distorsions de phases sont critiques comme dans le cas des transmissions de données.

2.4 Stabilité des filtres numériques

Un système physiquement réalisable doit obligatoirement satisfaire à la double condition de causalité et de stabilité.

- La causalité impose que la réponse à une excitation ne peut précéder cette excitation.
- La stabilité impose qu'une séquence d'entrée bornée (finie) générera une séquence de sortie également bornée.

En particulier, lorsque le signal d'entrée est une impulsion, le signal de sortie sera représenté par la réponse impulsionnelle h_k du filtre.

L'expression de la condition de stabilité du filtre numérique dans le domaine temporel s'exprimera par :

$$\lim_{k \rightarrow +\infty} h_k = 0 \quad (2.15)$$

Dans l'espace complexe Z , la fonction de transfert pourra toujours s'écrire sous la forme :

$$H(z) = A \frac{\prod_{i=1}^m (1 - Z_i z^{-1})}{\prod_{i=1}^n (1 - P_i z^{-1})} \quad (2.16)$$

où:

A représente un Gain
 Z_i les "Zéros" de $H(z)$
 P_i les "Pôles" de $H(z)$

le critère de stabilité dans le domaine Z s'exprimera alors par :

$$|P_i| < 1 \quad (2.17)$$

Un filtre numérique sera donc stable si et seulement si tous ses pôles sont situés à l'intérieur du cercle unité.

2.5 Classification des filtres numériques

On peut classer les filtres numériques en plusieurs familles selon le choix parmi de nombreux critères [10, 17].

Pour notre part, on s'intéressera principalement à la famille des filtres numériques causals, qu'on classera suivant deux critères, l'un temporel (réponse impulsionnelle), l'autre fréquentiel (réponse fréquentielle).

On distinguera alors, d'une part, pour le premier critère les deux classes de filtres suivantes :

- Les filtres non-récurrents ou à réponse impulsionnelle finie (RIF).
- Les filtres récurrents ou à réponse impulsionnelle infinie (RII).

et d'autre part, pour le deuxième critère :

- Les filtres passe-bas.

- Les filtres passe-haut.
- Les filtres passe-bandes.
- Les filtres rejeteurs de bandes, ... etc.

2.5.1 Les filtres RIF

Ils sont caractérisés par une réponse impulsionnelle h_k présentant un nombre fini n d'éléments non nuls (h_0, h_1, \dots, h_n) et où n représente l'ordre du filtre.

Les éléments des séquences de sortie $\{Y_k\}$ et d'entrée $\{u_k\}$ sont reliés par une relation simple :

$$y_k = \sum_{i=0}^n b_i u_{k-i} \quad (2.18)$$

où (b_0, b_1, \dots, b_n) sont les coefficients du filtre.

Leur fonction du transfert $H(z)$ est représentée par un polynôme d'ordre n .

$$H(z) = h_0 + h_1 z^{-1} + \dots + h_n z^{-n} = \prod_{i=1}^n (z - Z_i) \quad (2.19)$$

où Z_i sont les "zéros" de $H(z)$. A noter aussi que les coefficients du filtre RIF (b_0, b_1, \dots, b_n) ne sont en fait que les éléments (h_0, h_1, \dots, h_n) de sa réponse impulsionnelle.

$$b_i = h_i \quad \text{pour} \quad i = 0 \dots n \quad (2.20)$$

Les filtres RIF présentent n "pôles" nuls (situés donc à l'intérieur du cercle unité), leur stabilité est donc toujours garantie.

2.5.2 Les filtres R.I.I.

Ils sont caractérisés par l'équation aux différences définie en (2.1) :

Contrairement aux filtres non-récurrents, l'élément de la séquence de sortie du filtre RII à un instant donné dépend de l'élément de la séquence d'entrée lui correspondant et aussi des éléments de la séquence de sortie précédents.

Leur fonction de transfert est donnée par :

$$H(z) = \frac{B(z)}{A(z)} = \frac{\prod_{i=1}^m (z - Z_i)}{\prod_{i=1}^n (z - P_i)} \quad (2.21)$$

où Z_i sont les "zéros" et P_i les "pôles" de $H(z)$.

La stabilité des filtres récurrents RII est conditionnée par la position de ses "pôles" par rapport au cercle unité.

2.5.3 Comparaison entre filtres RIF et RII

Les deux types de filtres sont complémentaires, ils ont chacun leurs avantages et leurs inconvénients, c'est toujours l'application qui détermine le choix de l'un par rapport à l'autre.

Pour les filtres RII, une grande sélectivité peut être atteinte avec un nombre réduit de coefficients, par contre la même sélectivité ne peut être atteinte par un filtre de type RIF qu'avec l'utilisation d'un nombre beaucoup plus important de coefficients.

Ceci constitue un avantage certain pour les filtres RII sachant que le coût du filtre s'élève et sa vitesse de traitement diminue lorsque son ordre augmente.

Les filtres non-récurrents sont utilisés naturellement pour effectuer des intégrations numériques, simuler les équations différentielles, néanmoins leur avantage réside principalement dans leur stabilité quasi-absolue, leur simplicité, leur phase linéaire et la possibilité de réalisation en utilisant la Transformée de Fourier Rapide (TFR ou FFT) [6, 31].

2.6 Synthèse des filtres numériques

La synthèse d'un filtre numérique, c'est la détermination d'un nombre fini n de coefficients qui le caractérisent totalement. Il doit généralement, satisfaire à un certain nombre de contraintes exprimées souvent dans le domaine fréquentiel, exemple: ordre du filtre, fréquence de coupure, largeur de la bande de transition, ... etc. L'ensemble de toutes ces spécifications forment ce qu'on appelle le gabarit du filtre (figure 2.5).

Les méthodes de synthèse des filtres numériques sont nombreuses et très diversifiées, elle vont souvent de la simple résolution d'un système d'équations à

l'utilisation d'algorithmes et de procédés de plus en plus compliqués.

Selon l'approche utilisée lors de la synthèse, on peut grouper ces méthodes en deux classes :

1. Classe des méthodes utilisant l'arsenal des filtres analogiques prototypes [4].
2. Classe des méthodes utilisant directement le domaine Z .

Pour notre part, le travail effectué a été la synthèse de filtres récurrents (R.I.I), directement dans le domaine Z , par des méthodes d'optimisation basées sur le critère des moindres carrés.

Les méthodes propres à la synthèse des filtres non-récurrents (R.I.F) seront cependant brièvement présentées, car ce type de filtres nous a servi lors des étapes d'initialisation de nos programmes et aussi afin de comparer les différentes performances.

2.7 Synthèse des filtres non-récurrents (R.I.F)

2.7.1 Méthode d'échantillonnage fréquentiel

Le principe de cette méthode est de déduire la fonction de transfert $H(z)$ d'un filtre numérique à partir d'un nombre N d'échantillons $\tilde{H}(k)$ régulièrement espacés, prélevés sur la réponse en fréquence $H_d(\omega)$ du filtre désiré [30].

La fonction de transfert $H(z)$ du filtre cherché sera donnée par :

$$H(z) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{H}(k) \frac{1 - z^{-N}}{1 - z^{-1} e^{j2\pi \frac{k}{N}}} \quad (2.22)$$

où :

$$\tilde{H}(k) = H_d(e^{j\theta}) \Big|_{\theta=2\pi \frac{k}{N}} \quad k = 0, 1, \dots, N$$

2.7.2 Méthode de fenêtrage

Cette méthode consiste à prélever, au moyen d'une fenêtre temporelle w_k un nombre fini N d'éléments (h_0, h_1, \dots, h_N) de la réponse impulsionnelle h_k d'un filtre numérique constituant le modèle de base et répondant aux caractéristiques fréquentielles voulues [1, 47].

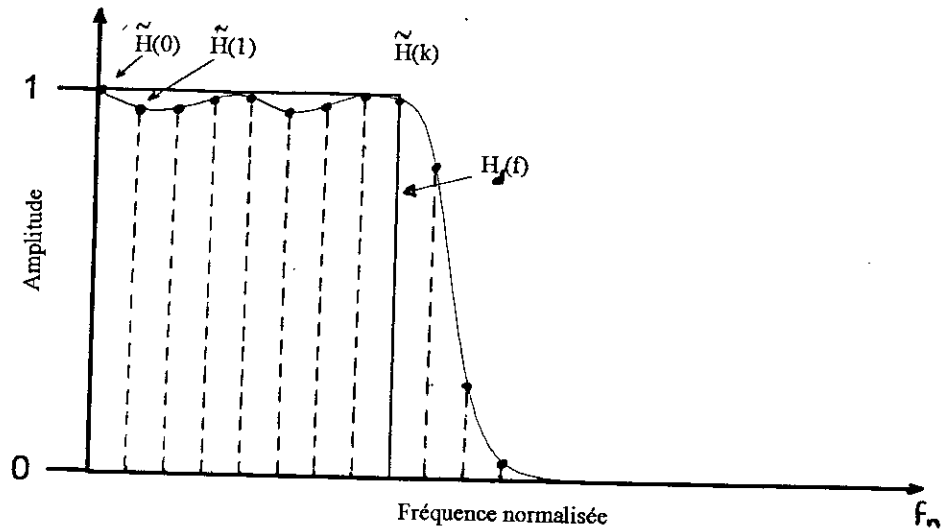


Figure 2.5: Echantillonnage de la réponse fréquentielle d'un filtre passe-bas

La réponse impulsionnelle \tilde{h}_k du filtre (RIF) à synthétiser sera :

$$\tilde{h}_k = h_k \cdot w_k \quad (2.23)$$

et sa réponse fréquentielle est donné par le produit de convolution :

$$\tilde{H}(\omega) = \frac{1}{2\pi} H(\omega) * W(\omega) \quad (2.24)$$

où bien :

$$H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \tilde{H}(\Omega) \cdot W(\omega - \Omega) d\Omega \quad (2.25)$$

La fenêtre la plus simple est certainement la fenêtre rectangulaire qui correspond en fait à une troncature simple de $h(k)$. Son inconvénient majeur est de présenter dans le domaine fréquentiel des perturbations, avec des niveaux relativement importants, connues sous le nom d'oscillations de GIBBS [39].

Dans le but de réduire ce phénomène, plusieurs autres fenêtres ont été développées et utilisées, comme par exemple, la fenêtre triangulaire ou aussi celles de Hanning, Hamming, Blackman, Kaiser, ... etc. [1, 4, 47].

2.8 Synthèse des filtres récurifs (R.I.I)

2.8.1 Méthode d'invariance de la réponse impulsionnelle

Cette méthode permet la synthèse d'un filtre numérique RII à partir d'un filtre analogique répondant aux exigences de la conservation de la réponse impulsionnelle en passant du domaine continu au domaine discret [1, 29, 44, 45].

Donc, ayant un filtre analogique connu, on peut déduire la fonction de transfert $H(z)$ du filtre numérique lui correspondant en suivant la procédure ci-après :

1. Trouver la réponse impulsionnelle $h_A(t)$ du filtre analogique.
2. Remplacer dans l'expression de $h_A(t)$, la variable t par nT , où T est la période d'échantillonnage.
3. Déduire la fonction de transfert $H(z)$ du filtre numérique cherché en déterminant la transformée en Z de $h_A(nT)$.

2.8.2 Méthode de la transformation bilinéaire

Cette méthode offre un autre moyen de synthèse de filtres numériques à partir de filtres analogiques connus [29, 44, 45].

Son principe repose sur l'approximation d'une équation différentielle du premier ordre par une équation aux différences [1].

La relation entre les réponses fréquentielles du filtre analogique original et celle du filtre numérique équivalent est déterminée par les correspondances entre les plans S et Z établies par cette transformation [1].

La procédure de synthèse d'un filtre numérique par cette méthode se résume en :

1. Trouver la fonction de transfert $H(s)$ du filtre analogique ayant les caractéristiques voulues.
2. Remplacer dans l'expression de $H(s)$ trouvée

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

où T représente la période d'échantillonnage.

3. Dédurre alors la fonction de transfert $\hat{H}(z)$ du filtre numérique cherché en faisant :

$$\hat{H}(z) = H(s) \Big|_{s=\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}}$$

2.8.3 Méthode d'optimisation

Lorsque les méthodes directes de synthèse des filtres numériques ne fournissent pas une solution satisfaisante, on s'oriente alors vers des méthodes plus complexes nécessitant des moyens de calculs plus performants.

Le principe des méthodes d'optimisation est la minimisation d'un critère exprimé sur la distance entre les paramètres du filtre cherché et ceux d'un filtre modèle.

Selon le critère adopté, de nombreuses méthodes de synthèse de filtres numériques sont disponibles [13, 21, 29].

Pour notre part et dans le cadre de ce travail, nous avons examiné et comparé les performances de deux méthodes basées sur le critère d'optimisation par les moindres carrés.

1. La méthode des moindres carrés modifiés (MCM).
2. La méthode des moindres carrés vrais (MCV).

Chapitre 3

Synthèse par les Méthodes de Padé et MCM

3.1 Introduction

Les méthodes de synthèse des filtres RII directement dans le domaine numérique sont nombreuses, on peut les classer selon plusieurs critères suivant l'utilisation ou non d'un critère d'optimisation, le type de données utilisées (réponse impulsionnelle ou fonction d'autocorrélation) ou aussi, le type d'équations à résoudre (linéaires ou non linéaires) [21, 46].

Les deux méthodes objet du présent chapitre sont :

- *La méthode de Padé.*
- *La méthode des Moindres Carrés Modifiés (MCM).*

La première appartient au groupe de méthodes qui n'utilisent aucun critère d'optimisation, la synthèse de filtres numériques par cette méthode s'effectue par la résolution directe de systèmes d'équations linéaires formés par les éléments des réponses impulsionnelles.

La seconde, appartient au groupe de méthodes dites d'optimisation linéaire. Elle utilise le critère des moindres carrés et nécessite des données telles que la réponse impulsionnelle et la fonction d'autocorrélation.

Dans la cas général, le problème de la synthèse d'un filtre RII par ces deux méthodes se pose comme suit :

Ayant un filtre prototype de départ, caractérisé par sa fonction de transfert $H(z)$ de la forme :

$$H(z) = \sum_{i=0}^{\infty} h_i z^{-i} \quad (3.1)$$

Trouver la fonction de transfert $\hat{H}(z)$ du filtre RII de la forme rationnelle suivante :

$$\hat{H}(z) = \frac{\hat{B}(z)}{\hat{A}(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad (3.2)$$

dont les caractéristiques s'approchent le plus de celles du filtre désiré de départ choisi comme prototype.

Pour le cas des méthodes d'optimisation, le problème revient donc à la minimisation d'une fonction de l'écart ϵ_k entre les sorties y_k et \hat{y}_k des deux filtres lorsqu'on applique à leurs entrées le même signal u_k (figure 3.1).

En particulier, pour le cas d'une minimisation par les moindres carrés, cette fonction est quadratique et sa formulation dépend de la méthode utilisée.

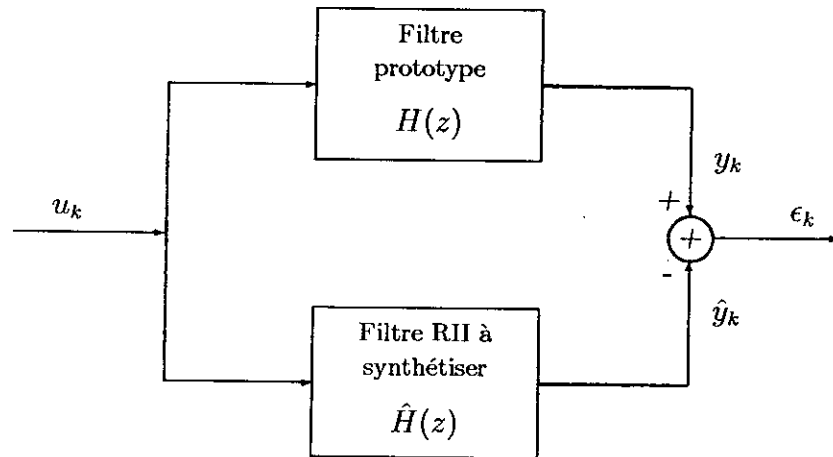


Figure 3.1: Modèle représentant l'erreur vraie

3.2 La méthode de Padé

3.2.1 Principe

Cette méthode utilise les éléments des réponses impulsionnelles h_k et \hat{h}_k du filtre prototype et du filtre RII à synthétiser, respectivement.

Le principe de cette méthode consiste à supposer que les $N = n + m + 1$ premiers éléments des réponses impulsionnelles h_k et \hat{h}_k sont égaux [12], i.e.

$$\hat{h}_k = h_k \quad k = 0, \dots, n + m \quad (3.3)$$

L'équation (3.2) donne :

$$\hat{B}(z) = \hat{A}(z)\hat{H}(z) \quad (3.4)$$

et dans le domaine temporel, celle-ci est équivalente au produit de convolution :

$$b_k = a_k * \hat{h}_k \quad k = 0, \dots, +\infty \quad (3.5)$$

en considérant seulement les $n + m + 1$ premières équations et en tenant compte de l'équation 3.3, on a :

$$b_k = \sum_{i=0}^n a_i h_{k-i} \quad k = 0, \dots, n + m \quad (3.6)$$

qui conduit après décomposition, au système d'équations linéaires suivant :

$$\sum_{i=0}^n a_i h_{k-i} = b_k \quad k = 0, \dots, m \quad (3.7)$$

$$\sum_{i=0}^n a_i h_{k-i} = 0 \quad k = m + 1, \dots, m + n \quad (3.8)$$

avec

$$h_{k-i} = 0 \quad \text{pour} \quad k < i$$

qui peut aussi s'écrire sous forme matricielle :

3.2 La méthode de Padé

3.2.1 Principe

Cette méthode utilise les éléments des réponses impulsionnelles h_k et \hat{h}_k du filtre prototype et du filtre RII à synthétiser, respectivement.

Le principe de cette méthode consiste à supposer que les $N = n + m + 1$ premiers éléments des réponses impulsionnelles h_k et \hat{h}_k sont égaux [12], i.e.

$$\hat{h}_k = h_k \quad k = 0, \dots, n + m \quad (3.3)$$

L'équation (3.2) donne :

$$\hat{B}(z) = \hat{A}(z)\hat{H}(z) \quad (3.4)$$

et dans le domaine temporel, celle-ci est équivalente au produit de convolution :

$$b_k = a_k * \hat{h}_k \quad k = 0, \dots, +\infty \quad (3.5)$$

en considérant seulement les $n + m + 1$ premières équations et en tenant compte de l'équation 3.3, on a :

$$b_k = \sum_{i=0}^n a_i h_{k-i} \quad k = 0, \dots, n + m \quad (3.6)$$

qui conduit après décomposition, au système d'équations linéaires suivant :

$$\sum_{i=0}^n a_i h_{k-i} = b_k \quad k = 0, \dots, m \quad (3.7)$$

$$\sum_{i=0}^n a_i h_{k-i} = 0 \quad k = m + 1, \dots, m + n \quad (3.8)$$

avec

$$h_{k-i} = 0 \quad \text{pour} \quad k < i$$

qui peut aussi s'écrire sous forme matricielle :

$$\begin{pmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{pmatrix} \begin{pmatrix} 1 \\ \tilde{\mathbf{a}} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \quad (3.9)$$

où

$$\tilde{\mathbf{a}} = (a_1, \dots, a_n)^t \quad (3.10)$$

$$\mathbf{b} = (b_0, \dots, b_m)^t \quad (3.11)$$

$\mathbf{0}$ est le vecteur nul de dimension $n - m$.

$$\mathbf{H}_0 = \begin{pmatrix} h_0 & 0 & \dots & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ h_m & h_{m-1} & \dots & h_0 & \dots & 0 \end{pmatrix} \quad (3.12)$$

et

$$\mathbf{H}_1 = \begin{pmatrix} h_{m+1} & h_m & \dots & h_0 & \dots & 0 \\ h_{m+2} & h_{m+1} & \dots & h_1 & \dots & 0 \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ h_{m+n} & h_{m+n-1} & \dots & h_{n-1} & \dots & h_m \end{pmatrix} \quad (3.13)$$

ce système peut être finalement décomposé en :

$$\mathbf{H}_0 \mathbf{a} = \mathbf{b} \quad (3.14)$$

$$\tilde{\mathbf{H}}_1 \tilde{\mathbf{a}} = -\mathbf{h} \quad (3.15)$$

où

$$\mathbf{a} = (1, \tilde{\mathbf{a}})^t = (a_0, \dots, a_n)^t \quad (3.16)$$

$$\mathbf{h} = (h_{m+1}, h_{m+2}, \dots, h_{m+n})^t \quad (3.17)$$

et

$$\tilde{\mathbf{H}}_1 = \begin{pmatrix} h_m & \dots & h_0 & \dots & 0 \\ h_{m+1} & \dots & h_1 & \dots & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ h_{m+n-1} & \dots & h_{n-1} & \dots & h_m \end{pmatrix} \quad (3.18)$$

Le nombre d'équations du système est $n+m+1$ qui est égal au nombre d'inconnues constituant les vecteurs $\tilde{\mathbf{a}}$ et \mathbf{b} . La résolution du système se fait directement sans aucune procédure d'optimisation.

Remarques

Concernant les performances de cette méthode, dont l'avantage principal est la simplicité, les remarques suivantes s'imposent :

- Les résultats obtenus ne sont pas toujours satisfaisants du fait principalement que seuls $n + m + 1$ éléments de la réponse impulsionnelle sont pris en considération.
- La stabilité des filtres synthétisés n'est pas garantie.
- Les performances de la méthode dépendent étroitement du nombre d'éléments de la réponse impulsionnelle utilisés, celle-ci est meilleure lorsque $m + n + 1$ augmente, c'est à dire lorsque l'ordre du filtre $\hat{H}(z)$ est élevé.

On notera par ailleurs, l'existence d'une variante de cette méthode dite Méthode de Padé Etendue [26] qui utilise un nombre d'équations supérieur au nombres d'inconnues. Dans ce cas, la détermination des coefficients du filtre $\hat{H}(z)$ se fait en minimisant un critère des moindres carrés.

3.2.2 Procédure de synthèse

Pour synthétiser un filtre numérique RII par la méthode de Padé, on suit généralement la procédure donnée ci-après :

Ayant pour données de départ la réponse fréquentielle du filtre désiré donnée généralement sous forme idéale ou aussi sous la forme d'un gabarit, on doit trouver les coefficients (a_1, \dots, a_n) et (b_0, \dots, b_m) caractérisant le filtre $\hat{H}(z)$. Pour ce faire, on procède comme suit :

1. Trouver la réponse impulsionnelle du filtre désiré.
2. Sélectionner, au moyen d'une fenêtre temporelle (rectangulaire, triangulaire, de Hamming, de Ham, de Kaiser, ... etc), les $N = n + m + 1$ éléments de la réponse impulsionnelle[4].
3. Effectuer un décalage vers la droite de ces éléments de la réponse impulsionnelle afin de satisfaire la condition de causalité du filtre désiré.
4. Former le système d'équations donné par (3.14) et (3.15).
5. Résoudre le système en calculant les éléments du vecteur \mathbf{a} à partir de l'équation (3.15) puis utiliser ces résultats afin de trouver les éléments du vecteur \mathbf{b} en résolvant l'équation (3.14).

3.3 Méthode des Moindres Carrés Modifiés

3.3.1 Principe

Le principe de synthèse des filtres numériques par la méthode des moindres carrés modifiés (MCM) est le même que celui de la méthode de Padé décrite précédemment. Toutefois, la méthode des moindres carrés modifiés utilise, en plus de la réponse impulsionnelle la densité spectrale ou bien la fonction d'autocorrélation [36, 41].

Par contre, les différences apparaissent essentiellement au niveau de la nature et du nombre d'équations utilisées lors de la synthèse.

En effet, la détermination des coefficients du filtre (b_0, \dots, b_m) et (a_0, \dots, a_n) par la méthode MCM se fait en considérant la totalité des éléments de la réponse impulsionnelle h à laquelle s'ajoute l'information du second ordre. Ceci étant, le nombre d'équations sera infini et la résolution de ces systèmes d'équations où les inconnues sont les coefficients a_i et b_i nécessitera l'emploi d'un critère d'optimisation, celui des Moindres Carrés [24] pour le cas qui nous intéresse.

3.3.2 Procédure de synthèse

Lorsqu'on applique à l'entrée d'un filtre numérique un bruit blanc de moyenne nulle et de variance unité ($\sigma^2 = 1$), la densité spectrale de la sortie correspondante est égale au carré du module de la réponse fréquentielle du filtre.

Donc, si on considère un filtre numérique stable caractérisé par sa réponse impulsionnelle h_k et sa fonction de transfert $H(z)$ et si u_k est une séquence décorrélée, de

moyenne nulle et de variance $\sigma^2 = 1$ appliquée à l'entrée du filtre alors, la séquence de sortie y_k correspondante est donnée par :

$$y_k = \sum_{i=0}^{\infty} h_i u_{k-i} \quad (3.19)$$

et les éléments de la fonction d'autocorrélation r_k , représentant l'information du second ordre du filtre est donnée par :

$$\begin{aligned} r_k &= E(y_n, y_{n+k}) \\ &= \frac{1}{2\pi} \int_{-\pi}^{+\pi} |H(e^{j\omega})|^2 e^{j\omega k} d\omega \\ &= \sum_{i=0}^{\infty} h_i h_{k-i} \end{aligned} \quad (3.20)$$

il est important de noter que

1. $r_k = r_{-k}$
2. Les séquences h_k et r_k sont, respectivement, les coefficients de Fourier de $H(e^{j\omega})$ et de $|H(e^{j\omega})|^2$ d'où l'appellation d'information du premier ordre pour h_k et d'information de second ordre pour r_k .

Enfin, le problème de synthèse d'un filtre numérique par la méthode des moindres carrés modifiés MCM s'énonce de la manière suivante :

Connaissant les éléments de la réponse impulsionnelle d'un filtre désiré donné, trouver les coefficients a_i et b_i du filtre RII caractérisé par sa fonction de transfert de la forme $\hat{H}(z) = \hat{B}(z)/\hat{A}(z)$ qui minimisent le critère suivant :

$$\|h - \hat{h}\|^2 = \sum_{k=0}^{\infty} (h_k - \hat{h}_k)^2 \quad (3.21)$$

$$= \frac{1}{2\pi} \int_{-\pi}^{+\pi} |H(e^{j\omega}) - \hat{H}(e^{j\omega})|^2 d\omega \quad (3.22)$$

$$= E(y_k - \hat{y}_k)^2 \quad (3.23)$$

$$= E(\epsilon_k)^2 \quad (3.24)$$

où y_k et \hat{y}_k sont respectivement, les sorties des filtres $H(z)$ et $\hat{H}(z)$ lorsqu'on applique à leurs entrées le même bruit blanc u_n de moyenne nulle et de variance unité (figure 3.1).

Malheureusement, le problème de minimisation de la fonction (3.24) conduit à un système d'équations non linéaires dont la résolution nécessite le recours aux méthodes de programmation non linéaires.

Afin d'éviter cela, une modification de la fonction (3.24) s'impose et permet de linéariser le problème. Cette modification consiste à considérer la minimisation d'une version modifiée de l'erreur ϵ'_k (figure 3.2).

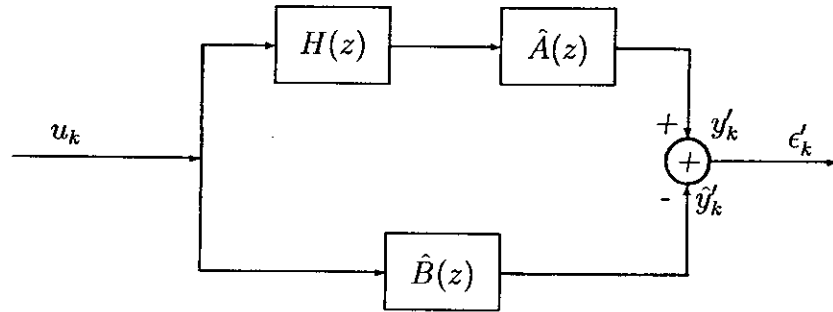


Figure 3.2: Modèle de l'erreur pour la méthode des moindres carrés modifiés (MCM)

Dans ce cas, l'erreur vraie $E(z)$ est :

$$E(z) = H(z) - \hat{H}(z) \quad (3.25)$$

$$= H(z) - \frac{\hat{B}(z)}{\hat{A}(z)} \quad (3.26)$$

et sa version modifiée $E'(z)$ s'écrira

$$E(z)\hat{A}(z) = H(z)\hat{A}(z) - \hat{B}(z) \quad (3.27)$$

$$E'(z) = H(z)\hat{A}(z) - \hat{B}(z) \quad (3.28)$$

Donc, pour la synthèse de filtre numérique par la méthode de moindres carrés modifiés, la fonction à minimiser est une version pondérée de l'erreur vraie.

$$E(\epsilon'_k)^2 = \frac{1}{2\pi} \int_{-\pi}^{+\pi} |H(e^{j\omega})\hat{A}(e^{j\omega}) - \hat{B}(e^{j\omega})|^2 d\omega \quad (3.29)$$

$$= \frac{1}{2\pi} \int_{-\pi}^{+\pi} |\hat{A}(e^{j\omega}) (H(e^{j\omega}) - \hat{H}(e^{j\omega}))|^2 d\omega \quad (3.30)$$

En passant au domaine temporel et d'après l'équation (3.28) l'erreur modifiée s'écrit :

$$\epsilon'_k = y'_k - \hat{y}'_k \quad (3.31)$$

et son énergie dans le domaine temporel est :

$$E(\epsilon'_k)^2 = \sum_{k=0}^{\infty} (\epsilon'_k)^2 = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{\infty} a_j h_{k-j} - b_k \right)^2 \quad (3.32)$$

où

$$\epsilon'_k = h_k * a_k - b_k \quad (3.33)$$

En considérant que $h_k = 0$ pour $k < 0$ alors :

$$E(\epsilon'_k)^2 = \sum_{k=0}^m b_k^2 - 2 \sum_{k=0}^m b_k \sum_{j=0}^n a_j h_{k-j} + \sum_{i,j=0}^n a_i a_j r_{|i-j|} \quad (3.34)$$

où r_k représentent les éléments de la fonction d'autocorrélation définis par l'équation (3.20).

La minimisation de ce critère par rapport à (b_0, b_1, \dots, b_m) donne :

$$b_k = \sum_{j=0}^n a_j h_{k-j} \quad 0 \leq k \leq m \quad (3.35)$$

en remplaçant dans l'équation (3.34), on obtient :

$$E(\epsilon'_k)^2 = \sum_{k=0}^m \left(\sum_{j=0}^n a_j h_{k-j} \right)^2 - 2 \sum_{k=0}^m \left(\sum_{j=0}^n a_j h_{k-j} \right) \left(\sum_{j=0}^n a_j h_{k-j} \right) + \sum_{i,j=0}^n a_i a_j r_{|i-j|} \quad (3.36)$$

ce qui donne finalement

$$E(\epsilon'_k)^2 = \sum_{i,j=0}^n a_i a_j r_{|i-j|} - \sum_{k=0}^m \left(\sum_{j=0}^n a_j h_{k-j} \right)^2 \quad (3.37)$$

cette dernière équation s'écrit sous la forme matricielle :

$$E(\epsilon'_k)^2 = (a_n, a_{n-1}, \dots, a_0) \mathbf{K} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix} \quad (3.38)$$

où \mathbf{K} est une matrice de dimensions $(n+1) \times (n+1)$ dont les éléments sont définis par l'expression suivante [36] :

$$k_{ij} = r_{|i-j|} - \sum_{k=0}^m h_{k-i} h_{k-j} \quad i, j = 0, \dots, n \quad (3.39)$$

sachant que $h_k = 0$ pour $k < 0$, la matrice \mathbf{K} s'écrit :

$$\mathbf{K} = \mathbf{R} - \mathbf{H}_0^t \mathbf{H}_0 \quad (3.40)$$

où

$$\mathbf{H}_0 = \begin{pmatrix} h_0 & 0 & \dots & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ h_m & h_{m-1} & \dots & h_0 & \dots & 0 \end{pmatrix} \quad (3.41)$$

et \mathbf{R} , une matrice de dimensions $(n+1) \times (n+1)$ de forme Toeplitz définie par :

$$\mathbf{R} = \begin{pmatrix} r_0 & r_1 & \dots & r_n \\ r_1 & r_0 & \dots & r_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_n & r_{n-1} & \dots & r_0 \end{pmatrix} \quad (3.42)$$

l'équation (3.38) peut être aussi écrite sous la forme :

$$\alpha_n = (a_0, a_1, \dots, a_n) \mathbf{J} \mathbf{K} \mathbf{J} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} \quad (3.43)$$

où \mathbf{J} est une matrice de permutation de la forme

$$\mathbf{J} = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \dots & 0 & 0 \end{pmatrix}$$

en considérant aussi que :

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = (a_0, a_1, \dots, a_n)^t$$

on aura alors

$$\alpha_n = \mathbf{a}^t \mathbf{J} \mathbf{K} \mathbf{J} \mathbf{a} \quad (3.44)$$

En remplaçant \mathbf{K} par son expression donnée par (3.40) et en tenant compte de sa structure l'équation précédente devient :

$$E(\epsilon'_k)^2 = \mathbf{a}^t (\mathbf{R} - \mathbf{H}_0^t \mathbf{H}_0) \mathbf{a} \quad (3.45)$$

La solution du problème de minimisation de $E(\epsilon'_k)^2$ sous la contrainte $a_0 = 1$ est obtenue en résolvant le système d'équations suivant :

$$(\mathbf{R} - \mathbf{H}_0^t \mathbf{H}_0) \mathbf{a} = \begin{pmatrix} \alpha_n \\ 0 \end{pmatrix} \quad (3.46)$$

pour \mathbf{a} et α_n , où α_n représente l'énergie minimale de l'erreur (variance) $E(\epsilon'_k)$.

Connaissant les éléments du vecteur \mathbf{a} , les coefficients (b_0, b_1, \dots, b_m) éléments du vecteur \mathbf{b} sont déduits à partir de :

$$\mathbf{H}_0 \mathbf{a} = \mathbf{b} \quad (3.47)$$

finale^{ment}, les équations (3.46) et (3.47) peuvent être regroupées pour former un seul système d'équations

$$\begin{pmatrix} \mathbf{R} & -\mathbf{H}_0^t \\ -\mathbf{H}_0 & \mathbf{I}_{(m+1)} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = \begin{pmatrix} \alpha_n \\ \mathbf{0} \end{pmatrix} \quad (3.48)$$

A noter que du fait que ce système est bien structuré, sa résolution peut se faire de manière rapide et efficace à l'aide de l'algorithme de Mullis-Roberts donné en [36] et en Annexe C.

Algorithme de synthèse

Pour synthétiser un filtre numérique RII par la méthode des moindres carrés modifiés (MCM), la procédure à suivre est résumée en ces points :

1. Calculer les éléments (h_0, h_1, \dots, h_n) de la réponse impulsionnelle du filtre désiré où n représente l'ordre du filtre RII à synthétiser.
2. Calculer les éléments (r_0, r_1, \dots, r_n) de la fonction d'autocorrélation en utilisant l'équation (3.20).
3. Former le système d'équations (3.48).
4. Utiliser l'algorithme de Mullis-Roberts donné en Annexe C pour résoudre ce système et déterminer les coefficients (a_1, \dots, a_n) et (b_0, b_1, \dots, b_n) qui caractérisent le filtre RII à synthétiser.

Remarques

- Le résultat de la minimisation de l'équation 3.34 donné par l'équation 3.35, exprime l'égalité des m premiers éléments des réponses impulsionnelles du filtre désiré et du filtre à synthétiser.

$$h_k = \hat{h}_k \quad \text{pour } 0 \leq k \leq m$$

- Les filtres synthétisés par la méthode MCM sont toujours stables [36].
- L'algorithme de Mullis-Roberts considère le cas particulier où $m = n$.

Chapitre 4

La Méthode des Moindres Carrés Vrais (MCV)

4.1 Introduction

La méthode des moindres carrés vrais (MCV), objet de notre étude, est en fait un cas particulier d'une méthode plus générale, celle des moindres carrés pondérés (MCP) [21, 29].

Cette technique a été introduite initialement par Steiglitz et McBride pour l'identification des systèmes en automatique [42].

Dans ce qui suivra, nous considérerons d'abord la solution du problème de synthèse des filtres numériques RII par la méthode MCP et à partir de cela, nous déduirons la solution particulière propre au cas des moindres carrés vrais.

4.2 Développement théorique du cas général MCP

4.2.1 Etude et résolution du problème

Considérons le problème de minimisation de la fonction suivante:

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} |W(e^{j\theta})|^2 |H(e^{j\theta})\hat{A}(e^{j\theta}) - \hat{B}(e^{j\theta})|^2 d\theta \quad (4.1)$$

qui représente l'énergie de l'erreur entre un filtre désiré connu $H(z)$ et son ap-

proximation caractérisée par sa fonction de transfert $\hat{H}(z)$ telle que:

$$\hat{H}(z) = \frac{\hat{B}(z)}{\hat{A}(z)} \quad (4.2)$$

où $\hat{B}(z)$ et $\hat{A}(z)$ sont des polynômes en z d'ordre n avec :

$$\mathbf{a} = (1, a_1, \dots, a_n)^t \quad (4.3)$$

$$\mathbf{b} = (b_0, b_1, \dots, b_n)^t \quad (4.4)$$

qui représentent les vecteurs dont les éléments sont les coefficients de $\hat{A}(z)$ et $\hat{B}(z)$ respectivement et $W(z)$ une fonction de pondération quelconque, qu'on pourra supposer de la forme :

$$W(z) = \frac{C(z)}{D(z)} = \frac{c_0 + c_1 z^{-1} + \dots + c_{n_c} z^{-n_c}}{d_0 + d_1 z^{-1} + \dots + d_{n_d} z^{-n_d}} \quad (4.5)$$

avec $d_0 = 1$ et $n_c \leq n_d$ afin de satisfaire la condition de causalité.

$W(z)$ peut être considérée soit comme une fonction de pondération qui donne plus ou moins de poids à certaines composantes de l'erreur ou bien aussi comme la fonction de transfert d'un préfiltre tel que présenté sur la figure 4.1.

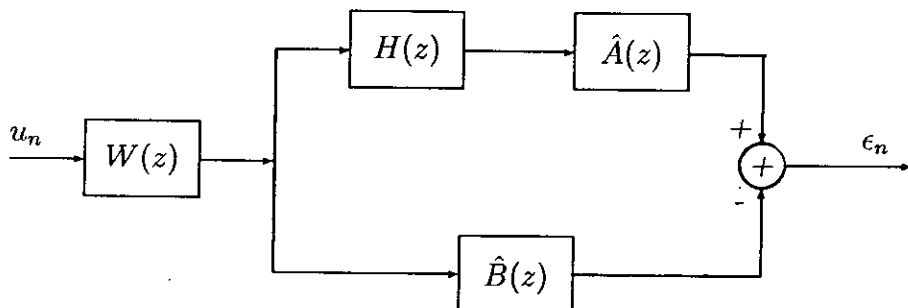


Figure 4.1: Modèle de l'erreur pondérée

A noter ici les deux expressions de $V(\mathbf{a}, \mathbf{b})$ relatives aux deux cas particuliers de la fonction $W(z)$

1. pour $W(z) = 1$

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} |H(e^{j\theta})\hat{A}(e^{j\theta}) - \hat{B}(e^{j\theta})|^2 d\theta$$

qui correspond au critère de la méthode des Moindres Carrés Modifiés (MCM).

2. pour $W(z) = \frac{1}{\hat{A}(z)}$

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \frac{1}{\hat{A}(e^{j\theta})} \right|^2 |H(e^{j\theta})\hat{A}(e^{j\theta}) - \hat{B}(e^{j\theta})|^2 d\theta$$

qui correspond au critère de la méthode des Moindres Carrés Vrais (MCV).

en conséquence, l'erreur générale E_g pour la méthodes MCP s'écrira :

$$E_g(e^{j\theta}) = W(e^{j\theta})E(e^{j\theta}) \quad (4.6)$$

où $E(e^{j\theta})$ représente l'erreur modifiée vue au chapitre 3 (MCM).

$$E(e^{j\theta}) = H(e^{j\theta})\hat{A}(e^{j\theta}) - \hat{B}(e^{j\theta}) \quad (4.7)$$

donc, la fonction $V(\mathbf{a}, \mathbf{b})$ s'écrira :

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} |E_g(e^{j\theta})|^2 d\theta \quad (4.8)$$

et le problème à résoudre sera donc :

Trouver pour une fonction $W(z)$ donnée et $a_0 = 1$, les vecteurs \mathbf{a} et \mathbf{b} définis en (4.3) et (4.4) qui minimisent la fonction $V(\mathbf{a}, \mathbf{b})$ donnée en (4.8) et représentant l'erreur quadratique pondérée.

D'après l'égalité de Parseval, ce problème de minimisation exprimé dans le domaine fréquentiel, peut être considéré et résolu dans le domaine temporel [40].

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} |E_g(e^{j\theta})|^2 d\theta = \sum_{k=0}^{\infty} (e_g)_k^2 \quad (4.9)$$

où $(e_g)_k$ est l'erreur MCP dont l'expression est donnée dans le domaine temporel par :

$$e_g = w_k * e_k = (w_k * (h_k * a_k - b_k)) \quad (4.10)$$

où w et h représentent, respectivement, les réponses impulsionnelles causales de $W(z)$ et $H(z)$ et e_k l'erreur MCM donnée par :

$$e_k = h_k * a_k - b_k \quad (4.11)$$

Le développement de ce produit de convolution conduit à la forme matricielle suivante :

$$\begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ \vdots \\ e_n \\ \vdots \end{pmatrix} = \begin{pmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_n & & \dots & h_0 \\ h_{n+1} & h_n & \dots & h_1 \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} 1 \\ a_0 \\ \vdots \\ a_n \end{pmatrix} - \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.12)$$

ou sous forme condensée :

$$\mathbf{e} = \mathbf{H}\mathbf{a} - \tilde{\mathbf{b}} \quad (4.13)$$

On peut également représenter l'erreur MCP sous forme matricielle en développant le produit de convolution

$$e_g = w_k * e_k \quad (4.14)$$

ce qui donne :

$$\begin{pmatrix} e_{g0} \\ e_{g1} \\ \vdots \\ e_{gn} \\ \vdots \end{pmatrix} = \begin{pmatrix} w_0 & 0 & \dots & 0 & \dots \\ w_1 & w_0 & \dots & 0 & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ w_n & w_{n-1} & \dots & w_0 & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \\ \vdots \end{pmatrix} \quad (4.15)$$

qu'on écrira aussi sous forme condensée

$$\mathbf{e}_g = \mathbf{W}^t \mathbf{e} \quad (4.16)$$

où

$$\mathbf{W} = \begin{pmatrix} w_0 & w_1 & \dots & w_n & \dots \\ 0 & w_0 & \dots & w_{n-1} & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ 0 & 0 & \dots & w_0 & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

La solution du problème de la minimisation de la fonction dans le domaine temporel est donnée par :

$$V(\mathbf{a}, \mathbf{b}) = \sum_{k=0}^{\infty} (e_g)_k^2 = \mathbf{e}_g^t \cdot \mathbf{e}_g \quad (4.17)$$

En remplaçant \mathbf{e}_g par son expression donnée par la combinaison de (4.13) et (4.16), on a :

$$V(\mathbf{a}, \mathbf{b}) = [\mathbf{W}^t(\mathbf{H}\mathbf{a} - \tilde{\mathbf{b}})]^t [\mathbf{W}^t(\mathbf{H}\mathbf{a} - \tilde{\mathbf{b}})] \quad (4.18)$$

qui donne après développement :

$$V(\mathbf{a}, \mathbf{b}) = (\mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{W}^t \tilde{\mathbf{b}})^t (\mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{W}^t \tilde{\mathbf{b}}) \quad (4.19)$$

en posant $\mathbf{W}^t \tilde{\mathbf{b}} = \mathbf{S}^t \mathbf{b}$ où \mathbf{S}^t est une matrice de dimensions $\infty \times (n+1)$ de la forme :

$$\mathbf{S}^t = \begin{pmatrix} w_0 & 0 & \dots & 0 \\ w_1 & w_0 & 0 & \ddots & \vdots \\ w_2 & w_1 & w_0 & \ddots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ w_n & w_{n-1} & \dots & \dots & w_0 \\ w_{n+1} & w_n & \dots & \dots & w_1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \end{pmatrix} \quad (4.20)$$

l'équation (4.19) devient alors :

$$V(\mathbf{a}, \mathbf{b}) = (\mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{S}^t \mathbf{b})^t (\mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{S}^t \mathbf{b}) \quad (4.21)$$

on peut écrire également

$$(\mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{S}^t \mathbf{b})^t = (\mathbf{a}^t \quad \mathbf{b}^t) \begin{pmatrix} \mathbf{H}^t \mathbf{W} \\ -\mathbf{S} \end{pmatrix} \quad (4.22)$$

$$(\mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{S}^t \mathbf{b}) = (\mathbf{W}^t \mathbf{H} \quad -\mathbf{S}^t) \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \quad (4.23)$$

et en remplaçant dans l'équation (4.21), celle-ci devient :

$$V(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^t \quad \mathbf{b}^t) \begin{pmatrix} \mathbf{H}^t \mathbf{W} \\ -\mathbf{S} \end{pmatrix} (\mathbf{W}^t \mathbf{H} \quad -\mathbf{S}^t) \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \quad (4.24)$$

d'où l'écriture finale, sous forme matricielle, de la fonction $V(\mathbf{a}, \mathbf{b})$ à minimiser

$$V(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^t \quad \mathbf{b}^t) \begin{pmatrix} \mathbf{H}^t \mathbf{W} \mathbf{W}^t \mathbf{H} & -\mathbf{H}^t \mathbf{W} \mathbf{S}^t \\ -\mathbf{S} \mathbf{W}^t \mathbf{H} & \mathbf{S} \mathbf{S}^t \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \quad (4.25)$$

les vecteurs \mathbf{a} et \mathbf{b} , solutions du problème de minimisation de la fonction $V(\mathbf{a}, \mathbf{b})$, sous la contrainte $a_0 = 1$, sont donnés par la résolution du système :

$$\begin{pmatrix} \mathbf{H}^t \mathbf{W} \mathbf{W}^t \mathbf{H} & -\mathbf{H}^t \mathbf{W} \mathbf{S}^t \\ -\mathbf{S} \mathbf{W}^t \mathbf{H} & \mathbf{S} \mathbf{S}^t \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = \begin{pmatrix} \alpha_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha_n \\ \mathbf{0} \end{pmatrix} \quad (4.26)$$

où α_n est la variance minimale de l'erreur, c'est à dire :

$$\alpha_n = [V(\mathbf{a}, \mathbf{b})]_{\min} \quad (4.27)$$

Les vecteurs solutions \mathbf{a} et \mathbf{b} sont alors obtenus en découplant le système d'équations précédent, ce qui donne :

$$\mathbf{H}^t \mathbf{W} \mathbf{W}^t \mathbf{H} \mathbf{a} - \mathbf{H}^t \mathbf{W} \mathbf{S}^t \mathbf{b} = \begin{pmatrix} \alpha_n \\ \mathbf{0} \end{pmatrix} \quad (4.28)$$

$$- \mathbf{S} \mathbf{W}^t \mathbf{H} \mathbf{a} + \mathbf{S} \mathbf{S}^t \mathbf{b} = \mathbf{0} \quad (4.29)$$

à partir de (4.29) nous aurons :

$$\mathbf{b} = (\mathbf{S} \mathbf{S}^t)^{-1} \mathbf{S} \mathbf{W}^t \mathbf{H} \mathbf{a} \quad (4.30)$$

et en remplaçant (4.30) dans (4.29), celle-ci donnera:

$$\mathbf{H}^t \left[\mathbf{W} \mathbf{W}^t - \mathbf{W} \mathbf{S}^t (\mathbf{S} \mathbf{S}^t)^{-1} \mathbf{S} \mathbf{W}^t \right] \mathbf{H} \mathbf{a} = \begin{pmatrix} \alpha_n \\ \mathbf{0} \end{pmatrix} \quad (4.31)$$

donc, la résolution du problème se fait en calculant d'abord les éléments du vecteur \mathbf{a} de l'équation (4.31) puis, déduire le vecteur \mathbf{b} à partir de (4.30).

Cependant, on peut déterminer simultanément les deux vecteurs solution \mathbf{a} et \mathbf{b} en résolvant le système d'équations donné par (4.26) pour lequel les éléments $\mathbf{H}^t \mathbf{W} \mathbf{W}^t \mathbf{H}$, $\mathbf{H}^t \mathbf{W} \mathbf{S}^t$ et $\mathbf{S} \mathbf{S}^t$ sont des blocs matrices de dimensions $(n+1) \times (n+1)$ composés par les éléments des fonctions d'autocorrélation et d'intercorrélations de $W(z)$ et $F(z)$ avec:

$$F(z) = H(z)W(z) \quad (4.32)$$

$$\mathbf{H}^t \mathbf{W} \mathbf{W}^t \mathbf{H} = \begin{pmatrix} \rho_0 & \rho_1 & \dots & \rho_n \\ \rho_1 & \rho_0 & \dots & \rho_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_n & \rho_{n-1} & \dots & \rho_0 \end{pmatrix} \quad (4.33)$$

$$\mathbf{H}^t \mathbf{W} \mathbf{S}^t = \begin{pmatrix} \eta_0 & \eta_1 & \dots & \eta_n \\ \eta_{-1} & \eta_0 & \dots & \eta_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{-n} & \eta_{-(n-1)} & \dots & \eta_0 \end{pmatrix} \quad (4.34)$$

$$\mathbf{SW}^t\mathbf{H} = \begin{pmatrix} \eta_0 & \eta_{-1} & \dots & \eta_{-n} \\ \eta_1 & \eta_0 & \dots & \eta_{-(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_n & \eta_{n-1} & \dots & \eta_0 \end{pmatrix} \quad (4.35)$$

$$\mathbf{SS}^t = \begin{pmatrix} q_0 & q_1 & \dots & q_n \\ q_1 & q_0 & \dots & q_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ q_n & q_{n-1} & \dots & q_0 \end{pmatrix} \quad (4.36)$$

où les éléments $\rho_k, \eta_k, \eta_{-k}$ et q_k sont donnés par :

$$\rho_k = \frac{1}{2\pi} \int_{-\pi}^{+\pi} F(e^{j\theta})F(e^{-j\theta})e^{jk\theta} d\theta \quad (4.37)$$

$$\eta_k = \frac{1}{2\pi} \int_{-\pi}^{+\pi} W(e^{j\theta})F(e^{-j\theta})e^{jk\theta} d\theta \quad (4.38)$$

$$\eta_{-k} = \frac{1}{2\pi} \int_{-\pi}^{+\pi} F(e^{j\theta})W(e^{-j\theta})e^{jk\theta} d\theta \quad (4.39)$$

$$q_k = \frac{1}{2\pi} \int_{-\pi}^{+\pi} W(e^{j\theta})W(e^{-j\theta})e^{jk\theta} d\theta \quad (4.40)$$

et l'équation (4.26) s'écrit donc :

$$\begin{pmatrix} \rho_0 & \rho_1 & \dots & \rho_n & -\eta_0 & -\eta_1 & \dots & -\eta_n \\ \rho_1 & \rho_0 & \dots & \rho_{n-1} & -\eta_{-1} & -\eta_0 & \dots & -\eta_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_n & \rho_{n-1} & \dots & \rho_0 & -\eta_{-n} & -\eta_{-(n-1)} & \dots & -\eta_0 \\ -\eta_0 & -\eta_{-1} & \dots & -\eta_{-n} & q_0 & q_1 & \dots & q_n \\ -\eta_1 & -\eta_0 & \dots & -\eta_{-(n-1)} & q_1 & q_0 & \dots & q_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\eta_n & -\eta_{n-1} & \dots & -\eta_0 & q_n & q_{n-1} & \dots & q_0 \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} \alpha_n \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.41)$$

Ce système est résolu en utilisant l'algorithme de Levinson à deux canaux (bi-canal) [40] donné en Annexe D, pour lequel on notera :

1. Pour former les matrices R_k , $k = 0, \dots, n$ utilisées par l'algorithme, on doit au préalable, calculer les éléments $\rho_k, \eta_k, \eta_{-k}$ et q_k définis par les équations (4.38 - 4.40).

2. Le calcul des éléments (a_1, a_2, \dots, a_n) et (b_1, b_2, \dots, b_n) se fera en utilisant la relation suivante :

$$(1 \ b_0 \ b_1 \ \dots \ a_n \ b_n) = (1 \ \gamma_1)(\mathbf{I}_2 \ A_1(n) \ A_2(n) \ \dots \ A_n(n)) \quad (4.42)$$

où γ_1 est un scalaire calculé à partir de

$$(1 \ \gamma_1)\tilde{\alpha}_n = (\alpha_n \ 0)$$

et les $A_i(k)$ $i = 1, \dots, n$ sont des matrices de dimension (2×2) qu'il faudra stocker au fur et à mesure de leur calcul.

4.2.2 Calcul des matrices R_k

La formation des matrices R_k $k = 0, 1, \dots, n$ de la forme :

$$R_k = \begin{pmatrix} \rho_k & -\eta_k \\ -\eta_{-k} & q_k \end{pmatrix}$$

utilisées par l'algorithme de Levinson à deux canaux nécessite la détermination des éléments $\rho_k, \eta_k, \eta_{-k}$ et q_k des fonctions d'autocorrélation et d'intercorrélations de $W(z)$ et $F(z)$.

Ceci est fait par la résolution d'un certain nombre de systèmes d'équations particuliers avec un nombre fini d'opérations en utilisant les algorithmes de Sylvester et de Jury, basés sur le théorème d'Euclide [19, 20] présentés en Annexe E et Annexe F.

a- Calcul des éléments des fonctions d'autocorrélation q_k et ρ_k

En supposant toujours que la fonction de pondération $W(z)$ est telle que donnée en (4.5), le passage au domaine temporel nous permet d'écrire la relation liant les coefficients c_i et d_i

$$\sum_{j=0}^{n_d} d_j w_{i-j} = \begin{cases} c_i & 0 \leq i \leq n_c \\ 0 & i > n_c \end{cases} \quad (4.43)$$

qui s'écrit sous forme matricielle :

$$\begin{pmatrix} w_0 & 0 & \dots & 0 \\ w_1 & w_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_d} & w_{n_d-1} & \dots & w_0 \\ w_{n_d+1} & w_{n_d} & \dots & w_1 \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} 1 \\ d_1 \\ \vdots \\ d_{n_c} \\ \vdots \\ d_{n_d} \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_{n_c} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.44)$$

où sous forme compacte

$$\tilde{\mathbf{S}}\mathbf{d} = \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix} \quad (4.45)$$

On multiplie les deux membres de l'égalité par la matrice de Toeplitz triangulaire supérieure \mathbf{W}

$$\mathbf{W}\tilde{\mathbf{S}}\mathbf{d} = \mathbf{W} \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix} \quad (4.46)$$

ce qui donne après développement :

$$\begin{pmatrix} q_0 & q_1 & \dots & q_{n_d} \\ q_1 & q_0 & \dots & q_{n_d-1} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n_d} & q_{n_d-1} & \dots & q_0 \\ q_{n_d+1} & q_{n_d} & \dots & q_1 \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} 1 \\ d_1 \\ \vdots \\ d_{n_d} \end{pmatrix} = \begin{pmatrix} w_0 & w_1 & \dots & w_{n_c} \\ 0 & w_0 & \dots & w_{n_c-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n_c} \end{pmatrix} \quad (4.47)$$

qu'on écrit sous la forme compacte

$$\begin{pmatrix} \tilde{\mathbf{Q}}_0 \\ \tilde{\mathbf{Q}}_1 \end{pmatrix} \mathbf{d} = \begin{pmatrix} \mathbf{W}_1 \mathbf{c} \\ \mathbf{0} \end{pmatrix} \quad (4.48)$$

où $\tilde{\mathbf{Q}}_0$ et $\tilde{\mathbf{Q}}_1$ sont des blocs de dimensions $(n_d + 1) \times (n_d + 1)$ et $\infty \times (n_d + 1)$ de la matrice $\tilde{\mathbf{Q}} = \mathbf{W}\tilde{\mathbf{S}}$

On constate que le système (4.47) présente un nombre infini d'équations parmi lesquelles on prélève $(n+1)$ suffisantes pour calculer les $(n+1)$ valeurs de q_k utilisées par l'algorithme de Levinson.

Pour le cas $n_d \geq n$, le système à résoudre sera donc constitué par les $n_d + 1$ premières équations de (4.47)

$$\begin{pmatrix} q_0 & q_1 & \dots & q_{n_d} \\ q_1 & q_0 & \dots & q_{n_d-1} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n_d} & q_{n_d-1} & \dots & q_0 \end{pmatrix} \begin{pmatrix} 1 \\ d_1 \\ \vdots \\ d_{n_d} \end{pmatrix} = \begin{pmatrix} w_0 & w_1 & \dots & w_{n_c} \\ 0 & w_0 & \dots & w_{n_c-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{n_c-n_d} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n_c} \end{pmatrix} \quad (4.49)$$

à noter que si $n_d < n$, les éléments q_{n_d+1}, \dots, q_n seront calculés en considérant les $n - n_d - 1$ équations suivantes qui donnent directement :

$$q_k = \sum_{i=0}^{n_c} c_i w_{i-k} - \sum_{i=1}^{n_d} d_i q_{k-i} \quad k = n_d + 1, \dots, n \quad (4.50)$$

Le système (4.49) peut être réorganisé et réécrit comme suit :

$$\mathbf{J}_r \mathbf{q} = \mathbf{z} \quad (4.51)$$

où \mathbf{J}_r est une matrice dite de Jury telle que :

$$\mathbf{J}_r = \begin{pmatrix} 1 & d_1 & \dots & d_{n_d-1} & d_{n_d} \\ d_1 & d_2 & \dots & d_{n_d} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{n_d} & 0 & \dots & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & d_{n_d-1} & \dots & 0 & 1 \end{pmatrix} \quad (4.52)$$

$$\mathbf{q} = (q_0, q_1, \dots, q_{n_d})^t$$

et

$$\mathbf{z} = (z_0, z_1, \dots, z_{n_c})^t$$

qui représente le second terme de l'équation (4.49) dont les éléments z_k sont donnés par:

$$z_k = \sum_{i=k}^{n_c} c_i w_{i-k} \quad k = 0, \dots, n_c \quad (4.53)$$

Finalement, le calcul des éléments de la fonction d'autocorrélation q_k se fait en résolvant le système d'équations donné par (4.51) en utilisant l'algorithme de Jury présenté en Annexe F.

Le calcul des éléments de la fonction d'autocorrélation ρ_k se fera de manière identique en suivant la même procédure.

b- Calcul des éléments des fonctions d'intercorrrelation η_k et η_{-k}

. Le calcul des éléments des fonctions d'intercorrrelation entre $W(z)$ et $F(z)$ se fera aussi par la résolution au moyen de l'algorithme de Sylvester, basé sur le théorème d'Euclide [19, 20] d'un système d'équations particulier connu sous le nom de Système de Sylvester.

Considérons donc les fonctions $W(z)$ définie en (4.5) et $F(z)$ définie par (4.32), qui peut aussi être écrite sous la forme d'un rapport de deux polynômes

$$F(z) = \frac{B(z) C(z)}{A(z) D(z)}$$

d'où

$$F(z) = \frac{T(z)}{S(z)} \quad (4.54)$$

avec :

$T(z)$ un polynôme de degré $n_t = n + n_c$

$S(z)$ un polynôme de degré $n_s = n + n_d$

tels que

$$\begin{aligned} T(z) &= \sum_{i=0}^{n_t} t_i z^{-i} \\ S(z) &= \sum_{i=0}^{n_s} s_i z^{-i} \quad \text{avec } s_0 = 1 \end{aligned}$$

et $n_t \leq n_s$

les coefficients t_i et s_i sont donnés par :

$$t_i = \sum_{j=0}^i b_j c_{i-j} \quad i = 0, \dots, n_t$$

$$s_i = \sum_{j=0}^i a_j d_{i-j} \quad i = 0, \dots, n_s$$

Considérons l'équation (4.45) et multiplions les deux membres de l'égalité par la matrice \mathbf{F} constituée par les éléments f_i de la réponse impulsionnelle de $F(z)$

$$\mathbf{F} = \begin{pmatrix} f_0 & f_1 & f_2 & f_3 & \dots \\ 0 & f_0 & f_1 & f_2 & \dots \\ 0 & 0 & f_0 & f_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

l'équation (4.45) devient :

$$\mathbf{F}\tilde{\mathbf{S}}\mathbf{d} = \mathbf{F} \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix} \quad (4.55)$$

ce qui donne le système d'équations infini suivant :

$$\sum_{i=0}^{n_d} d_i w_{i-k} = \begin{cases} \sum_{i=k}^{n_c} c_i f_{i-k} & k \leq n_c \\ 0 & k > n_c \end{cases} \quad (4.56)$$

d'autre part, en écrivant la relation liant les coefficients t_i et s_i de $F(z)$

$$\sum_{i=0}^{n_s} s_i f_{k-i} = \begin{cases} t_k & k \leq n_t \\ 0 & k > n_t \end{cases} \quad (4.57)$$

on obtient :

$$\begin{pmatrix} f_0 & 0 & \dots & 0 \\ f_1 & f_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_s} & f_{n_s-1} & \dots & f_0 \\ f_{n_s+1} & f_{n_s} & \dots & f_1 \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n_t} \\ \vdots \\ s_{n_s} \end{pmatrix} = \begin{pmatrix} t_0 \\ \vdots \\ t_{n_t} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.58)$$

où sous forme compacte

$$\tilde{\mathbf{F}}\mathbf{s} = \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix} \quad (4.59)$$

On multiplie les deux côtés de l'égalité par la matrice de Toeplitz triangulaire supérieure \mathbf{W}

$$\mathbf{W}\tilde{\mathbf{F}}\mathbf{s} = \mathbf{W} \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix} \quad (4.60)$$

qui donne après développement, le système d'équations suivant :

$$\sum_{i=0}^{n_s} s_i \eta_{k-i} = \begin{cases} \sum_{i=k}^{n_t} t_i w_{i-k} & k \leq n_t \\ 0 & k > n_t \end{cases} \quad (4.61)$$

Pour calculer les éléments $(\eta_{-n}, \dots, \eta_0, \dots, \eta_n)$ nécessaires a la formation des matrices R_k , on a besoin de $2n + 1$ équations qu'on prendra des systèmes (4.56) et (4.61).

Un bon choix du nombre suffisant d'équations conduit au système :

$$\mathbf{M}\boldsymbol{\eta} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} \quad (4.62)$$

où \mathbf{M} est la matrice $(n_d + n_s + 1) \times (n_d + n_s + 1)$ de la forme:

$$\mathbf{M} = \begin{pmatrix} 0 & \dots & 0 & 1 & d_1 & \dots & d_{n_d} & 0 \\ 0 & \dots & 1 & d_1 & \dots & d_{n_d} & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & d_1 & d_2 & \dots & d_{n_d} & 0 & \dots & 0 \\ s_{n_s} & s_{n_s-1} & \dots & s_1 & 1 & 0 & \dots & 0 \\ 0 & s_{n_s} & s_{n_s-1} & \dots & s_1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & s_{n_s} & s_{n_s-1} & \dots & s_1 & 1 \end{pmatrix} \quad (4.63)$$

$$\boldsymbol{\eta} = (\eta_{-n_s}, \dots, \eta_0, \dots, \eta_{n_d})^t \quad (4.64)$$

et $\mathbf{z}_1, \mathbf{z}_2$ deux vecteurs dont les éléments $(z_1)_i$ et $(z_2)_i$ sont donnés par :

$$(z_1)_i = \sum_{j=1}^{n_c} c_j f_{j-i} \quad i = 1, \dots, n_s \quad (4.65)$$

$$(z_2)_i = \sum_{j=i}^{n_t} t_j w_{j-i} \quad i = 0, \dots, n_d \quad (4.66)$$

un réarrangement des n_s premières équations de (4.62) nous permet de le réécrire sous la forme :

$$\begin{pmatrix} 1 & d_1 & d_2 & \dots & d_{n_d} & 0 & \dots & 0 \\ 0 & 1 & d_1 & d_2 & \dots & d_{n_d} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & d_1 & \dots & d_{n_d} & 0 \\ s_{n_s} & s_{n_s-1} & \dots & s_1 & 1 & 0 & \dots & 0 \\ 0 & s_{n_s} & s_{n_s-1} & \dots & s_1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & s_{n_s} & s_{n_s-1} & \dots & s_1 & 1 \end{pmatrix} \begin{pmatrix} \eta_{-n_s} \\ \eta_{-n_s+1} \\ \vdots \\ \eta_{-1} \\ \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{n_d} \end{pmatrix} = \begin{pmatrix} (z_1)_{n_s} \\ (z_1)_{n_s-1} \\ \vdots \\ (z_1)_1 \\ (z_2)_0 \\ (z_2)_1 \\ \vdots \\ (z_2)_{n_d} \end{pmatrix} \quad (4.67)$$

où sous forme compacte :

$$M' \eta = \begin{pmatrix} z'_1 \\ z_2 \end{pmatrix} \quad (4.68)$$

avec

$$z'_1 = J z_1$$

où J est une matrice de permutation définie par :

$$J = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix}$$

4.3 Etude de la méthode des moindres carrés vrais (MCV)

Comme il a été déjà précisé, la méthode MCV n'est qu'un cas particulier de la méthode des moindres carrés pondérés (MCP) pour laquelle la fonction de pondération $W(z)$ est de la forme :

$$W(z) = \frac{C(z)}{D(z)} = \frac{1}{\hat{A}(z)} \quad (4.69)$$

et la fonction $V(\mathbf{a}, \mathbf{b})$ à minimiser est :

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \frac{1}{\hat{A}(e^{j\theta})} \right|^2 |H(e^{j\theta})\hat{A}(e^{j\theta}) - \hat{B}(e^{j\theta})|^2 d\theta \quad (4.70)$$

d'autre part, on a aussi :

$$F(z) = \frac{T(z)}{S(z)} = \frac{\hat{B}(z)}{\hat{A}(z)D(z)} \quad (4.71)$$

et en conséquence de (4.69) et (4.71) on a les égalités simplificatrices suivantes :

$$\begin{aligned} C(z) &= 1 \\ D(z) &= \hat{A}(z) \\ T(z) &= \hat{B}(z) \\ S(z) &= \hat{A}(z)D(z) \\ n_c &= 0 \\ n_d &= n \\ n_t &= n \\ n_s &= 2n \\ c_0 &= 1 \\ d_i &= a_i \quad i = 1, \dots, n \\ t_i &= b_i \quad i = 1, \dots, n \end{aligned}$$

La résolution du problème de minimisation de la fonction $V(\mathbf{a}, \mathbf{b})$ au sens des moindres carrés vrais se fera en suivant la même démarche que celle utilisée pour le cas général MCP, en tenant compte des simplifications mentionnées auparavant.

La procédure à suivre peut être résumée en ces quelques points :

1. Calculer les éléments du polynôme $S(z)$ en effectuant le produit $\hat{A}(z)D(z)$.

2. Calculer les éléments nécessaires des réponses impulsionnelles w_k et f_k de $W(z)$ et $F(z)$ en utilisant les relations suivantes:

$$w_k = \begin{cases} -\sum_{i=1}^k d_i w_{k-i} & k = 1, \dots, n \\ c_0 & k = 0 \end{cases} \quad (4.72)$$

$$f_k = \begin{cases} b_k - \sum_{i=1}^k s_i f_{k-i} & k = 1, \dots, n \\ b_0 & k = 0 \end{cases} \quad (4.73)$$

3. Calculer les éléments de la fonction d'autocorrélation q_i $i = 1, \dots, n$ en résolvant le système d'équations suivant en utilisant l'algorithme de Jury donné en [20] et en Annexe F.

$$\begin{pmatrix} q_0 & q_1 & \dots & q_n \\ q_1 & q_0 & \dots & q_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ q_n & q_{n-1} & \dots & q_0 \end{pmatrix} \begin{pmatrix} 1 \\ d_1 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} c_0 w_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.74)$$

4. Calculer les éléments de la fonction d'autocorrélation ρ_i $i = 1, \dots, 2n$ en résolvant le système d'équations suivant en utilisant l'algorithme de Jury donné en [20] et en Annexe F.

$$\begin{pmatrix} \rho_0 & \rho_1 & \dots & \rho_{2n} \\ \rho_1 & \rho_0 & \dots & \rho_{2n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{2n} & \rho_{2n-1} & \dots & \rho_0 \end{pmatrix} \begin{pmatrix} 1 \\ s_1 \\ \vdots \\ s_{2n} \end{pmatrix} = \begin{pmatrix} f_0 & f_1 & \dots & f_n \\ 0 & f_0 & \dots & f_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} \quad (4.75)$$

5. Calculer les éléments de la fonction d'intercorrélacion $\eta_{-n}, \dots, \eta_0, \dots, \eta_n$ en résolvant le système d'équations suivant en utilisant l'algorithme de Sylvester donné en [20] et en Annexe E.

$$M' \eta = \begin{pmatrix} z'_1 \\ z_2 \end{pmatrix}$$

où la matrice M' est de la forme :

$$\mathbf{M}' = \begin{pmatrix} 1 & d_1 & d_2 & \dots & d_n & 0 & \dots & 0 \\ 0 & 1 & d_1 & d_2 & \dots & d_n & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & d_1 & \dots & d_n & 0 \\ s_{2n} & s_{2n-1} & \dots & s_1 & 1 & 0 & \dots & 0 \\ 0 & s_{2n} & s_{2n-1} & \dots & s_1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & s_{2n} & s_{2n-1} & \dots & s_1 & 1 \end{pmatrix} \quad (4.76)$$

et les éléments $(z'_1)_i$ et $(z_2)_i$ des vecteurs \mathbf{z}_1 et \mathbf{z}_2 sont déterminés à partir des relations suivantes :

$$(z'_1)_i = 0 \quad i = 1, \dots, 2n \quad (4.77)$$

$$(z_2)_i = \sum_{j=i}^n b_j w_{j-i} \quad i = 0, \dots, n \quad (4.78)$$

6. Former les matrices R_k , $k = 0, \dots, n$ et résoudre le problème en calculant les vecteurs solutions \mathbf{a} et \mathbf{b} en utilisant l'algorithme de Levinson à deux canaux donné en Annexe D.

4.4 Synthèse des filtres numériques RII par la méthode des moindres carrés vrais (MCV)

L'application de la méthode des moindres carrés vrais à la synthèse des filtres numériques RII, repose sur le principe décrit par Steiglitz et McBride en [42]. La procédure de synthèse consiste à résoudre de façon itérative le problème MCV vu précédemment en considérant que la fonction de pondération $W_{k+1}(z)$ à l'étape $k+1$ est donnée par :

$$W_{k+1}(z) = \frac{1}{\hat{A}_k(z)} \quad (4.79)$$

où $\hat{A}_k(z)$ est la solution du problème MCV à la $k^{\text{ième}}$ itération.

Cette procédure peut être résumée en ces quelques points :

Ayant un filtre prototype AR, MA ou ARMA d'ordre N caractérisé par sa fonction de transfert $H(z)$

1. Effectuer l'étape d'initialisation en résolvant le problème au sens des moindres carrés modifiés (MCM) pour trouver les n coefficients de $\hat{A}(z)$ et de $\hat{B}(z)$
2. Initialiser la procédure en faisant: (première itération, $k = 0$).

$$\begin{aligned}\hat{A}_0(z) &= \hat{A}(z) \\ \hat{B}_0(z) &= \hat{B}(z)\end{aligned}$$

3. Pour chaque étape faire : (étape k)

$$W_{k+1}(z) = \frac{1}{\hat{A}_k(z)}$$

4. Résoudre, en suivant la procédure donnée auparavant, le problème MCV en trouvant les coefficients (a_1, \dots, a_n) et (b_0, \dots, b_n) des polynômes $\hat{A}_{k+1}(z)$ et $\hat{B}_{k+1}(z)$ qui minimisent la fonction :

$$V(\mathbf{a}, \mathbf{b}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \frac{1}{\hat{A}_k(e^{j\theta})} \right|^2 \left| H(e^{j\theta}) \hat{A}_{k+1}(e^{j\theta}) - \hat{B}_{k+1}(e^{j\theta}) \right|^2 d\theta$$

5. Vérifier la stabilité du filtre obtenu en utilisant l'algorithme de Jury (inverse Levinson) donné en annexe F ou bien l'algorithme de Bistritz [5].
6. Calculer l'écart entre les coefficients de $\hat{A}_{k+1}(z)$ et $\hat{A}_k(z)$ si cet écart est suffisamment faible, la convergence est obtenue, sinon aller à l'étape 3 pour effectuer l'itération $k + 1$.

cette procédure est représentée sous forme graphique par l'organigramme de la figure 4.2.

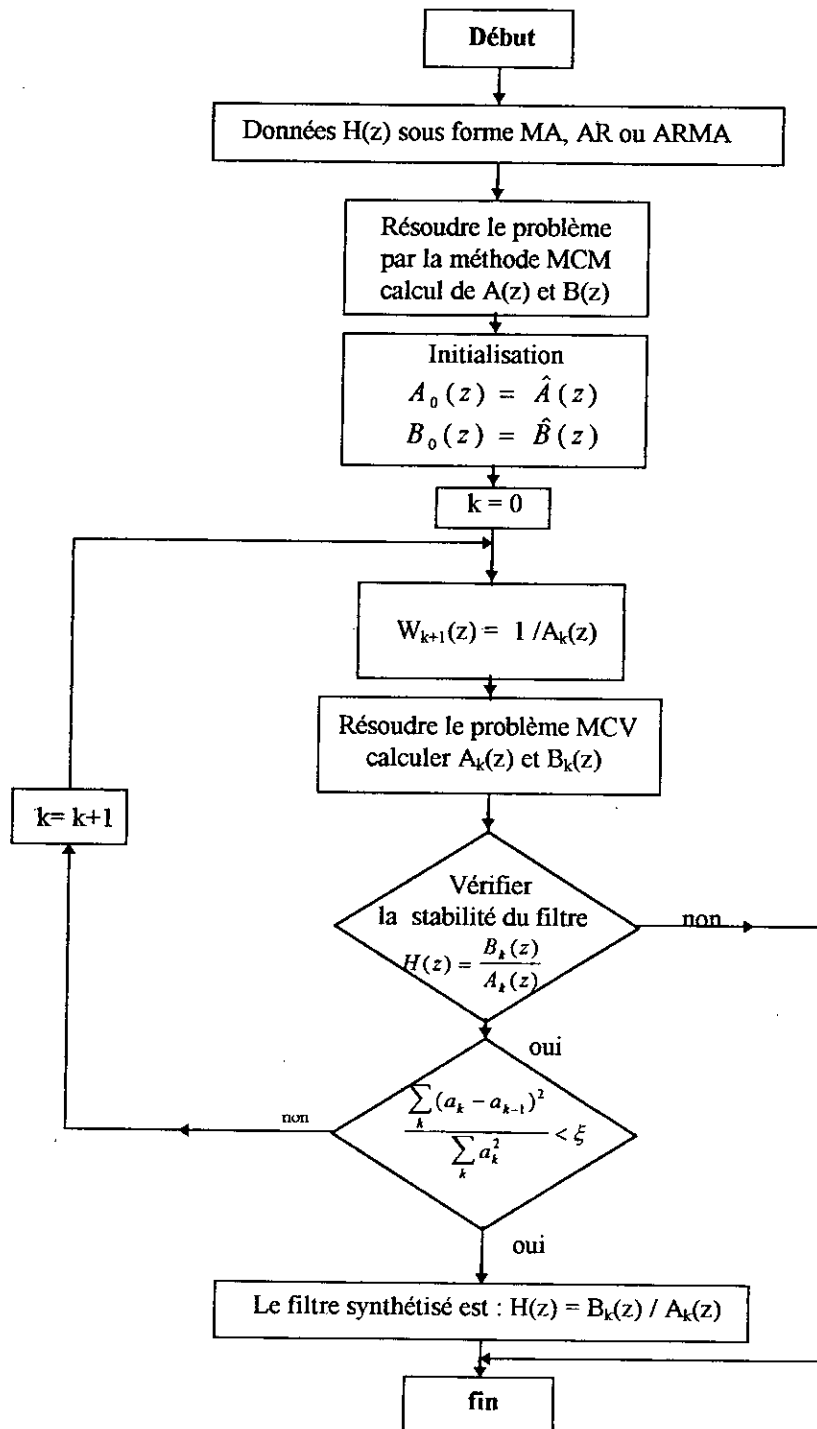


Figure 4.2: Procédure de synthèse d'un filtre numérique par la méthode des moindres carrés vrais (MCV)

Chapitre 5

Exemples de Synthèse

5.1 Introduction

Le but de ce chapitre est de présenter, sous forme graphique, les différents résultats obtenus par la mise en œuvre des programmes pour les trois méthodes étudiées.

- *La méthode de Padé*
- *La méthode MCM*
- *La méthode MCV*

Afin de comparer les performances des filtres synthétisés par ces trois méthodes et d'examiner particulièrement l'efficacité de la méthode des moindres carrés vrais (MCV), un certain nombre d'exemples, qui constituent un échantillon illustrant les différents cas de synthèse de filtres passe-bas, passe-haut et passe-bande ont été choisis, les spectres d'amplitude et de phase, calculés en utilisant l'algorithme de Horner donné en Annexe A sont tracés, pour chaque exemple et présentés sur la même figure (figures 5.x-a et 5.x-b).

Le groupe de figures (5.x-c) représentent les réponses impulsionnelles des différents filtres synthétisés.

L'évolution de l'erreur suivant le nombre d'itérations est représentée par les figures (5.x-d), elle permet de suivre la convergence de la méthode MCV.

Enfin, la possibilité d'application de la méthode MCV pour la réduction de modèles est aussi illustrée (figure 5.11).

5.2 Application à la synthèse des filtres

L'application de la méthode des moindres carrés vrais à la synthèse de filtres numériques RII, consiste à déterminer leur coefficients $(a_i, b_i \quad i = 0, \dots, n)$.

Le type (passe-bas, passe-haut ou passe-bande) et le gabarit (fréquence de coupure, atténuation) du filtre à synthétiser sont spécifiés par le filtre modèle de type AR ou MA d'ordre élevé.

L'initialisation de la procédure MCV se fait par un filtre intermédiaire ARMA, donné par la méthode des moindres carrés modifiés (MCM) ou de Padé.

5.2.1 Synthèse de filtres passe-bas

Exemple 1 (figure 5.1)

Filtre ARMA(10,10), de fréquence de coupure normalisée $f_c = 0.15$, d'atténuation 50 dB, synthétisé à partir d'un filtre AR(100).

Exemple 2 (figure 5.2)

Filtre ARMA(10,10), de fréquence de coupure normalisée $f_c = 0.05$, d'atténuation 30 dB, synthétisé à partir d'un filtre AR(150).

Exemple 3 (figure 5.3)

Filtre ARMA(10,10), de fréquence de coupure normalisée $f_c = 0.01$, d'atténuation 30 dB, synthétisé à partir d'un filtre AR(150).

Exemple 4 (figure 5.4)

Filtre ARMA(8,8), de fréquence de coupure normalisée $f_c = 0.10$, d'atténuation 40 dB, synthétisé à partir d'un filtre MA(51).

Exemple 5 (figure 5.5)

Filtre ARMA(8,8), de fréquence de coupure normalisée $f_c = 0.05$, d'atténuation 40 dB, synthétisé à partir d'un filtre MA(101).

5.2.2 Synthèse de filtres passe-haut**Exemple 6 (figure 5.6)**

Filtre ARMA(10,10), de fréquence de coupure normalisée $f_c = 0.15$, d'atténuation 30 dB, synthétisé à partir d'un filtre AR(180).

Exemple 7 (figure 5.7)

Filtre ARMA(12,12), de fréquence de coupure normalisée $f_c = 0.05$, d'atténuation 40 dB, synthétisé à partir d'un filtre AR(180).

Exemple 8 (figure 5.8)

Filtre ARMA(12,12), de fréquence de coupure normalisée $f_c = 0.01$, d'atténuation 30 dB, synthétisé à partir d'un filtre AR(180).

5.2.3 Synthèse de filtres passe-bande**Exemple 9 (figure 5.9)**

Filtre ARMA(10,10), de fréquences de coupures normalisées ($f_{11} = 0.05$, $f_{12} = 0.10$) (bande 1), d'atténuation 40 dB, synthétisé à partir d'un filtre AR(150).

Exemple 10 (figure 5.10)

Filtre ARMA(10,10), de fréquences de coupures normalisées ($f_{11} = 0.00$, $f_{12} = 0.05$) (bande 1) et ($f_{21} = 0.10$, $f_{22} = 0.15$) (bande 2) d'atténuation 40 dB, synthétisé à partir d'un filtre AR(150).

5.3 Application à la réduction de modèles

Concernant les performances de la méthode MCV appliquée à la réduction de modèles, celles-ci ont été testées en considérant des filtres ARMA synthétisés par les méthodes de transformations classiques à partir de filtres prototypes de Butterworth, Chebyshev, Cauer, . . . etc, [4] que nous avons approximés par des filtres ARMA d'ordres inférieurs. Un exemple illustrant cette application est donné par la figure 5.11.

Exemple 11 (figure 5.11)

Filtre ARMA(6,6), obtenu par réduction d'un filtre ARMA(10,10) (modèle), passe-bas de Butterworth, de fréquence de coupure normalisée $f_c = 0.25$.

5.4 Commentaires

La comparaison des spectres d'amplitude des filtres synthétisés par les méthodes de PADE, MCM et MCV fait apparaître les performances de cette dernière. Le spectre du filtre MCV s'approche sensiblement du filtre désiré dans la bande passante et dans la bande atténuée. A noter aussi que les résultats obtenus sont meilleurs lorsqu'on travaille avec des bandes passantes très étroites.

Les spectres de phases montrent que les filtres synthétisés par la méthode MCV conservent la propriété de linéarité de la phase à l'intérieur des bandes passantes lorsque celle-ci est vérifiée par le filtre modèle (AR ou MA).

La convergence de la méthode est rapide, moins de dix itérations sont généralement suffisantes.

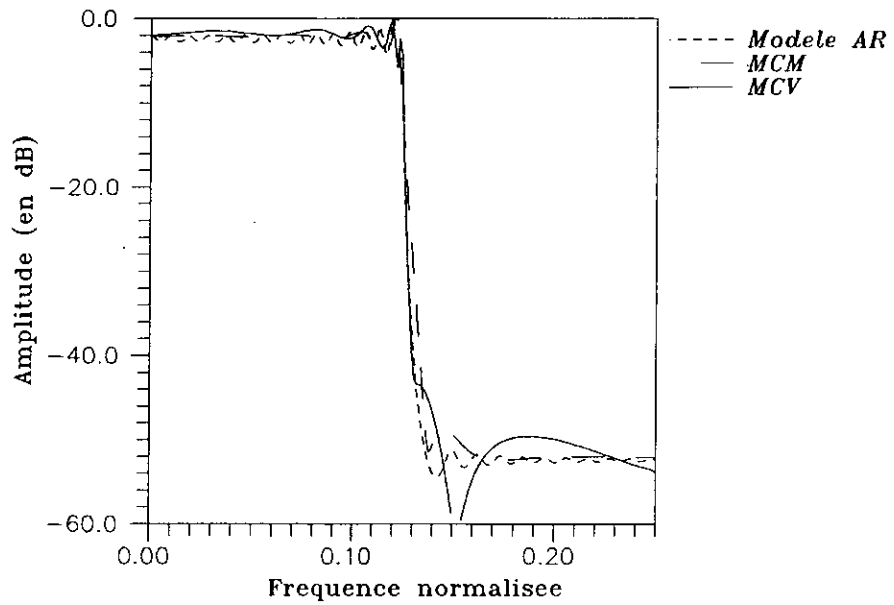


Figure 5.1-a: Spectres d'amplitude des filtres passe-bas AR(100) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

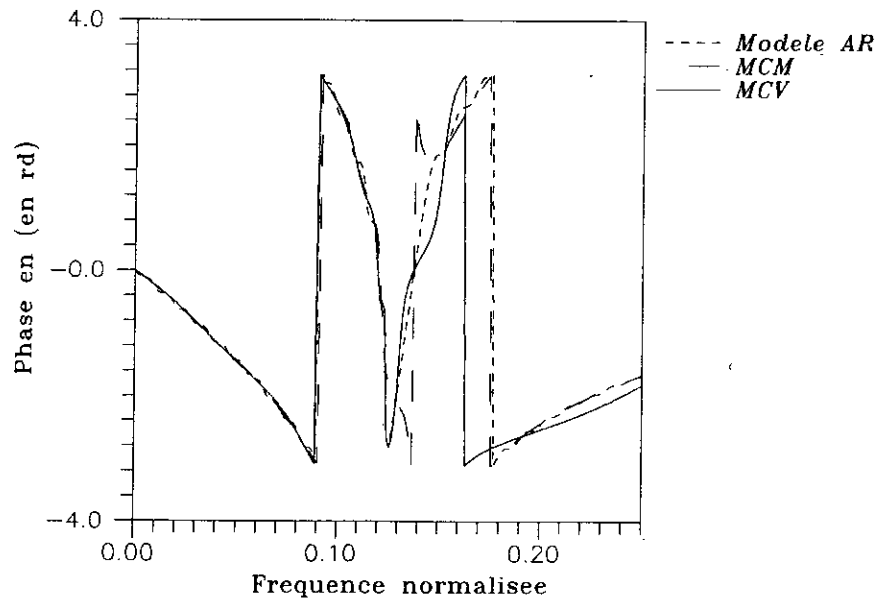


Figure 5.1-b: Spectres de phase des filtres passe-bas AR(100) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

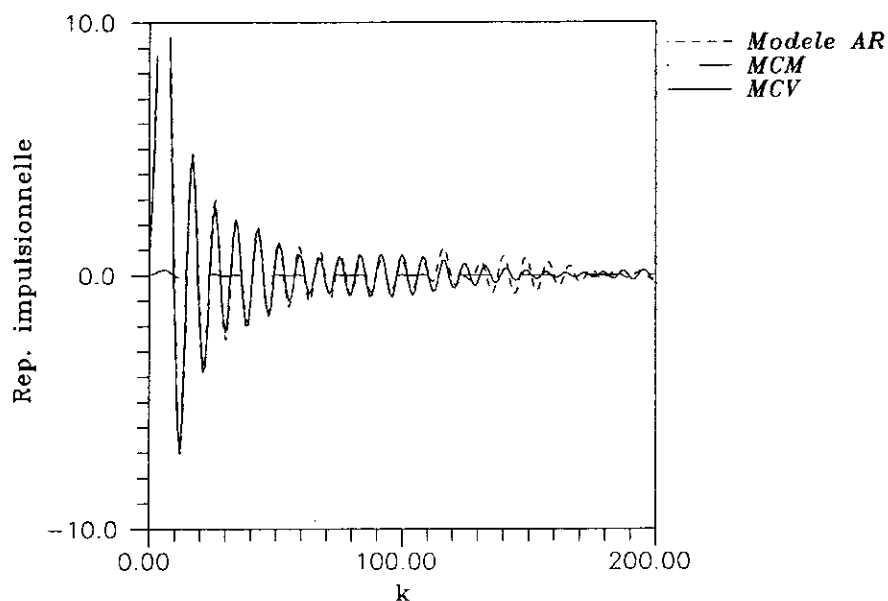


Figure 5.1-c: Réponses impulsionnelles des filtres passe-bas AR(100) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

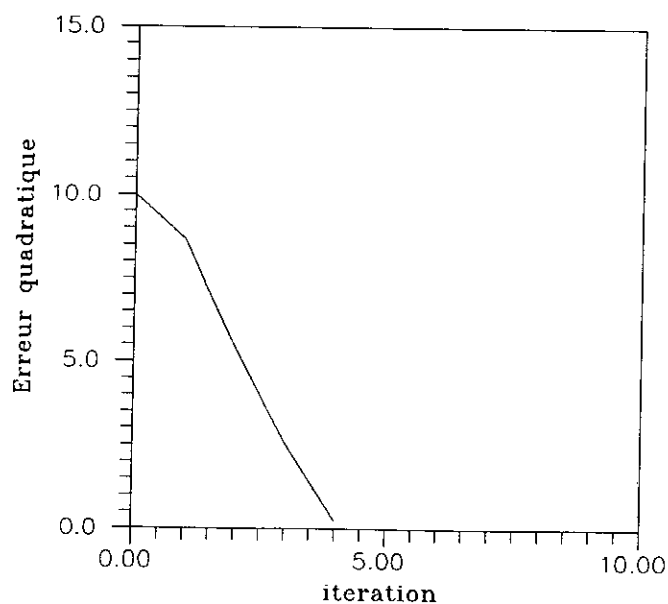


Figure 5.1-d: Evolution de l'erreur MCV.

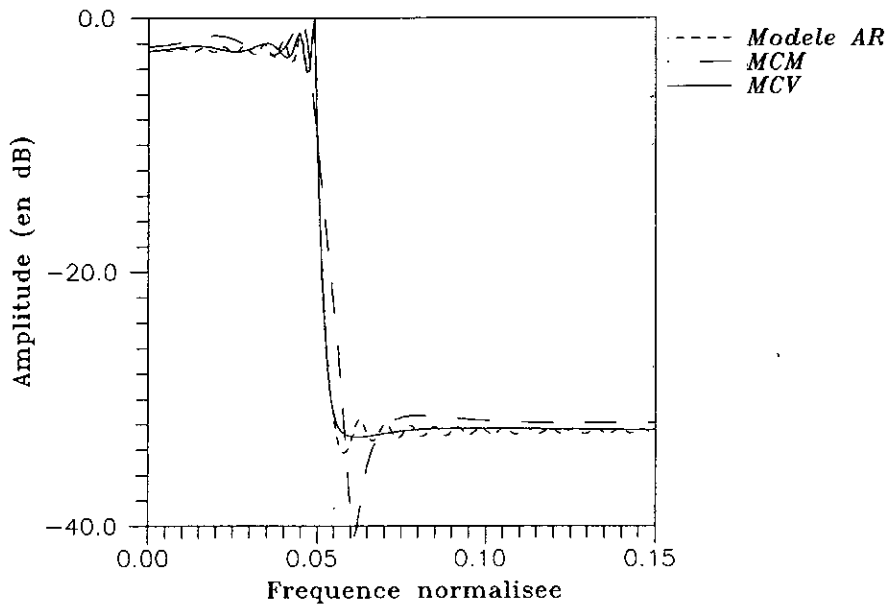


Figure 5.2-a: Spectres d'amplitude des filtres passe-bas AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

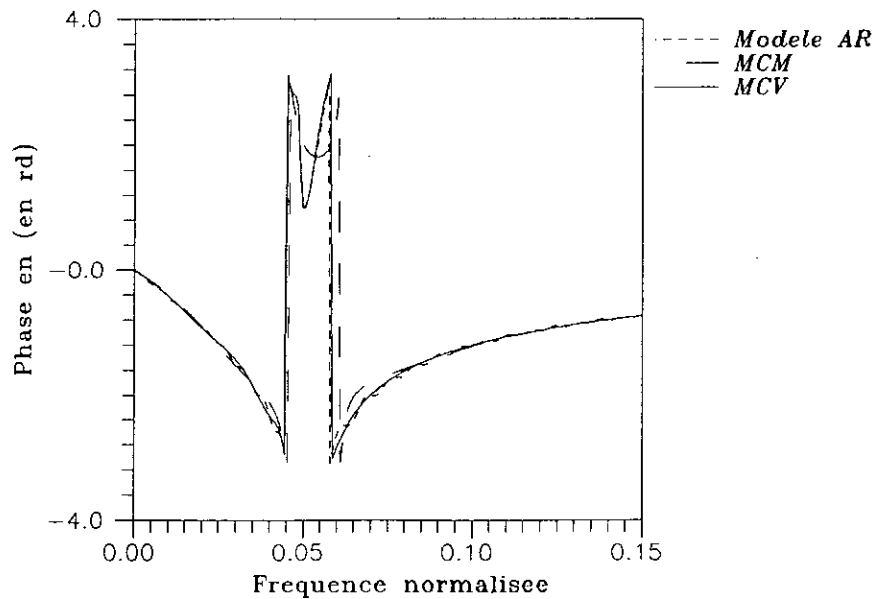


Figure 5.2-b: Spectres de phase des filtres passe-bas AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

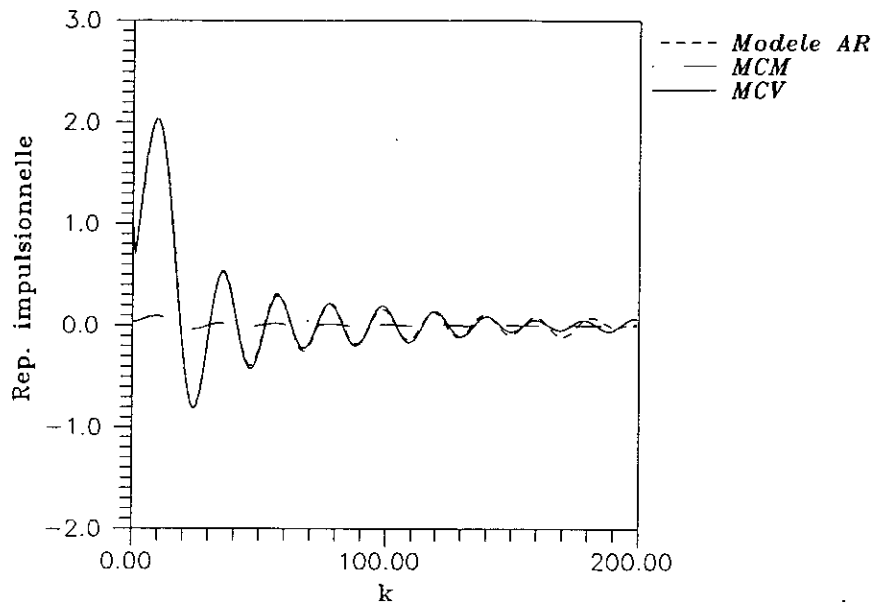


Figure 5.2-c: Réponses impulsionnelles des filtres passe-bas AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

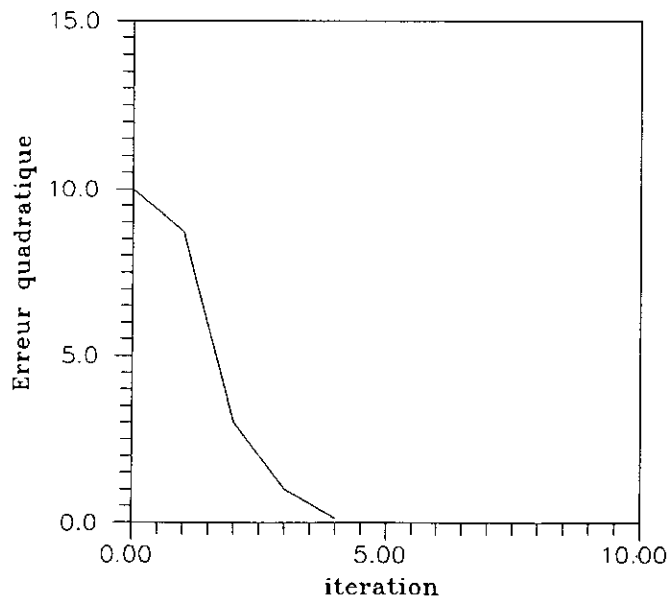


Figure 5.2-d: Evolution de l'erreur MCV.

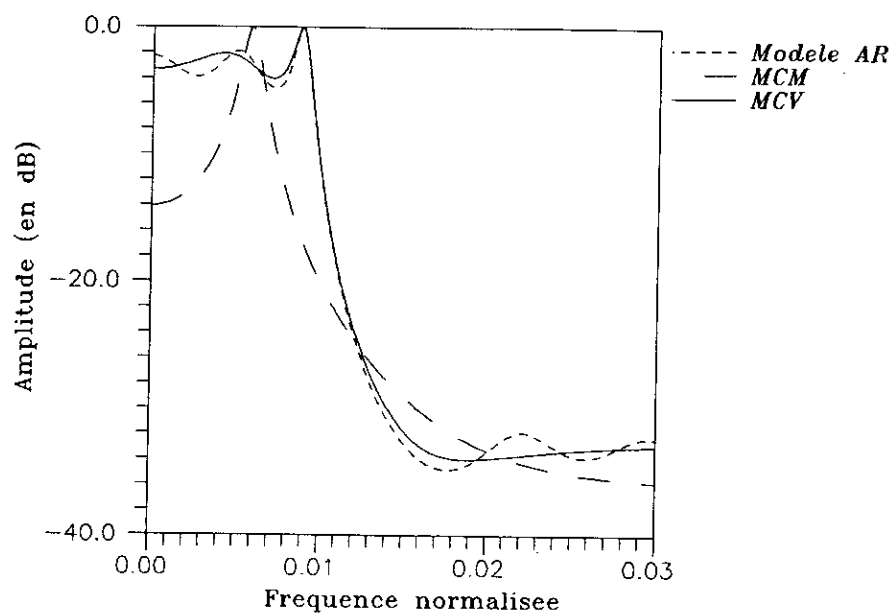


Figure 5.3-a: Spectres d'amplitude des filtres passe-bas AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

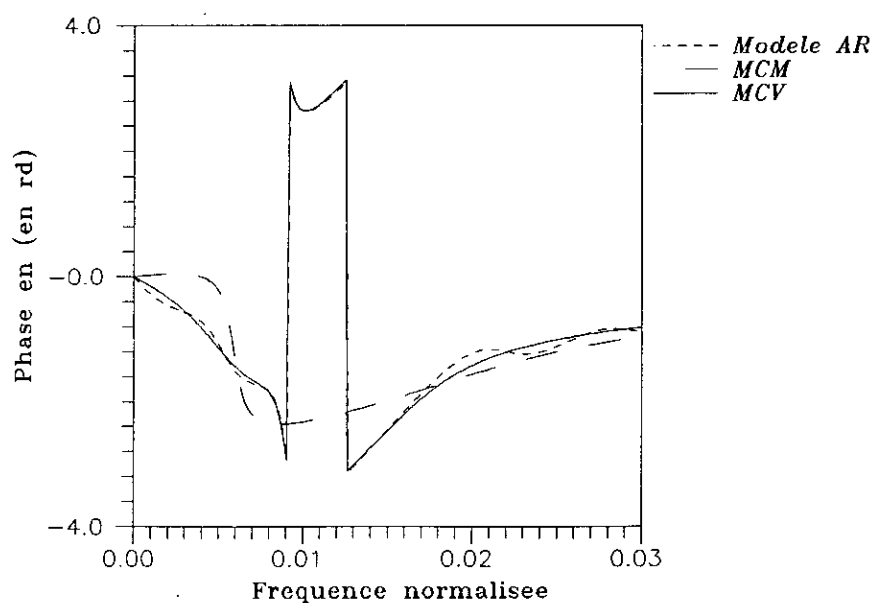


Figure 5.3-b: Spectres de phase des filtres passe-bas AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

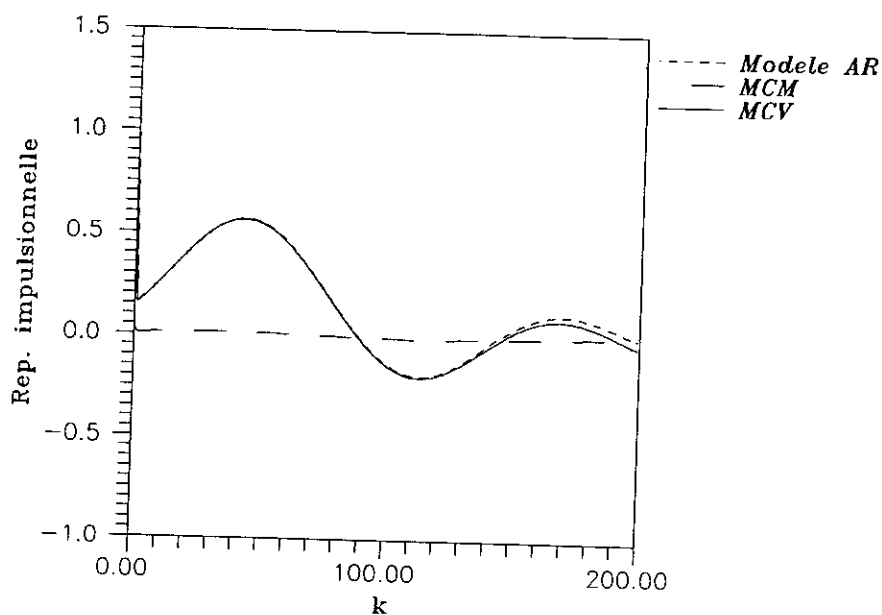


Figure 5.3-c: Réponses impulsionnelles des filtres passe-bas AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

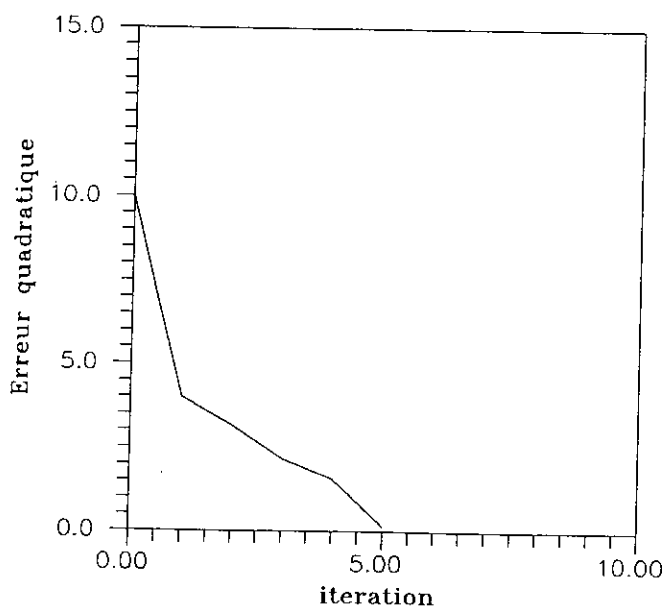


Figure 5.3-d: Evolution de l'erreur MCV.

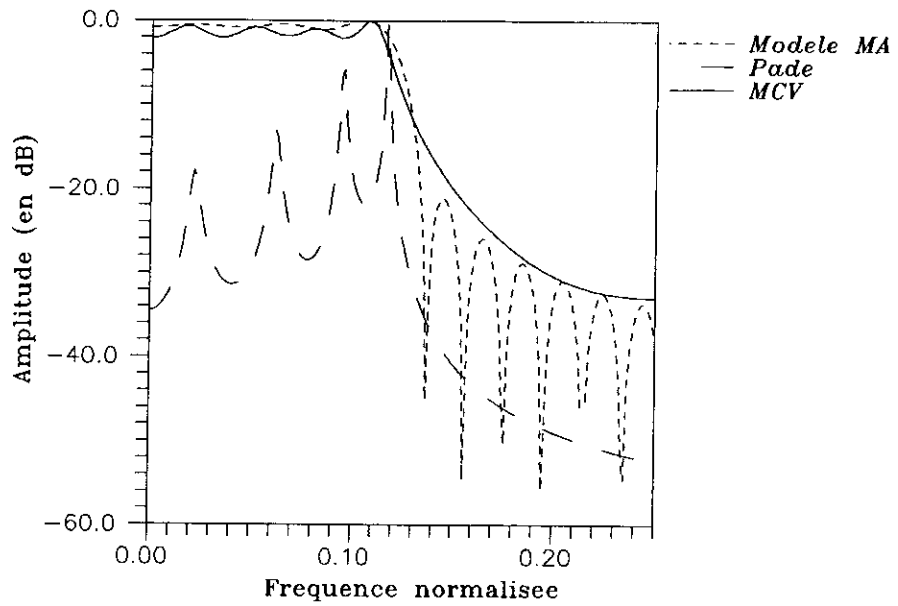


Figure 5.4-a: Spectres d'amplitude des filtres passe-bas MA(51) et ses approximations ARMA(8,8) synthétisés par les méthodes Padé et MCV.

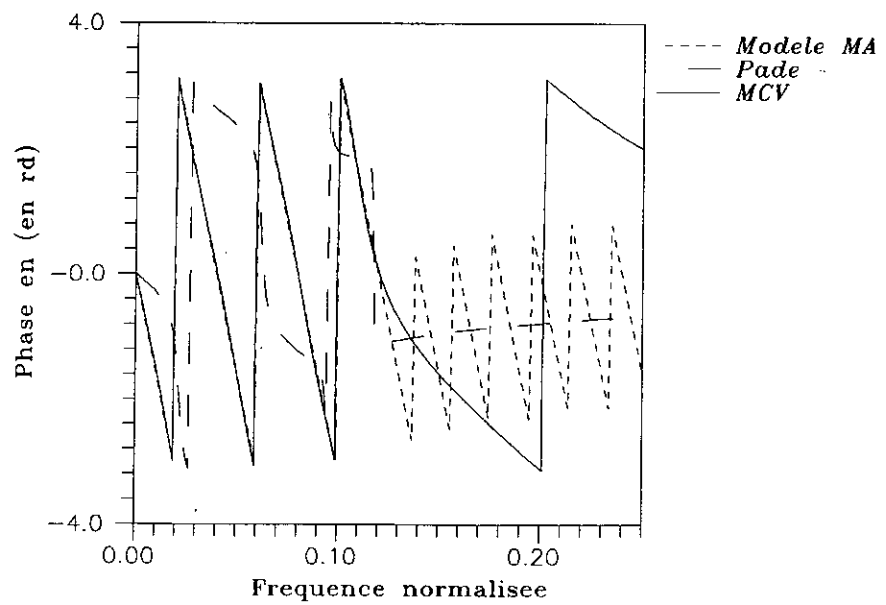


Figure 5.4-b: Spectres de phase des filtres passe-bas MA(51) et ses approximations ARMA(8,8) synthétisés par les méthodes Padé et MCV.

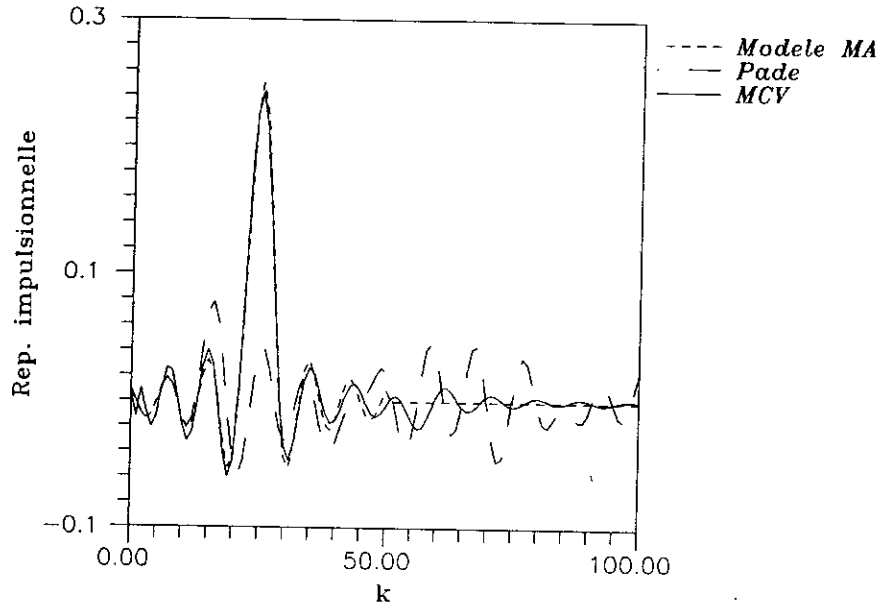


Figure 5.4-c: Réponses impulsionnelles des filtres passe-bas MA(51) et ses approximations ARMA(8,8) synthétisés par les méthodes Padé et MCV.

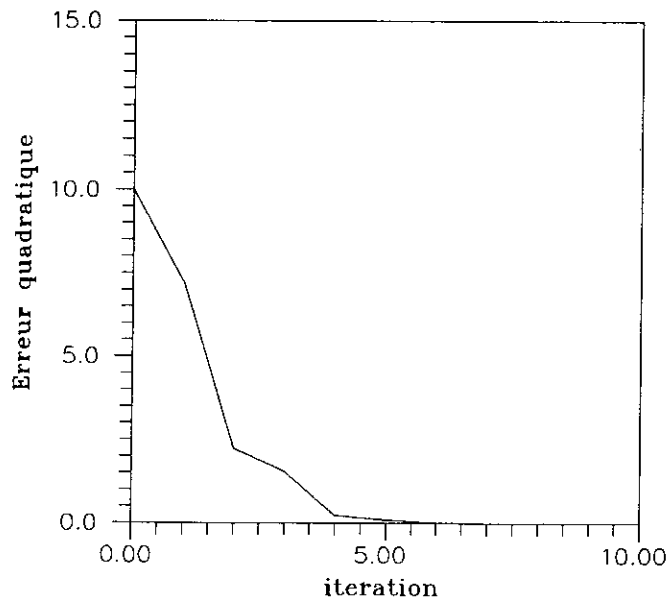


Figure 5.4-d: Evolution de l'erreur MCV.

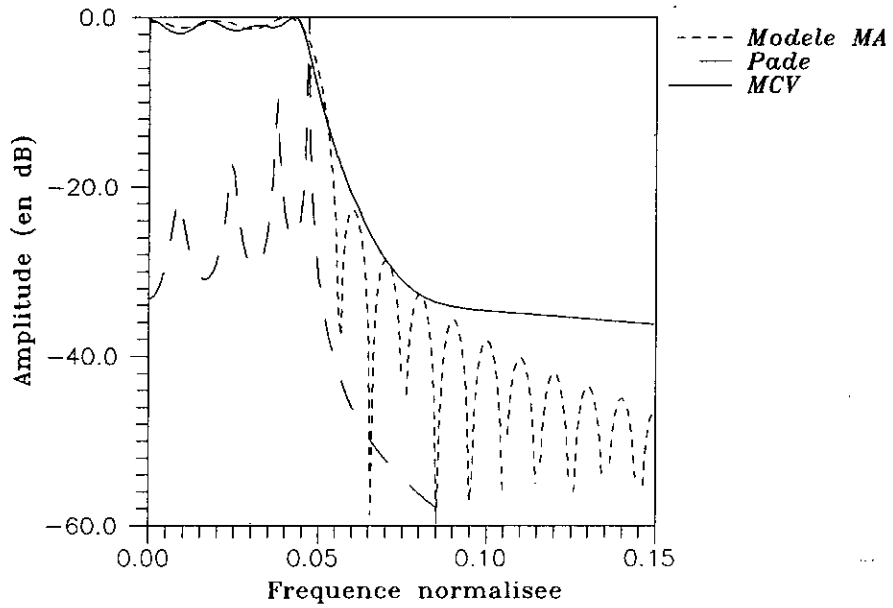


Figure 5.5-a: Spectres d'amplitude des filtres passe-bas MA(101) et ses approximations ARMA(8,8) synthétisés par les méthodes Padé et MCV.

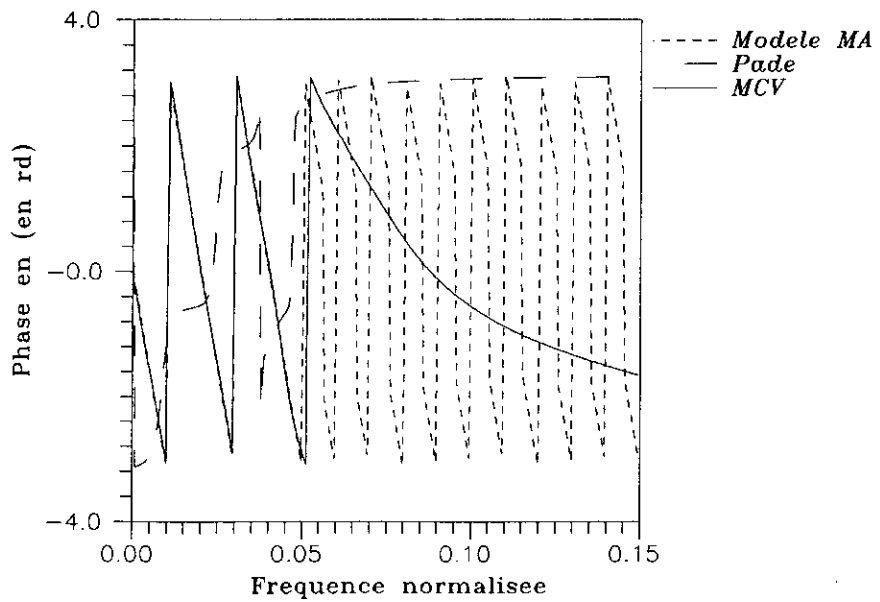


Figure 5.5-b: Spectres de phase des filtres passe-bas MA(101) et ses approximations ARMA(8,8) synthétisés par les méthodes Padé et MCV.

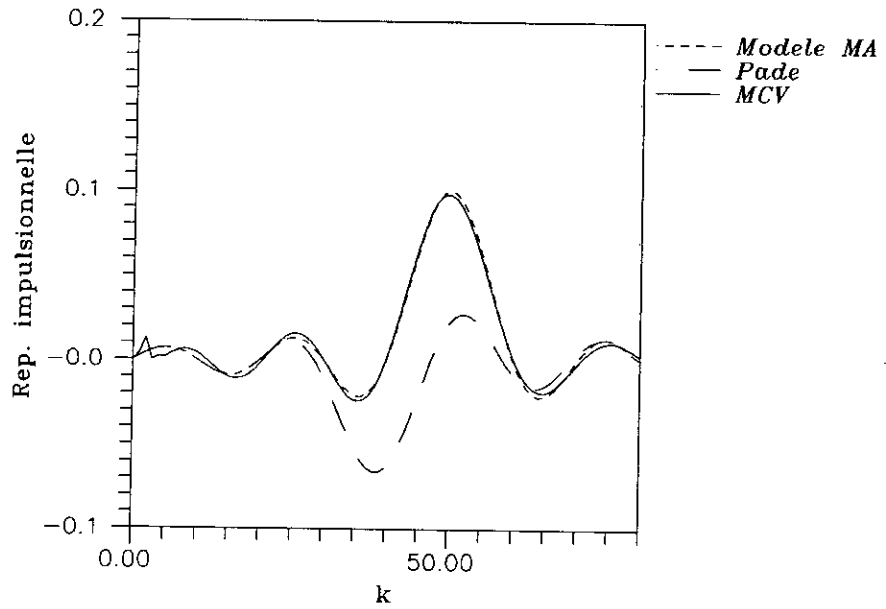


Figure 5.5-c: Réponses impulsionnelles des filtres passe-bas MA(101) et ses approximations ARMA(8,8) synthétisés par les méthodes Padé et MCV.

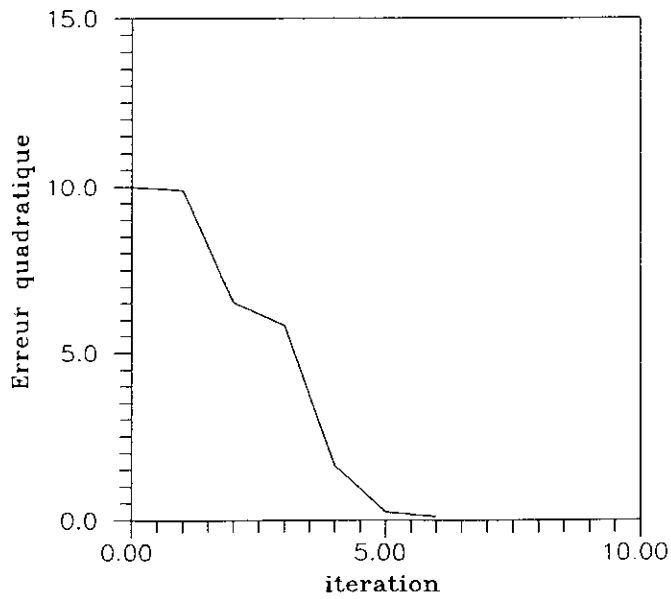


Figure 5.5-d: Evolution de l'erreur MCV.

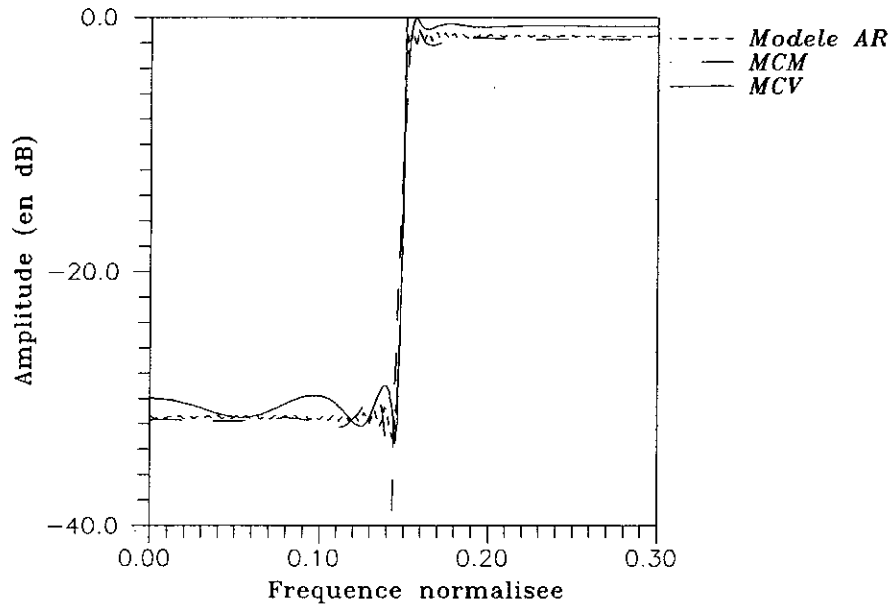


Figure 5.6-a: Spectres d'amplitude des filtres passe-haut AR(180) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

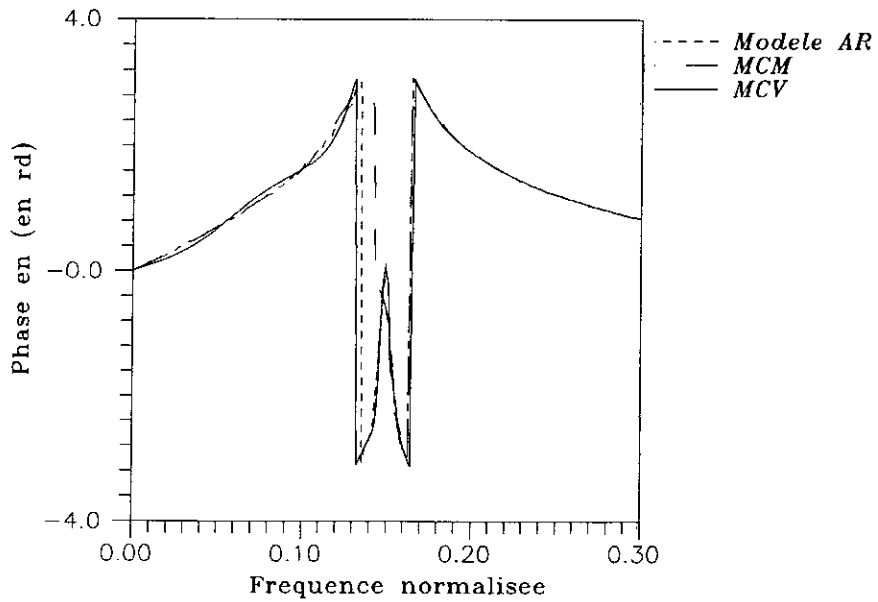


Figure 5.6-b: Spectres de phase des filtres passe-haut AR(180) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

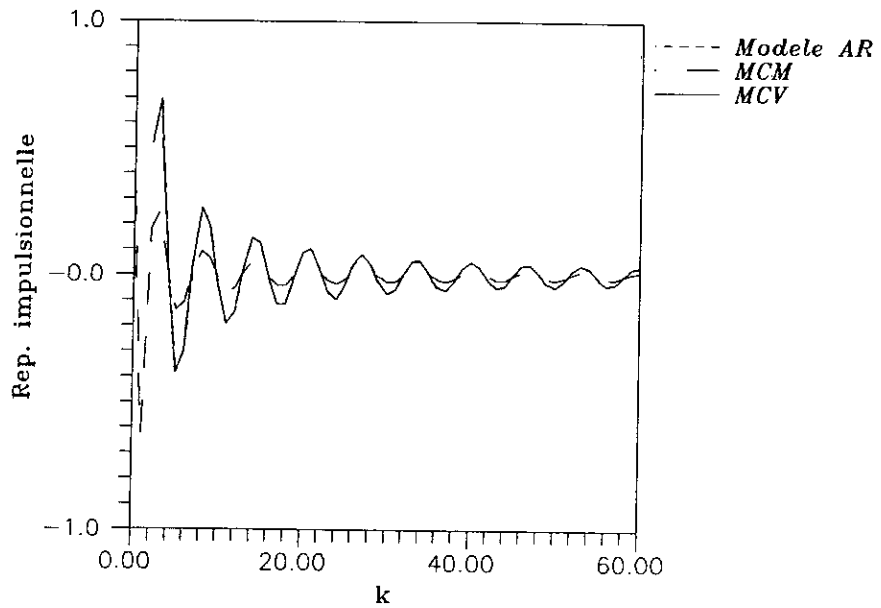


Figure 5.6-c: Réponses impulsionnelles des filtres passe-haut AR(180) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

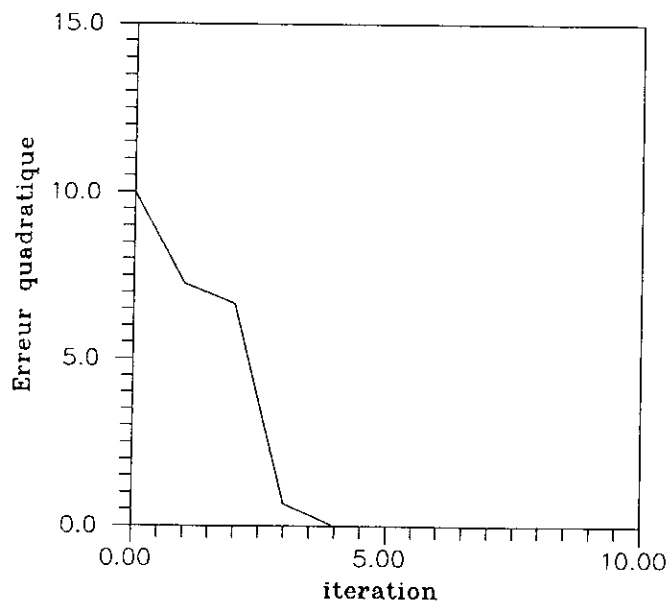


Figure 5.6-d: Evolution de l'erreur MCV.

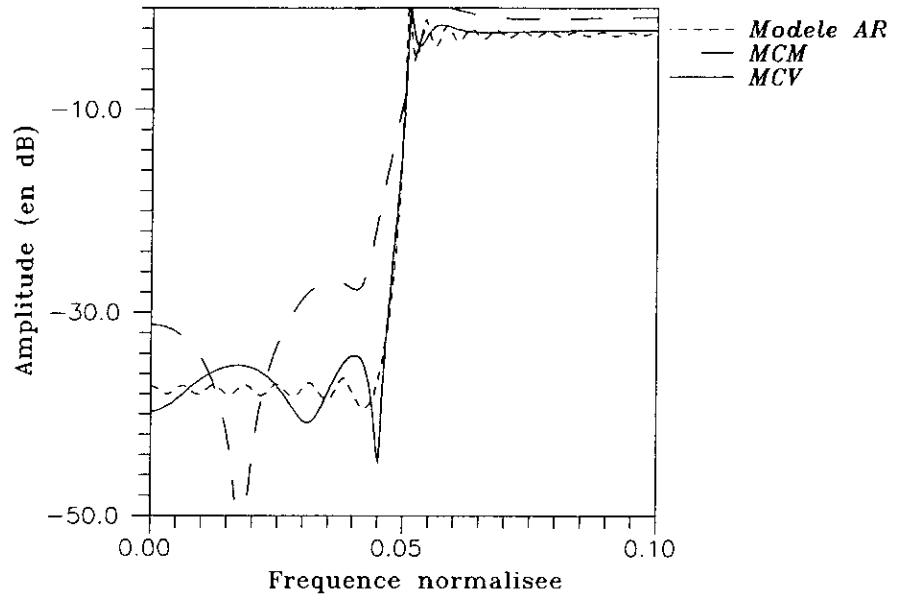


Figure 5.7-a: Spectres d'amplitude des filtres passe-haut AR(180) et ses approximations ARMA(12,12) synthétisés par les méthodes MCM et MCV.

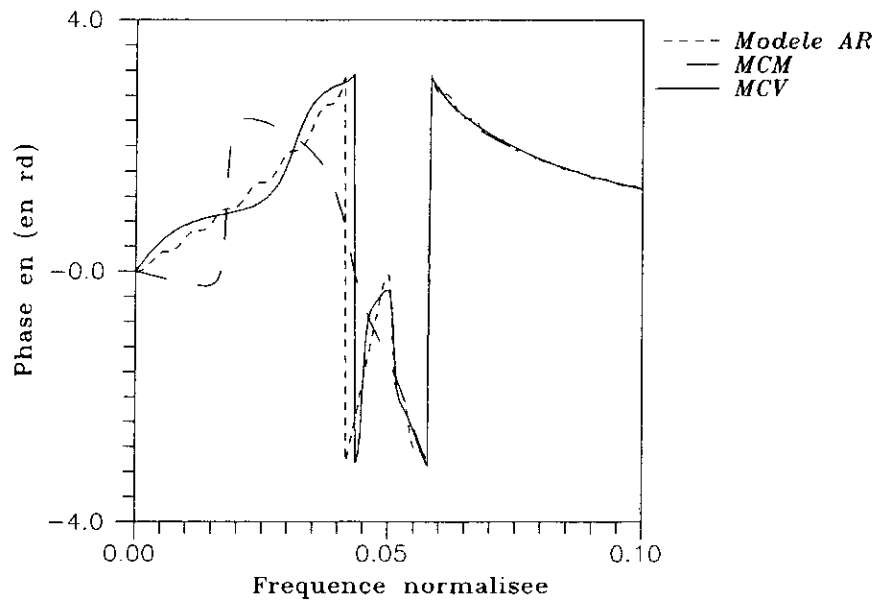


Figure 5.7-b: Spectres de phase des filtres passe-haut AR(180) et ses approximations ARMA(12,12) synthétisés par les méthodes MCM et MCV.

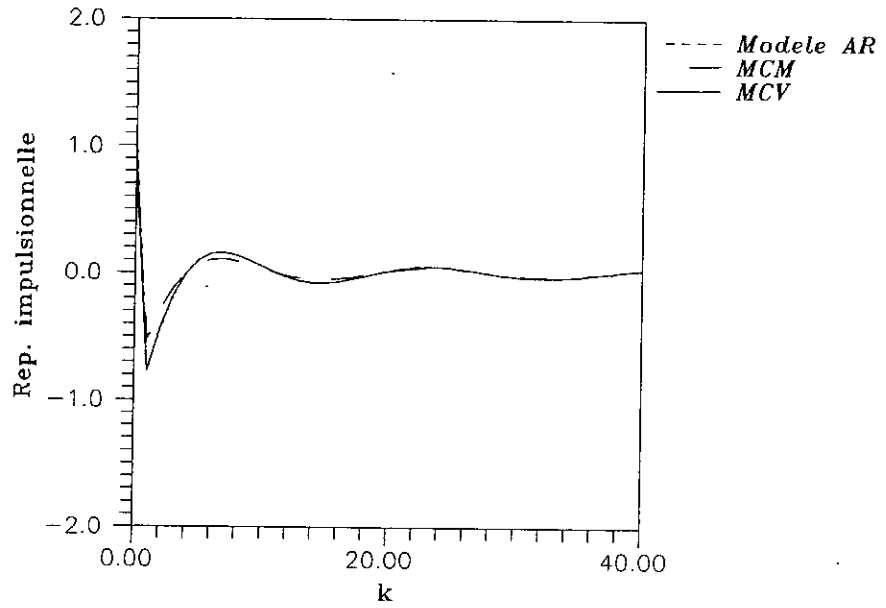


Figure 5.7-c: Réponses impulsionnelles des filtres passe-haut AR(180) et ses approximations ARMA(12,12) synthétisés par les méthodes MCM et MCV.

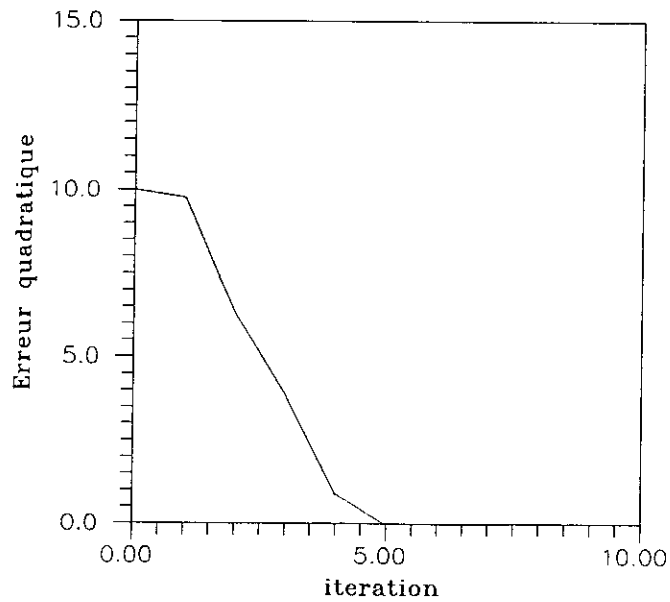


Figure 5.7-d: Evolution de l'erreur MCV.

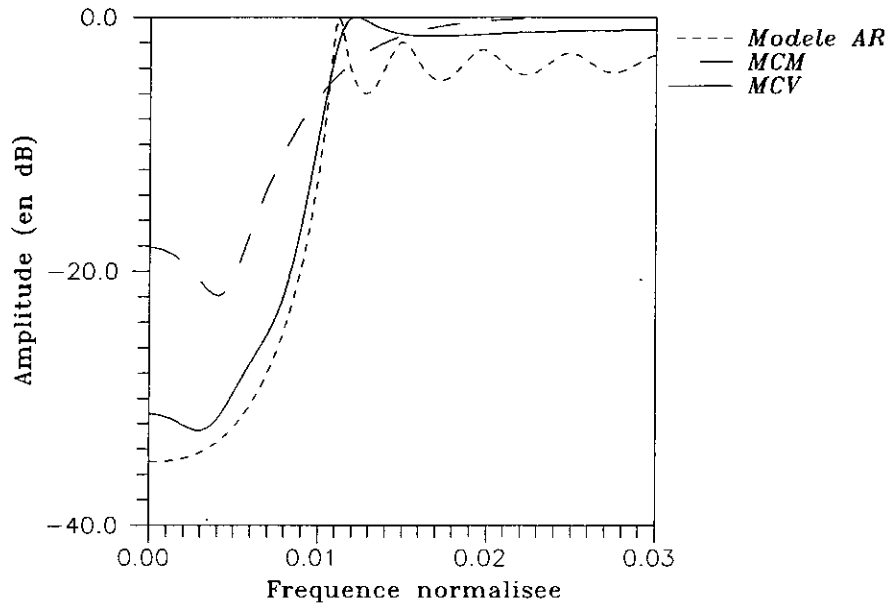


Figure 5.8-a: Spectres d'amplitude des filtres passe-haut AR(180) et ses approximations ARMA(12,12) synthétisés par les méthodes MCM et MCV.

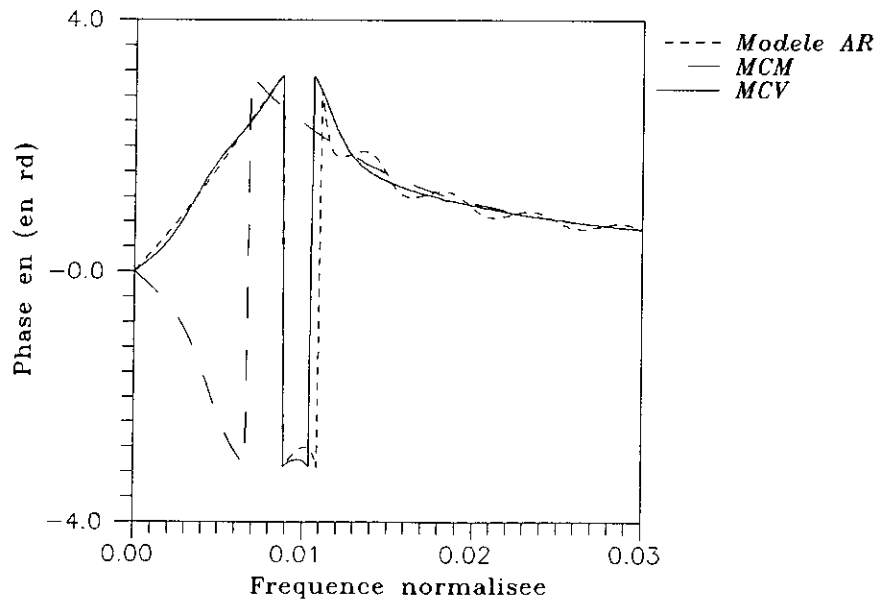


Figure 5.8-b: Spectres de phase des filtres passe-haut AR(180) et ses approximations ARMA(12,12) synthétisés par les méthodes MCM et MCV.

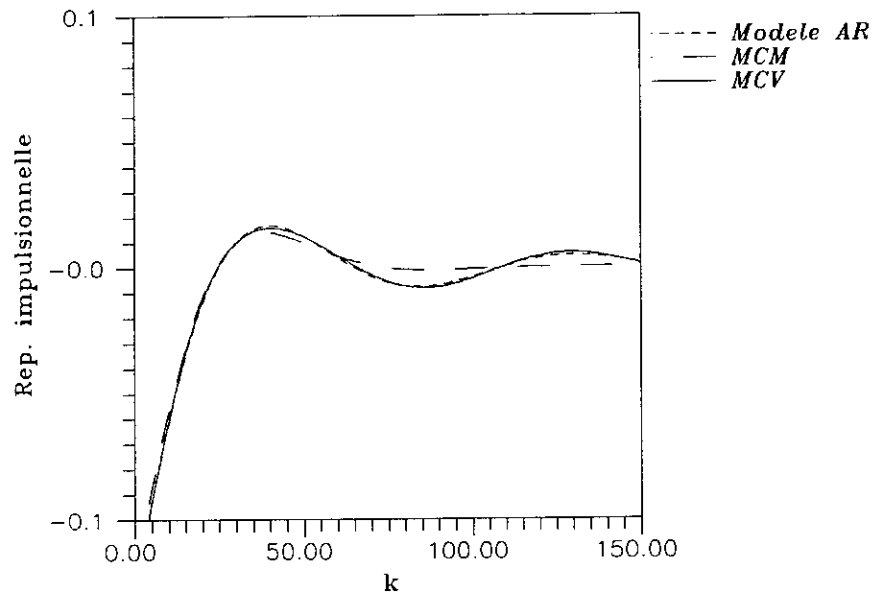


Figure 5.8-c: Réponses impulsionnelles des filtres passe-haut AR(180) et ses approximations ARMA(12,12) synthétisés par les méthodes MCM et MCV.

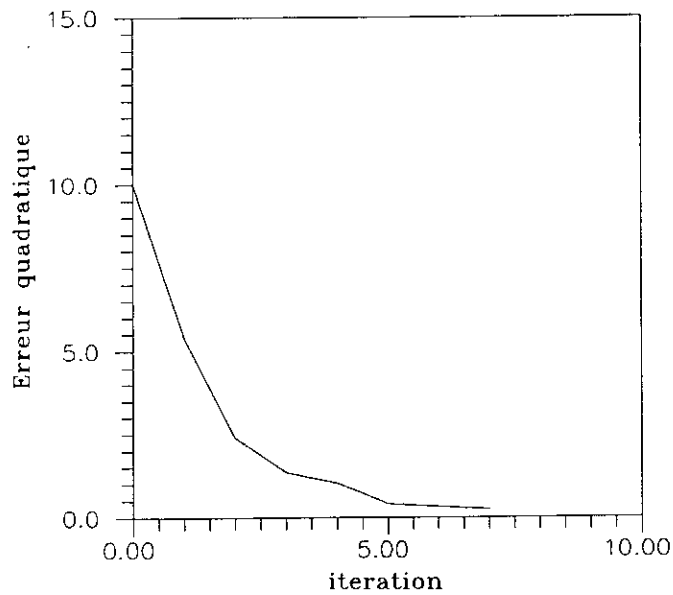


Figure 5.8-d: Evolution de l'erreur MCV.

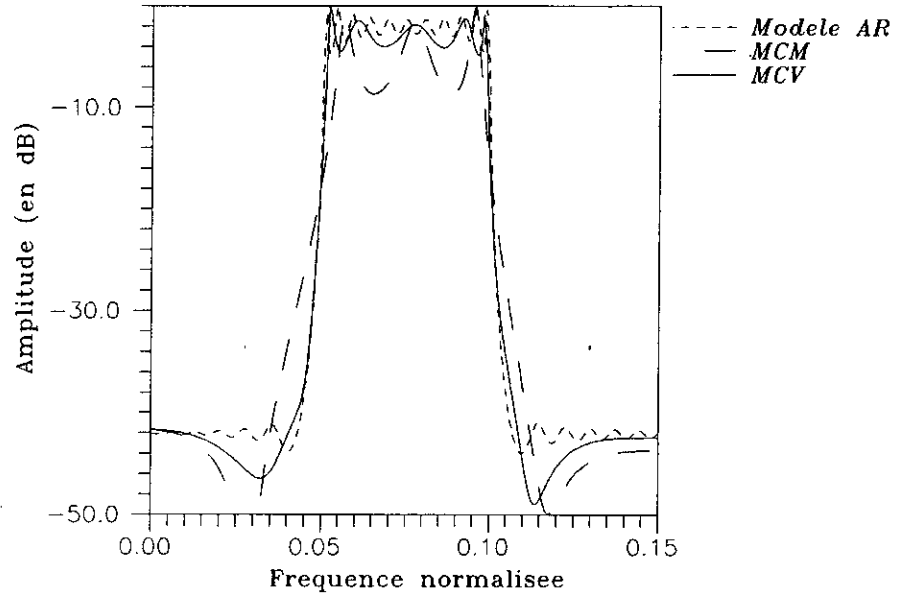


Figure 5.9-a: Spectres d'amplitude des filtres passe-bande AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

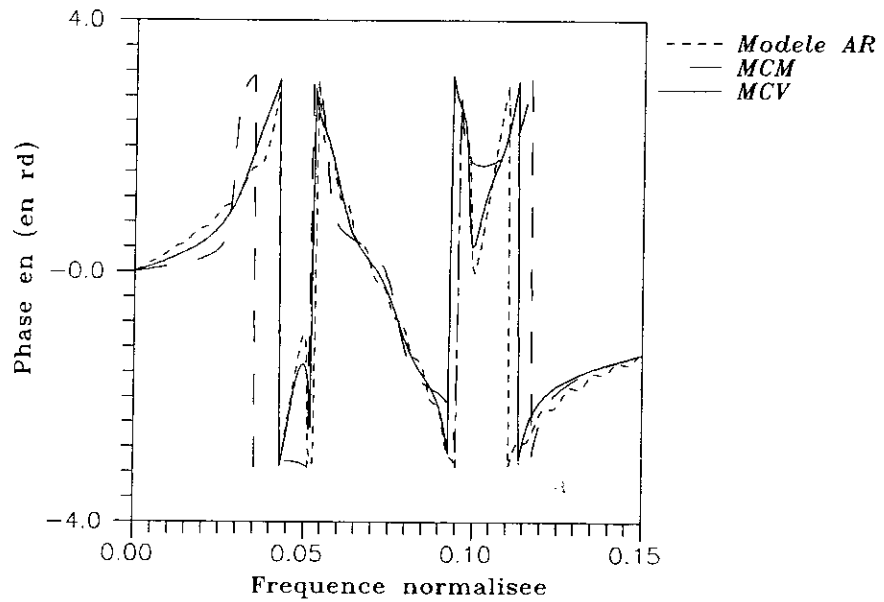


Figure 5.9-b: Spectres de phase des filtres passe-bande AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

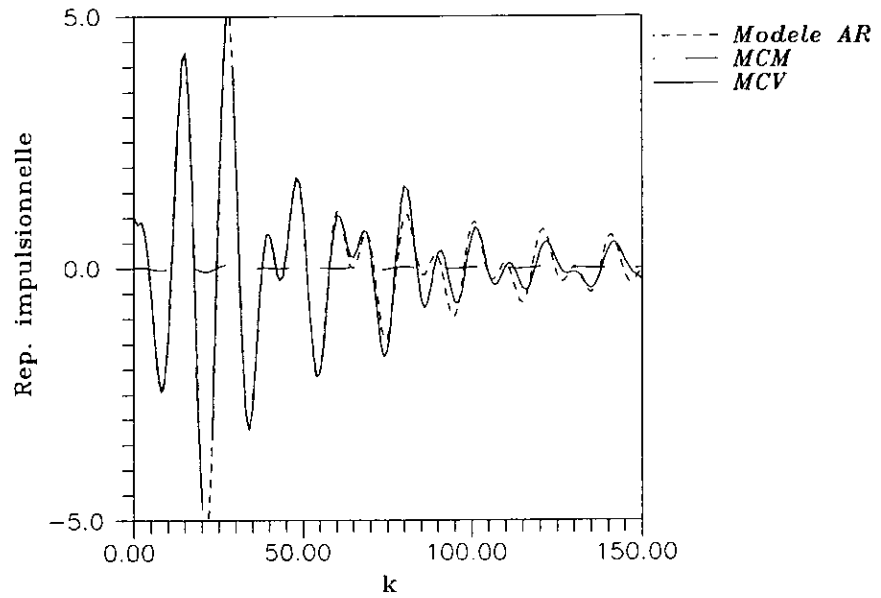


Figure 5.9-c: Réponses impulsionnelles des filtres passe-bande AR(150) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

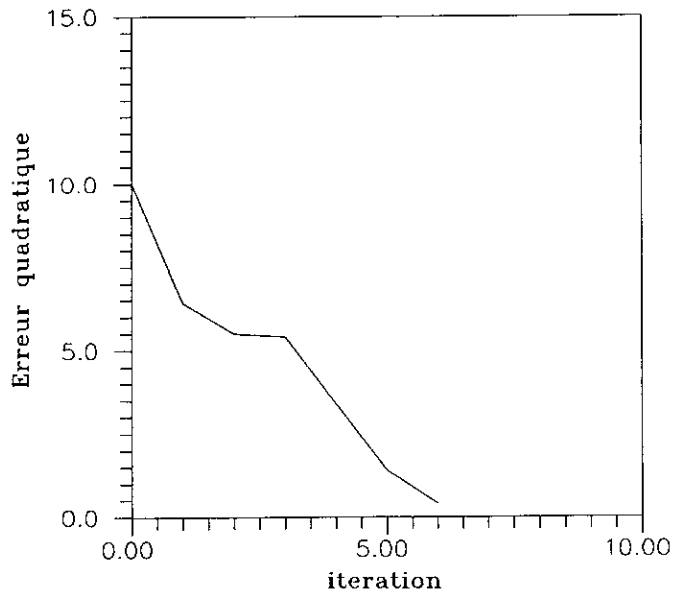


Figure 5.9-d: Evolution de l'erreur MCV.

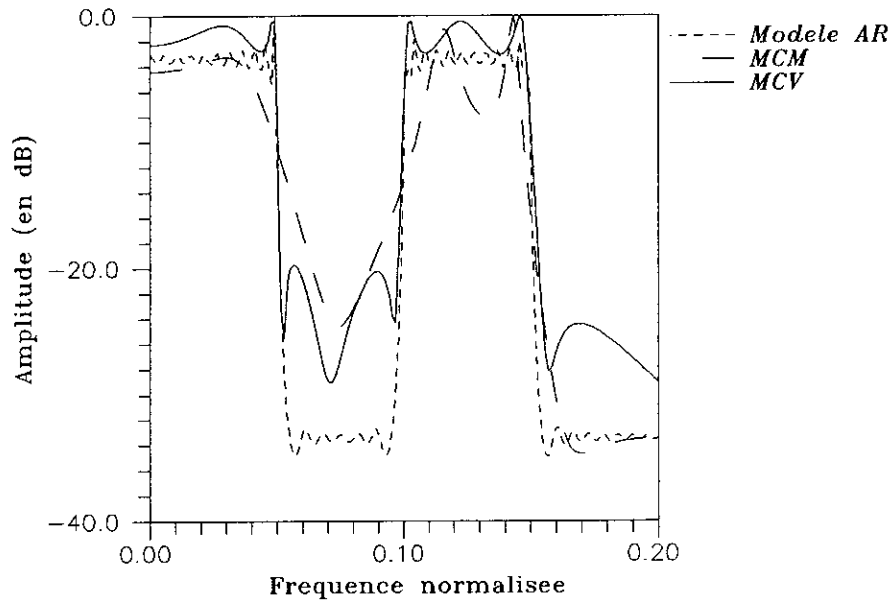


Figure 5.10-a: Spectres d'amplitude des filtres passe-bandes AR(190) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

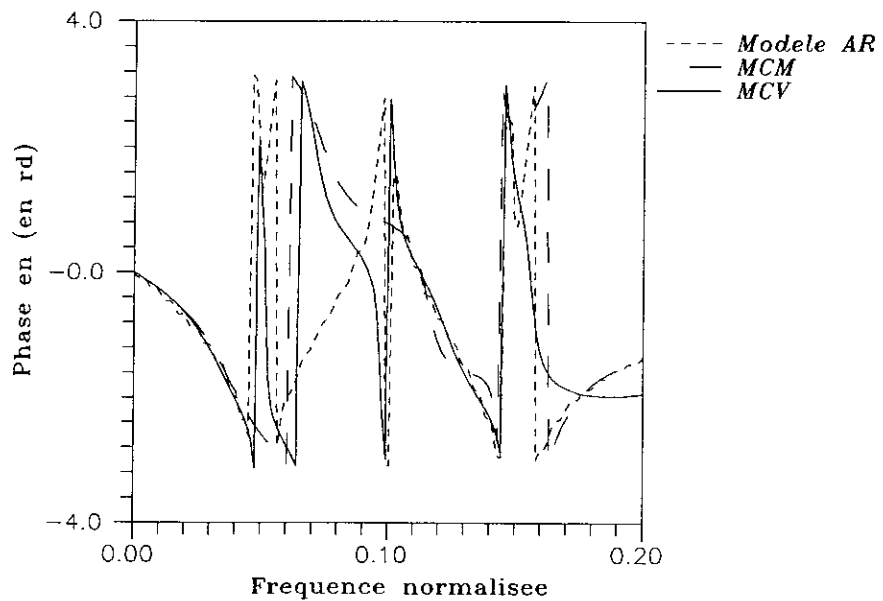


Figure 5.10-b: Spectres de phase des filtres passe-bandes AR(190) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

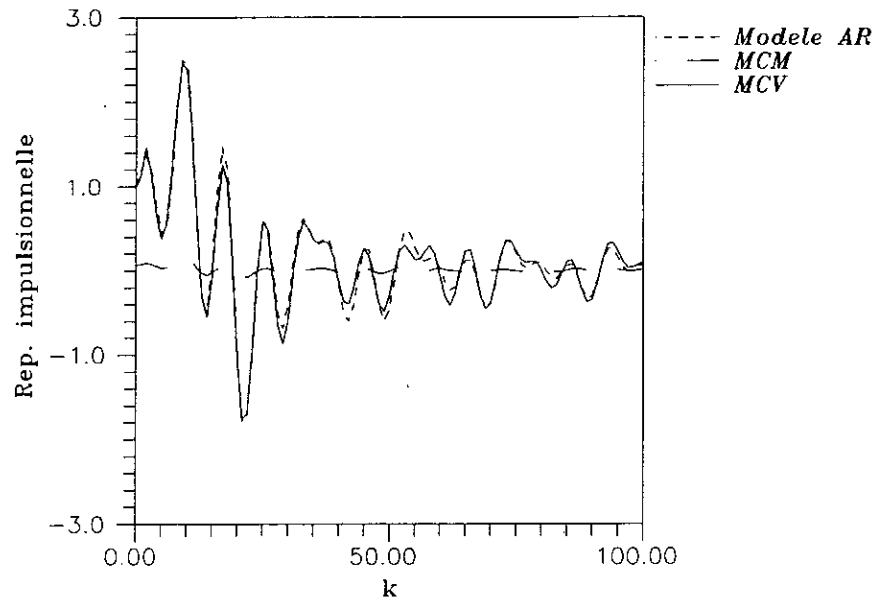


Figure 5.10-c: Réponses impulsionnelles des filtres passe-bandes AR(190) et ses approximations ARMA(10,10) synthétisés par les méthodes MCM et MCV.

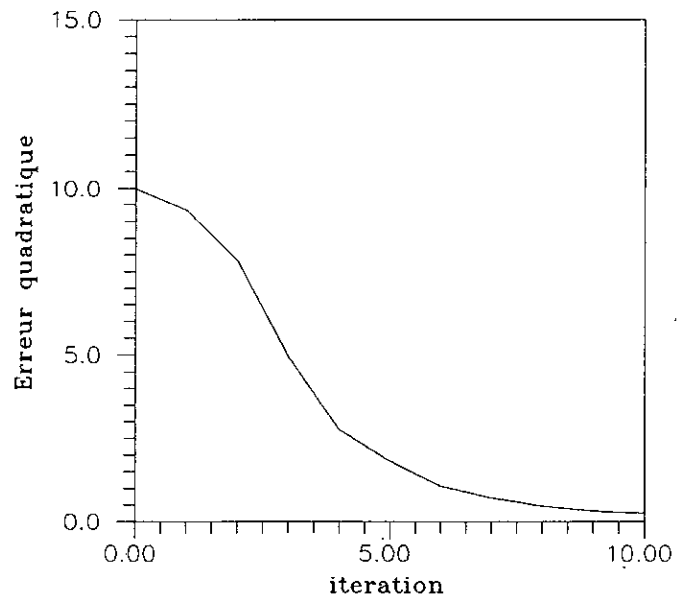


Figure 5.10-d: Evolution de l'erreur MCV.

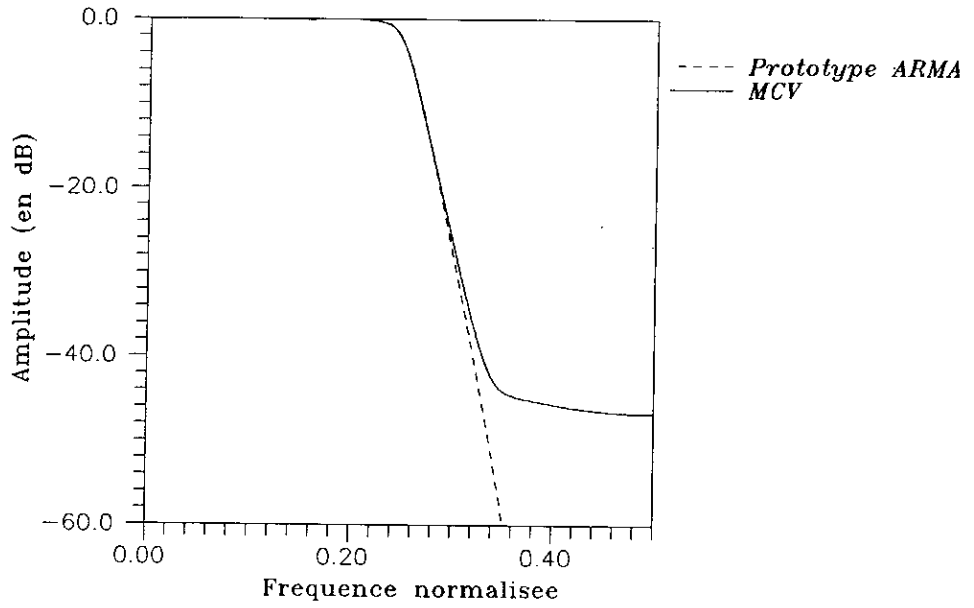


Figure 5.11-a: Spectres d'amplitude du filtre prototype passe-bas de Butterworth et son approximation ARMA(6,6) synthétisé par la méthode MCV.

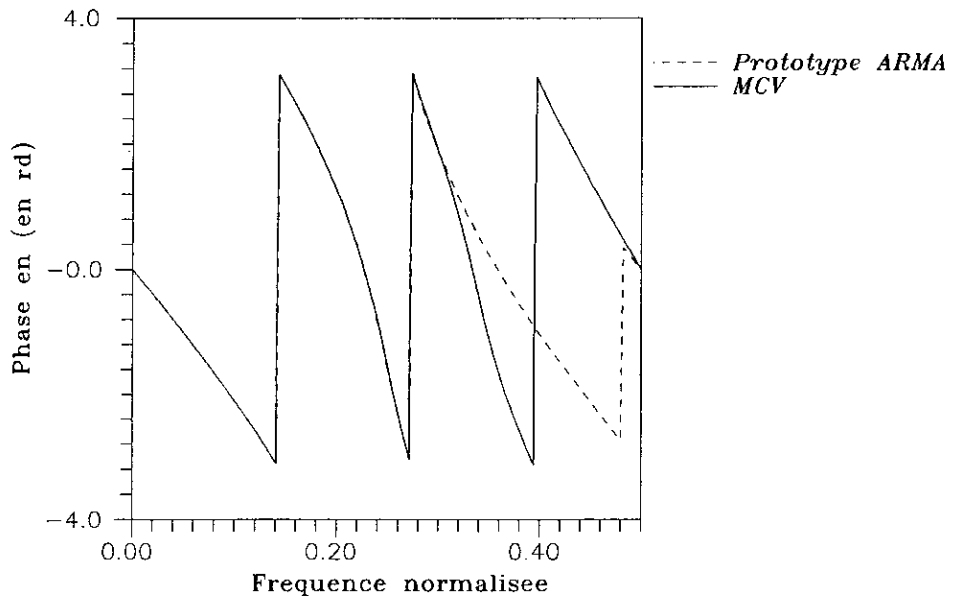


Figure 5.11-b: Spectres de phase du filtre prototype passe-bas de Butterworth et son approximation ARMA(6,6) synthétisé par la méthode MCV.

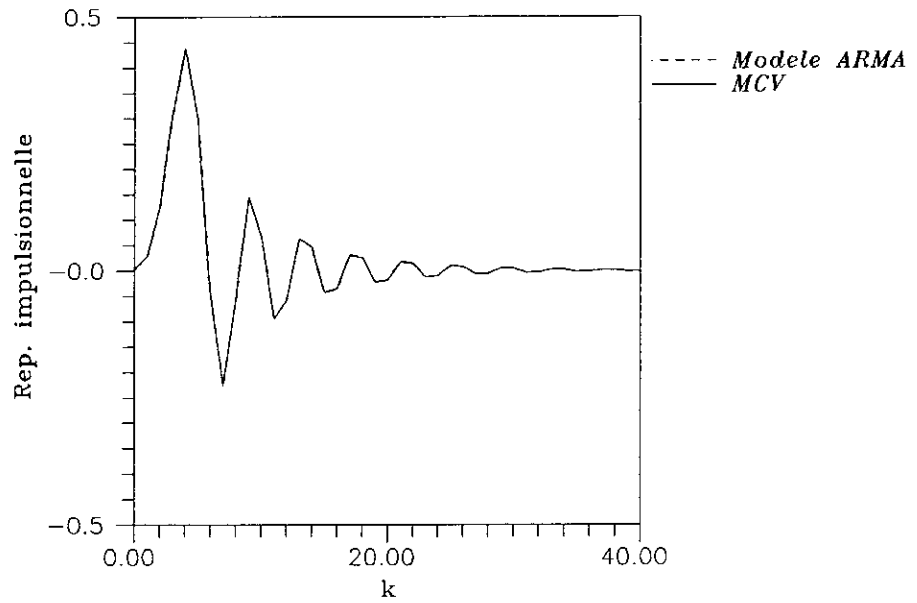


Figure 5.11-c: Réponses impulsionnelles du filtre prototype passe-bas de Butterworth et son approximation ARMA(6,6) synthétisé par la méthode MCV.

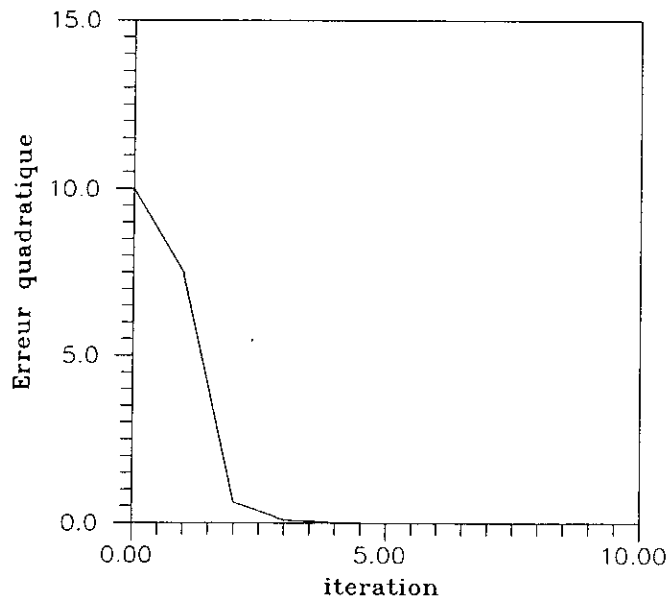


Figure 5.1-d: Evolution de l'erreur MCV.

Conclusion Générale

Le filtrage numérique occupe actuellement une place importante dans la technologie et son emploi se généralise au fur et à mesure que les techniques numériques se développent et remplacent progressivement leurs homologues analogiques.

De ce fait, les filtres numériques bénéficient d'une attention particulière, et la recherche, l'amélioration ou l'étude des méthodes de leur synthèse reste toujours d'actualité.

Dans le cadre de ce travail, nous avons étudié trois méthodes de synthèse de filtres numériques RII.

- *La méthode de Padé.*
- *La méthode de moindres carrées modifiées (MCM).*
- *La méthode de moindres carrées vrais (MCV).*

où les deux dernières sont basées sur le principe d'optimisation par le critère des moindres carrés.

La méthode MCV, objet de cette étude utilise également un processus itératif qui engendre un coût supplémentaire en temps de calcul lors des opérations de synthèse. Cependant, cet inconvénient est largement compensé par l'économie réalisée pour leur implementation et ce en raison du nombre réduit de coefficients les caractérisant.

D'après les résultats obtenus, nous pouvons conclure que la méthode M.C.V nous permet la synthèse de filtres, stables avec un nombre minimum de coefficients dont les performances égalent celles de filtres d'ordre plus élevé (jusqu'au double) obtenus par les méthodes de Padé et des moindres carrés modifiés (MCM).

L'insuffisance de l'espace mémoire disponible sur micro-ordinateur est le problème majeur rencontré lors de la mise en œuvre des programmes de synthèse des filtres. En effet le calcul des fonctions d'auto-corrélation et d'inter-corrélation exige

la résolution de systèmes d'équations de taille $(N + n + 1) \times (N + n + 1)$ où N représente l'ordre du filtre désiré et n celui du filtre à synthétiser.

La caractéristique du filtre RII synthétisé par la méthode des moindres carrés vrais est étroitement liée à celle du filtre désiré utilisé au départ qui elle, dépend directement de N .

Donc pour avoir un filtre avec de bonnes performances, l'ordre N de $H(z)$ doit être élevé car cette fonction est prise comme référence, mais ceci va aussi augmenter la taille des systèmes d'équations et par conséquent, l'espace mémoire occupé sera d'autant plus important.

Afin d'augmenter les performances des programmes, des solutions basées sur l'emploi d'algorithmes efficaces pour la résolution des systèmes d'équations ont été employées, particulièrement pour le calcul des fonctions d'auto-corrélation et d'inter-corrélation (Algorithmes de Jury et de Sylvester).

Les résultats obtenus sont satisfaisants et nous permettent de penser que le recours à des méthodes de programmation permettant une meilleure gestion de l'espace mémoire sont à envisager.

Bibliographie

- [1] A.Antoniou, *Digital Filters Analysis and Design*, Mc Graw-Hill. 1979.
- [2] M.F.Belbachir, J.Caelen, "Conception de Filtres Récursifs à Phase Linéaire," *Recherches, Traitement du Signal*, volume 5, No.5, pp.151-158, 1988.
- [3] M.Bellanger, *Traitement Numérique du Signal Théorie et Pratique*, Masson, 1990.
- [4] A.Belouchrani et L.Mitiche, "Etude Unifiée des Transformations Fréquentielles Numériques et Application a Quelques Filtres Prototypes" *Projet de fin d'études, Ingéniorat, E.N.P Alger, juin 1991*.
- [5] Y.Bistriz, "A Stability New Test for Linear Discret Systems in a Table Form," *IEEE Trans. Circuits and Systems, CAS.30, No.12, pp. 917-919, Dec. 1983*.
- [6] R.E.Blahut, *Fast Algorithms for Digital Signal Processing*, Addison Wesley, 1987.
- [7] H.J.Blinchikoff and A.Zverev, *Filtering in the Time and Frequency Domain*, John Wiley & Sons, 1979.
- [8] R.E.Bogner and A.G.Constantinides, *Introduction to Digital Filtering*, John Wiley & Sons, 1977.
- [9] R.Boite et M.Kunt, *Traitement de la Parole*, Presses Polytechniques Romandes, 1987.
- [10] R.Boite et H.Leich, *Les Filtres Numériques*, Masson, 1982.
- [11] C.A.Bozzo, *Le Filtrage Optimal et ses Applications aux Problèmes de Pour-suites*, Masson, 1980.
- [12] F.Brophy and A.C.Salazar, "Considerations of the Padé Approximant Technique in the Synthesis of Recursive Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-21, pp. 500-505, Dec. 1973.

- [13] J.A.Cadzow, "Recursive Digital Filter Synthesis Via Gradient Based Algorithms," *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. ASSP-24, pp. 349-355, October 1976.
- [14] M.A.Clements and J.W.Pease, "On Causal Linear Phase IIR Digital Filters," *IEEE Trans. Acoustics Speech, Signal Process.*, Vol. ASSP, pp. 479-482, April 1989.
- [15] R.E.Crochiere and L.R.Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, 1983.
- [16] A.Da Silva Passos, *Méthodes Mathématiques du Traitement Numérique du Signal*, Eyrolles, 1989.
- [17] D.De Brucq et G.Folliot, *Théorie du Signal Modélisation Statistique Automatique et Traitement*, Masson, 1988.
- [18] F.De Coulon, *Théorie et Traitement des Signaux*, Presses Polytechniques Romandes, 1984.
- [19] C.J.Demeure and C.T.Mullis, "On the Fast Computation of Cross-covariance and Auto-covariance Sequences," *Electrical and Computer Engineering Department, Univ. of Colorado, Boulder*.
- [20] C.J.Demeure and C.T.Mullis, "The Euclid Algorithm and the Fast Computation of Cross-covariance and Auto-covariance Sequences," *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. 37, April 1989.
- [21] B.Derras, "The Approximation of Discrete Time Linear Systems Based on the Use of Projections," *Ph.D Thesis, Dept. of Electrical and Computer Engineering, Univ. of Colorado, Boulder, Co, 1988*.
- [22] P.Fondanèche et P.Gilbertas, *Filtres Numériques Principes et Réalisations*, Masson, 1981.
- [23] R.A.Gabel and R.A.Roberts, *Signals and Linear Systems*, John Wiley & Sons, 1987.
- [24] A.A.Giordano and F.M.Hsu, *Least Square Estimation with Application to Digital Signal Processing*, John Wiley & Sons, 1985.
- [25] A,E.R.Hanafi "Application de la Prédiction Linéaire à la synthèse des Filtres Numériques," *Thèse de Docteur Ingénieur, Ecole Nationale Supérieure des Télécommunications, Dept. Systèmes et Communications, 1982*.
- [26] R.Hastings-James and S.K.Mehra, "Extension of the Padé Approximant Technique for the Design of Recursive Digital Filters," *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol ASSP-25, pp. 501-509, Dec. 1977.

- [27] S.Haykin, *Communication System*, John Wiley & Sons, 1983
- [28] M.Kunt, *Traitement Numérique des Signaux*, Dunod, 1981
- [29] M.Labarrère, J.P.Krief et B.Gimonet, *Le Filtrage et ses Applications*, Cepadues Editions, 1988.
- [30] H.Y-F.Lam, *Analog and Digital Filters Design and Realisation*, Prentice Hall, 1979.
- [31] B.Liu, *Digital Filters and the Fast Fourier Transform*, Prentice Hall, New Jersey, 1978.
- [32] J.Max et col., *Méthodes et Techniques de Traitement du Signal et Applications aux Mesures Physiques, (Tome 1)*, Masson 1981.
- [33] R.N.McDonough and W.H.Huggins, "Best Least-Squares Representation of Signals by Exponentials," *IEEE Trans. Automatic, Control*, Vol AC-13, pp. 408-412, August 1968.
- [34] G.Miller, "Least-Squares Rational Z-transform Approximation," *Journal of the Franklin Institute*, Vol. 295, January 1973.
- [35] C.T.Mullis and C.J.Demeure, "The Jury Matrix and a Newton-Raphson Procedure for MA Spectral Factorization," *Proc. XIXth Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, Cal., Nov 1986.
- [36] C.T.Mullis and R.A.Roberts, "The Use of Second Order Information in the Approximation of Discrete Time Linear Systems," *IEEE Trans Acoustics, Speech, Signal Process.*, Vol. ASSP-24, pp. 226-238, June 1976.
- [37] M.Najim, *Modélisation et Identification en Traitement du Signal*, Masson, 1988.
- [38] A.V.Oppenheim and R.W.Schafer, *Digital Signal Processing*, Prentice Hall, New Jersey, 1975.
- [39] L.R.Rabiner and B.Gold, *Theory and Application of Digital Signal Processing*, Prentice Hall, New Jersey, 1975.
- [40] R.A.Roberts and C.T.Mullis, *Digital Signal Processing*, Addison Wesley, Readings, Massachusetts, 1987.
- [41] L.L.Scharf and J.C.Luby, "Statistical Design of Auto-regressive Moving Average Digital Filters," *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. ASSP-27, pp. 240-247, June 1979.
- [42] K.Steiglitz and L.E.McBride, "A Technique for the Identification of Linear Systems," *IEEE Trans. Automatic, Control*, pp. 461-464, October 1965.

- [43] P.Stoica and T.Soderstrom, em "The Steiglitz - McBride Algorithm Revisted Convergence Analysis and Accuracy Aspects," *IEEE Trans. Automatic Control*, Vol. AC-26, pp. 712-717, June 1981.
- [44] T.J.Terrell, *Introduction to Digital Filters*, McMillan, 1980.
- [45] C.S.Williams, *Designing Digital Filters*, Prentice Hall, New Jersey, 1986.
- [46] T.Yoshida, A.Nishihara and N.Fujii "Magnitude and Phase Approximation of IIR Digital Filters Using Linear Programming," *Trans. of IEICE*, vol. E72, No.10, pp. 1097-1103, October 1989.
- [47] M.Zait et S.Chouai, "Etude et Simulation des Filtres Numériques RIF Synthétisés par la Méthode de Fenêtrage." *Projet de fin d'études, Ingéniorat, E.N.P Alger, Juillet 1993.*

Annexe A

Algorithme de Horner

Cet algorithme permet l'évaluation rapide d'un polynôme $P(z)$ d'ordre n de la forme

$$P(z) = \sum_{i=0}^n a_i z^{n-i} \quad z = e^{j\omega}$$

ayant : a_0, a_1, \dots, a_n et θ
on cherche : α et β tels que :

$$P(e^{j\theta}) = \alpha + j\beta$$

Initialisation :
 $C = \cos \theta$
 $S = \sin \theta$
 $D = 2C$
 $V_1 = 0$
 $V_2 = 0$

Iterations : De $k = 0$ a n faire
 $V_3 = a_k - V_1 + DV_2$
 $V_1 \leftarrow V_2$
 $V_2 \leftarrow V_3$
(fin de la boucle en k)

$$\alpha = V_2 - CV_1$$
$$\beta = SV_1$$

Annexe B

Algorithme de Levinson-Durbin

Initialisation :

$$\begin{aligned}y^{(1)} &= R_x(1)/R_x(0) = y(0) \\h(0) &= R_{xp}(0)/R_x(0) \\h^{(1)} &= h(0) = R_{px}(0)/R_x(0) \\\beta_0 &= R_x(0) \\\alpha_0 &= R_x(1)/R_x(0) = y(0)\end{aligned}$$

faire : De $k = 1$ a $N - 1$ faire :

$$\begin{aligned}\beta_k &= (1 - \alpha_{k-1}^2)\beta_{k-1} \\h(k) &= \frac{R_{px}(0) - r^{(k)t} J_k h^{(k)}}{\beta_k} \\\nu^{(k)} &= h^{(k)} - J_k y^{(k)} h(k) \\h^{(k+1)} &= \begin{pmatrix} \nu^{(k)} \\ h(k) \end{pmatrix}\end{aligned}$$

Si $k < N - 1$ alors :

$$\begin{aligned}y(k) &= \alpha_k = \frac{R_x(k+1) - r^{(k)t} J_k y^{(k)}}{\beta_k} \\Z^{(k)} &= y^{(k)} - \alpha_k J_k y^{(k)} \\y^{(k+1)} &= \begin{pmatrix} Z^{(k)} \\ y(k) \end{pmatrix}\end{aligned}$$

(fin de la boucle en k)

Annexe C

Algorithme de Mullis-Roberts

Initialisation :

$$\begin{aligned}\alpha_0 &= r_0 - h_0^2 \\ a_0(0) &= 1 \\ b_0(0) &= 1/\alpha_0 \\ c_0(0) &= h_0/\alpha_0 \\ \delta_0 &= 1 + h_0^2/\alpha_0\end{aligned}$$

recursion :

$$\begin{aligned}\beta_n &= \sum_{k=0}^n a_k(n)r_{n+1-k} \\ \gamma_n &= -\sum_{k=0}^n a_k(n)h_{n+1-k} \\ \theta_n &= \beta_n b_0(n) + \gamma_n c_0(n) \\ \phi_n &= \beta_n c_0(n) + \gamma_n \delta_n \\ \alpha_{n+1} &= \alpha_n - \beta_n \theta_n - \gamma_n \phi_n \\ \delta_{n+1} &= \delta_n + \phi_n^2/\alpha_{n+1}\end{aligned}$$

De $k = 0$ a $n + 1$ faire :

$$\begin{aligned}a_k(n+1) &= a_k(n) - \beta_n b_{n+1-k}(n) - \gamma_n c_{n+1-k}(n) \\ b_k(n+1) &= b_k(n) - (\theta_n/\alpha_{n+1}) \cdot a_{n+1-k}(n+1) \\ c_k(n+1) &= c_k(n) - (\phi_n/\alpha_{n+1}) \cdot a_{n+1-k}(n+1)\end{aligned}$$

(fin de la boucle en k)

Annexe D

Algorithme de Levinson à deux canaux (bicanal)

Données :

$$R_k = \begin{pmatrix} p_k & -\eta_k \\ -\eta_{-k} & q_k \end{pmatrix} \quad k = 0, 1, \dots, n$$

Initialisation : $\tilde{\alpha}_0 + \tilde{\beta}_0 = R_0$
 $A_0(0) = B_0(0) = I_2$, $X_0 = R_1$

faire : De $k = 0$ a $n - 1$ faire :

$$\begin{aligned} C_{k+1} &= -X_k \tilde{\beta}_k^{-1} \\ D_{k+1} &= -X_k^t \tilde{\alpha}_k^{-1} \\ \tilde{\alpha}_{k+1} &= \tilde{\alpha}_k + C_{k+1} X_k^t \\ \tilde{\beta}_{k+1} &= \tilde{\beta}_k + D_{k+1} X_k \end{aligned}$$

De $i = 1$ a $k + 1$ faire :

$$\begin{aligned} A_i(k+1) &= A_{i-1}(k) + C_{k+1} B_{i-1}(k) \\ B_i(k+1) &= D_{k+1} A_{i-1}(k) + B_{i-1}(k) \end{aligned}$$

(fin de la boucle en i)

$$X_{k+1} = \sum_{i=0}^{k+1} A_i(k+1) R_{k+2-i}$$

(fin de la boucle en k)

Annexe E

Algorithme de Sylvester

Cet algorithme permet la résolution d'un système d'équations de Sylvester.

Donnees : $a[0, n_1], c[0, n_2], y[0, n_1 + n_2]$ avec $n_1 \geq n_2$ et $a[0] = c[0] = 1$

Initialisation : $a^{n_1}[0, \dots, n_1] = a[0, \dots, n_1], c^{n_2}[0, \dots, n_2] = c[0, \dots, n_2],$

Factorisation : De $i = n_1 - 1$ a n_2 faire

De $j = 0$ a $i + 1$ faire

$$a^i[j] = a^{i+1}[j] - a^{i+1}[i + 1].c[i - j + 1]$$

(fin de la boucle en j)

(fin de la boucle en i)

De $i = n_2$ a 1 faire

$$\alpha \leftarrow 1 - a^i[i].c^i[i]$$

De $j = 0$ a i faire

$$a^{i-1}[j] = (a^i[j] - a^i[i].c^i[i - j])/\alpha$$

$$c^{i-1}[j] = (c^i[j] - c^i[i].a^i[i - j])/\alpha$$

(fin de la boucle en j)

(fin de la boucle en i)

Solution : De $i = n_1$ a $n_2 + 1$ faire

De $j = n_1 + 1$ a $n_1 + n_2$ faire

$$y[j] \leftarrow y[j] - a^i[i].y[j - i]$$

(fin de la boucle en j)

(fin de la boucle en i)

De $i = n_2$ a 1 faire

$$\alpha \leftarrow 1 - a^i[i].c^i[i]$$

De $j = n_1 - i + 1$ a n_1 faire
 $temp \leftarrow y[j]$
 $y[j] \leftarrow (y[j] - c^i[i].y[j+i])/\alpha$
 $y[j+i] \leftarrow (y[j+i] - a^i[i].temp)/\alpha$
 (fin de la boucle en j)
 (fin de la boucle en i)

De $i = n_1 + 2$ a $n_1 + n_2$ faire

$$y[i] \leftarrow y[i] - \sum_{j=n_1+1}^{i-1} a^{i-n_1-1}[i-j].y[j]$$

(fin de la boucle en i)

De $i = n_1 - 1$ a 1 faire

$$k \leftarrow \min(n_1 - i, n_2)$$

$$y[i] \leftarrow y[i] - \sum_{j=i-1}^{n_1} c^k[j-i].y[j]$$

(fin de la boucle en i)

Annexe F

Algorithme de Jury

Cet algorithme permet la résolution d'un système d'équations de Jury.

Donnees : $a[0, n], c[0, y[0, n]$ avec $a[0] = 1$

Initialisation : $a^n[0, \dots, n] = a[0, \dots, n]$

Factorisation : De $i = n - 1$ a 1 faire

$$\alpha \leftarrow 1 - a^{i+1}[i + 1].a^{i+1}[i + 1]$$

De $j = 1$ a i faire

$$a^i[j] = (a^{i+1}[j] - a^{i+1}[i + 1].a[i - j + 1])/\alpha$$

(fin de la boucle en j)

(fin de la boucle en i)

Solution : De $i = n$ a 1 faire

$$\alpha \leftarrow 1 - a^i[i].a^i[i]$$

De $j = 0$ a $i/2$ faire

$$temp \leftarrow y[j]$$

$$y[j] \leftarrow (y[j] - a^i[i].y[i - j])/\alpha$$

(fin de la boucle en j)

(fin de la boucle en i)

De $i = n_2$ a 1 faire

$$\alpha \leftarrow 1 - a^i[i].c^i[i]$$

De $j = n_1 - i + 1$ a n_1 faire

$$temp \leftarrow y[j]$$

$$y[j] \leftarrow (y[j] - c^i[i].y[j + i])/\alpha$$

$$y[j + i] \leftarrow (y[j + i] - a^i[i].temp)/\alpha$$

$$\text{si } j \neq i - j \text{ alors } y[i - j] \leftarrow (y[i - j] - a^i[i].temp)/\alpha$$

(fin de la boucle en j)
(fin de la boucle en i)

De $i = 2$ a n faire

$$y[i] \leftarrow y[i] - \sum_{j=1}^{i-1} a^{i-1}[j] \cdot y[i-j]$$

(fin de la boucle en i)