

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE**



UNIVERSITE DE DJELFA – ZIANE ACHOUR –

Département d'Electronique

**Mémoire de Magister en Electronique
Option : signaux et systèmes**

**Analyse et Synthèse des Filtres RII 2-D
d'Ordre réduit**

par **KACEM Omar**

Ingénieur d'Etat en Electronique

Soutenu le :

Devant le jury :

- | | |
|----------------------------|---|
| • A. Gueddim (M.C.) | (Président, Univ. de Djelfa) |
| • L. Mitiche (M.C.) | (Directeur de thèse, Univ. de Djelfa) |
| • A. Mitiche (M.C.) | (Co-directeur de thèse, Univ. de Djelfa) |
| • M. Bensebti (Pr.) | (Examineur, Univ. de Blida) |
| • D. Mahi (Pr.) | (Examineur, Univ. de Laghouat) |

Remerciements

J'adresse mes remerciements à Monsieur Dr. Lahcene Mitiche de m'avoir proposé et d'avoir dirigé ce travail .

Je remercie vivement Monsieur Gueddin d'avoir accepté de présider le jury, ainsi que Messieurs les professeurs M. Bensebti de l'Université de Blida et D. Mahi de l'Université de Laghouat, d'avoir accepté d'examiner mon travail.

Je tiens à remercier Mme Dr. Amel Mitiche, pour sa formation en matière d'Automatique et son aide précieuse durant ma recherche.

Cette page ne saurait être complète sans remercier mes camarades M. Boranan, B. Assali, Y. Khenin, S. Larbi, et F. Mamri.

Enfin, je tiens à remercier tous ceux qui ont contribués de près ou loin, à la concrétisation de ce travail.

Dédicace

Je dédie tout d'abord ce modeste travail à
mes très chers parents

A mes chers frères et chère sœur
chacun à son nom

A tous mes amis et à toutes
les personnes que j'aime et
qui m'aiment.

Résumé

Le travail présenté dans ce mémoire concerne l'analyse et la synthèse des filtres à réponse impulsionnelle infinie bidimensionnelle (RII 2-D) d'ordre réduit. La synthèse est faite dans les deux domaines, spatial (temporel) avec deux méthodes, la méthode de Prony (Prony modifié) et méthode Itérative, et dans le domaine fréquentiel on utilise la méthode de programmation itérative semi-définie (SDP). Après la synthèse on fait une réduction d'ordre du système par la méthode de Quasi-Gramians.

Mots clés : Stabilité, Calcul récursive, Phase linéaire, SDP, Espace d'état, Réalisation équilibrée, Réduction de modèles.

Abstract

The work presented in this memory concerns model order reduction of the two-dimensional infinite impulse response filters (2-D IIR). The synthesis is performed in the two domains, spatial (temporal) with two methods, Prony's method (modified Prony), Iterative method, and in the frequency domain we use the method of Semi-Definite iterative Programming (SDP). After the synthesis, we make an order reduction of the model by Quasi-Gramians method.

Key words: Stability, Recursively computation, Linear phase, SDP, State space, Balanced realization, Model order reduction.

ملخص

العمل المبين في هذه الرسالة معني بتحليل وتركيب المرشحات ثنائية البعد ذات الاستجابة غير المحدودة ذو الدرجة المخفضة. تم التركيب في مجالين: في المجال الزمني بواسطة طريقتين، طريقة برونى (برونى المعدلة) وطريقة التكرار. أما في مجال الترددات استعملنا طريقة البرمجة المكررة نصف المعرفة. بعد عملية التركيب قمنا بتخفيض درجة المرشح باستعمال طريقة شبه الغراميان. كلمات مفتاحيه : استقرار، حساب تراجمي، طور خطي، ب ن م، فضاء الحالة، تحقيق متوازن، نموذج مخفض الدرجة.

Notation

$x(n_1, n_2)$: Séquence (signal) bidimensionnelle.

n_1, n_2 : Variables temporelles discrètes.

$\delta(n_1, n_2)$: Impulsion du Kronecker.

$h(n_1, n_2)$: Réponse impulsionnelle.

R : Région du support.

z_1, z_2 : Variables complexes.

ROC : Région de convergence (*region of convergence*).

$H(w_1, w_2)$: Réponse fréquentielle.

w_1, w_2 : Pulsations, Variables contenues.

$ARMA$: Modèle Auto régressif à Moyenne Adaptée.

p, q : Nombre des paramètres inconnus.

SDP : Programmation semi-définie (Semidefinite programming).

LMI : Matrices d'inégalité linéaire.

g_k : Gradient à l'itération k .

$F(x)$: Matrice semi-définie positive.

Ω : Espace fréquentielle.

$W(w_1, w_2)$: Fonction de pondération.

\Re : Partie réel.

P : Matrice définie positive.

I : Matrice identité.

ε : Prescriptive tolérance.

x^h : Vecteur d'état horizontal de dimension n .

x^v : Vecteur d'état vertical de dimension m .

u : Vecteur d'entrée de dimension l .

y : Vecteur de sortie de dimension p .

T : Matrice de transformation du modèle.

(r_1, r_2) : Ordre de modèle réduit .

$\{A, B, C, d\}$: Matrices d'état du modèle numérique d'ordre (n, m) .

$\{A_r, B_r, C_r, d\}$: Matrices d'état du modèle numérique d'ordre réduit (r_1, r_2) .

Sommaire

INTRODUCTION GENERALE

Chapitre 1 Signaux et Systèmes Numériques Bidimensionnels

1.1 Introduction	6
1.2 Signaux et Systèmes Bidimensionnelles	6
1.2.1 Les Signaux	6
1.2.2 Les Systèmes	8
1.3 Convolution 2-D	9
1.4 Systèmes stables et systèmes de support spécial	10
1.4.1 Systèmes stables	10
1.4.2 Systèmes de support spécial (causalité)	10
1.5 Équation aux différences à coefficients constants linéaires	11
1.5.1 Équations aux différences dans systèmes linéaires et invariant dans le temps	12
1.5.2 Calcul récursive	12
1.6 Transformée en z	13
1.6.1 Propriétés de la transformée en z 2-D	13
1.6.2 Transformée en z inverse 2-D	15
1.7 Fonctions transfert d'un système 2-D	15
1.8 Transformée de Fourier 2-D	16
1.8.1 Propriétés de la transformée de Fourier	16
1.9 Conclusion	18

Chapitre 2 Caractéristiques des Filtrés Numériques Bidimensionnels

2.1 Introduction	19
2.2 Filtrés à phase linéaire	19
2.3 Caractéristiques des filtrés	21
2.4 Classification des filtrés numériques	22
2.5 Comparaison des filtrés	23
2.6 Conclusion	23

Chapitre 3 Synthèse des Filtrés Numériques RII 2-D

3.1 Introduction	24
3.2 Problème de synthèse	24
3.3 Problème de stabilité	25
3.3.1 Théorème de stabilité	26
3.4 Synthèse dans le domaine spatial	27
3.4.1 Algorithme linéaire de forme fermée	27
a) <i>La méthode de Prony</i>	29
b) <i>Algorithme Itérative</i>	31
c) <i>Synthèse de filtre à phase linéaire</i>	32
3.5 Synthèse dans le domaine fréquentielle	34
3.5.1 Synthèse par la programmation itérative semi-définie	34
3.6 Conclusion	38

Chapitre 4**Réduction d'ordre**

4.1 Introduction.....	39
4.2 Modèle d'espace d'état	39
4.2.1 Représentation du modèle dans l'espace d'état	42
4.2.2 Réalisation minimale dans l'espace d'état.....	43
4.3 Dérivation de la matrice de la fonction transfert à partir du modèle d'état	46
4.4 Méthodes de réduction.....	49
4.4.1 Trois types de gramians pour les systèmes discrets 2-D.....	50
4.5 Algorithme itérative pour le cas Quasi-Gramians	52
4.6 Conclusion.....	53

Chapitre 5**Simulation et interprétation**

5.1 Introduction.....	54
5.2.1 Synthèse : Méthode de Prony (Prony Modifiée)	55
5.2.2 Synthèse : Méthode Itérative.....	60
5.2.3 Synthèse : Méthode SDP.....	65
5.3 Réduction	70
5.3.1 Réduction: Méthode Itérative.....	70
5.3.2 Réduction : Méthode SDP.....	75
5.4 Interprétation et conclusion	80

CONCLUSION GENERALE.....	81
---------------------------------	-----------

<i>Références Bibliographiques.....</i>	82
--	-----------

INTRODUCTION GENERALE

Les champs du traitement numérique du signal bidimensionnel et d'image numérique ont maintenu la vitalité énorme pendant les dernières quatre décennies et il y a chaque indication que cette tendance continuera. Amélioration en technologie fournissent des puces et des microprocesseurs de traitement des signaux qui ont été précédemment associés aux ordinateurs. Ces avances permettent le traitement des signaux sophistiquées et les algorithmes à traitement d'images à mettre en application en temps réel à un coût sensiblement réduit.

Des nouvelles applications continuent à être trouvées et les applications existantes continuent à augmenter dans des secteurs divers tels que des communications, électronique grand public, la médecine, la robotique, et la géophysique. Avec des avances en technologie de matériel et l'expansion dans les applications, de nouveaux algorithmes sont développés et des algorithmes existants mieux compris, qui en retour aboutissent à autre expansion dans les applications et donner un grand motif pour d'autres avances en technologie de matériel.

A un niveau conceptuel, il y a beaucoup de similitude entre le traitement des signaux unidimensionnels et le traitement des signaux bidimensionnels. Dans le traitement des signaux unidimensionnels, les concepts discutés sont le filtrage, transformé de Fourier, transformée de Fourier discrète, et ainsi de suite. Dans le traitement des signaux bidimensionnels, nous encore sommes concernés par les mêmes concepts. Par conséquent, les concepts généraux que nous développons dans le traitement du signal bidimensionnel peuvent être visualisés en tant qu'extensions directes des résultats dans le traitement du signal unidimensionnel.

A un niveau d'analyse détaillé, cependant, une grande différence existe entre le traitement du signal unidimensionnel et bidimensionnel. Par exemple, une différence principale est le nombre de données concerné dans des applications typiques (dans le traitement de la parole, en général 10.000 points d'informations traité dans une seconde, cependant, dans le traitement de vidéo, nous aurions 7.5 millions de points d'informations au traiter par seconde).

Une autre différence principale vient du fait que les mathématiques utilisées pour le traitement du signal unidimensionnel sont souvent plus simples que cela utilisée pour le traitement du signal bidimensionnel. Par exemple, beaucoup de systèmes unidimensionnels sont décrits par des équations différentielles, alors que beaucoup de systèmes bidimensionnels sont décrits par des équations aux différences partielles, généralement beaucoup plus facile de résoudre des équations différentielles que des équations aux différences partielles.

Un autre exemple est l'absence du théorème fondamental de l'algèbre pour des polynômes bidimensionnels. Pour des polynômes unidimensionnels peut être factorisé comme produit des polynômes de 1^{ère} (2^{ème}) ordre, la transformée en z de la réponse impulsionnelle du filtre numérique est factorisée comme produit des polynômes de 1^{ère} (2^{ème}) ordre et implémenter ce filtre en cascade. La transformée en z de la réponse impulsionnelle d'un filtre numérique bidimensionnel ne peut pas, généralement être factorisée comme un produit des polynômes et la structure en cascade n'est pas donc une structure générale pour une réalisation de filtre numérique bidimensionnelle.

Une autre conséquence de la difficulté de factorisation d'un polynôme bidimensionnel est la difficulté de la stabilité du système. Dans un système unidimensionnel, les emplacements des pôles peuvent être déterminés facilement, et un système instable peut être stabilisé sans affecter la réponse d'amplitude par une simple manipulation des emplacements de pôles. Dans un système bidimensionnel, parce que les pôles sont des surfaces plutôt que des points et n'est aucun théorème fondamental d'algèbre, il est extrêmement difficile de déterminer les emplacements des pôles.

En conséquence, le contrôle de la stabilité d'un système bidimensionnel et la stabilisation d'un système bidimensionnel instable sans affecter la réponse d'amplitude sont extrêmement difficiles. [1]

Un ensemble de cinq chapitres est présenté dans ce travail pour l'analyse et la synthèse des filtres à réponse impulsionnelle infinie (RII) bidimensionnelle (2-D) d'ordre réduit.

La synthèse est terminée dans les deux domaines, spatial (temporel) avec deux méthodes, la méthode de Prony (Prony modifiée) et méthode Itérative, et Programmation itérative semi-définie (SDP) dans le domaine fréquentielle. Après la synthèse on fait une réduction d'ordre du système par la méthode de Quasi-Gramians.

Le chapitre 1 est consacré à l'étude de signaux et systèmes bidimensionnelle, avec rappels sur la convolution 2-D, la stabilité, les systèmes à support spécial, et sur les propriétés principales de la transformée en z , et transformée de Fourier.

Dans le chapitre 2 on a résumée les caractéristiques générales du filtre numérique bidimensionnel, comme le filtre à phase linéaire, la classification des filtres selon le comportement fréquentiel et temporel.

On a étudié dans la troisième chapitre la synthèse des filtres numériques RII 2-D, et les problèmes que on peut trouver dans la procédure de la synthèse, comme la stabilité et le filtre à phase linéaire. Deux méthodes dans le domaine spatial peuvent résoudre ces problèmes, et un filtre stable et à phase approximativement linéaire peut être obtenue par la méthode de SDP dans le domaine fréquentielle, ensuite une réduction d'ordre fait par la méthode de quasi-Gramians, via la modélisation dans l'espace d'état dans le quatrième chapitre.

On fait les simulations à base du logiciel Matlab (version 7.1) suivis de leurs interprétations dans le chapitre 5.

Une conclusion générale est donnée à la fin de ce travail.

Chapitre 1

Signaux et Systèmes Numériques Bidimensionnels

1.1 Introduction

Un élément majeur en étudiant le traitement du signal est l'analyse, la synthèse, et l'implémentation d'un système qui transforme un signal d'entrée à un signal de sortie plus désirable pour une application donnée. Quand nous développons des résultats théoriques autour des systèmes, nous fréquemment imposons les contraintes des linéarités et invariance dans le temps. Bien que ces contraintes soient très restrictives, les résultats théoriques obtenus ainsi s'appliquent dans la pratique au moins approximativement à beaucoup de systèmes.

1.2 Signaux et systèmes bidimensionnelles

1.2.1 Les Signaux

Les signaux que nous considérons sont des signaux de l'espace discret. Un signal bidimensionnel dans l'espace discret (notée par $x(n_1, n_2)$) représente une séquence qui est défini pour toutes les valeurs entières n_1 et n_2 . Un exemple d'une séquence $x(n_1, n_2)$ est dessinée dans la figure 1.1, dans la figure, le sommet à (n_1, n_2) représente l'amplitude à (n_1, n_2) . Il est difficile de dessiner un signal (séquence) 2-D dans une figure à trois dimensions comme suivant les indications de la figure 1.1.

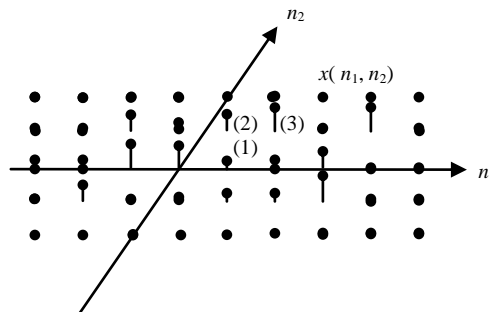


Figure 1.1 Séquence bidimensionnelle $x(n_1, n_2)$.

Beaucoup de séquences que nous utilisons ont des amplitudes de 0 ou de 1 pour de grandes régions de (n_1, n_2) , dans ce cas, des cercles remplis avec les valeurs d'amplitude représentant entre parenthèses, et ont une amplitude de 1 au point sans spécification. La figure 1.2 donne le résultat quand cette simplification supplémentaire est faite à la séquence dans la figure 1.1.

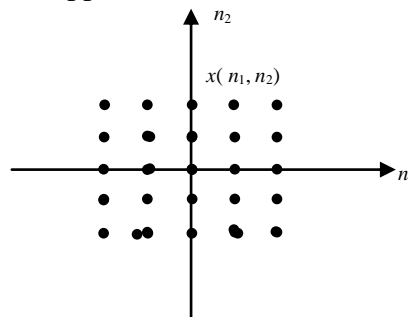


Figure 1.2 Alternatives méthode de dessinée un 2-D signal (séquence).

Exemples des séquences

Quelques classes des séquences jouent un rôle particulièrement important dans le traitement du signal 2-D, tel que l'impulsion du Kronecker.

Impulsion du Kronecker : dénoté par le $\delta(n_1, n_2)$ est défini par:

$$\delta(n_1, n_2) = \begin{cases} 1, & n_1 = n_2 = 0 \\ 0, & \text{ailleurs} \end{cases}$$

l'impulsion $\delta(n_1, n_2)$ est dessinée sur la figure 1.3, joue un rôle assimilé au d'impulsion $\delta(n)$ dans traitement du signal unidimensionnelle.

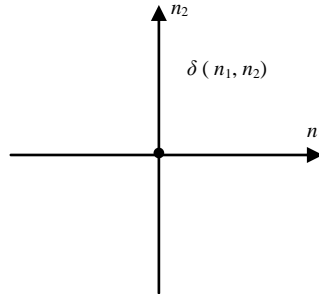


Figure 1.4 Impulsion $\delta(n_1, n_2)$.

N'importe quelle séquence peut être représentée comme combinaison linéaire des impulsions décalées comme suit :

$$x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2) \quad (1.1)$$

La représentation de $x(n_1, n_2)$ par (1.1) est très utile dans l'analyse des systèmes.

Les séquences séparables : une séquence $x(n_1, n_2)$ est dite séparable s'il peut être exprimé comme $x(n_1, n_2) = f(n_1)g(n_2)$ où $f(n_1)$ et $g(n_2)$ sont des fonctions de n_1 et n_2 seulement, respectivement.

Une séquence 2-D typiquement n'est pas une séquence séparable. Malgré le fait que les séquences séparables constituent une classe très spéciale des séquences 2-D, ils jouent un rôle important dans le traitement du signal 2-D. Dans cas où les résultats qui s'appliquent aux séquences 1-D n'étendent pas généralement aux séquences 2-D d'une façon direct, ils font souvent pour les séquences séparables 2-D. Plus que ça, la séparabilité d'une séquence peut être exploitée afin de réduire le calcul dans divers contextes, tels que le filtrage et le calcul de la transformée de Fourier discrète.

Les séquences périodiques : une séquence $x(n_1, n_2)$ est dit périodique avec une période de $N_1 \times N_2$ si $x(n_1, n_2)$ remplit la condition suivante : $x(n_1, n_2) = x(n_1 + N_1, n_2) = x(n_1, n_2 + N_2)$ pour tout les (n_1, n_2) où N_1 et N_2 sont des nombres entiers positifs.

Par exemple, $\cos(\pi n_1 + (\pi/2)n_2)$ est une séquence périodique avec une période de 2×4 , depuis $\cos(\pi n_1 + (\pi/2)n_2) = \cos(\pi(n_1+2) + (\pi/2)n_2) = \cos(\pi n_1 + (\pi/2)(n_2+4))$ pour tout (n_1, n_2) .

1.2.2 Les Systèmes

Système linéaire discret invariant dans le temps (LIT)

Une relation d'entrée-sortie s'appelle un système s'il y a une seule sortie pour n'importe quelle d'entrée donnée.

Un système T qui associe une entrée $x(n_1, n_2)$ à une sortie $y(n_1, n_2)$ est représenté par:

$$y(n_1, n_2) = T[x(n_1, n_2)]$$

Cette définition d'un système est très large. Sans aucune restriction, la caractérisation d'un système n'exige une relation complète d'entrée-sortie.

Connaître la sortie d'un système à un ensemble d'entrées ne nous permet pas généralement de ne déterminer la sortie du système à aucun autre ensemble d'entrées [1].

Deux types de restriction qui simplifient considérablement la caractérisation et l'analyse d'un système sont des linéarités et invariance dans le temps. Dans la pratique, heureusement, beaucoup de systèmes peuvent être approximer par un système linéaire et invariant dans le temps.

La linéarité d'un système T est définie par :

$$\text{Linéarité} \Leftrightarrow T[ax_1(n_1, n_2) + bx_2(n_1, n_2)] = ay_1(n_1, n_2) + by_2(n_1, n_2) \quad (1.2)$$

avec $T[x_1(n_1, n_2)] = y_1(n_1, n_2)$, $T[x_2(n_1, n_2)] = y_2(n_1, n_2)$, a et b sont des constantes,

Cette condition s'appelle le *principe de la superposition*.

L'invariance dans le temps (IT) ou l'invariance dans l'espace d'un système est définie par :

$$\text{Invariant dans le temps} \Leftrightarrow T[x(n_1 - m_1, n_2 - m_2)] = y(n_1 - m_1, n_2 - m_2) \quad (1.3)$$

avec $y(n_1, n_2) = T[x(n_1, n_2)]$ et m_1 et m_2 sont des nombres entiers.

Considérer un système linéaire T . Utiliser (1.1) et (1.2), nous pouvons exprimer la sortie $y(n_1, n_2)$ pour une entrée $x(n_1, n_2)$ par :

$$\begin{aligned} y(n_1, n_2) = T[x(n_1, n_2)] &= T \left[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2) \right] \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) T[\delta(n_1 - k_1, n_2 - k_2)] \end{aligned} \quad (1.4)$$

De (1.4), un système linéaire peut être complètement caractérisé par la réponse du système au d'impulsion $\delta(n_1, n_2)$ et à son de décalages $\delta(n_1 - k_1, n_2 - k_2)$.

Si nous savons $T[\delta(n_1 - k_1, n_2 - k_2)]$ pour toutes les valeurs entières de k_1 et de k_2 , la sortie du système linéaire à n'importe quelle entrée $x(n_1, n_2)$ peut être obtenue à partir (1.4).

La caractérisation de système est encore simplifiée si nous imposons la restriction supplémentaire de l'invariance dans le temps.

Supposer que nous indiquons la réponse d'un système T à une entrée $\delta(n_1, n_2)$ par $h(n_1, n_2)$:

$$h(n_1, n_2) = T[\delta(n_1, n_2)] \quad (1.5)$$

De (1.3) et (1.5),

$$h(n_1 - k_1, n_2 - k_2) = T[\delta(n_1 - k_1, n_2 - k_2)] \quad (1.6)$$

Pour un système linéaire discret invariant dans le temps (LIT), puis, de (1.4) et (1.6), la relation d'entrée-sortie est donnée par:

$$y(n_1, n_2) = T[x(n_1, n_2)] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) \quad (1.7)$$

L'équation (1.7) dit qu'un système de LIT est complètement caractérisée par la réponse d'impulsion $h(n_1, n_2)$. Spécifiquement, pour un système de LIT, la connaissance de $h(n_1, n_2)$ seul nous permet de déterminer la sortie du système à n'importe quelle entrée de (1.7).

L'équation (1.7) désigné sous le nom de *la convolution 2-D* est dénotée par l'opérateur "*" comme suit :

Pour un système LIT

$$\begin{aligned} y(n_1, n_2) &= x(n_1, n_2) * h(n_1, n_2) \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) \end{aligned} \quad (1.8)$$

Systèmes séparables (filtres)

Les systèmes séparables forment une classe importante des filtres 2-D. Ils sont employés couramment dans la synthèse, l'analyse, et l'implémentation des systèmes 2-D, principalement due à la simplicité de l'analyse qu'ils offrent. Au plus de ça, le test de la stabilité des filtres séparables est équivalent au test de deux systèmes 1-D [2].

1.3 Convolution 2-D

La convolution dans (1.8) a un certain nombre de propriétés qui sont des extensions directes de cas unidimensionnel. Une partie du plus important est énumérée ci-dessous.

Commutativité

$$x(n_1, n_2) * y(n_1, n_2) = y(n_1, n_2) * x(n_1, n_2) \quad (1.9)$$

Associativité

$$(x(n_1, n_2) * y(n_1, n_2)) * z(n_1, n_2) = x(n_1, n_2) * (y(n_1, n_2) * z(n_1, n_2)) \quad (1.10)$$

Distributivité

$$x(n_1, n_2) * (y(n_1, n_2) + z(n_1, n_2)) = (x(n_1, n_2) * y(n_1, n_2)) + (x(n_1, n_2) * z(n_1, n_2)) \quad (1.11)$$

Convolution avec l'impulsion décalée

$$x(n_1, n_2) * \delta(n_1 - m_1, n_2 - m_2) = x(n_1 - m_1, n_2 - m_2) \quad (1.12)$$

La convolution de deux séquences $x(n_1, n_2)$ et $h(n_1, n_2)$ de (1.8) peut être interprétée en tant que multiplier deux séquences $x(n_1, n_2)$ et $h(n_1 - k_1, n_2 - k_2)$, qui sont des fonctions des variables k_1 et k_2 , et addition du produit au-dessus de toutes les valeurs entières de k_1 et k_2 .

Un système de LIT est dit séparable, si sa réponse d'impulsion $h(n_1, n_2)$ est une séquence séparable. Pour un système séparable, il est possible réduit le nombre d'opérations arithmétiques exigées pour calculer la somme de convolution. Pour des grands nombres de données, comme typiquement trouvés dans les images, la réduction de calcul peut être considérable. Pour montrer ça, considérer une séquence d'entrée $x(n_1, n_2)$ du $N \times N$ points et d'une réponse d'impulsion $h(n_1, n_2)$ du $M \times M$ points:

$$x(n_1, n_2) = 0 \text{ dehors } 0 \leq n_1, n_2 \leq N - 1, \text{ et}$$

$$h(n_1, n_2) = 0 \text{ dehors } 0 \leq n_1, n_2 \leq M - 1.$$

où $N \gg M$ dans des cas typiques. Si (1.8) est utilisé directement pour calculer $y(n_1, n_2)$, approximativement $(N + M - 1)^2 M^2$ opérations arithmétiques (une opération arithmétique = une multiplication et une addition) sont exigées puisque le nombre de points différents de zéro de sortie est $(N + M - 1)^2$ et calculant chaque point de sortie exige M^2 des opérations arithmétiques.

Si $h(n_1, n_2)$ est une séquence séparable, il peut être exprimé par :

$$\begin{aligned} h(n_1, n_2) &= h(n_1)h(n_2), \\ h(n_1) &= 0 \text{ dehors } 0 \leq n_1 \leq M-1, \\ h(n_2) &= 0 \text{ dehors } 0 \leq n_2 \leq M-1. \end{aligned}$$

(1.8) rendre

$$\begin{aligned} y(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_1(n_1 - k_1) h_2(n_2 - k_2) \\ &= \sum_{k_1=-\infty}^{\infty} h_1(n_1 - k_1) \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2) \end{aligned} \quad (1.13)$$

Pour un k_1 fixe, $\sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2)$ dans (1.13) est une convolution 1-D de $x(n_1, n_2)$ et de $h_2(n_2)$, exploitant la séparabilité de $h(n_1, n_2)$, exige approximativement $NM(N + M - 1) + M(N + M - 1)^2$ opérations arithmétiques. Ceci peut être une économie de calcul considérable au-dessus de $(N + M - 1)^2 M^2$. Si nous supposons $N \gg M$, exploitant la séparabilité de $h(n_1, n_2)$, réduit le nombre d'opérations arithmétiques par approximativement un facteur de $M/2$ [1].

1.4 Systèmes stables et systèmes de support spécial

Pour des raisons pratiques, il est souvent approprié imposer des contraintes supplémentaires à la classe des systèmes que nous considérons. Les systèmes stables et les systèmes de support spécial ont de telles contraintes.

1.4.1 Systèmes stables

Un système est considéré stable dans le sens (BIBO) si et seulement si à toute entrée bornée correspond une sortie bornée. La stabilité est une contrainte désirable à imposer, puisqu'un système instable peut produire d'une sortie illimitée, qui peut cause la surcharge de système ou d'autres difficultés.

De ces définition et (1.8), il peut montrer qu'un état nécessaire et suffisant pour qu'un système de LIT soit stable est que sa réponse d'impulsion $h(n_1, n_2)$ soit absolument sommable :

$$\text{Stabilité d'un système LIT} \Leftrightarrow \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty \quad (1.14)$$

(1.14) est une extension directe des résultats de système 1-D, les systèmes 2-D diffèrent considérablement des systèmes 1-D quand la stabilité de système est testée [1].

1.4.2 Systèmes de support spécial (causalité)

Un système 1-D est dit causal si et seulement si la sortie actuelle $y(n)$ ne dépend d'aucune valeur future d'entrée, par exemple, $x(n+1)$, $x(n+2)$, $x(n+3)$, . . . , utilisation de cette définition, nous pouvons prouver qu'un état nécessaire et suffisant pour qu'un système de LIT 1-D soit causal est que sa réponse d'impulsion $h(n)$ soit zéro pour $n < 0$. La causalité est une contrainte désirable à imposer en synthèse les systèmes 1-D. Un système non causal exigerait éléments de retard, qui sont indésirable dans des applications telles que le traitement de parole en temps réel.

Dans les applications typiques de traitement du signal 2-D telles que traitement d'images, la contrainte de causalité peut ne pas être nécessaire. A un moment donné, une trame complète d'une image peut être disponible pour le traitement, et elle peut être traitée de gauche à droite, de haut en bas, ou dans n'importe quelle direction on choisit.

Bien que la notion de la causalité ne pas être utile dans le traitement du signal 2-D, il est utile d'étendre la notion qu'un système (LIT) 1-D causal à une réponse d'impulsion $h(n)$ dont les valeurs différentes de zéro se situent dans une région particulière.

Un système (LIT) 2-D dont la réponse impulsionnelle $h(n_1, n_2)$ a toutes ses valeurs différentes de zéro dans une région particulière s'appelle un système de support spécial.

Un système (LIT) 2-D est dit un système de support de quart de cercle (*quadrant support*) quand sa réponse impulsionnelle $h(n_1, n_2)$ est une séquence de support de quart de cercle.

Une séquence de support de quart de cercle, est un qui a toutes ses valeurs différentes de zéro dans un quart de cercle. Un exemple d'une séquence de support de premier quart de cercle est la séquence $u(n_1, n_2)$ l'échelon d'unité.

Un système (LIT) 2-D est dit un système de support intercalé (*wedge support*) quand sa réponse impulsionnelle $h(n_1, n_2)$ est une séquence de support intercalé entre deux lignes.

Considérer deux lignes émanant de l'origine. Si toutes les valeurs différentes de zéro dans une séquence se situent dans la région bornée par ces deux lignes, et l'angle entre les deux lignes est moins que 180° , la séquence s'appelle une séquence de support intercaler.

Des séquences de support de quart de cercle et les séquences de support intercalé sont étroitement liées. Une séquence de support de quart de cercle est toujours une séquence de support intercalé. Il peut démontrer que n'importe quel séquence de support de intercaler peut toujours être dessinée à un premier quart de cercle par une linéaire transformation sans affecter sa stabilité ce qui est très utile en étudiant la stabilité d'un 2-D système [1].

1.5 Équation aux différences à coefficients constants linéaires

Les équations aux différences jouent un rôle plus important pour des systèmes d'espace discret que font les équations différentielles pour les systèmes analogiques. En plus de représenter une classe large des systèmes d'espace discret, les équations aux différences peuvent être utilisées pour produire récursivement de leurs solutions. Ce dernier avantage peut être exploité en réalisant des filtres numériques à réponses impulsionnelle infini.

On considère une équation aux différences à coefficient constantes linéaire de la formule suivante :

$$\sum_{(k_1, k_2) \in R_a} a(k_1, k_2) y(n_1 - k_1, n_2 - k_2) = \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \quad (1.15)$$

où $a(k_1, k_2)$ et $b(k_1, k_2)$ sont des séquences réel connus avec un nombre fini des amplitudes différentes de zéro, R_a représente la région dans (k_1, k_2) tels que $a(k_1, k_2)$ est différent de zéro, et R_b est assimilé défini. Pour la convenance, nous nous référerons à une équation aux différences à coefficients constants linéaire simplement comme équation aux différences, et nous supposons que $a(k_1, k_2)$ est zéro pour $(k_1, k_2) \notin R_a$, et $b(k_1, k_2)$ est zéro pour $(k_1, k_2) \notin R_b$.

Une équation aux différences ne spécifie pas seulement un système, puisqu'il y a beaucoup de solutions de $y(n_1, n_2)$ de (1.15) pour un $x(n_1, n_2)$ donné. Pour spécifier une solution unique, un ensemble de conditions initiales est nécessaire.

Puisque les conditions initiales doivent donner des informations suffisantes pour que nous déterminions les formules fonctionnelles spécifiques et les constantes associées aux fonctions, ils se composent typiquement d'un nombre infini des valeurs dans la sortie $y(n_1, n_2)$. Avec cette remarque, le cas 2-D diffère fondamentalement du cas 1-D.

Dans le cas 1-D, une équation aux différences spécifie une solution dans des constantes arbitraires. Par exemple, une équation aux différences 1-D d'ordre N spécifie une solution avec des N constantes, ainsi des N conditions initiales (les valeurs de N dans la sortie $y(n)$) sont généralement suffisantes pour spécifier une solution unique. Dans le cas 2-D, nous avons besoin typiquement des conditions initiales qui comportent un nombre infini des points dans la sortie $y(n_1, n_2)$ [1].

Puisqu'un filtre numérique est un système linéaire et invariant dans le temps (LIT), nous avons besoin de forcer l'équation aux différences pour devenir un système de LIT. Ceci peut être fait en choisissant un ensemble approprié des conditions initiales.

1.5.1 Équations aux différences dans systèmes linéaires et invariant dans le temps

La seule méthode pour forcer une équation aux différences avec des conditions initiales pour être un système de LIT est de suivre les trois étapes suivantes en choisissant les conditions initiales :

Étape 1 : interprètent l'équation aux différences comme procédure de calcul spécifique.

Étape 2 : déterminent R_{BC} , la région (n_1, n_2) pour lequel les conditions initiales sont appliquées, comme suit :

(a) Déterminer R_h , la région du support de la réponse impulsionnelle $h(n_1, n_2)$.

(b) Déterminer R_y , la région du support de la sortie $y(n_1, n_2)$, à partir du R_h et du R_x , où R_x est assimilé défini.

(c) R_{BC} : tout le $(n_1, n_2) \notin R_y$.

Étape 3 : les conditions initiales: mettre $y(n_1, n_2) = 0$ pour $(n_1, n_2) \in R_{BC}$.

Étape 1 s'assure que nous choisirons un sur les nombreux systèmes qui peuvent être dérivés d'une équation aux différences donnée. Étape 2 s'assure que les conditions initiales seront choisies telles qu'ils décaleront que l'entrée est décalée, et ils ne dépasser pas la limite avec le résultat de convolution de l'entrée avec la réponse impulsionnelle du système. Étape 3 s'assure que les conditions initiales que nous imposons sont des zéros.

Quand nous choisissons les conditions initiales après ces trois étapes, nous constaterons que l'équation aux différences en résultant avec des conditions initiales deviendra un système de LIT [1].

1.5.2 Calcul récursif

L'équation aux différences avec des conditions initiales joue un rôle particulièrement important dans le traitement numérique des signaux, puisque c'est la seule méthode en pratique de réaliser un filtre numérique RII [1]. Comme nous avons discuté dans les sections précédentes, l'équation aux différences peut être utilisée comme procédure récursive pour calculant la sortie.

Nous définissons un système pour être récursivement calculables (*recursively computable*) quand là existe un chemin que nous pouvons suivre en calculant chaque point de sortie récursivement, un point calculable à la fois. De la définition d'un système récursivement calculable, il peut démontrer que non toutes les procédures de calcul résultant des équations aux différences sont récursivement calculables.

Il peut montrer que si un système est récursivement calculable, si la sortie a support intercaler. Pour une entrée de séquence $x(n_1, n_2)$, la sortie de support intercaler est non seulement nécessaire, mais également suffisant pour que le système pour être récursivement calculable [1].

1.6 Transformée en z

La transformée en z d'une séquence $x(n_1, n_2)$ est dénotée par $X(z_1, z_2)$ et est définie par

$$X(z_1, z_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) z_1^{-n_1} z_2^{-n_2} \quad (1.16)$$

où z_1 et z_2 sont des variables complexes. Puisque chacune des deux variables complexe est évaluées dans espace de deux dimension 2-D, l'espace pour évaluées (z_1, z_2) est espace de quatre dimension 4-D. Il est ainsi extrêmement difficile de visualiser des points ou des segments de $X(z_1, z_2)$.

Supposer que $X(z_1, z_2)$ de (1.16) est évalué sur $(z_1 = r_1 e^{jw_1}, z_2 = r_2 e^{jw_2})$, où r_1 et w_1 sont le rayon et l'argument dans le plan de z_1 , et r_2 et w_2 sont le rayon et l'argument dans le plan de z_2 , la fonction $X(z_1, z_2)$ peut être exprimée par :

$$\begin{aligned} X(z_1, z_2) \Big|_{z_1=r_1 e^{jw_1}, z_2=r_2 e^{jw_2}} &= \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) r_1^{-n_1} r_2^{-n_2} e^{-jw_1 n_1} e^{-jw_2 n_2} \\ &= F \left[x(n_1, n_2) r_1^{-n_1} r_2^{-n_2} \right] \end{aligned} \quad (1.17)$$

où $F \left[x(n_1, n_2) r_1^{-n_1} r_2^{-n_2} \right]$ représente la transformée de Fourier du $x(n_1, n_2) r_1^{-n_1} r_2^{-n_2}$. Puisque la convergence uniforme de $F \left[x(n_1, n_2) \right]$ exige que $x(n_1, n_2)$ absolument sommable, de (1.17), la convergence uniforme de la transformée en z de $x(n_1, n_2)$ exige que $x(n_1, n_2) r_1^{-n_1} r_2^{-n_2}$ absolument sommable:

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |x(n_1, n_2) r_1^{-n_1} r_2^{-n_2}| < \infty \quad (1.18)$$

où $r_1 = |z_1|$ et $r_2 = |z_2|$, de (1.18), la convergence de $X(z_1, z_2)$ dépendra généralement de la valeur de $r_1 = |z_1|$ et $r_2 = |z_2|$. Par exemple, pour la séquence l'échelon d'unité $u(n_1, n_2)$, $r_1^{-n_1} r_2^{-n_2} u(n_1, n_2)$ est absolument sommable seulement pour $(|z_1| > 1, |z_2| > 1)$, ainsi sa transformée en z converge seulement pour $(|z_1| > 1, |z_2| > 1)$.

La région dans le plan (z_1, z_2) où $X(z_1, z_2)$ converge uniformément s'appelle la région de la convergence (*region of convergence: ROC*) [1].

La région de la convergence joue un rôle important dans la représentation de transformée en z d'une séquence.

1.6.1 Propriétés de la transformée en z 2-D

Beaucoup de propriétés de la transformée en z peuvent être obtenues à partir de la définition de transformée en z de (1.16). Certaines des propriétés plus importantes sont énumérées dans le Tableau 1.1. Toutes les propriétés excepté la propriété 3 et la propriété 7 peuvent être visualisées en tant qu'extensions direct du cas unidimensionnel. La propriété 3 s'applique aux séquences séparables, et la propriété 7 peut être utilisée dans la détermination transformée en z d'une séquence de support de premier quart de cercle obtenu par la transformation linéaire des variables d'une séquence de support intercaler [1].

Tableau 1.1 Propriétés de la transformée en z 2-D

Propriété	la région de la convergence <i>ROC</i>
$x(n_1, n_2) \leftrightarrow X(z_1, z_2), ROC: R_x$ $y(n_1, n_2) \leftrightarrow Y(z_1, z_2), ROC: R_y$ <i>Propriété 1 Linéarité:</i> $ax(n_1, n_2) + by(n_1, n_2) \leftrightarrow aX(z_1, z_2) + bY(z_1, z_2)$ <i>Propriété 2 Convolution :</i> $x(n_1, n_2) * y(n_1, n_2) \leftrightarrow X(z_1, z_2)Y(z_1, z_2)$ <i>Propriété 3 Séquence séparable :</i> $x_1(n_1) x_2(n_2) \leftrightarrow X_1(z_1)X_2(z_2)$	 au moins $R_x \cap R_y$ au moins $R_x \cap R_y$ $ z_1 \in ROC$ de $X_1(z_1)$ et $ z_2 \in ROC$ of $X_2(z_2)$.
<i>Propriété 4 Décalage:</i> $x(n_1 - m_1, n_2 - m_2) \leftrightarrow X(z_1, z_2) z_1^{-m_1} z_2^{-m_2}$	R_x à l'exception possible de $ z_1 = 0, \infty$, $ z_2 = 0, \infty$.
<i>Propriété 5 Différentiation :</i> a) $-n_1 x(n_1, n_2) \leftrightarrow \frac{\partial X(z_1, z_2)}{\partial z_1} z_1$, b) $-n_2 x(n_1, n_2) \leftrightarrow \frac{\partial X(z_1, z_2)}{\partial z_2} z_2$,	R_x , R_x ,
<i>Propriété 6 Symétrie :</i> a) $x^*(n_1, n_2) \leftrightarrow X^*(z_1^*, z_2^*)$, b) $x(-n_1, -n_2) \leftrightarrow X(z_1^{-1}, z_2^{-1})$,	R_x $ z_1^{-1} , z_2^{-1} $ dans R_x
<i>Propriété 7 Transformation linéaire des variables :</i> $x(n_1, n_2) = y(n_1, n_2) \Big _{m_1=l_1n_1+l_2n_2, m_2=l_3n_1+l_4n_2} \leftrightarrow Y(z_1, z_2) = X(z_1^{l_1} z_2^{l_3}, z_1^{l_2} z_2^{l_4})$	$ROC: z_1^{l_1} z_2^{l_3} , z_1^{l_2} z_2^{l_4} $ dans R_x .

1.6.2 Transformée en z inverse 2-D

La définition de transformée en z de (1.16) peut être utilisée en déterminant la transformée en z et le *ROC* d'une séquence 2-D. Comme dans le cas unidimensionnelle, utiliser (1.16) et l'intégral du Cauchy, nous pouvons déterminer la relation de transformée en z inverse qui exprime $x(n_1, n_2)$ en fonction de $X(z_1, z_2)$ et de son *ROC*.

$$x(n_1, n_2) = \frac{1}{(2\pi j)^2} \oint_{C_1} \oint_{C_2} X(z_1, z_2) z_1^{-n_1-1} z_2^{-n_2-1} dz_1 dz_2 \quad (1.19)$$

où C_1 et C_2 sont deux contours dans le *ROC* de $X(z_1, z_2)$, C_1 est un contour fermée qui encercle dans le sens antihoraire dans le plan z_1 pour n'importe quel z_2 fixe, sur le C_2 , et le C_2 est un contour fermée qui encercle dans le sens antihoraire dans le plan z_2 pour n'importe

quel z_1 fixe, sur le C_1 , une séquence $x(n_1, n_2)$ n'est pas uniquement spécifiée par $X(z_1, z_2)$ seulement.

Depuis les contours C_1 et C_2 peuvent être déterminés du *ROC* de $X(z_1, z_2)$, cependant, une séquence $x(n_1, n_2)$ est uniquement spécifiée par $X(z_1, z_2)$ et son *ROC* :

$$x(n_1, n_2) \leftrightarrow X(z_1, z_2), \text{ ROC}$$

Théoriquement, (1.19) peut être utilisé directement en déterminant $x(n_1, n_2)$ de $X(z_1, z_2)$ et son *ROC*. Dans la pratique, cependant, il est extrêmement pénible d'exécuter l'intégration de contour même pour des problèmes simples [1].

Pour les signaux 1-D, des approches qui ont été utiles en exécutant le fonctionnement de transformée en z inverse sans évaluation explicite l'intégration de contour comme expansion de série, fraction partielle, et utilisation les propriétés de transformée en z . Pour les signaux 2-D, malheureusement, la méthode fraction partielle n'est pas une procédure générale. Dans le cas 1-D, la factorisation de tout 1-D polynôme $D(z)$ comme produit des polynômes d'ordres inférieur est garanti par le théorème fondamental de l'algèbre. Un 2-D polynôme, cependant, ne peut pas en général être factorisé comme produit des polynômes de d'ordres inférieur. Par conséquent, il n'est pas généralement possible d'utiliser une procédure analogue.

Car de la difficulté impliquée dans la méthode fraction partielle, aucune méthode en pratique connue pour exécuter la transformée en z inverse d'une séquence général 2-D. Ce limitation d'utilité de la transformée en z de séquence 2-D est comparable à la transformée en de séquence 1-D [1].

1.7 Fonctions transfert d'un système 2-D

Dans les sections précédentes, nous avons prouvé qu'avec un choix approprié des conditions initiales, une équation aux différences avec des conditions initiales peut être considérée un système de LIT. Considérer une équation aux différences de la forme

$$\sum_{(k_1, k_2) \in R_a} a(k_1, k_2) y(n_1 - k_1, n_2 - k_2) = \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \quad (1.20)$$

où $a(k_1, k_2)$ et $b(k_1, k_2)$ sont des séquences fini avec des amplitudes différentes de zéro en R_a et R_b respectivement.

Par la transformée en z de deux côtés de (1.20), nous avons

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{\sum_{(n_1, n_2) \in R_b} b(n_1, n_2) z_1^{-n_1} z_2^{-n_2}}{\sum_{(n_1, n_2) \in R_a} a(n_1, n_2) z_1^{-n_1} z_2^{-n_2}} \quad (1.21)$$

Le dénominateur et le numérateur de (1.21) sont la transformée en z de $a(n_1, n_2)$ et de $b(n_1, n_2)$, et peut donc être exprimé comme $A(z_1, z_2)$ et $B(z_1, z_2)$ respectivement. La fonction transfert $H(z_1, z_2)$ peut alors être exprimée par

$$H(z_1, z_2) = \frac{B(z_1, z_2)}{A(z_1, z_2)} = \frac{\sum_{(n_1, n_2) \in R_b} b(n_1, n_2) z_1^{-n_1} z_2^{-n_2}}{\sum_{(n_1, n_2) \in R_a} a(n_1, n_2) z_1^{-n_1} z_2^{-n_2}} \quad (1.22)$$

Les fonctions $A(z_1, z_2)$ et $B(z_1, z_2)$ sont des polynômes d'ordre fini en z_1 et z_2 . La fonction $H(z_1, z_2)$, qui est une rapport de deux polynômes d'ordre fini, s'appelle une fonction rationnelle. Si $A(z_1, z_2)$ ne peut pas être écrit comme un produit de deux polynômes d'ordre fini, il est dit *irréductible*. S'il y a aucun facteur commun ce ne divise $A(z_1, z_2)$ et $B(z_1, z_2)$, $A(z_1, z_2)$ et $B(z_1, z_2)$ sont dits co-premier.

Une fonction rationnelle $H(z_1, z_2)$ est dit irréductible si $A(z_1, z_2)$ et $B(z_1, z_2)$ sont co-premiers.

1.8 Transformée de Fourier 2-D

C'est un fait remarquable que n'importe quelle séquence stable $x(n_1, n_2)$ peut être obtenue en combinant convenablement des exponentiels complexes de la forme $X(w_1, w_2) e^{jw_1 n_1} e^{jw_2 n_2}$.

La fonction $X(w_1, w_2)$, qui représente l'amplitude associée à l'exponentiel complexe $e^{jw_1 n_1} e^{jw_2 n_2}$ peut être obtenue à partir de $x(n_1, n_2)$. Les relations entre $x(n_1, n_2)$ et $X(w_1, w_2)$ sont donnés par:

Paire de transformée de Fourier de l'espace discret

$$X(w_1, w_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) e^{-jw_1 n_1} e^{-jw_2 n_2} \quad (1.23a)$$

$$x(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{w_1=-\pi}^{\pi} \int_{w_2=-\pi}^{\pi} X(w_1, w_2) e^{jw_1 n_1} e^{jw_2 n_2} dw_1 dw_2 \quad (1.23b)$$

L'équation (1.23a) montre comment l'amplitude $X(w_1, w_2)$ associée à l'exponentiel $e^{jw_1 n_1} e^{jw_2 n_2}$ peut être déterminée de $x(n_1, n_2)$. La fonction $X(w_1, w_2)$ s'appelle la *transformée de Fourier de l'espace discret* de $x(n_1, n_2)$, ou brièvement la transformée de Fourier. L'équation (1.23b) montre comment les exponentiels complexes $X(w_1, w_2) e^{jw_1 n_1} e^{jw_2 n_2}$ ont spécifiquement combiné pour former $x(n_1, n_2)$. La séquence $x(n_1, n_2)$ s'appelle *transformée de Fourier de l'espace discret inverse* ou brièvement la transformée de Fourier inverse de $X(w_1, w_2)$. De (1.23), il peut voir que $X(w_1, w_2)$ est en général complexe, quoique $x(n_1, n_2)$ être réel.

Il est préférable d'exprimer $X(w_1, w_2)$ en termes de son amplitude $|X(w_1, w_2)|$ et de phase $\theta_X(w_1, w_2)$ ou en termes de sa partie réelle $X_R(w_1, w_2)$ et partie imaginaire $X_I(w_1, w_2)$ par :

$$X(w_1, w_2) = |X(w_1, w_2)| e^{j\theta_X(w_1, w_2)} = X_R(w_1, w_2) + jX_I(w_1, w_2) \quad (1.24)$$

De (1.23), il peut également voir que $X(w_1, w_2)$ est une fonction des variables continues w_1 et w_2 , bien que $x(n_1, n_2)$ soit une fonction des variables discrètes n_1 et du n_2 .

$X(w_1, w_2)$ est toujours périodique avec une période de $2\pi \times 2\pi$; c'est-à-dire,

$X(w_1, w_2) = X(w_1 + 2\pi, w_2) = X(w_1, w_2 + 2\pi)$ pour tous les w_1 et w_2 . Nous pouvons également prouver que la transformée de Fourier converge uniformément pour des séquences stables.

On dit que la transformée de Fourier de $x(n_1, n_2)$ converge uniformément quand $X(w_1, w_2)$ est fini et

$$\lim_{N_1 \rightarrow \infty} \lim_{N_2 \rightarrow \infty} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} x(n_1, n_2) e^{-jw_1 n_1} e^{-jw_2 n_2} = X(w_1, w_2) \text{ pour tous } w_1 \text{ et } w_2$$

Quand la transformée de Fourier de $x(n_1, n_2)$ converge uniformément, $X(w_1, w_2)$ est une fonction analytique et est infiniment différentiable par rapport w_1 et w_2 .

1.8.1 Propriétés de la transformée de Fourier

Nous pouvons dériver un certain nombre de propriétés utiles des paires de transformée de Fourier du (1.23), certaines des propriétés plus importantes, utile dans la pratique, sont énumérés dans le Tableau 1.2. Les plupart sont essentiellement des extensions direct des propriétés de la transformée de Fourier 1-D. La seule exception est la propriété 4, qui s'applique aux séquences séparables. Si une séquence 2-D $x(n_1, n_2)$ peut être écrit comme $x_1(n_1)x_2(n_2)$, alors sa transformée de Fourier, $X(w_1, w_2)$, est donnée par $X_1(w_1)X_2(w_2)$ où $X_1(w_1)$ et $X_2(w_2)$ représentent la transformée de Fourier 1-D de $x_1(n_1)$ et $x_2(n_2)$ respectivement. Noter que cette propriété est très différente de la propriété 3, la propriété de multiplication.

Dans la propriété de multiplication, $x(n_1, n_2)$ et $y(n_1, n_2)$ sont des séquences 2-D. Dans la propriété 4, $x_1(n_1)$ et $x_2(n_2)$ sont séquences 1-D, et leur produit $x_1(n_1)x_2(n_2)$ forment un séquence 2-D [1].

Tableau 1.2 Propriétés de la transformée de Fourier 2-D

Propriétés	Correspondance mathématique
$x(n_1, n_2) \leftrightarrow X(w_1, w_2)$ $y(n_1, n_2) \leftrightarrow Y(w_1, w_2)$	
<i>Propriété 1 Linéarité</i>	$ax(n_1, n_2) + by(n_1, n_2) \leftrightarrow aX(w_1, w_2) + bY(w_1, w_2)$
<i>Propriété 2 Convolution</i>	$x(n_1, n_2)*y(n_1, n_2) \leftrightarrow X(w_1, w_2)Y(w_1, w_2)$
<i>Propriété 3 Multiplication</i>	$(n_1, n_2)y(n_1, n_2) \leftrightarrow X(w_1, w_2)\Theta Y(w_1, w_2)$ $= \frac{1}{(2\pi)^2} \int_{\theta_1=-\pi}^{\pi} \int_{\theta_2=-\pi}^{\pi} X(\theta_1, \theta_2)Y(w_1 - \theta_1, w_2 - \theta_2)d\theta_1d\theta_2$
<i>Propriété 4 Séquence Séparable</i>	$x(n_1, n_2) = x_1(n_1)x_2(n_2) \leftrightarrow X(w_1, w_2) = X_1(w_1)X_2(w_2)$
<i>Propriété 5 Décalage</i>	a) $x(n_1 - m_1, n_2 - m_2) \leftrightarrow X(w_1, w_2) e_1^{-jw_1m_1} e_2^{-jw_2m_2}$ b) $e_1^{jv_1n_1} e_2^{jv_2n_2} x(n_1, n_2) \leftrightarrow X(w_1 - v_1, w_2 - v_2)$
<i>Propriété 6 Différentiation</i>	a) $-jn_1x(n_1, n_2) \leftrightarrow \frac{\partial X(w_1, w_2)}{\partial w_1}$, b) $-jn_2x(n_1, n_2) \leftrightarrow \frac{\partial X(w_1, w_2)}{\partial w_2}$.
<i>Propriété 7 Valeur Initial</i>	a) $x(0, 0) = \frac{1}{(2\pi)^2} \int_{w_1=-\pi}^{\pi} \int_{w_2=-\pi}^{\pi} X(w_1, w_2)dw_1dw_2$, b) $X(0, 0) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2)$.
<i>Propriété 8 Théorème de Parseval</i>	a) $\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2)y^*(n_1, n_2)$ $= \frac{1}{(2\pi)^2} \int_{w_1=-\pi}^{\pi} \int_{w_2=-\pi}^{\pi} X(w_1, w_2)Y^*(w_1, w_2)dw_1 dw_2$, b) $\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) ^2 = \frac{1}{(2\pi)^2} \int_{w_1=-\pi}^{\pi} \int_{w_2=-\pi}^{\pi} X(w_1, w_2) ^2 dw_1 dw_2$.
<i>Propriété 9 Symétrie</i>	a) $x(-n_1, n_2) \leftrightarrow X(-w_1, w_2)$, b) $x(n_1, -n_2) \leftrightarrow X(w_1, -w_2)$, c) $x(-n_1, -n_2) \leftrightarrow X(-w_1, -w_2)$, d) $x^*(n_1, n_2) \leftrightarrow X^*(-w_1, -w_2)$, e) $x(n_1, n_2)$: réel $\leftrightarrow X(w_1, w_2) = X^*(-w_1, -w_2)$.

1.9 Conclusion

Un système linéaire et invariant dans le temps qui présente un filtre dans notre cas peut traduire par une réponse impulsionnelle ou une fonction transfert, nous avons besoin de l'un ou de l'autre pour simuler la réponse fréquentielle d'un tel filtre. Dans la pratique on utilise les coefficients de numérateur et/ou de dénominateur pour réaliser un filtre, et pour déterminer ces coefficients nous utilisons les méthodes de synthèse qui peuvent être dans les deux domaines, temporel et fréquentiel, et ces méthodes diffèrent selon le type de filtre que l'on peut synthétiser. Il y a deux types différents de filtres selon leur réponse impulsionnelle, le filtre à réponse impulsionnelle finie (RIF), et un filtre à réponse impulsionnelle infinie (RII), ils sont complètement différents dans la synthèse (selon les critères de la stabilité et la linéarité de la phase) et l'implémentation (réalisation en cascade ou en parallèle).

Chapitre 2

Caractéristiques des Filtres Numériques Bidimensionnels

2.1 Introduction

Trois étapes sont généralement suivies en utilisant des filtres numériques. Dans la première étape, nous spécifions les caractéristiques requises du filtre. Le cahier des charges de filtre dépend, naturellement, de l'application destinée. Par exemple, si nous souhaitons restaurer un signal qui a été dégradé par bruit, les caractéristiques de filtre exigées dépendent des caractéristiques spectrales du signal et du bruit. La deuxième étape est synthèse de filtre. Dans cette étape, nous déterminons $h(n_1, n_2)$, la réponse impulsionnelle du filtre, ou sa fonction de transfert $H(z_1, z_2)$, qui satisfait à cahier des charges des caractéristiques. La troisième étape est implémentation de filtre, dans laquelle nous réalisons un système discret avec $h(n_1, n_2)$ donné ou $H(z_1, z_2)$. Ces trois étapes sont corrélées, elle ne pas raisonnable, par exemple, de synthétiser un filtre qui ne peut pas être réalisable. Pour des raisons pratiques, nous nous limiterons à une certaine classe des filtres numériques. Une restriction est ce $h(n_1, n_2)$ doit être réel. Dans la pratique, nous traiterons de données réelles. Pour s'assurer que le signal traité sera réel, nous exigerons de $h(n_1, n_2)$ d'être réel. Une autre restriction est la stabilité de $h(n_1, n_2)$, c.-à-d., $\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty$. Dans la pratique, une sortie pas bornée peut occasionner beaucoup de difficultés, par exemple, surcharge de système. Nous limiterons notre discussion, alors, à la classe des filtres numériques dont la réponse d'impulsion $h(n_1, n_2)$ est réelle et stable [1].

2.2 Filtres à phase linéaire

On dit qu'un filtre numérique $h(n_1, n_2)$ à phase linéaire (ou *zero-phase*) quand sa réponse fréquentielle $H(w_1, w_2)$ est une fonction réelle

$$H(w_1, w_2) = H^*(w_1, w_2) \quad (2.1)$$

Ne pas tous les filtres avec de réponse en fréquentielle réelle sont nécessairement des filtres à phase linéaire, depuis $H(w_1, w_2)$ peuvent être négatifs. Dans la pratique, les régions de fréquence pour lequel $H(w_1, w_2)$ est négative typiquement correspondre aux régions de bande coupée, et à une phase de 180° dans les régions de bande coupée a peu de signification.

Des propriétés de symétrie de la transformée de Fourier, (2.1) est équivalent dans le domaine spatial à:

$$h(n_1, n_2) = h^*(-n_1, -n_2) \quad (2.2)$$

Puisque nous considérons seulement $h(n_1, n_2)$ réelle, (2.2) réduit à

$$h(n_1, n_2) = h(-n_1, -n_2) \quad (2.3)$$

L'équation (2.3) dit que la réponse impulsionnelle d'un filtre à phase linéaire est symétrique par rapport à l'origine. Une caractéristique d'un filtre à phase linéaire est sa tendance de préserver la forme du composant de signal dans la région de bande passante du filtre. Dans les applications telles que le traitement de la parole, la phase linéaire caractéristique d'un filtre n'est pas très critique. Le système auditif humain répond aux caractéristiques spectrales de grandeur de brève durée, ainsi la forme d'onde de la parole peut parfois changer rigoureusement sans auditeur humain pouvant la distinguer de l'original. Dans à traitement d'images, la caractéristique de phase linéaire semble être plus importante. Notre monde visuel se compose des lignes, des gribouillages, etc. Une phase non linéaire déformée les différents composants de fréquence qui composent les lignes et les gribouillages. Ceci déformer la forme de signal dans diverses voies, y compris le brouillage.

La caractéristique de phase linéaire est tout à fait utile dans les applications telles que le traitement d'images, et il est très facile réaliser filtre a phase linéaire avec des filtres a réponse impulsionnelle fini (RIF). La synthèse et l'implémentation sont souvent simplifiées si nous avons besoin de filtre à phase linéaire [1].

Considérer une réponse impulsionnelle de phase linéaire $h(n_1, n_2)$, de (2.3) et la définition de transformée de Fourier de (1.23), la réponse fréquentielle $H(w_1, w_2)$ peut être exprimée par

$$\begin{aligned} H(w_1, w_2) &= \sum_{(n_1, n_2) \in R_h} h(n_1, n_2) e^{-jw_1 n_1} e^{-jw_2 n_2} \\ &= h(0, 0) + \sum_{(n_1, n_2) \in R_h'} h(n_1, n_2) e^{-jw_1 n_1} e^{-jw_2 n_2} + \sum_{(n_1, n_2) \in R_h''} h(n_1, n_2) e^{-jw_1 n_1} e^{-jw_2 n_2} \end{aligned} \quad (2.4)$$

où R_h est la région du support de $h(n_1, n_2)$ et se compose de trois régions : $(0, 0)$, R_h' et R_h'' . La région R_h'' est renversée de R_h' par rapport à l'origine. Combinaison des deux termes dans l'expression de côté droit de (2.4) et noter que $h(n_1, n_2) = h(-n_1, -n_2)$, nous peut exprimer (2.4) par :

$$\begin{aligned} H(w_1, w_2) &= h(0, 0) + \sum_{(n_1, n_2) \in R_h'} (h(n_1, n_2) e^{-jw_1 n_1} e^{-jw_2 n_2} + h(-n_1, -n_2) e^{jw_1 n_1} e^{jw_2 n_2}) \\ &= h(0, 0) + \sum_{(n_1, n_2) \in R_h'} 2h(n_1, n_2) \cos(w_1 n_1 + w_2 n_2) \end{aligned} \quad (2.5)$$

De (2.5), $H(w_1, w_2)$ pour un filtre RIF à phase linéaire peut toujours être exprimé comme combinaison linéaire des termes de cosinus de la forme $\cos(w_1 n_1 + w_2 n_2)$ [1].

Le filtre $h(n_1, n_2)$ de phase linéaire est symétrique par rapport à l'origine, approximativement le moitié des points dans $h(n_1, n_2)$ sont indépendant. En conséquence, on dit que le filtre a la symétrie double (*twofold*). Dans quelques applications, telles que la conception des filtres circulairement symétriques, il peut être utile d'imposer des symétries supplémentaires à $h(n_1, n_2)$. Une telle contrainte est symétrie quadruple (*fourfold*) donnée par

$$h(n_1, n_2) = h(-n_1, n_2) = h(n_1, -n_2) \quad (2.6)$$

Dans le domaine de transformée de Fourier, (2.6) est équivalent à

$$H(w_1, w_2) = H(-w_1, w_2) = H(w_1, -w_2) \quad (2.7)$$

Une autre de ce genre contraint de symétrie est une symétrie octuple (*eightfold*) donnée par

$$h(n_1, n_2) = h(-n_1, n_2) = h(n_1, -n_2) = h(n_2, n_1) \quad (2.8)$$

Dans le domaine de transformée de Fourier, (2.8) est équivalent à

$$H(w_1, w_2) = H(-w_1, w_2) = H(w_1, -w_2) = H(w_2, w_1) \quad (2.9)$$

Les points indépendants de $h(n_1, n_2)$ dans les symétries doubles, quadruples, et octuples sont dessinés dans la figure 2.1 pour le cas quand la région du support de $h(n_1, n_2)$ est 11×11 points avec la forme carrée.

Pour une région donnée de support de $h(n_1, n_2)$, les contraintes imposantes de symétrie réduit le nombre de paramètres indépendants qui doivent être estimés dans la synthèse et réduit le nombre d'opérations arithmétiques exigées dans l'implémentation [1].

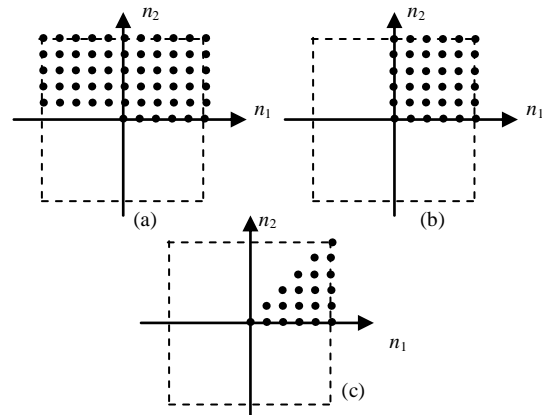


Figure 2.1 11×11 points indépendants de $h(n_1, n_2)$ avec (a) symétrie double, (b) symétrie quadruples, et (c) symétrie octuple.

2.3 Caractéristiques des filtres

Comme les filtres numériques 1-D, les filtres numériques 2-D sont généralement spécifiés dans le domaine de fréquence. Depuis $H(w_1, w_2) = H(w_1 + 2\pi, w_2) = H(w_1, w_2 + 2\pi)$ pour tous (w_1, w_2) , et pour $-\pi \leq w_1 \leq \pi, -\pi \leq w_2 \leq \pi$, $H(w_1, w_2)$ est complètement spécifié. En plus, depuis $h(n_1, n_2)$ est assumé réel, $H(w_1, w_2) = H^*(-w_1, -w_2)$.

Spécifiant $H(w_1, w_2)$ pour $0 \leq w_1 \leq \pi, 0 \leq w_2 \leq \pi$ donc spécifient complètement $H(w_1, w_2)$ pour tous w_1 et w_2 . On dit qu'un filtre a une réponse fréquentielle circulairement symétrique, si $H(w_1, w_2)$ est une fonction de $w_1^2 + w_2^2$ pour $\sqrt{w_1^2 + w_2^2} \leq \pi$ et est constant en dehors du région $-\pi \leq w_1 \leq \pi, -\pi \leq w_2 \leq \pi$. On dit qu'un filtre a une réponse impulsionnelle circulairement symétrique, si $h(n_1, n_2)$ est une fonction de $n_1^2 + n_2^2$, symétrie circulaire de $H(w_1, w_2)$ implique la symétrie circulaire de $h(n_1, n_2)$. La symétrie circulaire de $h(n_1, n_2)$, cependant, n'implique pas la symétrie circulaire de $H(w_1, w_2)$.

Les réponses fréquentielles des filtres passe-bas, passe-haut, passe-bande, et coupe-bande idéaux circulairement symétriques sont affichées dans les figures 2.2 (a), (b), (c), et (d), respectivement. Les régions ombragées dans les figures ont l'amplitude 1, et les régions non ombragées ont l'amplitude 0.

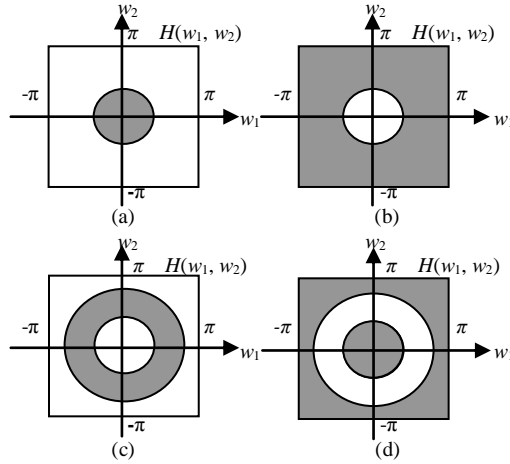


Figure 2.2 les réponses fréquentielles des filtres idéaux circulairement symétriques. (a) Filtre passe-bas ;(b) filtre passe-haut ; (c) filtre passe-bande ;(d) filtre coupe-bande

Leurs réponses impulsionnelles correspondantes sont données par :

$$h_{pb}(n_1, n_2) = \frac{R}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(R\sqrt{n_1^2 + n_2^2}) \quad (2.10a)$$

$$h_{ph}(n_1, n_2) = \delta(n_1, n_2) - h_{pb}(n_1, n_2) \quad (2.10b)$$

$$h_{pbn}(n_1, n_2) = \frac{R_2}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(R_2\sqrt{n_1^2 + n_2^2}) \quad (2.10c)$$

$$- \frac{R_1}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(R_1\sqrt{n_1^2 + n_2^2})$$

$$h_{cb}(n_1, n_2) = \delta(n_1, n_2) - h_{pbn}(n_1, n_2) \quad (2.10d)$$

où $h_{pb}(n_1, n_2)$, $h_{ph}(n_1, n_2)$, $h_{pbn}(n_1, n_2)$ et $h_{cb}(n_1, n_2)$ représentent les filtres passe-bas, passe-haut, passe-bande, et coupe-bande, respectivement, et $J_1(x)$ est la fonction Bessel de la première espèce et première ordre [1].

Depuis $H(w_1, w_2)$ est en général une fonction complexe de (w_1, w_2) , nous doit spécifier l'amplitude et la phase de $H(w_1, w_2)$. Pour le filtre RIF, nous avons besoin de la phase linéaire, ainsi nous devons spécifier seulement la grandeur de l'amplitude.

La méthode la plus utilisée généralement avec la caractéristiques de l'amplitude s'appelle le schéma de tolérance (*tolerance scheme*). Pour illustrons ce schéma, considèrent la caractéristiques d'un filtre passe-bas. Dans le meilleur des cas, un filtre passe-bas a seulement une région de bande passante et une région de bande coupée. Dans la pratique, il est existe un bande de transition entre les deux régions ne peut pas être réalisée, la région de bande passante correspond $(w_1, w_2) \in R_p$ et la région de bande coupée correspond $(w_1, w_2) \in R_s$, suivant les indications de la figure 2.3. La région de fréquence entre la bande passante et la bande coupée s'appelle la bande de transition.

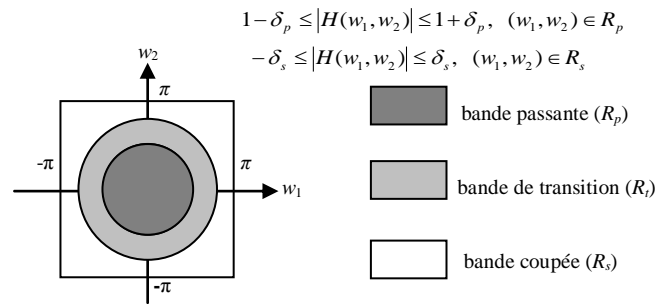


Figure 2.3 schéma de tolérance du caractéristiques de filtre passe-bas 2-D.

Idéalement, l'amplitude est unité dans la région de bande passante et zéro dans la région de bande coupée. Dans la pratique, on requit de $1 - \delta_p \leq |H(w_1, w_2)| \leq 1 + \delta_p$ pour $w_1, w_2 \in R_p$ et $|H(w_1, w_2)| \leq \delta_s$ pour $(w_1, w_2) \in R_s$. Les constants δ_p et δ_s s'appellent la tolérance de bande passante et la tolérance de bande coupée, respectivement.

Un filtre passe-bas, alors, est complètement spécifié par δ_p et δ_s, R_p, R_s . D'autres filtres peuvent également être spécifiés d'une façon analogue.

2.4 Classification des filtres numériques 2-D

Les filtres numériques 2-D peuvent être classifiés dans deux groupes. Dans le premier groupe, $h(n_1, n_2)$ est une séquence fini, ainsi les filtres dans ce groupe s'appellent les filtres à réponse impulsionnelle fini (RIF). Dans le deuxième groupe, $h(n_1, n_2)$ est une séquence infinie, ainsi des filtres dans ce groupe s'appellent les filtres à réponse impulsionnelle infini (RII), comme dans 1-D, les filtres RIF diffèrent considérablement dans la synthèse et l'implémentation des filtres RII.

Un filtre à réponse impulsionnelle infini à une séquence $h(n_1, n_2)$ qui est infinie. En conséquence, un filtre RII diffère à quelques majeurs aspects que d'un filtre RIF. Par exemple, la stabilité n'est jamais un problème pour des filtres RIF dans la synthèse et l'implémentation, alors que c'est un problème très importante et problématique pour des filtres RII. D'une part, un filtre RII exige typiquement d'un nombre sensiblement plus petit des coefficients pour répondre à une caractéristique particulière de l'amplitude que fait un filtre RIF.

Comme des filtres RIF, l'utilisation des filtres RII comporte des caractéristiques, la synthèse et l'implémentation de filtre. Dans le cahier des charges de filtre RII, nous devons spécifier l'amplitude et la phase de la réponse fréquentielle $H(w_1, w_2)$. Pour un filtre RIF, il est très facile de réaliser un filtre avec phase linéaire, avec un filtre RII, cependant, le contrôle de la caractéristique de phase est très difficile, en conséquence, seulement la réponse de l'amplitude est spécifiée quand un filtre RII est synthétiser. La caractéristique de phase du filtre en résultant est alors considérée comme acceptable.

Si la phase linéaire est nécessaire, deux filtres RII ou plus sont combinés pour obtenir la phase linéaire pour le filtre global [1].

2.5 Comparaison des filtres

Les filtres RIF ont beaucoup d'avantages par rapport aux filtres RII. La stabilité n'est pas un problème dans la synthèse et l'implémentation de filtre de RIF. Pour des filtres RII, cependant, le test de la stabilité de filtre et la stabilisation d'un filtre instable sans affecter de manière significative l'amplitude sont des grands problèmes. La phase linéaire est simple à réaliser pour les filtres RIF. Synthèse des filtres RII à phase linéaire est possible, mais est plus impliqué que synthèse des filtres de RIF à phase linéaire. En plus, les méthodes de synthèse sont plus simples pour les filtres RIF que pour les filtres RII.

L'avantage principal d'un filtre RII par rapport un filtre RIF est le nombre plus petit d'opérations arithmétiques exigées pour l'implémentation directe. Pour répondre aux mêmes caractéristiques d'amplitude, un filtre RIF exige de plus opérations arithmétiques (typiquement par un facteur de 5 à 30) pour un échantillon de sortie qu'un filtre RII.

Les filtres RII sont utiles, donc, principalement dans applications où l'implémentation peu coûteuse est très importante. En raison de leur avantage général par rapport aux filtres RII, les filtres RIF sont beaucoup plus communs dans la pratique. Cependant, la synthèse et l'implémentation des filtres 2-D RIF et RII sont des domaines de recherche actifs.

Les différences important entre les filtres 1-D et 2-D existent également dans la synthèse et l'implémentation de filtres numériques. Dans 1-D, il y a des méthodes pratiques pour la synthèse les filtres optimaux des filtres RIF et RII. Dans le 2-D, des méthodes pratiques qui peuvent être sûrement employées pour synthétisent les filtres optimaux de RIF ou RII n'ont pas été encore développées [1]. Dans 1-D, l'implémentation en cascade et en parallèles est des méthodes générales de réalisation pour des fonctions transferts rationaux. Dans le 2-D, l'implémentation en cascade et en parallèles ne peuvent pas généralement être utilisée pour la réalisation des fonctions transferts rationaux. La synthèse et l'implémentation sont plus complexes pour les filtres numériques 2-D que pour les filtres 1-D [1].

2.6 Conclusion

Ce chapitre était consacré à l'étude des filtres à phase linéaire, et les caractéristiques d'un filtre dans le domaine de fréquence. L'analyse, la synthèse, et l'implémentation d'un filtre RII peut être plus difficile que d'un filtre RIF a cause des propriétés de stabilité et la phase linéaire, qui sont ne pas problème dans la synthèse pour les filtre RIF.

Chapitre 3

Synthèse des Filtres Numériques RII-2D

3.1 Introduction

Un filtre RII avec une réponse impulsionnelle arbitraire $h(n_1, n_2)$ ne peut pas être réalisé, depuis calculer chaque échantillon de sortie exige un grand nombre d'opérations arithmétiques. En conséquence, en plus d'exiger de $h(n_1, n_2)$ d'être réel et stable, nous exigeons la transformée en z de $h(n_1, n_2)$ être rationnel et récursivement calculable. Spécifiquement, nous exigeons de $H(z_1, z_2)$, la transformée en z de $h(n_1, n_2)$, d'être une fonction rationnel de la formule similaire que l'équation (1.22).

En plus, nous exigeons une région du support intercalé pour la sortie de system [1].

3.2 Problème de synthèse

Le problème de synthèse d'un filtre RII est de déterminer une fonction transfert $H(z_1, z_2)$ rationnel et stable avec une région du support intercalé de sortie qui répond à caractéristiques de filtre. En d'autres termes, nous souhaitons déterminer une procédure de calcul de sortie de filtre stable qui est récursivement calculable. Dans la discussion le problème de synthèse de filtre, il est utile d'utiliser une fonction de transfert et une procédure de calcul l'un pour l'autre.

Cependant, une fonction transfert rationnel donnée $H(z_1, z_2)$ peut mener à beaucoup de procédures de calcul différentes [1]. Pour rendre la relation unique, nous adopterons une convention dans l'expression de $H(z_1, z_2)$. Spécifiquement, nous supposerons que $a(0, 0)$ de (3.1) est toujours 1, ainsi $H(z_1, z_2)$ sera alors sous la forme

$$H(z_1, z_2) = \frac{\sum_{(k_1, k_2) \in R_b} b(k_1, k_2) z_1^{-k_1} z_2^{-k_2}}{1 + \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) z_1^{-k_1} z_2^{-k_2}} \quad (3.1)$$

où $R_a - (0, 0)$ représente la région du support de $a(k_1, k_2)$ excepté l'origine $(0, 0)$.

La seule procédure de calcul correspondant à (3.1) est alors donnée par :

$$y(n_1, n_2) \leftarrow - \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) y(n_1 - k_1, n_2 - k_2) + \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \quad (3.2)$$

Avec les séquences $a(n_1, n_2)$ et $b(n_1, n_2)$ sont les coefficients de filtre RII.

La première étape dans la synthèse d'un filtre RII est habituellement déterminer les régions de R_a et R_b . Les régions du support de $a(k_1, k_2)$ et $b(k_1, k_2)$ sont déterminées par plusieurs considérations, comme le système en résultant doit être récursivement calculable, ceci exige de $a(k_1, k_2)$ d'être une séquence de support intercalé. Une autre considération est la région approximative du support $h(n_1, n_2)$ que nous souhaitons pour le filtre résultant.

Si nous déterminons les coefficients de filtre en essayant de rapprocher une réponse impulsionnelle désiré $h_d(n_1, n_2)$ dans le domaine spatial, nous voudrions choisir le R_a et R_b tels que $h(n_1, n_2)$ aura au moins approximativement la même région du support que le $h_d(n_1, n_2)$.

Si R_a et R_b ont le même support intercalé, alors il peut montrer que $h(n_1, n_2)$ a également exactement le même support intercalé. Si nous souhaitons avoir une région de support premier quart de cercle de $h(n_1, n_2)$, donc, un choix raisonnable serait la même région de support pour $a(k_1, k_2)$ et $b(k_1, k_2)$.

Une autre considération associe aux paramètres de filtre. Dans la synthèse de filtre passe-bas, par exemple, un petit δ_p , δ_s , et bande de transition exigeront généralement un plus grand nombre des coefficients de filtre. Il est difficile de déterminer le nombre de coefficients de

filtre exigés pour répondre à un caractéristique donné de filtre pour un algorithme de synthèse particulier, et une procédure itérative peut devenir nécessaire [1].

Une différence principale entre les filtres RII et RIF est sur le problème lié à la stabilité. Un filtre de RIF est toujours stable tant que $h(n_1, n_2)$ est bornée (fini) pour tous (n_1, n_2) , ainsi la stabilité n'est jamais un problème. Avec un filtre RII, cependant, assurer la stabilité est une mission principale.

Une manière est constituer à synthèse un filtre stable RII par imposer une structure spéciale à $H(z_1, z_2)$ tels que testant la stabilité et en stabilisant un filtre instable devenir relativement une mission simple. Cependant, une telle approche, tendance à imposer une contrainte stricte à l'algorithme de synthèse ou à limiter fortement la classe des filtres qui peuvent être synthétisé. Par exemple, si $H(z_1, z_2)$ a un polynôme séparable de dénominateur de la forme $A_1(z_1)A_2(z_2)$, testant la stabilité et stabilisant un $H(z_1, z_2)$ instable sans affecter l'amplitude est un problème 1-D. Cependant, la classe des filtres qui peuvent être synthétisé avec un polynôme séparable de dénominateur sans gain important dans le nombre de coefficients dans le polynôme de numérateur de $H(z_1, z_2)$ est restreinte. Une alternative manière est de synthèse un filtre sans considérer le problème de stabilité, et puis teste la stabilité du filtre résultant et essaye de le stabiliser si est instable. Le test de stabilité et stabilisation un filtre instable ne soient pas des problèmes faciles, cette approche n'impose pas des contraintes strictes aux algorithmes de synthèse et ne limite pas la classe des filtres qui peuvent être synthétisé.

Dans le cas 1-D, il y a deux approches standard de synthèse les filtres RII. On est de synthèse le filtre à partir la fonction transfert dans le domaine analogique (filtres prototypes), et l'autre est de synthèse directement dans le domaine discret. La première méthode est en général beaucoup plus simple et plus utile que la seconde. Utiliser la fonction transfert d'un filtre analogique elliptique et la transformation bilinéaire, par exemple, nous pouvons synthétiser les filtres passe-bas, passe-haut, passe-bande, et coupe-bande en suivant quelques étapes simples. Malheureusement, cette méthode n'est pas utile dans le cas 2-D. Dans le cas 1-D, cette méthode exploite la disponibilité de beaucoup de méthodes simples pour synthèse des filtres analogue 1-D. Ces méthodes n'existent pas pour la synthèse des filtres analogiques 2-D. La deuxième méthode, synthèse un filtre RII directement dans le domaine discret, peut être classifiée dans deux catégories. Le premier est de synthèse dans le domaine spatiale, où les filtres sont synthétisés en utilisant un critère d'erreur dans le domaine spatial. La seconde est de synthèse dans le domaine de fréquence, où des filtres sont synthétisés en utilisant un critère d'erreur dans le domaine de fréquence. La grandeur d'amplitude d'un filtre RII est spécifiée en utilisant le schéma de tolérance, et les performances de différents algorithmes de synthèse sont souvent comparées à la base de schéma de tolérance. Par conséquent, le critère d'erreur de Tchebychev, connu sous le nom de critère d'erreur min-max, est un choix normal pour synthèse les filtres RII. Un critère d'erreur de ce type, cependant, mène à un problème fortement non linéaire [1].

3.3 Problème de stabilité

Dans un problème typique de stabilité, nous sommes données la fonction transfert d'un filtre $H(z_1, z_2)$ et des informations sur son *ROC*, et nous souhaitons déterminer la stabilité. Si le *ROC* est explicitement donné, la détermination de la stabilité de filtre est donnée. C'est parce qu'un filtre est stable si et seulement si le *ROC* inclut la surface d'unité ($|z_1|=1, |z_2|=1$). Malheureusement, le *ROC* est rarement donné explicitement et habituellement seulement les informations implicites sur le *ROC* sont disponibles. Quand la fonction transfert de filtre est obtenue, la région du support de sa réponse impulsionnelle $h(n_1, n_2)$ est typiquement connue

de l'étape de synthèse de filtre. Puisque notre intérêt principal teste la stabilité des filtres numériques, nous considérerons le problème de tester la stabilité de $H(z_1, z_2)$ et la région du support de $h(n_1, n_2)$.

De (1.22) nous supposons que $A(z_1, z_2)$ et $B(z_1, z_2)$ sont co-premiers, ainsi $H(z_1, z_2)$ est irréductible. Une autre restriction est que le système doit être récursivement calculable.

Avec ces deux restrictions, le système peut être réalisé par une équation aux différences avec des conditions initiales et la sortie peut être calculée récursivement, une valeur à la fois. Dans le cas 1-D, quand la fonction transfert du système $H(z)$ est exprimé comme $B(z)/A(z)$, où il n'y a aucun facteur commun entre $A(z)$ et $B(z)$, $B(z)$ n'affecte pas la stabilité du système. Dans le cas 2-D, cependant, la présence de $B(z_1, z_2)$ de (1.22) peut stabiliser un système instable, même lorsqu'il n'y a aucun facteur commun entre $A(z_1, z_2)$ et $B(z_1, z_2)$. Dans le cas 1-D, le problème de tester la stabilité d'un système causal dont la fonction transfert est donné par $H(z) = 1/A(z)$ est tout à fait directement. Depuis un polynôme 1-D $A(z)$ peut toujours être factorisé simplement comme produit des polynômes de premier ordre, nous peut facilement déterminer les pôles de $H(z)$. La stabilité du système causal est équivalente à avoir tous les pôles à l'intérieur du cercle d'unité. La manière précédente ne peut pas être utilisée en testant la stabilité d'un système 2-D de support de premier quart de cercle. Cette méthode exige de spécifier l'emplacement de tous les pôles. En partie parce qu'un polynôme $A(z_1, z_2)$ ne peut pas en général être factorisé comme produit des polynômes de d'ordres inférieure, il est extrêmement difficile de déterminer toutes les surfaces de pôles de $H(z_1, z_2) = 1/A(z_1, z_2)$ et la méthode basée sur la détermination de toutes les surfaces de pôles n'a pas mené aux procédures pratiques réussies pour tester la stabilité du système 2-D [1].

3.3.1 Théorème de stabilité

C'est un théorème qui peut être utilisé pour développer des procédures pratiques pour tester la stabilité d'un système 2-D de support de premier quart de cercle sans détermination explicite de tous les pôles.

Théorème

Ce théorème est initialement développé par DeCarlo-Strintzis, peut être écrit comme suit :

$$\begin{aligned} \text{Stabilité} \Leftrightarrow & \quad (a) \quad A(z_1, z_2) \neq 0, \quad |z_1| = |z_2| = 1 \\ & \quad \text{et} \quad (b) \quad A(z_1, 1) \neq 0, \quad |z_1| \geq 1 \\ & \quad \text{et} \quad (c) \quad A(1, z_2) \neq 0, \quad |z_2| \geq 1. \end{aligned} \quad (3.3)$$

Algorithmes pour tester la stabilité

Nous pouvons développer beaucoup de différentes méthodes utiles pour tester la stabilité du filtre 2-D selon le théorème précédent. Une telle méthode est Indiquée dans la figure 3.1.

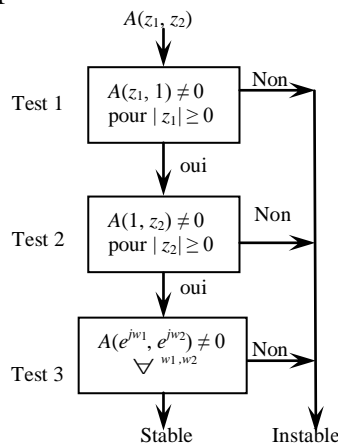


Figure 3.1 tester la stabilité d'un système 2-D de support de premier quart de cercle $H(z_1, z_2) = 1/A(z_1, z_2)$.

Test1 dans la figure pour assurer la condition (b) du théorème. Test 2 assurer la condition (c) du théorème. Test 3 assuré (a) du théorème. Si le système franchir chacun des trois tests, il est stable. Autrement, il est instable.

Les procédures de stabilité sont basées sur la notion que le test de stabilité d'un système 2-D peut être réalisé par un grand ensemble de tests de stabilité d'un système 1-D.

La stabilité n'est pas absolument garantie avec un numéro fini des tests de stabilité d'un système 1-D, même avec la précision infinie dans les opérations arithmétiques [1].

3.4 Synthèse dans le domaine spatial

Dans la synthèse du filtre RII dans le domaine spatial, quelques réponses désirées d'un système dans le domaine spatial à entrée connue est assumée donnée. Les coefficients de filtre sont estimés tels que la réponse du filtre synthétisée à l'entrée connue est étroitement analogue que possible à la réponse désirée. Ceci, naturellement, ne réduit pas au minimum la norme d'erreur de Tchebychev, mais c'est une méthode normale si nous souhaitons le filtre préserver quelques propriétés désirables dans ce domaine. L'entrée utiliser dans la synthèse de filtre RII est $\delta(n_1, n_2)$, et la réponse impulsionnelle désirée qui est assumée que donné est dénoté par $h_d(n_1, n_2)$.

La synthèse dans le domaine spatial peut être visualisée comme problème d'identification de système. Supposer que nous avons un système inconnu que nous souhaitons modeler avec une fonction de transfert rationnel $H(z_1, z_2)$. Une manière à estimer les paramètres de modèle de système [coefficients de filtre $a(n_1, n_2)$ et $b(n_1, n_2)$ dans notre cas] est d'exiger de la réponse impulsionnelle du système d'être étroitement analogue que possible au $h_d(n_1, n_2)$.

Dans l'identification de système, modelant un système par $H(z_1, z_2)$ comme (3.2) s'appelle modélisation ARMA (auto-regressive moving-average).

Parmi beaucoup de possibilités, un critère d'erreur utilisé dans la synthèse de filtre est donner par :

$$\text{Erreur} = \sum_{(n_1, n_2) \in R_e} e^2(n_1, n_2) \quad (3.4a)$$

$$\text{avec} \quad e(n_1, n_2) = h_d(n_1, n_2) - h(n_1, n_2) \quad (3.4b)$$

et R_e est la région du support de la séquence d'erreur. Idéalement, R_e est toutes les valeurs de (n_1, n_2) . Dans la pratique, cependant, il étend seulement d'une région finie de (n_1, n_2) , où $h_d(n_1, n_2)$ a l'énergie significative. L'erreur de moindre carrée de (3.4) est choisie parce qu'elle est appliquée largement dans les problèmes d'identification de système et de simple modification de (3.4) assister a développée quelques méthodes simples pour estimer les paramètres de système.

La minimisation de l'erreur de (3.4) par apport $a(n_1, n_2)$ et $b(n_1, n_2)$ est un problème non linéaire.

Une approche alternative pour minimiser l'erreur de (3.4) est de modifier légèrement l'erreur, de sorte que l'algorithme en résultant mène aux solutions de forme fermie (*closed form*) qui exige résoudre seulement des ensembles d'équations linéaires [1].

3.4.1 Algorithme linéaire de forme fermée

L'algorithme que nous discutons dans cette section conduire aux solutions de forme fermée qui exigent résoudre seulement des ensembles d'équations linéaires. Elle est considérablement plus simple de calcul que l'Algorithme itérative discutée dans la prochaine section.

Cependant, aucun de eux ne minimiser l'erreur de (3.4).

La méthode discutée dans cette section est basée sur l'observation qu'une modification raisonnable de l'expression d'erreur de (3.4) transforme un problème non linéaire à linéaire.

Considérer la procédure calculable suivante :

$$y(n_1, n_2) \leftarrow - \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) y(n_1 - k_1, n_2 - k_2) + \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \quad (3.5)$$

Nous supposons que nombre des valeurs inconnues de $a(n_1, n_2)$ est p , et $q + 1$ valeurs inconnu pour $b(n_1, n_2)$, et ainsi un total des coefficients de filtre $N = p + q + 1$ à déterminer.

Remplaçant $x(n_1, n_2)$ par $\delta(n_1, n_2)$ et $y(n_1, n_2)$ avec $h(n_1, n_2)$ dans (3.5) et note que $\sum_{(k_1, k_2) \in R_b} b(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)$ est $b(n_1, n_2)$, on a

$$h(n_1, n_2) \leftarrow - \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) h(n_1 - k_1, n_2 - k_2) + b(n_1, n_2) \quad (3.6)$$

Si nous remplaçons $h(n_1, n_2)$ dans les deux côtés de (3.6) par $h_d(n_1, n_2)$ donné, alors l'équation de (3.6) ne se devenir pas correct :

$$h_d(n_1, n_2) \leftarrow - \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2) + b(n_1, n_2) \quad (3.7)$$

Puisque nous souhaitons approximer $h_d(n_1, n_2)$ par $h(n_1, n_2)$, il est raisonnable de définir une séquence d'erreur $e_M(n_1, n_2)$ comme la différence entre le côté gauche et droite de (3.7) :

$$e_M(n_1, n_2) = h_d(n_1, n_2) + \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2) - b(n_1, n_2) \quad (3.8)$$

Il est clair que $e_M(n_1, n_2)$ dans (3.8) ne soit pas identique que $e(n_1, n_2)$ dans (3.4b). L'indice M à $e_M(n_1, n_2)$ est une indication que cette $e_M(n_1, n_2)$ est une modification de $e(n_1, n_2)$.

La minimisation de $e_M(n_1, n_2)$ par apport les coefficients inconnus de filtre $a(n_1, n_2)$ et $b(n_1, n_2)$ est un problème linéaire.

On a $H(z_1, z_2) = B(z_1, z_2)/A(z_1, z_2)$ donc $b(n_1, n_2) = h(n_1, n_2) * a(n_1, n_2)$, (3.8) peut être récrit par

$$e_M(n_1, n_2) = a(n_1, n_2) * h_d(n_1, n_2) - a(n_1, n_2) * h(n_1, n_2) \quad (3.9)$$

De (3.9), il est clair que $e_M(n_1, n_2)$ dans (3.8) ne soit pas définie dans le domaine du $h_d(n_1, n_2)$ et du $h(n_1, n_2)$. Au lieu de ça, il est défini dans un nouveau domaine où $h_d(n_1, n_2)$ et $h(n_1, n_2)$ filtré avec $a(n_1, n_2)$. Pour cette raison, des méthodes basées à l'équation (3.8) ou d'une manière équivalente de (3.9) s'appellent les méthodes de modélisation du signal indirect [1].

La figure 3.2 donné la différence entre $e(n_1, n_2)$ de (3.4b) et $e_M(n_1, n_2)$ de (3.8).

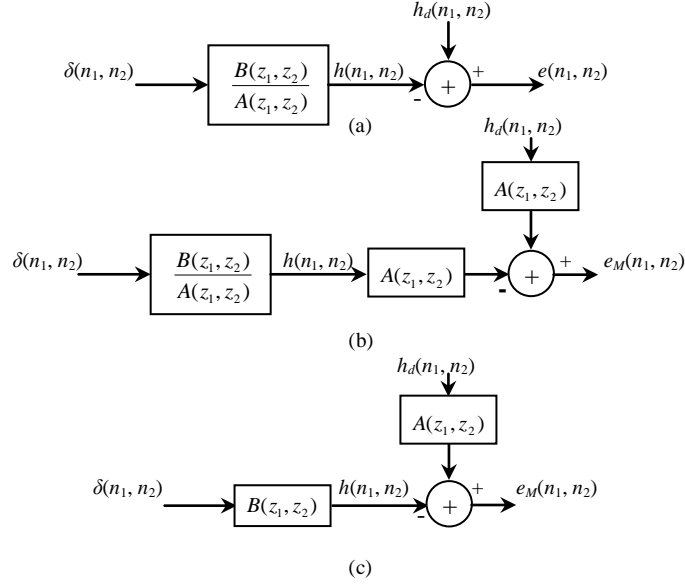


Figure 3.2 Erreurs de moindre carrée et moindre carrée modifiées.
 (a) $e(n_1, n_2)$ utilisé dans le critère d'erreur de moindre carrée ;
 (b) $e_M(n_1, n_2)$ utilisée dans le critère d'erreur modifié de moindre carrée;
 (c) simplification de (b).

a) La méthode de Prony

Dans la méthode de Prony, l'expression d'erreur à minimiser est :

$$\text{Erreur} = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} e_M^2(n_1, n_2) \quad (3.10)$$

où $e_M(n_1, n_2)$ est donnée par (3.8).

L'erreur de (3.10) est une formule quadratique de l'inconnu $a(n_1, n_2)$ et $b(n_1, n_2)$, et minimisation de l'expression d'erreur par rapport aux paramètres inconnus exige résoudre un ensemble d'équations linéaires.

Les équations linéaires $p + q + 1$ sont données par :

$$\frac{\partial \text{Erreur}}{\partial a(n_1, n_2)} = 0, \quad (l_1, l_2) \in R_a - (0, 0) \quad (3.11a)$$

et
$$\frac{\partial \text{Erreur}}{\partial b(n_1, n_2)} = 0, \quad (l_1, l_2) \in R_b \quad (3.11b)$$

L'observation de l'erreur de (3.10) prouve que (3.11) peut être résolue par premièrement des équations linéaires p pour $a(n_1, n_2)$ et puis résoudre des équations linéaires $q + 1$ pour $b(n_1, n_2)$. Il est utile de réécrire (3.10) par

$$\text{Erreur} = E_1 + E_2 \quad (3.12a)$$

avec
$$E_1 = \sum_{(n_1, n_2) \in R_b} e_M^2(n_1, n_2) \quad (3.12b)$$

et
$$E_2 = \sum_{(n_1, n_2) \notin R_b} e_M^2(n_1, n_2) \quad (3.12c)$$

L'expression E_1 dans (3.12b) se compose $q + 1$ termes, et E_2 dans (3.12c) se compose d'un grand nombre de termes. Considérer E_2 d'abord, de (3.8) et (3.12c), E_2 peut être exprimé par :

$$E_2 = \sum_{(n_1, n_2) \notin R_b} \sum (h_d(n_1, n_2) + \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2) - b(n_1, n_2))^2 \quad (3.13)$$

Pour $(n_1, n_2) \notin R_b$, $b(n_1, n_2) = 0$, et donc (3.13) simplifie à

$$E_2 = \sum_{(n_1, n_2) \notin R_b} \sum (h_d(n_1, n_2) + \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2))^2 \quad (3.14)$$

Nous remarquons que E_2 ne dépend pas de $b(n_1, n_2)$. Considérer maintenant E_1 , puisque E_1 se compose de $q + 1$ termes qui sont quadratiques dans $a(n_1, n_2)$ et $b(n_1, n_2)$ et il y a $q + 1$ termes de $b(n_1, n_2)$, nous peut choisir $b(n_1, n_2)$ tels que E_1 est zéro pour $a(n_1, n_2)$. Ceci signifie que la minimisation de l'erreur de (3.12a) par apport $a(n_1, n_2)$ est équivalente la minimisation de E_2 par apport $a(n_1, n_2)$.

Minimisation de E_2 dans (3.12) par apport $a(n_1, n_2)$ donné un p équations linéaires pour p inconnus, donnés par :

$$\sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) r(k_1, k_2; l_1, l_2) = -r(0,0; l_1, l_2), \quad (l_1, l_2) \in R_a - (0, 0) \quad (3.15a)$$

$$\text{avec} \quad r(k_1, k_2; l_1, l_2) = \sum_{(n_1, n_2) \notin R_b} h_d(n_1 - k_1, n_2 - k_2) h_d(n_1 - l_1, n_2 - l_2) \quad (3.15b)$$

Une fois que $a(n_1, n_2)$ est déterminé, nous pouvons minimiser l'erreur dans (3.12a) par apport $b(n_1, n_2)$. Puisque E_2 ne dépend pas de $b(n_1, n_2)$, minimisation l'erreur dans (3.12a) par apport $b(n_1, n_2)$ est équivalent la minimisation de E_1 dans (3.12b) par apport $b(n_1, n_2)$. L'erreur E_1 peut être minimiser en différenciant E_1 par apport $b(n_1, n_2)$ et en plaçant le résultat à zéro, ou alternativement on remarque que E_1 peut être réduit à zéro par :

$$e_M(n_1, n_2) = 0 \text{ for } (n_1, n_2) \in R_b \quad (3.16)$$

Dans l'un du deux cas, les équations résultant pour $b(n_1, n_2)$ sont données par :

$$b(n_1, n_2) = h_d(n_1, n_2) + \sum_{(k_1, k_2) \in R_a - (0,0)} a(k_1, k_2) h_d(n_1 - k_1, n_2 - k_2) \quad , (n_1, n_2) \in R_b \quad (3.17)$$

Nous pouvons voir de (3.17) que $b(n_1, n_2)$ n'exige pas de résoudre les équations linéaires $q + 1$ simultanément, et les coefficients de $b(n_1, n_2)$ peuvent être trouver un par un. La méthode discutée ci-dessus peut être visualisée comme application directe d'une méthode d'identification de système développée par Prony au problème de synthèse un filtre RII 2-D. Puisque cette méthode a été la première fois appliquée au problème de synthèse de filtre RII 2-D par [Shanks et autres], elle désigné parfois sous le nom de la méthode de Shanks [1]. Puisque la méthode de Prony essaye de minimiser l'erreur carrée, le filtre en résultant est peut être stable [1].

Il y a quelque nombre de variations à la méthode décrite ci-dessus. Par exemple, $a(n_1, n_2)$ et $b(n_1, n_2)$ ont été estimés par la minimisation de l'erreur dans (3.10). En raison du choix spécifique de $e_M(n_1, n_2)$, cependant, $b(n_1, n_2)$ a été déterminé essentiellement à partir les $q + 1$ termes de $e_M(n_1, n_2)$. Une autre méthode à améliorer l'évaluation de $b(n_1, n_2)$ par d'utiliser l'expression initiale d'erreur dans (3.4).

Quoique la minimisation de l'erreur dans (3.4) par apport $a(n_1, n_2)$ et $b(n_1, n_2)$ soit un problème non linéaire, elle devient linéaire si nous supposons que $a(n_1, n_2)$ est donné. Ceci est illustré dans la figure 3.3.

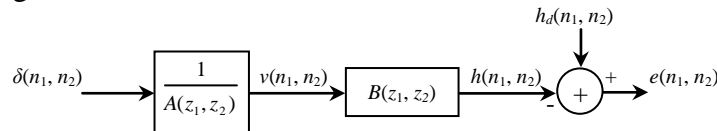


Figure 3.3 le signal de l'erreur $e(n_1, n_2)$ utilisé dans l'estimation $b(n_1, n_2)$ dans la méthode du Prony modifié.

Nous estimons d'abord $a(n_1, n_2)$ suivre la méthode de Prony. Puis, supposant que $a(n_1, n_2)$ est donné, nous calculons $v(n_1, n_2)$, la sortie de $1/A(z_1, z_2)$ avec l'entrée du $\delta(n_1, n_2)$. Si $1/A(z_1, z_2)$ est instable, $v(n_1, n_2)$ est probablement augmenter dans l'amplitude, et nous ne pourra pas continuer. Dans ce cas ci, il n'est pas utile d'estimer $b(n_1, n_2)$. Au lieu de ça, nous devrions estimer $a(n_1, n_2)$ meilleur. Supposant que nous pouvons procéder, nous pouvons récrire (3.4) en termes de $v(n_1, n_2)$ par

$$\begin{aligned} \text{Erreur} &= \sum_{(n_1, n_2) \in R_e} \sum (h_d(n_1, n_2) - v(n_1, n_2) * b(n_1, n_2))^2 \\ &= \sum_{(n_1, n_2) \in R_e} \sum (h_d(n_1, n_2) - \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) v(n_1 - k_1, n_2 - k_2))^2 \end{aligned} \quad (3.18)$$

Alors que $h_d(n_1, n_2)$ et $v(n_1, n_2)$ sont connus, l'erreur dans (3.18) est une forme quadratique des paramètres inconnus $b(n_1, n_2)$ et peut donc être estimé par résoudre un ensemble d'équations linéaires. Différenciant l'erreur dans (3.18) par rapport $b(l_1, l_2)$ pour $(l_1, l_2) \in R_b$

$$\begin{aligned} \sum_{(k_1, k_2) \in R_b} b(k_1, k_2) \left(\sum_{(n_1, n_2) \in R_e} v(n_1 - k_1, n_2 - k_2) v(n_1 - l_1, n_2 - l_2) \right) \\ = \sum_{(n_1, n_2) \in R_e} h_d(n_1, n_2) v(n_1 - l_1, n_2 - l_2) \end{aligned} \quad (3.19)$$

La relation (3.19) est une $q + 1$ équations linéaires pour $q + 1$ paramètres inconnus $b(n_1, n_2)$.

b) Algorithme Itérative

C'est une extension d'une méthode d'identification de système 1-D développée par Steiglitz et McBride [1].

De (3.9), $e(n_1, n_2) = h_d(n_1, n_2) - h(n_1, n_2)$ est lié à $e_M(n_1, n_2)$ par

$$e_M(n_1, n_2) = a(n_1, n_2) * e(n_1, n_2) \quad (3.20)$$

L'équation (3.20) peut être réécrite par

$$e(n_1, n_2) = v(n_1, n_2) * e_M(n_1, n_2) \quad (3.21a)$$

$$\text{avec} \quad V(z_1, z_2) = \frac{1}{A(z_1, z_2)} \quad (3.21b)$$

La séquence $v(n_1, n_2)$ est l'inverse de $a(n_1, n_2)$. De (3.12) et (3.21a),

$$\begin{aligned} e(n_1, n_2) &= v(n_1, n_2) * e_M(n_1, n_2) \\ &= v(n_1, n_2) * (a(n_1, n_2) * h_d(n_1, n_2) - b(n_1, n_2)) \end{aligned} \quad (3.22)$$

L'équation (3.22) peut également être dérivée de

$$\begin{aligned} E(z_1, z_2) &= H_d(z_1, z_2) - H(z_1, z_2) \\ &= H_d(z_1, z_2) - \frac{B(z_1, z_2)}{A(z_1, z_2)} \\ &= \frac{1}{A(z_1, z_2)} (A(z_1, z_2) H_d(z_1, z_2) - B(z_1, z_2)) \\ &= V(z_1, z_2) (A(z_1, z_2) H_d(z_1, z_2) - B(z_1, z_2)) \end{aligned} \quad (3.23)$$

De (3.22), si $v(n_1, n_2)$ est donné de façon ou d'autre, alors $e(n_1, n_2)$ est linéaire dans $a(n_1, n_2)$ et $b(n_1, n_2)$, ainsi minimisation $\sum_{n_1} \sum_{n_2} e^2(n_1, n_2)$ par rapport $a(n_1, n_2)$ et $b(n_1, n_2)$ est un problème linéaire. Évidemment, si $v(n_1, n_2)$ est donné, alors $a(n_1, n_2)$ automatiquement sera donné par (3.21b).

Algorithme :

Etape 1 : On commence par une estimation initiale de $a(n_1, n_2)$, obtenue suivre une méthode telle que Prony.

Etape 2 : Obtenir $v(n_1, n_2)$ de $a(n_1, n_2)$.

Etape 3 : Minimiser $\sum_{n_1} \sum_{n_2} e^2(n_1, n_2)$ par rapport $a(n_1, n_2)$ et $b(n_1, n_2)$ en résolvant un ensemble d'équations linéaires.

Etape 4 : Nous avons maintenant une nouvelle estimation de $a(n_1, n_2)$, et le processus continue jusqu'à nous obtenions $a(n_1, n_2)$ et $b(n_1, n_2)$ désirer.

La méthode est illustrée dans la figure 3.4.

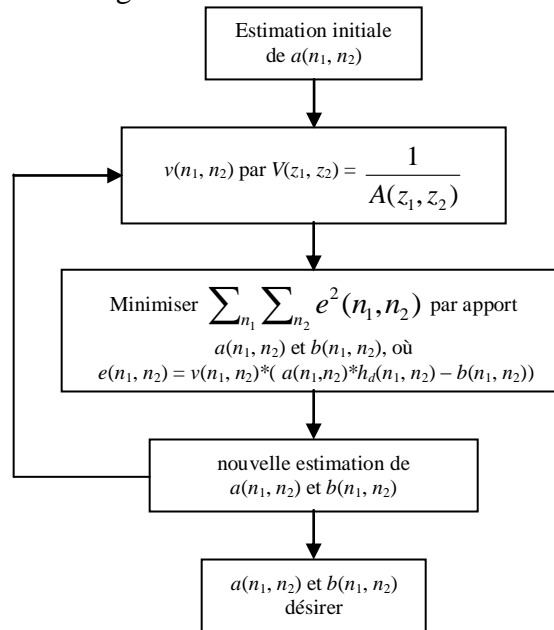


Figure 3.4 méthode itérative.

La convergence de l'algorithme itérative discuté dans cette section ne peut pas être garantie. En plus, les conditions dans lesquelles l'algorithme converge ne sont pas connues. La méthode exige de calcul $v(n_1, n_2)$, l'inverse de $a(n_1, n_2)$, dans chaque itération. Si $1/A(z_1, z_2)$ est instable à un point quelconque, l'algorithme itérative ne peut pas contenue.

Malgré de ces problèmes qui ont été produits dans la pratique, l'algorithme itérative a été avec succès employée dans la synthèse des filtres RII par Shaw et Mersereau. L'algorithme est habituellement terminé avant convergence. Le filtre synthétiser, quand la méthode est réussie, semble être « bon » dans les limites de quelques itérations, et de l'erreur basée sur $v(n_1, n_2)$ dans (3.22) diminue très lentement après quelques itérations [1].

c) Synthèse de filtre à phase linéaire

Dans quelques applications, synthèse de filtre à phase linéaire peut être nécessaire. Il est simple de synthèse des filtres RIF à phase linéaire. Il est impossible de synthèse un seul filtre RII avec une phase linéaire. Pour avoir la phase linéaire, $h(n_1, n_2)$ doit être égal à $h(-n_1, -n_2)$. Un filtre RII exige une séquence infinie $h(n_1, n_2)$, et une sortir a un support intercalé. Ces conditions ne peuvent pas tous être satisfaites en même temps. Il est possible, cependant, pour réaliser un filtre a phase linéaire on utilisant plus d'un filtre RII.

Une méthode particulièrement bien adaptée dans le domaine spatial est de diviser $h_d(n_1, n_2)$ en différentes régions, synthétiser un filtre RII pour approximer $h_d(n_1, n_2)$ dans chaque région, et puis combinent les filtres en utilisant une structure parallèle [1].

Supposer que nous avons un $h_d(n_1, n_2)$ désiré, on suppose que

$$h_d(n_1, n_2) = h_d(-n_1, -n_2) \quad (3.24)$$

Nous pouvons diviser $h_d(n_1, n_2)$ en nombre pair des régions: deux, quatre, six, huit, ou plus.

Supposer que nous divisons $h_d(n_1, n_2)$ en quatre régions

$$h_d^I(n_1, n_2) = h_d(n_1, n_2)w(n_1, n_2) \quad (3.25a)$$

$$h_d^{II}(n_1, n_2) = h_d(n_1, n_2)w(-n_1, n_2) \quad (3.25b)$$

$$h_d^{III}(n_1, n_2) = h_d(n_1, n_2)w(-n_1, -n_2) \quad (3.25c)$$

$$\text{et} \quad h_d^{IV}(n_1, n_2) = h_d(n_1, n_2)w(n_1, -n_2) \quad (3.25d)$$

où $w(n_1, n_2)$ est une séquence de support de premier quart de cercle donné par :

$$w(n_1, n_2) = \begin{cases} 1, & n_1 \geq 1, n_2 \geq 1 \\ \frac{1}{2}, & n_1 \geq 1, n_2 = 0 \\ \frac{1}{2}, & n_1 = 0, n_2 \geq 1 \\ \frac{1}{4}, & n_1 = n_2 = 0 \end{cases} \quad (3.26)$$

La séquence $w(n_1, n_2)$ est illustrer dans la figure 3.5. On le choisit tels qu'il y aura symétrie maximum et la somme de quatre fenêtres égal 1.

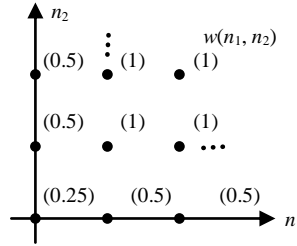


Figure 3.5 la séquence $w(n_1, n_2)$ de (3.26).

De (3.25) et (3.26), il est clair que

$$h_d(n_1, n_2) = h_d^I(n_1, n_2) + h_d^{II}(n_1, n_2) + h_d^{III}(n_1, n_2) + h_d^{IV}(n_1, n_2) \quad (3.27)$$

Depuis $h_d^I(n_1, n_2)$, $h_d^{II}(n_1, n_2)$, $h_d^{III}(n_1, n_2)$ et $h_d^{IV}(n_1, n_2)$ sont des séquences de support de quart de cercle, ils peuvent être implémentés au moyen de systèmes récursivement calculables.

Supposer que nous utilisons un des techniques de synthèse dans la domaine spatiales pour synthétiser $H^I(z_1, z_2)$ qui approximer $h_d^I(n_1, n_2)$. De même, supposer que nous avons synthétiser $H^{II}(z_1, z_2)$ qui approximer $h_d^{II}(n_1, n_2)$. De (3.24) et (3.25),

$$h_d^{III}(n_1, n_2) = h_d^I(-n_1, -n_2) \quad (3.28)$$

Par conséquent, $H^{III}(z_1, z_2)$ qui approximer $h_d^{III}(n_1, n_2)$ peut être obtenu à partir de $H^I(z_1, z_2)$ par

$$H^{III}(z_1, z_2) = H^I(z_1^{-1}, z_2^{-1}) \quad (3.29)$$

De même, $H^{IV}(z_1, z_2)$ peut être obtenu à partir de $H^{II}(z_1, z_2)$ par :

$$H^{IV}(z_1, z_2) = H^{II}(z_1^{-1}, z_2^{-1}) \quad (3.30)$$

Depuis $H^I(z_1, z_2)$, $H^{II}(z_1, z_2)$, $H^{III}(z_1, z_2)$, et $H^{IV}(z_1, z_2)$ approximer $h_d^I(n_1, n_2)$, $h_d^{II}(n_1, n_2)$, $h_d^{III}(n_1, n_2)$ et $h_d^{IV}(n_1, n_2)$, respectivement, de (3.27), $h_d(n_1, n_2)$ seront approximer par :

$$\begin{aligned} H(z_1, z_2) &= H^I(z_1, z_2) + H^{II}(z_1, z_2) + H^{III}(z_1, z_2) + H^{IV}(z_1, z_2) \\ &= H^I(z_1, z_2) + H^{II}(z_1, z_2) + H^I(z_1^{-1}, z_2^{-1}) + H^{II}(z_1^{-1}, z_2^{-1}) \end{aligned} \quad (3.31)$$

Chacun des quatre filtres approximer une séquence de support d'un quart de cercle et est récursivement calculable. $H(z_1, z_2)$ à phase linéaire car $H(z_1, z_2) = H(z_1^{-1}, z_2^{-1})$. Le système dans (3.31) peut être implémenté en utilisant une structure parallèle comme la figure 3.6 indiqué. L'entrée est filtrée par chacun des quatre systèmes, et les résultats sont combinés pour produire la sortie.

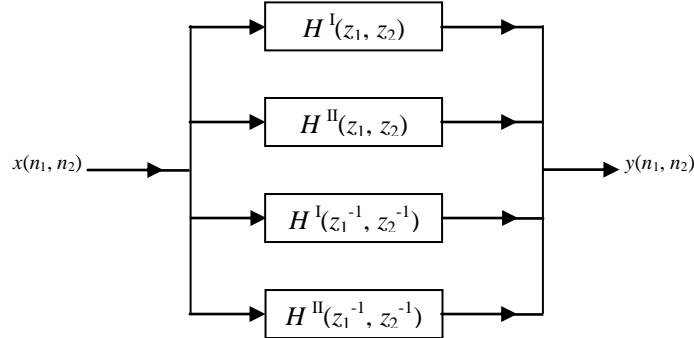


Figure 3.6 l'implémentation de $H(z_1, z_2)$ par la structure parallèle.

Si $h_d(n_1, n_2)$ a la symétrie quadruple,

$$h_d^{II}(n_1, n_2) = h_d^I(-n_1, n_2) \quad (3.32)$$

et donc $H^{II}(z_1, z_2)$ peut être déterminé de $H^I(z_1, z_2)$ par :

$$H^{II}(z_1, z_2) = H^I(z_1^{-1}, z_2) \quad (3.33)$$

Dans ce cas, $H(z_1, z_2)$ dans (3.31) est donné par :

$$H(z_1, z_2) = H^I(z_1, z_2) + H^I(z_1^{-1}, z_2) + H^I(z_1^{-1}, z_2^{-1}) + H^I(z_1, z_2^{-1}) \quad (3.34)$$

De (3.34), dans ce cas seulement un filtre doit être synthétisé [1].

3.5 Synthèse dans le domaine fréquentielle

3.5.1 Synthèse par la programmation itérative semi-définie

La programmation semi-définie (Semidefinite programming : SDP) a récemment attiré beaucoup d'intérêt de recherches. Un des autres, est un outil d'optimisation s'applique aux synthèses des filtres numériques RIF. Une tentative sur étendre l'approche de filtres numériques RIF au filtres RII 2-D. On peut formuler le problème de synthèse d'un filtre RII stable comme une itérative de SDP dans le sens de min-max.

Les contraintes de stabilité sont exprimées comme inégalités linéaires formulées comme matrice qui s'insèrent dans la configuration de SDP [3].

Dans cette section on assume que les filtres RII ont des dénominateurs séparables. Cette supposition impose simplement une contrainte au type de filtres RII étant symétrie quadruple. Néanmoins cette classe des filtres est couvre pratiquement tous les types de filtres RII qui ont été trouvés utiles dans le traitement d'image/vidéo [4].

Méthode de synthèse en sens Min-Max

La méthode est essentiellement un algorithme de minimisation séquentiel et est comme suit : A l'itération de $k^{\text{ème}}$ une approximation linéaire de la fonction de transfert est obtenue par l'utilisation l'expansion de Taylor. De cette façon, on peut déduire une fonction objective qui est une limite supérieure de l'erreur approximer dans la région de fréquence d'intérêt.

La fonction objective est alors minimiser par rapport à plusieurs contraintes. Ces contraintes incluent les contraintes de stabilité. La contrainte de stabilité est convertie à une matrice d'inégalité linéaire (LMI) et on montrer que cette matrice dans chaque itération peut être exécutée par utilisation la programmation semi-définie (SDP).

A. Approximation linéaire de la fonction de transfert

Considérer un filtre numérique RII quadratiquement symétrique dont la fonction de transfert est donnée par :

$$H(z_1, z_2) = \frac{B(z_1, z_2)}{A(z_1)A(z_2)} \quad (3.35)$$

$$\text{où } B(z_1, z_2) = \sum_{k_1=0}^n \sum_{k_2=0}^n b(k_1, k_2) z_1^{-k_1} z_2^{-k_2} \quad \text{et} \quad A(z) = \sum_{k=0}^r a(k) z^{-k}, \quad a(0) = 1.$$

Puisque le filtre est quadratiquement symétrique, nous avons $b(k_1, k_2) = b(k_2, k_1)$. En conséquence, il y a seulement de $r + (n + 1)(n + 2)/2$ variables dans (3.35), qui forment un vecteur à dimension $[r + 0.5(n + 1)(n + 2)]$

$$x = [a_1 \cdots a_r b_{00} \cdots b_{nn} b_{10} b_{20} b_{21} \cdots b_{n0} \cdots b_{n,n-1}]^T \quad (3.36)$$

Dénoter le vecteur x dans l'itération $k^{\text{ème}}$ comme x_k et la réponse fréquentielle du filtre pour $x = x_k$ par $H(e^{jw_1}, e^{jw_2}, x_k)$. À environs du x_k , la variable de synthèse peut être exprimée par :

$$x = x_k + \delta$$

La réponse fréquentielle peut être approximer en termes de fonction linéaire de δ par :

$$H(e^{jw_1}, e^{jw_2}, x) \approx H(e^{jw_1}, e^{jw_2}, x_k) + g_k^T \delta \quad (3.37)$$

où g_k est le gradient de $H(e^{jw_1}, e^{jw_2}, x)$ pour $x = x_k$, c.-à-d.,

$$g_k = \nabla H(e^{jw_1}, e^{jw_2}, x_k) \quad (3.38)$$

à partir (3.35), le vecteur de gradient peut être évalué par :

$$\frac{\partial H}{\partial x_i} = \begin{cases} -\frac{H}{A_1} e^{-jw_1} - \frac{H}{A_2} e^{-jw_2} & \text{pour } 1 \leq i \leq r \\ \frac{1}{A_1 A_2} e^{-j(w_1 + w_2)} & \text{pour } r + 1 \leq i \leq r + n + 1 \\ \frac{1}{A_1 A_2} [e^{-j(lw_1 + mw_2)} + e^{-j(mw_1 + lw_2)}] & \text{ailleurs} \end{cases} \quad (3.39)$$

où H, A_1, A_2 dénotent les valeurs $H(e^{jw_1}, e^{jw_2}), A(e^{jw_1}),$ et $A(e^{jw_2})$ évalué à $x = [x_1 x_2 \cdots x_i \cdots x_{\hat{n}}]^T, \hat{n} = r + 0.5(n + 1)(n + 2)$ respectivement.

B. Algorithme de synthèse

Programmation semi-définie

Un problème de SDP peut être formulé de plusieurs méthodes. Dans le cas de la synthèse de filtre, on peut prendre la formulation suivant

$$\text{minimiser } c^T x \quad (3.40a)$$

$$\text{soumis à } F(x) \succeq 0 \quad (3.40b)$$

$$F(x) = F_0 + \sum_{i=1}^n x_i F_i \quad (3.40c)$$

où, $c \in \mathbb{R}^{n \times 1}, F_i \in \mathbb{R}^{n \times n}$ pour $0 \leq i \leq n$ sont des matrices constantes avec F_i symétrique, et $F(x) \succeq 0$ indiquer que $F(x)$ est semi-défini positive.

Formulation de problème

La synthèse en sens min-max peut être obtenue comme solution du problème d'optimisation contrainte

$$\min_x \left[\max_{(w_1, w_2) \in \Omega} e(w_1, w_2, x) \right] \quad (3.41a)$$

$$\text{soumis à: } H(e^{jw_1}, e^{jw_2}, x) \text{ représente un filtre stable} \quad (3.41b)$$

$$\text{où } e(w_1, w_2, x) = W(w_1, w_2) \left| H(e^{jw_1}, e^{jw_2}, x) - H_d(w_1, w_2) \right|^2 \quad (3.41c)$$

$\Omega = \{(w_1, w_2) : -\pi \leq w_1, w_2 \leq \pi\}$, $H_d(w_1, w_2)$ est la réponse fréquentielle désirée, et $W(\omega_1, \omega_2) \geq 0$ est une fonction de pondération. Le problème en (3.41) peut être reformulé comme suivant :

$$\text{minimise } \mu \quad (3.42a)$$

$$\text{soumis à: } e(w_1, w_2, x) \leq \mu \text{ pour } (w_1, w_2) \in \Omega \quad (3.42b)$$

$$H(e^{jw_1}, e^{jw_2}, x) \text{ représente un filtre stable} \quad (3.42c)$$

où μ est traité comme un *variable supplémentaire* de synthèse. En utilisant l'approximation linéaire pour $H(e^{jw_1}, e^{jw_2}, x)$ dans (3.37), on à

$$e(w_1, w_2, x) \approx \delta^T Q_k \delta + 2\delta^T q_k + c_k \quad (3.43a)$$

$$\text{où } Q_k = W(w_1, w_2) \Re(\bar{g}_k g_k^T) \quad (3.43b)$$

$$q_k = W(w_1, w_2) \Re \left\{ \left[H(e^{jw_1}, e^{jw_2}, x_k) - H_d(w_1, w_2) \right] \bar{g}_k \right\} \quad (3.43c)$$

$$c_k = e(w_1, w_2, x_k) \quad (3.43d)$$

et la contrainte dans (3.42b) peut être approximer par :

$$(\mu - 2\delta^T q_k - c_k) - (Q_k^{\frac{1}{2}} \delta)^T (Q_k^{\frac{1}{2}} \delta) \geq 0 \quad (3.44)$$

pour $(w_1, w_2) \in \Omega$ où $Q_k^{\frac{1}{2}}$ est la racine carrée asymétrique de Q_k .

On peut démontrer que (3.44) reste vérifier si et seulement si :

$$\Phi_k(w_1, w_2) = \begin{bmatrix} I & Q_k^{\frac{1}{2}} \delta \\ \delta^T Q_k^{\frac{1}{2}} & \mu - 2\delta^T q_k - c_k \end{bmatrix} \succeq 0 \quad (3.45)$$

Si nous dénotons le vecteur complexe de gradient par $g_k = g_{kr} + jg_{ki}$, alors (3.43b) conduire à :

$$Q_k = W(w_1, w_2) (g_{kr} g_{kr}^T + g_{ki} g_{ki}^T) = G_k^T G_k \quad (3.46a)$$

$$\text{où } G_k = W^{\frac{1}{2}}(w_1, w_2) \begin{bmatrix} g_{kr} & g_{ki} \end{bmatrix}^T \quad (3.46b)$$

et (3.45) devient

$$\Phi_k(w_1, w_2) = \begin{bmatrix} I & G_k \delta \\ \delta^T G_k & \mu - 2\delta^T q_k - c_k \end{bmatrix} \succeq 0 \quad (3.47)$$

pour $(w_1, w_2) \in \Omega$. De (3.46) et (3.47) il est évident que $\Phi_k(w_1, w_2)$ soit une matrice symétrique de la dimension 3×3 .

Dans l'itération $k^{\text{ème}}$ le problème d'optimisation de (3.42) peut être formulé comme

$$\text{minimiser } \hat{c}^T \hat{\delta} \quad (3.48a)$$

$$\text{soumis à: } S_k \succeq 0 \quad (3.48b)$$

$$H(e^{jw_1}, e^{jw_2}, x) \text{ représente un filtre stable} \quad (3.48c)$$

$$\text{où} \quad \hat{c} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \hat{\delta} = \begin{bmatrix} \mu \\ \delta \end{bmatrix} \quad (3.48d)$$

$$\text{et} \quad S_k = \text{diag} \{ \Phi_k(w_1^{(1)}, w_2^{(1)}), \dots, \Phi_k(w_1^{(L)}, w_2^{(M)}) \} \quad (3.48e)$$

avec $\{(w_1^{(l)}, w_2^{(m)}) \text{ for } 1 \leq l \leq L, 1 \leq m \leq M\}$ l'ensemble de fréquences dans la région de d'intérêt.

Contraintes de stabilité

La contrainte de stabilité dans (3.48c) a besoin de modification approprié pour la rendre traitable pour l'optimisation numérique. Dénoter les vecteurs formés des premiers r composants du $x_k + \delta$ par $a_k + \delta_1$. Car le dénominateur de $H(z_1, z_2)$ est séparable, il peut être montré que le filtre RII avec de vecteur de coefficient $x_k + \delta$ est stable si et seulement si les modules des valeurs propres de la matrice

$$D_k = \begin{bmatrix} -(a_k + \delta_1)^T \\ \hat{I}_r \end{bmatrix} \quad (3.49)$$

sont tous strictement inférieur à 1, où \hat{I}_r dénote une matrice de la taille $(r-1) \times r$ obtenu en augmentant la matrice d'identité avec une colonne des zéros du côté droit. En ce cas, D_k est dit une matrice stable. Appliquant la théorie bien connue de Lyapunov [5], on conclut que la matrice D_k est stable si et seulement si l'existe une matrice définie positive P tels que

$$P - D_k^T P D_k \succ 0 \quad (3.50)$$

où $M \succ 0$ dénote que la matrice M est définie positive. Il peut aisément vérifier que l'inégalité de matrice dans (3.50) reste vérifier si et seulement si :

$$\begin{bmatrix} P^{-1} & D_k \\ D_k^T & P \end{bmatrix} \succ 0 \quad (3.51)$$

Pour assurer une marge de stabilité pour le filtre RII, les contraintes dans (3.51) sont modifiées à :

$$Y_k = \begin{bmatrix} P^{-1} - \tau I_r & D_k \\ D_k^T & P - \tau I_r \end{bmatrix} \succeq 0 \quad (3.52)$$

où τ est une scalaire positive qui contrôle la marge de stabilité du filtre.

Le problème dans (3.48) peut maintenant être converti en

$$\text{minimiser } \hat{c}^T \hat{\delta} \quad (3.53a)$$

$$\text{soumis à : } \begin{bmatrix} S_k & 0 \\ 0 & Y_k \end{bmatrix} \succeq 0 \quad (3.53b)$$

La matrice P de (3.52) n'est pas considérée comme variable de synthèse, mais cette matrice est fixée dans chaque itération et peut être obtenue en résolvant l'équation de Lyapunov suivant :

$$P - \hat{D}_k^T P \hat{D}_k = I \quad (3.54a)$$

$$\text{où} \quad \hat{D}_k = \begin{bmatrix} -a_k^T \\ \hat{I}_r \end{bmatrix} \quad (3.54b)$$

Il est bien connu que si le filtre RII avec le vecteur de coefficient x_k est stable, alors la solution de l'équation (3.54) est seule et définie positive [5].

Avec P fixe dans Y_k , le problème de minimisation dans (3.53) est un problème de SDP de la taille $1 + r + 0.5(n + 1)(n + 2)$.

Étapes de synthèse

Donner l'ordre du filtre RII 2-D (n, r) et la réponse fréquentielle désirée.

Étape 1 : La méthode de synthèse proposée commence par un point initial x_0 qui correspond à un filtre stable obtenue suivre une méthode conventionnelle. Par exemple, on peut prendre un filtre RIF d'ordre n approximer $H_d(w_1, w_2)$ et placer simplement $A(z_1) \equiv 1$ et $A(z_2) \equiv 1$ comme un point initiale.

Étape 2 : Avec ce x_0 , une matrice définie positive P peut être obtenue en résolvant l'équation de Lyapunov de (3.54), et les quantités G_k , q_k , et c_k peuvent être évaluées par l'utilisation (3.43) et (3.46).

Étape 3 : Ensuite le problème de SDP dans (3.53) est résolu.

Étape 4 : La solution $\hat{\delta}^* = [\mu^* \quad \delta^{*T}]^T$ obtenue peut être employée pour mettre à jour x_0 à $x_1 = x_0 + \delta^*$. L'itération continue jusqu'à $\|\delta^*\|$ est devenir inférieure à prescriptive tolérance ε .

3.6 Conclusion

Dans ce chapitre nous essayions de trouvée un méthode de synthèse de filtre numérique RII qui produit un filtre avec le minimum d'ordre et préserve la stabilité, avec une phase linéaire, dans le domaine spatial on peut diviser la réponse impulsionnelle désirer à 4 régions ensuite on peut synthétiser 4 filtres (un filtre dans le cas où il existe un symétrie quadruple) pour chaque région par la méthode de Prony ou Itérative, le résultat est une filtre à phase linéaire et peut être un filtre stable. Un filtre stable avec une phase approximativement linéaire est synthétisé dans le domaine fréquentiel par la méthode de programmation itérative semi-définie (SDP).

Chapitre 4

Réduction d'ordre

4.1 Introduction

Il est souvent désirable de représenter un système d'ordre élevé par un système d'ordre inférieur. Dans la plupart des cas, de tels modèles de d'ordre inférieure donnée l'exactitude raisonnable pour la réalisation, la commande, et/ou les objectives de calcul.

Une procédure de réduction appropriée devrait produire un modèle qui approximer l'original bien, une modèle stable à partir d'un original stable, et il devrait pouvoir être implémenter sur un ordinateur avec le minimum des éléments de stockage [2].

La réduction d'ordre dans l'environnement de l'espace d'état a quelques des avantages. De telles réalisations peuvent transférer facilement à la réalisation pratique, elles peuvent être facilement généralisées au cas d'un système invariant dans le temps. Il est possible d'appliquer la vaste connaissance de la théorie des matrices dans l'analyse, et avec le non unicité des réalisations de l'espace d'état fournissent le choix approprié pour utiliser.

Ce choix peut être régi par des erreurs de troncature, des problèmes avec la sensibilité des coefficients après la quantification, etc.

Malheureusement, dans le cas 2-D, la stabilité du modèle réduit n'est pas garantie, en général. Mais on démontre que la procédure de réduction peut avec succès être appliquée à système séparable dans le dénominateur [2].

4.1 Modèle d'espace d'état

Le modèle du l'espace d'état offre la grande utilité dans la formulation et l'analyse des systèmes linéaires. Des systèmes linéaires qui sont décrits par des fonctions de transfert, les équations aux différences sont facilement formulés dans une représentation du l'espace d'état. Une fois formulées, beaucoup de techniques connues peuvent être appliquées pour analyser systématiquement le modèle. En conséquence, le modèle du l'espace d'état est un général et un puissant outil qui est utilisé pour unifier la recherche et l'étude des systèmes linéaires [6].

Il y a plusieurs auteurs : Attasi [7], Fornasini-Marchesini [8] et Givone-Roesser [9] ont proposé différents modèles du l'espace d'état pour les systèmes 2-D et ont proposé quelques extensions des notions habituelles dans le cas 1-D tel que la commandabilité, et l'observabilité au cas 2-D. Cependant, ces résultats ne sont pas tout à fait satisfaisants, ils soient manquent de la motivation pour les modèles du l'espace d'état introduits ou soit la notion du l'espace d'état est incorrectement définie. Il apercevoir que le modèle de Givone-Roesser est le plus satisfaisant, il est également le plus général puisque les modèles d'Attasi et de Fornasini-Marchesini peuvent être inclus dans le modèle de Givone-Roesser [10].

Givone et Roesser ont utilisé une « approche de circuit » au problème de la réalisation de l'espace d'état pour les systèmes 2-D. Ils présentent un modèle dans lequel l'état local est divisé en un état horizontal et un état vertical qui sont propagé, respectivement, horizontalement et verticalement par des équations aux différences de premier ordre. Cependant, Givone et Roesser n'ont pas fourni beaucoup de motivation pour introduire d'un tel modèle puisqu'ils ne pouvaient pas implémenté une fonction de transfert 2-D arbitraire, $H(z_1, z_2) = B(z_1, z_2)/A(z_1, z_2)$, où on assume que $A(z_1, z_2)$ est irréductible, avec leur modèle du l'espace d'état. Mitra donné une réponse à ce problème dans [11] en présentant une méthode d'implémenté pour les fonctions de transfert 2-D avec quelques éléments de retards z_1^{-1} et z_2^{-1} .

Le modèle de Givone-Roesser est le suivant :

$$x(i, j) = \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} \quad (4.1a)$$

où x est l'état local, x^h est vecteur d'état horizontal de dimension n , x^v est vecteur d'état vertical de dimension m

$$\begin{bmatrix} x^h(i+1, j) \\ x^v(i, j+1) \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(i, j) \quad (4.1b)$$

$$y(i, j) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} + du(i, j), \quad (i, j) \geq (0, 0) \quad (4.1c)$$

où u vecteur d'entrée de dimension l , et y vecteur de sortie de dimension p . Clairement x^h est l'état horizontal est propagé horizontalement, x^v l'état vertical est propagé verticalement par des équations aux différences de premier ordre.

Maintenant, de la figure 4.1 suivant d'implémentation de circuit (avec $d = 0$)

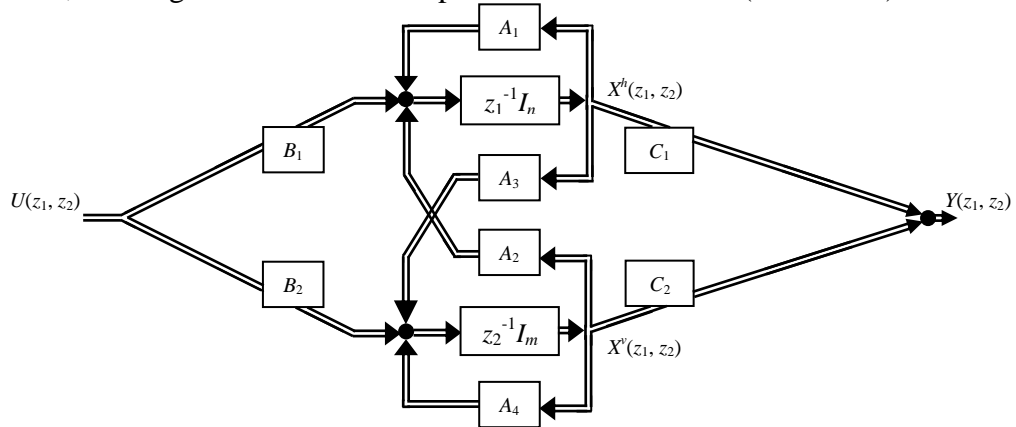


Figure 4.1 implémentation de circuit (avec $d = 0$).

on peut écrire directement l'équation suivante:

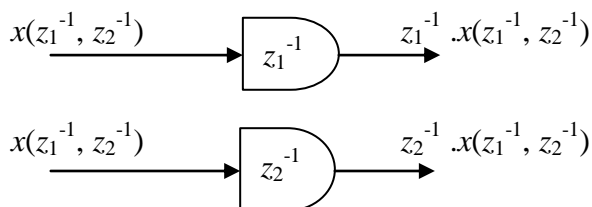
$$H(z_1, z_2) = C \left[\begin{pmatrix} z_1 I & 0 \\ 0 & z_2 I \end{pmatrix} - A \right]^{-1} B \quad (4.2)$$

Il est clair qu'il y ait une correspondance linéaire entre le modèle de Givone-Roesser et les réalisations de circuit avec les éléments de retard z_1^{-1} et z_2^{-1} .

Réalisation de Circuit

Dans le cas du temps discret 1-D, les éléments dynamiques utilisés sont des éléments de retards (de temps).

Pour la réalisation d'une 2-D fonction de transfert, une différence importante est que deux types d'éléments dynamiques sont nécessaires :



Ils se indiquent «l'élément de retards horizontal» par (z_1^{-1}) et «l'élément de retards vertical» par (z_2^{-1}) , respectivement.

Un problème important est celui de la façon employée les éléments dynamiques, multiplicateurs et additionneurs, pour réaliser un filtre numérique 2-D avec la fonction de transfert :

$$H(z_1^{-1}, z_2^{-1}) = \frac{B(z_1^{-1}, z_2^{-1})}{A(z_1^{-1}, z_2^{-1})} = \frac{\sum_{k_1=0}^n \sum_{k_2=0}^m b_{k_1 k_2} z_1^{-k_1} z_2^{-k_2}}{\sum_{k_1=0}^n \sum_{k_2=0}^m a_{k_1 k_2} z_1^{-k_1} z_2^{-k_2}} \quad (4.3)$$

Nous pouvons faire ceci dans deux étapes. D'abord nous récrivons (4.3) dans la représentation rationnel-gain, c.-à-d.

$$H(z_1^{-1}, z_2^{-1}) = \frac{\sum_{k_1=0}^n b_{k_1} (z_2^{-1}) z_1^{-k_1}}{\sum_{k_1=0}^n a_{k_1} (z_2^{-1}) z_1^{-k_1}} \quad (4.4)$$

Sans perte de généralité, nous pouvons supposer que $a_{00} = 1$, alors

$$a_0(z_2^{-1}) = 1 + \tilde{a}_0(z_2^{-1})$$

Ainsi, utilisation la technique de la réalisation 1-D, nous notons la réalisation montrée dans figure 4.2, où les gains des multiplicateurs sont représentés dans les polynômes $F[z_2^{-1}]$.

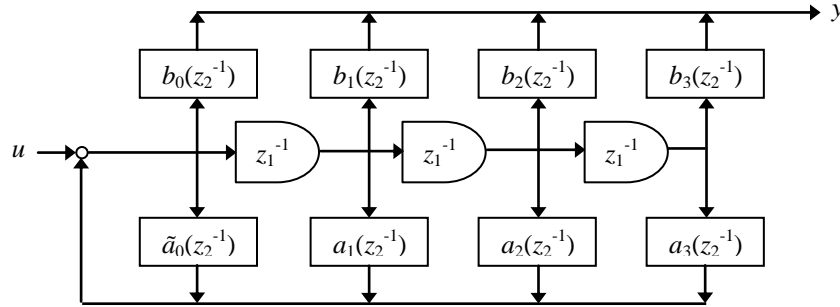


Figure 4.2 réalisation de la fonction transfert (4.4)

En plus de n éléments de retards horizontaux de figure 4.2, nous avons besoin m éléments de retards verticaux pour mettre en application les gains de rétroaction (feedback) $\{a_i(z_2^{-1}), i = 0, 1, \dots, m\}$ et d'autres m éléments de retards verticaux pour mettre en application les gains de lecture $\{b_i(z_2^{-1}), i = 0, 1, \dots, m\}$.

Ainsi, la réalisation complète exige $n + 2m$ éléments dynamiques.

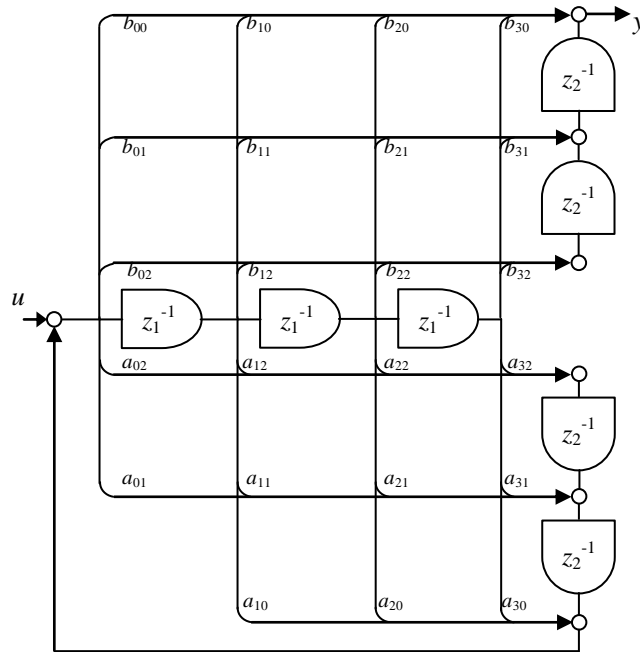


Figure 4.3 Réalisation de Circuit 2-D.

Cette réalisation est (canonique) standard, sa structure est très simple et elle implique seulement des gains réels.

4.1.1 Représentation du modèle dans l'espace d'état

Les réalisations de circuit avec les éléments de retards z_1^{-1} et z_2^{-1} sont dans une correspondance linéaire avec des modèles dans l'espace-état du type de Roesser. Les sorties des éléments de retards z_1^{-1} sont les états horizontaux et les sorties des éléments de retards z_2^{-1} sont les états verticaux.

Ainsi, l'implémentation de figure 4.3 peut être transformée simplement en modèle suivant :

$$\begin{matrix} n\{ \\ m\{ \\ m\{ \end{matrix} \begin{bmatrix} x^h(i+1, j) \\ x^{v1}(i, j+1) \\ x^{v2}(i, j+1) \end{bmatrix} = A \begin{bmatrix} x^h(i, j) \\ x^{v1}(i, j) \\ x^{v2}(i, j) \end{bmatrix} + Bu(i, j) \quad (4.5a)$$

$$y(i, j) = C \begin{bmatrix} x^h(i, j) \\ x^{v1}(i, j) \\ x^{v2}(i, j) \end{bmatrix} + du(i, j) \quad (4.5b)$$

où

$$B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \dots \\ a_{01} \\ \vdots \\ a_{0m} \\ \dots \\ b_{01} \\ \vdots \\ b_{0m} \end{bmatrix} \quad C = [\tilde{b}_{10} \quad \dots \quad \tilde{b}_{n0} \quad \vdots \quad -b_{00} \quad 0 \quad \dots \quad 0 \quad \vdots \quad 1 \quad 0 \quad \dots \quad 0], \quad d = b_{00}$$

et

$$A_{11} = \begin{bmatrix} -a_{10} & -a_{20} & \dots & -a_{n0} \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix}, \quad A_{12} = \begin{bmatrix} -1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix}, \quad A_{13} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} \tilde{a}_{11} & \dots & \tilde{a}_{n1} \\ \vdots & \tilde{a}_{ij} & \vdots \\ \tilde{a}_{1m} & \dots & \tilde{a}_{nm} \end{bmatrix}, \quad A_{22} = \begin{bmatrix} -a_{01} & 1 & 0 & \dots & 0 \\ -a_{02} & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 1 \\ -a_{0m} & 0 & \dots & \dots & 0 \end{bmatrix}, \quad A_{23} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

$$A_{31} = \begin{bmatrix} \tilde{b}_{11} & \dots & \tilde{b}_{n1} \\ \vdots & \tilde{b}_{ij} & \vdots \\ \tilde{b}_{1m} & \dots & \tilde{b}_{nm} \end{bmatrix}, \quad A_{32} = \begin{bmatrix} -b_{01} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -b_{0m} & 0 & \dots & 0 \end{bmatrix}, \quad A_{33} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ \vdots & \ddots & & & 0 \\ & & & & 1 \\ 0 & \dots & & & 0 \end{bmatrix}$$

avec

$$\tilde{a}_{ij} = a_{ij} - a_{i0}a_{0j}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq m$$

$$\tilde{b}_{ij} = b_{ij} - a_{i0}b_{0j}, \quad 1 \leq i \leq n, \quad 0 \leq j \leq m.$$

4.1.2 Réalisation Minimale dans l'espace d'état

Est un algorithme présenté pour la réalisation minimale dans l'espace d'état d'une fonction de transfert bidimensionnelle pour le cas où le dénominateur de la fonction de transfert est factorable (dénominateur séparable). La représentation dans l'espace d'état est directement dérivée par correspondance (*inspection*) d'une réalisation de schéma fonctionnel de circuit du système 2-D. L'algorithme n'exige pas que le polynôme de dénominateur soit factorisé.

La notion de la réalisation minimal du modèle d'espace d'état, joue un rôle important dans l'analyse et de la synthèse des systèmes 2-D, en raison du grand nombre de données impliquées dans le traitement numérique des signaux. La réalisation minimale de l'espace d'état ne sont pas toujours possibles à toutes les fonctions de transfert rationnel 2-D [10] [12]. Cependant, des réalisations minimales de l'espace d'état ont été déterminées pour les cas spéciaux suivants :

- a) Systèmes avec fonctions de transfert de la forme $H(z_1, z_2) = 1/A(z_1, z_2)$, ou de la forme $H(z_1, z_2) = B(z_1, z_2)$ [13].
- b) Systèmes avec le dénominateur séparable [14].

D'abord, on introduit un test de factorabilité qui permet nous de déterminer si le dénominateur peut être factorisé. Ensuite, on présente un algorithme basé sur une réalisation de schéma fonctionnel de circuit de la fonction de transfert et donne les matrices du modèle par correspondance (*inspection*).

Considérer le système linéaire et invariant dans le temps, décrit par la fonction de transfert suivant :

$$H(z_1^{-1}, z_2^{-1}) = \frac{\sum_{k_1=0}^n \sum_{k_2=0}^m b(k_1, k_2) z_1^{-k_1} z_2^{-k_2}}{\sum_{k_1=0}^n \sum_{k_2=0}^m a(k_1, k_2) z_1^{-k_1} z_2^{-k_2}} \quad (4.6)$$

Le problème considéré est de déterminer un modèle minimal de l'espace d'état du système (4.6) pour le cas suivant :

Les coefficients de numérateur de (4.6) peuvent être arbitraires, alors que les coefficients de dénominateur satisfont la relation suivant :

$$a(k_1, k_2) = a(k_1, 0)a(0, k_2) \quad (4.7)$$

avec $a(0, 0) = 1$.

Le modèle de l'espace d'état est du type de Givone-Roesser [9] décrit dans (4.1) où $l = p = 1$, et $A_1, A_2, A_3, A_4, B_1, B_2, C_1$ et C_2 ont la dimension $(n \times n)$, $(n \times m)$, $(m \times n)$, $(m \times m)$, $(n \times 1)$, $(m \times 1)$, $(n \times 1)$ et $(m \times 1)$, respectivement.

Réalisation dans l'espace d'état

La fonction de transfert (4.6), où $a(k_1, k_2) = a(k_1, 0)a(0, k_2)$ peut être présentée sous une formule de schéma fonctionnel comme dans la figure 4.4.

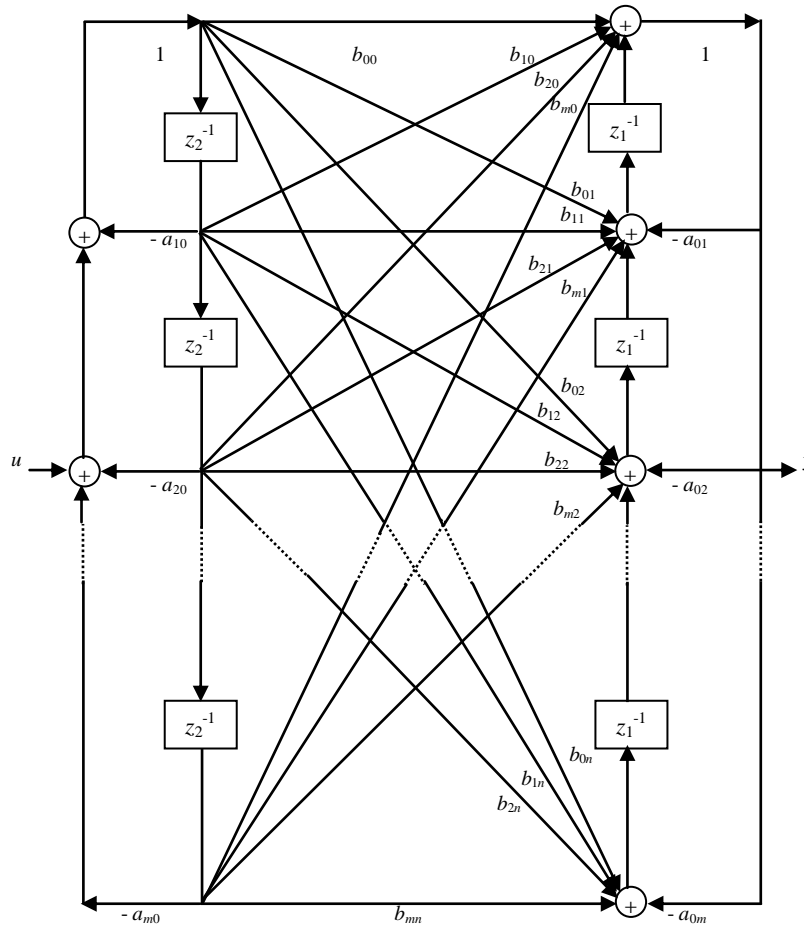


Figure 4.4 le schéma fonctionnel d'une fonction de transfert avec le dénominateur factorable.

Alors la réalisation minimale dans ce cas exige seulement $n + m$ éléments dynamiques.

Une réalisation de l'espace d'état peut être écrite par :

$$A_1 = \begin{bmatrix} -a(0,1) & \cdot & \cdots & \cdots & \cdot & -a(m,1) \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & \cdot \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \cdot & \cdots & 0 & 1 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}, A_3 = \begin{bmatrix} h(1,1) & \cdots & h(1,n) \\ \vdots & & \vdots \\ h(m,1) & \cdots & h(m,n) \end{bmatrix}$$

avec

$$\begin{aligned} h(1, 1) &= b(1, 1) + a(1, 0)a(0, 1)b(0, 0) - [b(1, 0)a(0, 1) + a(1, 0)b(1, 0)] \\ &\vdots \\ h(1, n) &= b(1, n) + a(1, 0)a(0, n)b(0, 0) - [b(1, 0)a(0, n) + a(1, 0)b(0, n)] \\ &\vdots \\ h(m, 1) &= b(m, 1) + a(m, 0)a(0, 1)b(0, 0) - [b(m, 0)a(0, 1) + a(m, 0)b(0, 1)] \\ &\vdots \\ h(m, n) &= b(m, n) + a(m, 0)a(0, n)b(0, 0) - [b(m, 0)a(0, n) + a(m, 0)b(0, n)] \end{aligned}$$

et

$$A_4 = \begin{bmatrix} -a(1,0) & 1 & 0 & \cdots & 0 \\ \cdot & 0 & \ddots & \ddots & \cdot \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \cdot & \cdot & & \ddots & 0 \\ \cdot & \cdot & & \ddots & 1 \\ -a(m,0) & 0 & \cdot & \cdots & 0 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} b(1,0) - a(1,0)b(0,0) \\ \vdots \\ \vdots \\ b(m,0) - a(m,0)b(0,0) \end{bmatrix}$$

$$C_1 = [b(0,1) - a(0,1)b(0,0), \quad \cdots, \quad b(0,n) - a(0,n)b(0,0)]$$

$$C_2 = [1 \quad 0, \quad \cdots, \quad 0]$$

et

$$d = b(0, 0).$$

4.3 Dérivation la matrice de la fonction de transfert à partir du modèle d'état

Deux étapes principales en développant un algorithme sont comme suit : D'abord, la matrice de fonction transfert est reformulée en termes de polynômes caractéristiques de plusieurs matrices qui dépendent d'une variable complexe. En second, un algorithme au lequel identifier les coefficients d'un polynôme 1-D d'ordre n quand ses valeurs $(n + 1)$ des points sur le cercle d'unité sont connus. L'algorithme nécessite de résoudre un système des équations linéaires dont la matrice est une matrice unitaire de Vandermonde [15].

Deux algorithmes pour la dérivation de la matrice de fonction transfert d'un système linéaire 2-D, invariant dans le temps, discret, de modèle du Givone-Roesser sont développés.

Considérer le modèle de l'espace d'état de Givone-Roesser à unique entrée, et à unique sortie (SISO) décrit comme de (4.1), et si nous définissons

$$I(z_1, z_2) = z_1 I \oplus z_2 I$$

où \oplus dénote la somme directe,

alors la matrice de fonction transfert du système est donnée par :

$$H(z_1, z_2) = [C_1 \quad C_2] \begin{bmatrix} z_1 I - A_1 & -A_2 \\ -A_3 & z_2 I - A_4 \end{bmatrix}^{-1} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} + d \quad (4.8)$$

$$= C[I(z_1, z_2) - A]^{-1} B + d$$

$H(z_1, z_2)$ de (4.8) peut être écrit par :

$$H(z_1, z_2) = \frac{\sum_{k=0}^m q_k(z_2) z_1^k}{\sum_{k=0}^m p_k(z_2) z_1^k} \quad (4.9)$$

où k est un nombre entier, $q_k(z_2)$ et $p_k(z_2)$ sont des polynômes de z_2 d'ordre non plus grand que n , et

$$\sum_{k=0}^m p_k(z_2) z_1^k = \det \begin{bmatrix} z_1 I - A_1 & -A_2 \\ -A_3 & z_2 I - A_4 \end{bmatrix}$$

avec

$$p_m(z_2) = \det(z_2 I - A_4) = P(z_2, A_4) \quad (4.10)$$

où $P(z_2, A_4)$ dénote le polynôme caractéristique de A_4 de variable z_2 . De plus, de (4.8) et la formule de l'inversion de matrice [16], la fonction de transfert $H(z_1, z_2)$ peut être exprimée par :

$$H(z_1, z_2) = l(z_2) + g(z_2)[z_1 I - E(z_2)]^{-1} f(z_2) \quad (4.11)$$

où

$$\begin{aligned} E(z_2) &= A_1 + A_2(z_2 I - A_4)^{-1} A_3 \\ f(z_2) &= B_1 + A_2(z_2 I - A_4)^{-1} B_2 \\ g(z_2) &= C_1 + C_2(z_2 I - A_4)^{-1} A_3 \\ l(z_2) &= d + C_2(z_2 I - A_4)^{-1} B_2 \end{aligned}$$

Noter que $(z_2 I - A_4)^{-1}$ est un facteur commun dans $E(z_2)$, $f(z_2)$, $g(z_2)$, et $l(z_2)$ par conséquent, les équations ci-dessus peut être exprimé par :

$$\begin{bmatrix} E(z_2) & f(z_2) \\ g(z_2) & l(z_2) \end{bmatrix} = \begin{bmatrix} A_1 & b_1 \\ c_1 & d \end{bmatrix} + \begin{bmatrix} A_2 \\ c_2 \end{bmatrix} (z_1 I - A_4)^{-1} [A_3 \quad b_2] \quad (4.12)$$

En utilisant une formule bien connue pour la fonction de transfert d'un modèle du l'espace d'état dans le cas 1-D (voir [16, annexe A.13]), (4.11) peut être récrit par :

$$\begin{aligned} H(z_1, z_2) &= \frac{\det[z_1 I - E(z_2) + f(z_2)g(z_2)]}{\det[z_1 I - E(z_2)]} + l(z_2) - 1 \\ &= \frac{P[z_1, E(z_2) - f(z_2)g(z_2)]}{P[z_1, E(z_2)]} + l(z_2) - 1 \end{aligned} \quad (4.13)$$

où $P[z_1, E(z_2)]$ et $P[z_1, E(z_2) - f(z_2)g(z_2)]$ sont des polynômes caractéristiques de $E(z_2)$ et $E(z_2) - f(z_2)g(z_2)$, respectivement. Le dénominateur dans (4.13) est un polynôme en z_1 mais le dénominateur dans (4.9) est un polynôme en z_1 avec $p_m(z_2)$ comme coefficient de z_1^m . Cette observation avec (4.10) donné :

$$\sum_{k=0}^m q_k(z_2) z_1^k = P(z_2, A_4) \{ P[z_1, E(z_2) - f(z_2)g(z_2)] + [l(z_2) - 1] P[z_1, E(z_2)] \} \quad (4.14a)$$

$$\sum_{k=0}^m p_k(z_2) z_1^k = P(z_2, A_4) P[z_1, E(z_2)] \quad (4.14b)$$

A. Algorithme pour un modèle SISO de Givone-Roesser

L'algorithme est dérivé de (4.14a) et (4.14b), et une méthode efficace pour la détermination d'un polynôme 1-D comme suivent:

1) Déterminer les coefficients d'un polynôme 1-D :

$$p(z_2) = \alpha_n z_2^n + \dots + \alpha_1 z_2 + \alpha_0$$

$p(z_2)$ est une polynôme d'ordre n avec des coefficients $\alpha_n, \dots, \alpha_1, \alpha_0$.

et $\{z_2(l), 0 \leq l \leq n\}$ soit les $(n+1)$ points qui sont uniformément distribués sur le cercle d'unité dans le plan complexe z_2 ,

$$z_2(l) = e^{j2\pi l/(n+1)}, \quad 0 \leq l \leq n \quad (4.15)$$

Si les valeurs $\{p_l = p[z_2(l)], 0 \leq l \leq n\}$ sont connues, alors les coefficients $\{\alpha_l, 0 \leq l \leq n\}$ peuvent être déterminés par :

$$\alpha = V^{-1}(z_2) q \quad (4.16)$$

où $\alpha = [\alpha_n, \dots, \alpha_1, \alpha_0]^T$, $q = [p_0, p_1, \dots, p_n]^T$, et $V(z_2)$ est la matrice de Vandermonde de dimension $(n+1) \times (n+1)$ dont le second à la dernière colonne est :

$$z_2 = [z_2(0) \quad z_2(1) \quad \dots \quad z_2(n)]^T$$

c'est

$$V(z_2) = \begin{bmatrix} z_2(0)^n & \cdots & z_2(0) & 1 \\ \vdots & & & \\ z_2(n)^n & \cdots & z_2(n) & 1 \end{bmatrix}$$

Depuis $z_2(l)$ ($0 \leq l \leq n$) sont différents, $V(z_2)$ est toujours une matrice non-singuliers. Et on a

$$V^H(z_2)V(z_2) = (n+1)I \quad (4.17)$$

où $V^H(z_2)$ dénote le conjugué et transposé de $V(z_2)$. Par conséquent, $V(z_2)/\sqrt{n+1}$ est une matrice unitaire et (4.16) peut être donner par :

$$\alpha = \frac{1}{n+1} V^H(z_2)q \quad (4.18)$$

L'équation (4.18) donner une formule efficace pour la détermination de polynôme $p(z_2)$.

2) Détermination les coefficients du $p_k(z_2)$ and $q_k(z_2)$:

Dans la section précédente on le suppose que la matrice A_4 n'a aucune valeur propre sur le cercle d'unité, qui est le cas où le système est stable [17]. Le cas où A_4 a des valeurs propres sur le cercle d'unité sera considéré dans la section B. Donnée un point z_2 sur le cercle d'unité, il suit de (4.12) que $E(z_2)$, $f(z_2)$, $g(z_2)$, et $l(z_2)$ peuvent être évalués de (4.14) pour obtenir les valeurs du $p_k(z_2)$ et $q_k(z_2)$ pour $0 \leq k \leq m$ au point donné z_2 .

Si cette procédure est appliquée à chacun du $n+1$ points définis par (4.15), alors les valeurs de chaque $p_k(z_2)$ et $q_k(z_2)$ sur $\{z_2(l), 0 \leq l \leq n\}$ peut être obtenu par l'algorithme suivant :

Algorithme:

Etape 1 : Utilisation (4.12) pour évaluer $E(z_2)$, $f(z_2)$, $g(z_2)$, et $l(z_2)$ dans l'ensemble de points définis par (4.15).

Etape 2 : Calculer le déterminante de $(z_2I - A_4)$ et les équations caractéristiques $E(z_2)$, et $E(z_2) - f(z_2)g(z_2)$ pour $z_2 = z_2(l)$, $0 \leq l \leq n$.

Etape 3 : Utiliser (4.14) pour obtenir $p_k(z_2)$ et $q_k(z_2)$ pour $0 \leq l \leq n$, $0 \leq k \leq m$.

Etape 4 : Pour chaque k ($0 \leq k \leq m$), forment des vecteurs $p = [p_0, p_1, \dots, p_n]^T$ et déterminent les polynômes $p_k(z_2)$ et $q_k(z_2)$ par l'utilisation (4.18).

B. Le cas du système est instable

Si A_4 a des valeurs propres avec du module égale l'unité, le système est instable. Dans ce cas, le $n+1$ points définis par (4.15) peut modifier par

$$z_2(l) = re^{j2\pi l/(n+1)}, \quad 0 \leq l \leq n \quad (4.19)$$

où $r > 0$ dénote le rayon d'un cercle dans le plan de z_2 où A_4 n'a aucune valeur propre.

Avec $q = [p_0, p_1, \dots, p_n]^T$, (4.19) devient

$$\alpha = V_r^{-1}(z_2)q$$

où

$$\begin{aligned} V_r(z_2) &= \begin{bmatrix} r^n z_2(0)^n & \cdots & r z_2(0) & 1 \\ \vdots & & \vdots & \vdots \\ r^n z_2(n)^n & \cdots & r z_2(n) & 1 \end{bmatrix} \\ &= V(z_2) \text{diag}\{r^n, \dots, r, 1\} \end{aligned}$$

et $\text{diag}\{r^n, \dots, r, 1\}$ est la matrice diagonale avec $r^n, \dots, r, 1$ comme les éléments de sa diagonale principale.

par (4.17),

$$V_r(z_2)^H V_r(z_2) = (n+1) \text{diag}\{r^{2n}, \dots, r^2, 1\}$$

ce qui implique

$$V_r^{-1}(z_2) = \frac{1}{n+1} \text{diag}\{r^{-2n}, \dots, r^{-2}, 1\} V_r^H(z_2)$$

Par conséquent, (4.18) est modifié à

$$\begin{aligned} \alpha &= \frac{1}{n+1} \text{diag}\{r^{-2n}, \dots, r^{-2}, 1\} V_r^H(z_2) q \\ &= \frac{1}{n+1} \text{diag}\{r^{-n}, \dots, r^{-1}, 1\} V^H(z_2) q \end{aligned} \quad (4.20)$$

Noter que (4.18) est un cas spécial de (4.20) avec $r=1$.

4.4 Méthodes de Réduction

Deux problèmes importants dans l'évaluation de méthode de réduction sont la précision d'approximation et la stabilité du système réduit. Pour les systèmes unidimensionnels 1-D, des méthodes qui rapportent l'excellente précision d'approximation avec la stabilité garantie ont été établies dans [18] [19]. Les techniques de réduction modèles les plus populaires sont basées sur le concept de la réalisation équilibrée qui a été initialement proposée par Moore [18]. Donné un système discret, sa réalisation équilibrée dans une représentation dans l'espace d'état dans laquelle l'importance de la variable d'état $i^{\text{ème}}$ peut être mesurée par la valeur singulière de Hankel. Ceci propose d'une manière que d'obtenir une approximation d'ordre réduit d'un modèle d'espace d'état soit de former une réalisation équilibrée et puis de maintenir les r états correspondant aux plus grandes valeurs singulières de Hankel où r est l'ordre du système réduit.

Pour les systèmes 2-D, le problème de réduction a été étudié par beaucoup de chercheurs pendant ces dernières années, et les résultats de la recherche obtenus ont été trouvés utiles dans, par exemple, la synthèse de filtre numérique [20]. Un des problèmes dans l'étude de réduction de modèle est d'étendre le concept de réalisation équilibré au cas 2-D. Puisqu'une réalisation équilibrée est essentiellement déterminée par les gramians de contrôlabilité et d'observabilité du système, et puisqu'il y a plusieurs types de Gramians qui peuvent être définis pour un système 2-D donné, il y a différents types de réalisations équilibrées pour un système discret donné, conduire à différentes approximations équilibrées [21].

Examen du cas d'un système 1-D

Commençons par un bref examen du concept de Gramians pour un système 1-D représenté par :

$$x_{k+1} = Ax_k + Bu_k \quad (4.21a)$$

$$y_k = Cx_k + du_k \quad (4.21b)$$

On suppose que le système est contrôlable, observable, et stable. Dans ces suppositions la réalisation (4.21) est dite équilibrée si

$$P = Q = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_n\} \quad (4.22)$$

où

$$P = \sum_{k=0}^{\infty} A^k B B^T (A^T)^k, Q = \sum_{k=0}^{\infty} A^k C^T C A^k \quad (4.23)$$

s'appellent la contrôlabilité et l'observabilité Gramians du système, respectivement. Il y a deux avantages associés à ces Gramians. Premièrement, pour une réalisation équilibrée, les Gramians définie par (4.22) donnent un ensemble des valeurs singulières de Hankel monotone décroissante $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ qui indiquent la signification des variables d'état associées, conduisant à la méthode bien connue de la réduction équilibrée de modèle [18]. En second, si la représentation de l'espace d'état donnée par (4.21) n'est pas encore équilibrée, alors les Gramians P et Q définis par (4.23) peuvent être employés pour trouver une transformation T tels que la réalisation caractérisée par $(T^{-1}AT, T^{-1}B, CT, d)$ est équilibrée. Après avoir obtenu une réalisation équilibrée, un système réduit d'ordre r peut être obtenu par troncature comme (A_r, B_r, C_r, d) où $A_r = A(1:r, 1:r)$, $B_r = B(1:r)$, et $C_r = C(1:r)$. Deux caractéristiques importantes des méthodes de réduction de modèles équilibrés sont que le système réduit reste stable si le système initial est stable, et que l'erreur d'approximation a une limite supérieure, qui est déterminée par les dernières $n - r$ valeurs singulières de Hankel, et est donnée par :

$$\|H(z) - H_r(z)\|_\infty = \max_{|z|=1} |H(z) - H_r(z)| \leq \sum_{k=r+1}^n \sigma_k \quad (4.24)$$

où $H(z) = C(zI - A)^{-1}B + d$ et $H_r(z) = C_r(zI - A)^{-1}B_r + d$ sont les fonctions de transfert du système initial et du système réduit, respectivement. En plus de (4.23), il y a deux autres méthodes pour évaluer P et Q : une implique les intégrales complexes

$$P = \frac{1}{2\pi j} \oint_{|z|=1} F(z)F^*(z) \frac{dz}{z}, Q = \frac{1}{2\pi j} \oint_{|z|=1} G^*(z)G(z) \frac{dz}{z} \quad (4.25)$$

où

$$F(z) = (zI - A)^{-1}B \quad (4.26a)$$

$$G(z) = C(zI - A)^{-1} \quad (4.26b)$$

et l'autre utilise les équations de Lyapunov :

$$APA^T - P = -BB^T \quad (4.27a)$$

$$A^TQA - Q = -C^TC \quad (4.27b)$$

Parmi les trois méthodes d'évaluation (4.23), (4.25) et (4.27), la méthode d'équations de Lyapunov demande la quantité minimum de calculs.

4.4.1 Trois types de Gramians pour les systèmes discrets 2-D

Considérons le modèle d'espace d'état de Givone-Roesser (SISO) décrit dans (4.1), et on définit

$$F(z_1, z_2) = (I(z_1, z_2) - A)^{-1}B \quad (4.28a)$$

$$G(z_1, z_2) = C(I(z_1, z_2) - A)^{-1} \quad (4.28b)$$

où $I(z_1, z_2) = z_1I_n \oplus z_2I_m$.

Le premier type de Gramians, connu sous le nom de pseudo contrôlabilité et observabilité Gramians [22], sont obtenus par la généralisation (4.25) et sont définis par :

$$P^p = \frac{1}{(2\pi j)^2} \oint_{|z_1|=1} \oint_{|z_2|=1} F(z_1, z_2)F^*(z_1, z_2) \frac{dz_2}{z_2} \frac{dz_1}{z_1}$$

$$Q^p = \frac{1}{(2\pi j)^2} \oint_{|z_1|=1} \oint_{|z_2|=1} G^*(z_1, z_2)G(z_1, z_2) \frac{dz_2}{z_2} \frac{dz_1}{z_1}$$

Le deuxième type de Gramians, connu sous le nom de contrôlabilité et observabilité Gramians structurées [23], sont définis par les solutions positivement définies P^s et Q^s de inégalités de Lyapunov

$$AP^sA^T - P^s + BB^T < 0 \quad (4.29a)$$

$$A^TQ^sA - Q^s + C^TC < 0 \quad (4.29b)$$

Noter que les matrices positivement définies (*positive definite matrices*) P^s et Q^s , si elles existent, ne sont pas uniques. Ceci conduit à la non-unicité des réalisations équilibre structurel qui sont basées sur P^s et Q^s , et donc on pose la question comment trouver une réalisation équilibre structurel qui rapporte le meilleur système réduit selon certains critères de performance [12].

Le troisième type de Gramians, connu sous le nom de quasi-Gramians [24], sont définis par les matrices diagonales positivement définies $P^q = \text{diag}\{P_1, P_2\}$ et $Q^q = \text{diag}\{Q_1, Q_2\}$ où P_i et Q_i ($i = 1, 2$) satisfont les équations de Lyapunov

$$A_1P_1A_1^T - P_1 + B_1B_1^T + A_2P_2A_2^T = 0 \quad (4.30a)$$

$$A_4P_2A_4^T - P_2 + B_2B_2^T + A_3^TP_1A_3 = 0 \quad (4.30b)$$

$$A_1^TQ_1A_1 - Q_1 + C_1^TC_1 + A_3^TQ_2A_3 = 0 \quad (4.30c)$$

$$A_4^TQ_2A_4 - Q_2 + C_2^TC_2 + A_2^TQ_1A_2 = 0 \quad (4.30d)$$

En termes de complexité de calcul, les Gramians structurés sont le plus cher d'évaluer tandis que les quasi-Gramians sont les plus économiques.

Approximations équilibrées

Comme dans le cas 1-D, des réalisations équilibrées d'un système discret 2-D sont obtenues en s'appliquant des transformations appropriées aux variables d'état, et ces transformations sont déterminées en utilisant la contrôlabilité et l'observabilité Gramians. Puisqu'il y a au moins trois types de Gramians, on peut en conséquence dériver trois types différents de réalisations équilibrées. En effet, une fois qu'un type de Gramians est choisi, les cases diagonales gauches supérieures et droites inférieures du Gramians sont employées pour calculer la matrice de transformation $T = T_1 \oplus T_2$ en utilisant, par exemple, l'algorithme de Laub [25] tels que la réalisation caractérisée par $\{T^1AT, T^1B, BT, d\}$ est équilibrée. Une fois qu'une réalisation équilibrée est obtenue, un système réduit d'ordre (r_1, r_2) , dénoté par $\{A_r, B_r, C_r, d\}$, peut être obtenu en tronquant les matrices A , B , et C avec

$$A_r = \begin{bmatrix} A_{1r} & A_{2r} \\ A_{3r} & A_{4r} \end{bmatrix}, \quad B_r = \begin{bmatrix} B_{1r} \\ B_{2r} \end{bmatrix}, \quad C_r = [C_{1r} \quad C_{2r}]$$

avec

$$A_{1r} = A_1(1:r_1, 1:r_1), \quad A_{2r} = A_2(1:r_1, 1:r_2),$$

$$A_{3r} = A_3(1:r_2, 1:r_1), \quad A_{4r} = A_4(1:r_2, 1:r_2),$$

$$B_{1r} = B_1(1:r_1), \quad B_{2r} = B_2(1:r_2)$$

$$C_{1r} = C_1(1:r_1), \quad C_{2r} = C_2(1:r_2)$$

Limites d'Erreur et Stabilité des Systèmes Réduits

1. cas de pseudo équilibre

Un système d'ordre réduit obtenu en utilisant l'approximation pseudo équilibrée préservé la stabilité si le système initial à dénominateur séparable [2]. Pour le système avec le dénominateur non séparable, bien que la stabilité du système réduit peut être préservée dans beaucoup de circonstances. Excepté quelques limites supérieures pour l'erreur d'approximation pour des types particuliers des systèmes 2-D séparables, aucune limite d'erreur n'est disponible pour la classe de système discret général quand la méthode pseudo équilibrée de réduction est utilisée.

2. cas de d'équilibre structurel

Bien que l'approximation équilibre structurel exige les calculs les plus intensifs, le système réduit obtenu préserve toujours la stabilité. D'ailleurs, il peut donner

$$\|H(z_1, z_2) - H_r(z_1, z_2)\|_\infty \leq 2 \left(\sum_{i=r_1+1}^{n_1} \sigma_{1i} + \sum_{i=r_2+1}^{n_2} \sigma_{2i} \right) \quad (4.31)$$

où $H(z_1, z_2)$ et $H_r(z_1, z_2)$ sont les fonctions de transfert des systèmes d'original et réduits, respectivement, et $\{\sigma_{1i}, i = r_1 + 1, \dots, n_1\}$ et $\{\sigma_{2i}, i = r_2 + 1, \dots, n_2\}$ sont le dernier $n_i - r_i$ ($i = 1, 2$) valeurs singulières de Hankel s'est associé à la dimension i [11]. Les valeurs singulières de Hankel apparaissant du côté droit de (4.31) ne sont pas uniques.

3. cas de quasi-équilibrées

Bien que la méthode quasi-équilibrée d'approximation ne préserve pas la stabilité du système réduit généralement, le système réduit reste stable si le système initial a un dénominateur séparable. Pour des systèmes avec des dénominateurs séparables, donner également plusieurs limites d'erreur pour l'erreur d'approximation [24].

On s'intéresse dans la prochaine section à le dernier cas (quasi-équilibrées) en raison de sa simplicité pour la réduction d'ordre, et on appliquer cette méthode pour les deux types de filtres RII, filtre à dénominateur séparable, et filtre à dénominateur ne pas séparable, et pour ça on proposer l'algorithme suivant :

4.5 Algorithme Itérative pour le cas Quasi-Gramians

Une méthode itérative pour le calcul des 2-D quasi-gramians est développée, où chaque itération impliquent de résoudre deux équations 1-D de Lyapunov. Pour système 2-D stable, l'algorithme itérative converge à la solution désirée très rapidement [26].

Algorithme

Étape 1 : on pose $P_2^{(0)} = Q_2^{(0)} = 0$ et $k = 1$.

Étape 2 : Résoudre les équations suivantes de Lyapunov pour $P_1^{(k)}$ et $Q_1^{(k)}$:

$$A_1 P_1^{(k)} A_1^T - P_1^{(k)} + F_1 = 0 \quad (4.32a)$$

$$A_1^T Q_1^{(k)} A_1 - Q_1^{(k)} + G_1 = 0 \quad (4.32c)$$

où

$$F_1 = B_1 B_1^T + A_2 P_2^{(k-1)} A_2^T$$

$$G_1 = C_1^T C_1 + A_3^T Q_2^{(k-1)} A_3$$

Étape 3 : Résoudre les équations suivantes de Lyapunov pour $P_2^{(k)}$ et $Q_2^{(k)}$:

$$A_4 P_2^{(k)} A_4^T - P_2^{(k)} + F_2 = 0 \quad (4.32b)$$

$$A_4^T Q_2^{(k)} A_4 - Q_2^{(k)} + G_2 = 0 \quad (4.32d)$$

où

$$F_2 = B_2 B_2^T + A_3 P_1^{(k)} A_3^T$$

$$G_2 = C_2^T C_2 + A_2^T Q_1^{(k)} A_2$$

Étape 4 : on pose $k = k + 1$ et répéter l'étape 2 et 3 jusqu'à

$$\|P_i^{(k)} - P_i^{(k-1)}\| < \varepsilon, \quad (i = 1, 2)$$

$$\|Q_i^{(k)} - Q_i^{(k-1)}\| < \varepsilon, \quad (i = 1, 2)$$

où le ε est une tolérance prescriptif.

Évidemment, si $P_i^{(k)} \rightarrow P_i$ et $Q_i^{(k)} \rightarrow Q_i$ pour $i = 1, 2$ comme $k \rightarrow \infty$, puis P_i et Q_i satisfont (4.30), donc, ils sont les quasi-gramians du filtre 2-D.

4.6 Conclusion

Les diversités du gramians sont venir a partir la méthode de calcule des matrices P et Q , des intégrales complexes de deux variables produit les matrices de gramians pseudo contrôlable et observable, la solution des inégalités de Lyapunov conduira à des gramians de contrôlabilité et observabilité structurées, et les quasi-gramians obtenue par les équations de Lyapunov, mais en termes de complexité de calcul, le gramians structuré sont le plus cher d'évaluer tandis que les quasi-Gramians sont les plus économiques.

Chapitre 5

Simulation et interprétation

5.1 Introduction

L'objectif de ce chapitre est de faire des simulations pour la synthèse et la réduction d'ordre. Pour cela, on devise les simulations en deux parties :

1^{ère} partie : la synthèse

On fait la synthèse dans les deux domaines :

Domaine temporel (spatial): par les deux méthodes : méthode de Prony et la méthode Itérative on génère les matrices du numérateur $b(n_1, n_2)$, et du dénominateur $a(n_1, n_2)$; ensuite, on utilise la fonction `Impulse_2D.m` pour produire la réponse impulsionnelle, et la réponse fréquentielle pour les différents types de filtres (passe-bas, passe-haut, passe-bande, et coupe-bande).

Domaine fréquentiel: par la méthode de programmation semi-définie (SDP), on fait le même travail pour les 4 différents types de filtres.

2^{ème} partie : la réduction

Pour la réduction d'ordre, on applique la méthode de quasi-gramians au filtre synthétisé en 1^{ère} partie .

- Les simulations seront suivies de leurs interprétations.

5.2.1 Synthèse : Méthode de Prony (Prony Modifiée)

Soit les filtres numériques RII donnés par les gabarits suivants :

- l'ordre de filtre $n=5$, $m=5$, (dimension de matrice a , et b).
- les régions normalisée de la bande passante et de la bande coupée correspondant pour chaque type de filtre sont :

Passe-bas : $Rp = [0 \ 0.4]$, et $Rs = [0.5 \ 1]$.

Passe-haut : $Rp = [0.6 \ 1]$, et $Rs = [0 \ 0.5]$.

Passe-bande : $Rp = [0.4 \ 0.6]$, $Rs_1 = [0 \ 0.3]$ et $Rs_2 = [0.7 \ 1]$.

Coupe-bande : $Rp = [0.3 \ 0.6]$, $Rs_1 = [0 \ 0.2]$ et $Rs_2 = [0.7 \ 1]$.

Remarque:

- le nombre totale de coefficients est égale à $n \times m \times (2 \text{ matrices } a, \text{ et } b) \times (4 \text{ filtres}) = 200$.

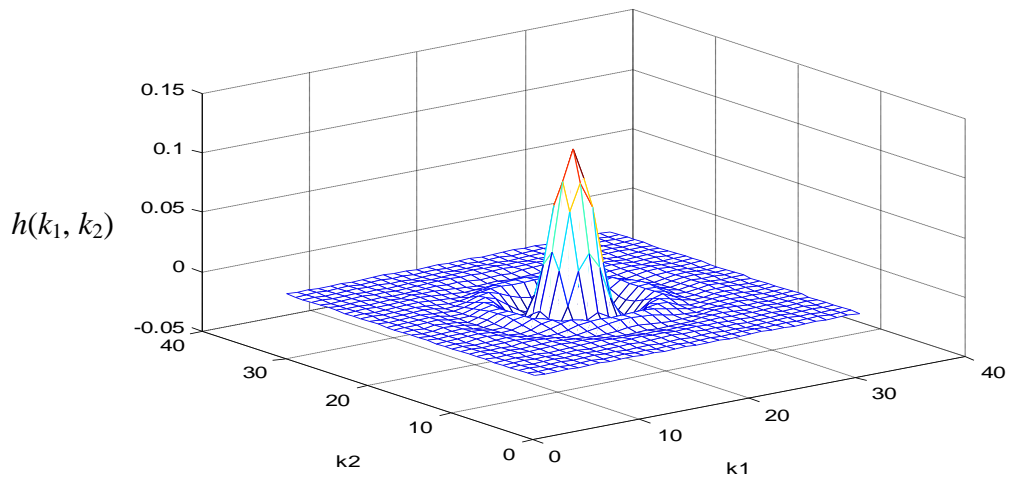


Figure (5.1) : réponse impulsionnelle du filtre RII passe-bas, méthode de Prony Modifiée.

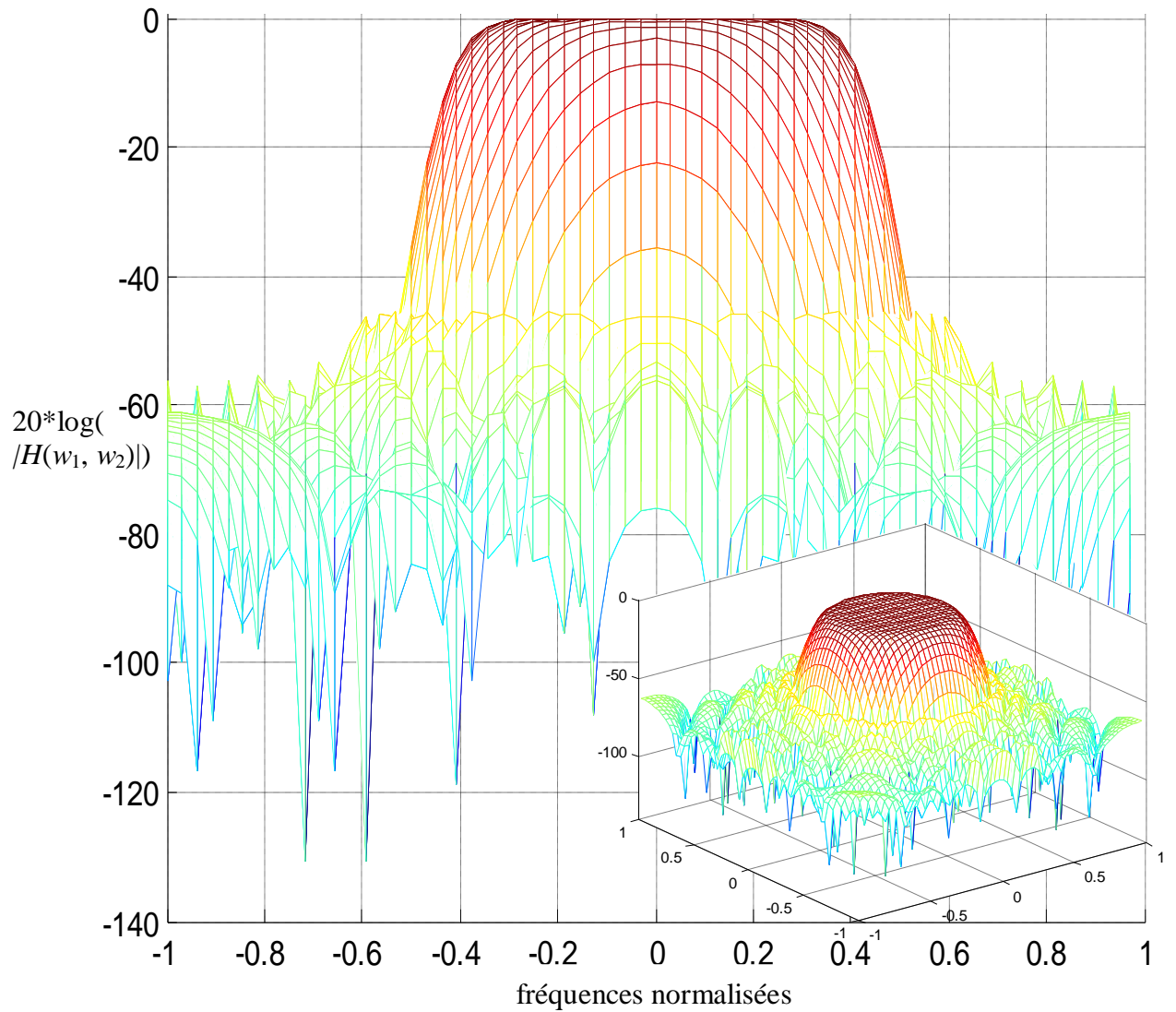


Figure (5.2) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-bas.

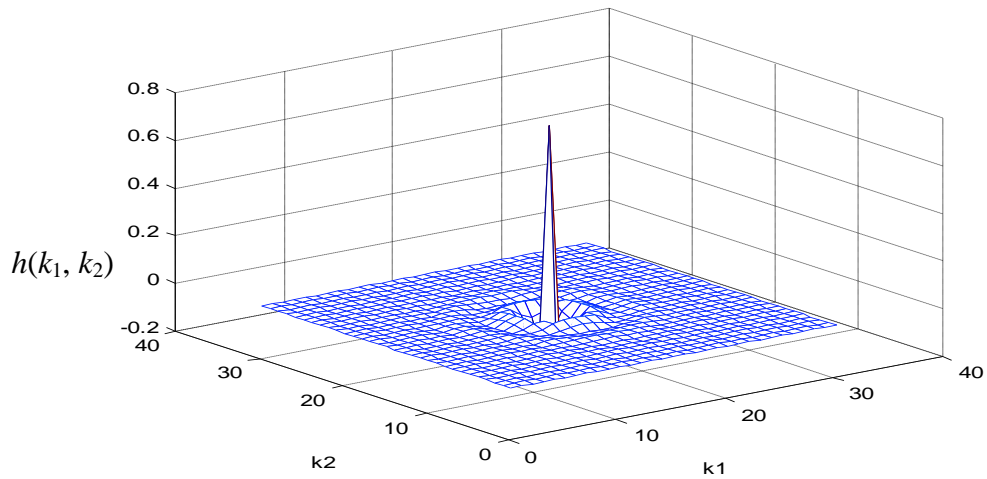


Figure (5.3) : réponse impulsionnelle du filtre RII passe-haut, méthode de Prony Modifiée.

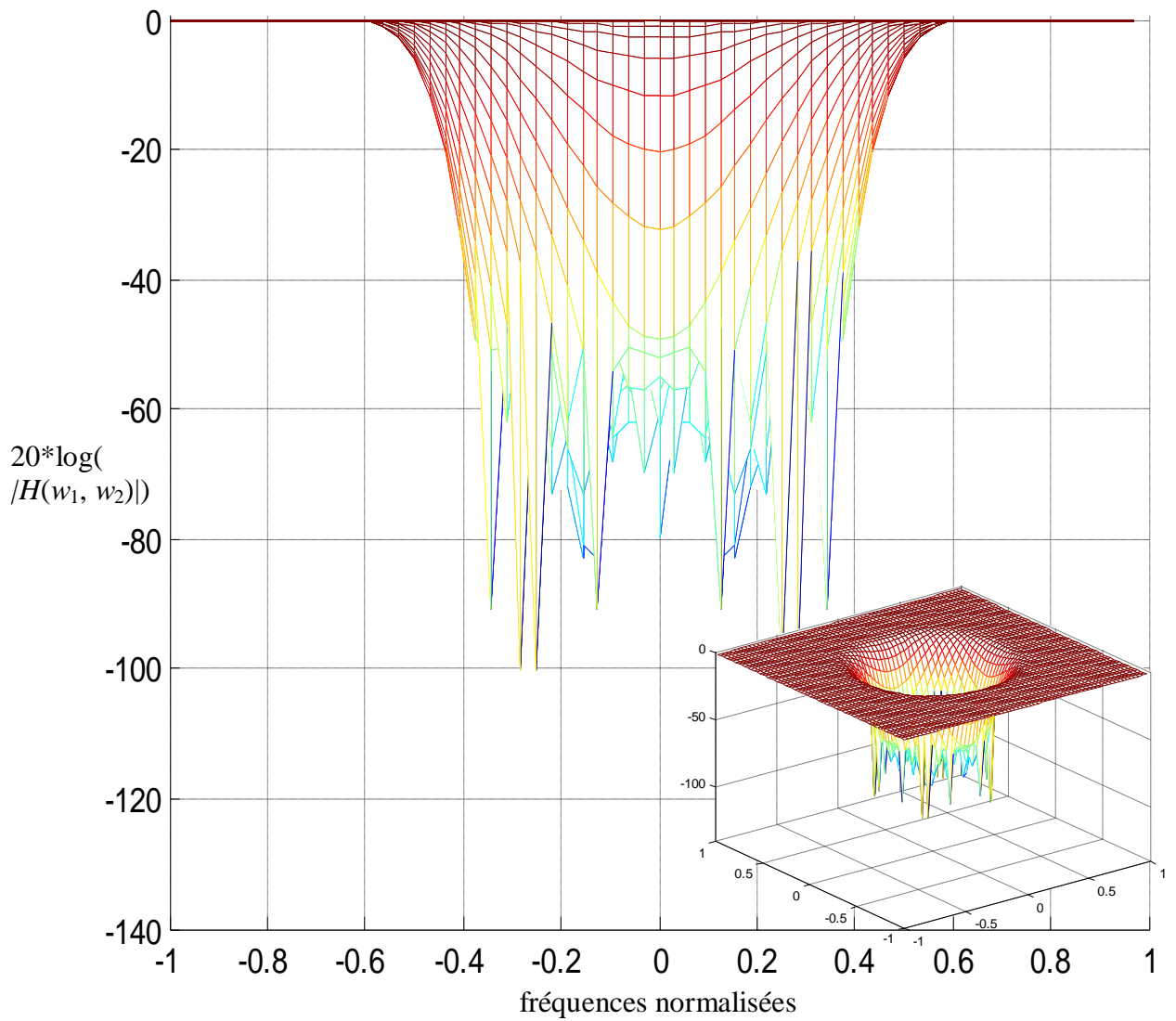


Figure (5.4) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-haut.

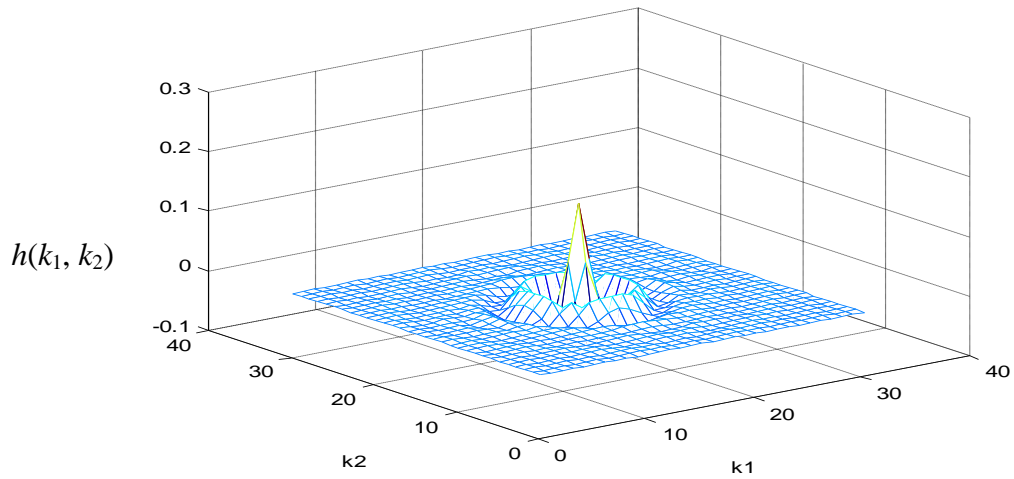


Figure (5.5) : réponse impulsionnelle du filtre RII passe-bande, méthode de Prony Modifiée.

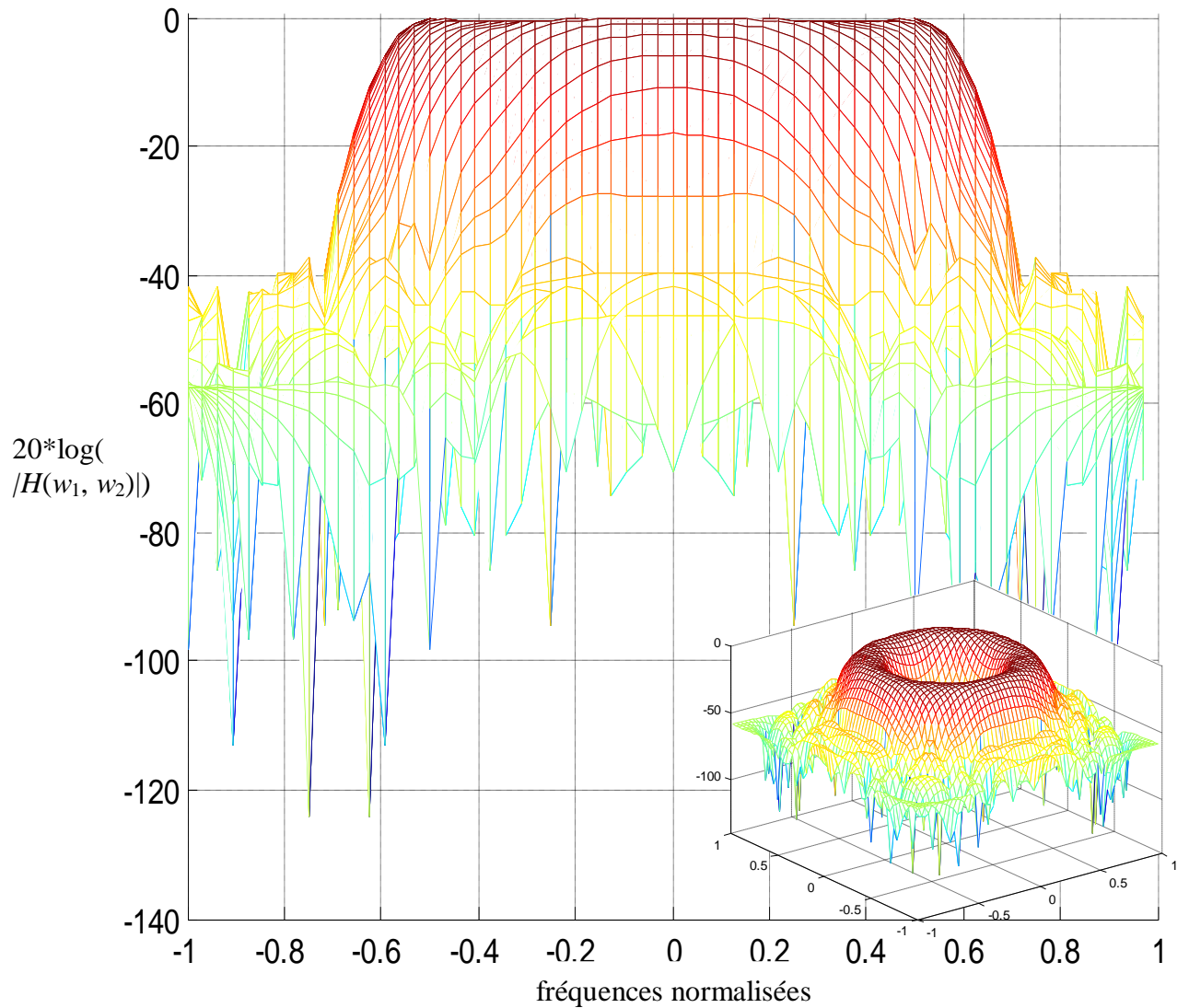


Figure (5.6) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-bande.

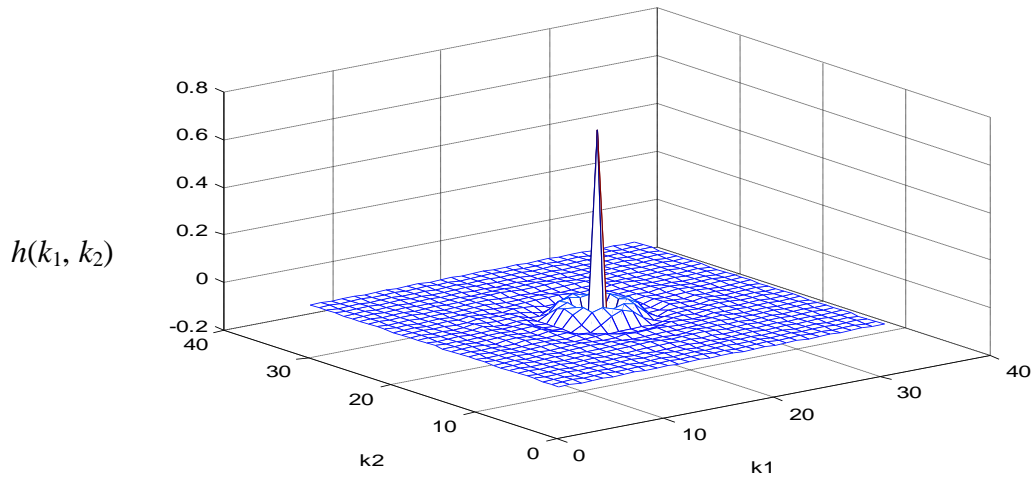


Figure (5.7) : réponse impulsionnelle du filtre RII coupe-bande, méthode de Prony Modifiée.

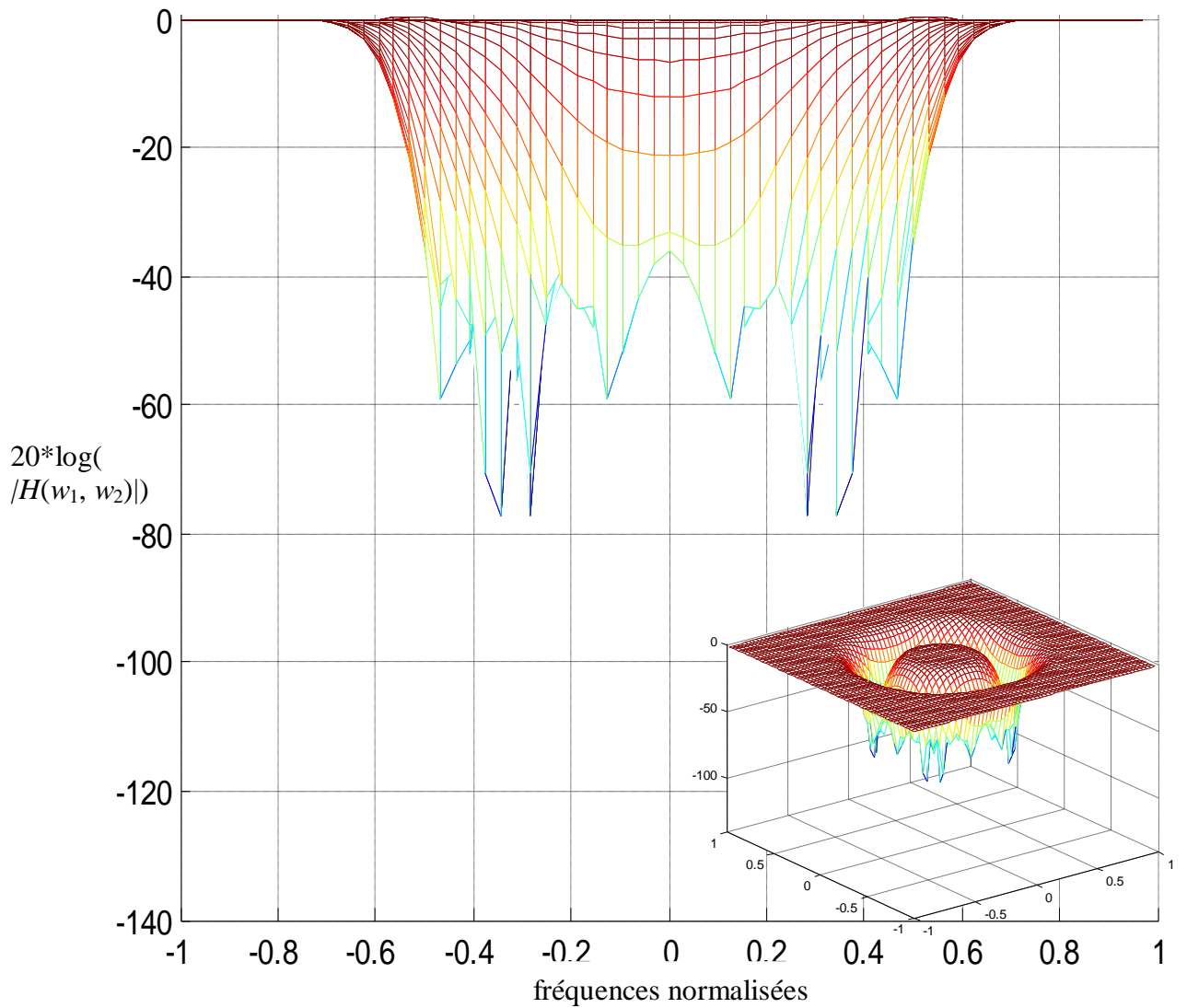


Figure (5.8) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII coupe-bande.

5.1.2 Synthèse : Méthode Itérative

Soit les filtres numériques RII donnés par les gabarits suivants :

- l'ordre de filtre $n=5$, $m=5$, (dimension de matrice a , et b).
- les régions normalisées de la bande passante et de la bande coupée correspondant pour chaque type de filtre égales :

Passe-bas : $R_p = [0 \ 0.4]$, et $R_s = [0.5 \ 1]$.

Passe-haut : $R_p = [0.6 \ 1]$, et $R_s = [0 \ 0.5]$.

Passe-bande : $R_p = [0.4 \ 0.6]$, $R_{s1} = [0 \ 0.3]$ et $R_{s2} = [0.7 \ 1]$.

Coupe-bande : $R_p = [0.3 \ 0.6]$, $R_{s1} = [0 \ 0.2]$ et $R_{s2} = [0.7 \ 1]$.

- nombre de itérations égale $it=10$.

Remarque:

- le nombre totale de coefficients est égale à $n \times m \times (2 \text{ matrices } a, \text{ et } b) \times (4 \text{ filtres}) = 200$.

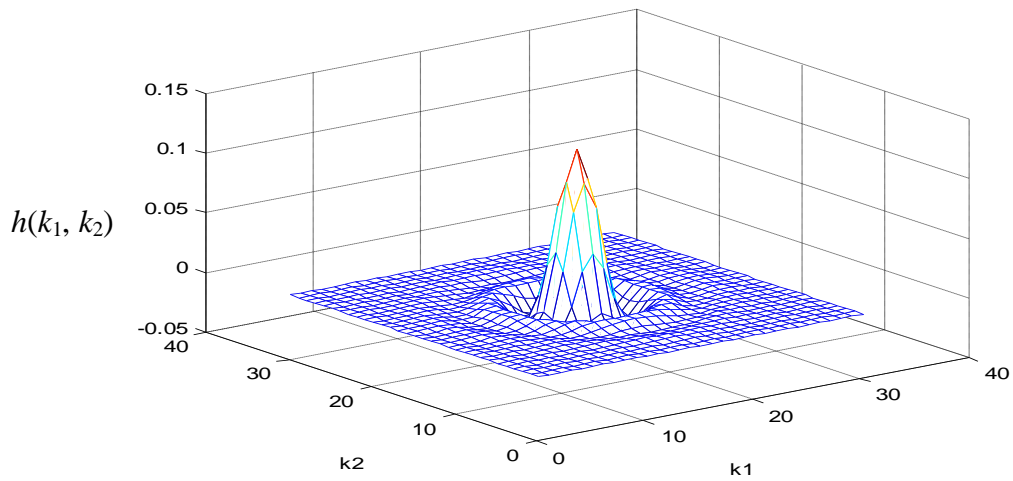


Figure (5.9) : réponse impulsionnelle du filtre RII passe-bas, méthode Itérative.

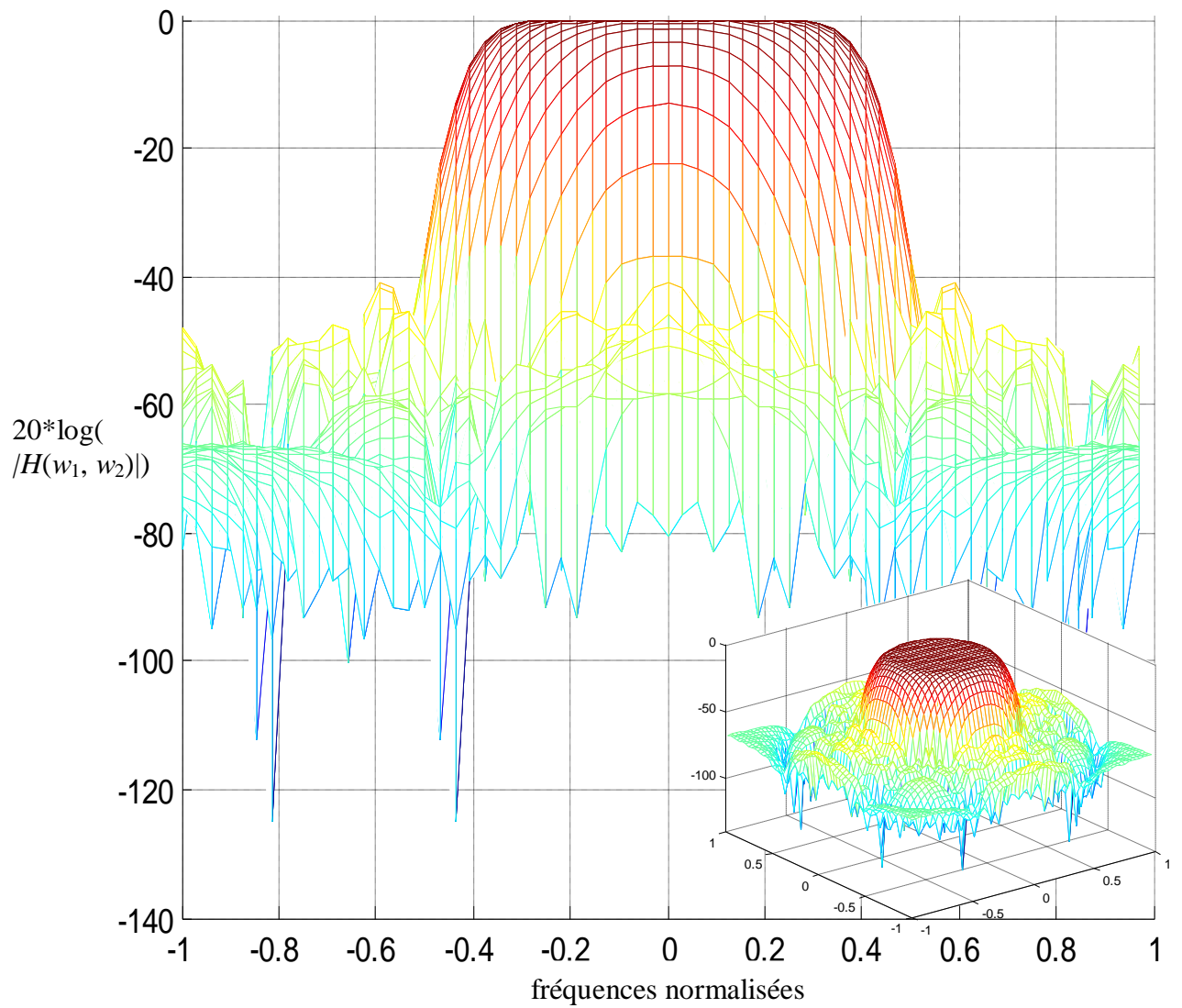


Figure (5.10) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-bas.

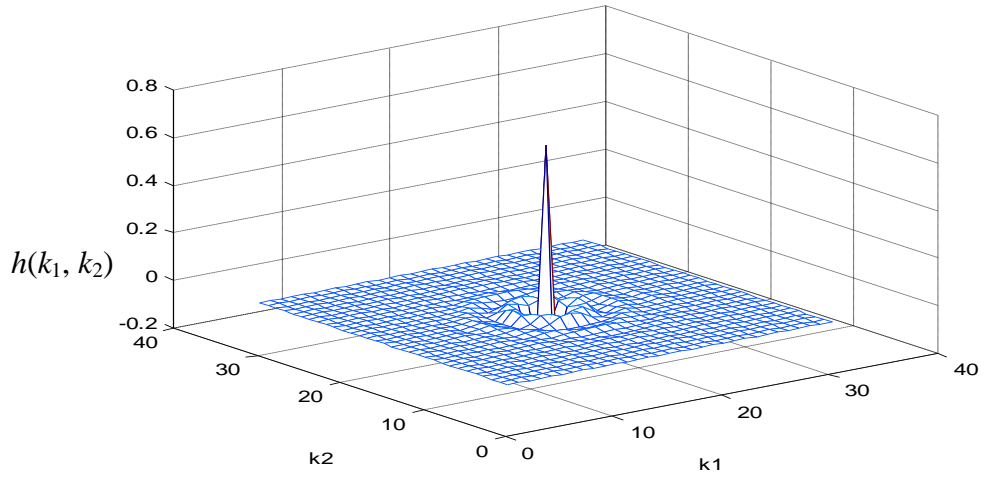


Figure (5.11) : réponse impulsionnelle du filtre RII passe-haut, méthode Itérative.

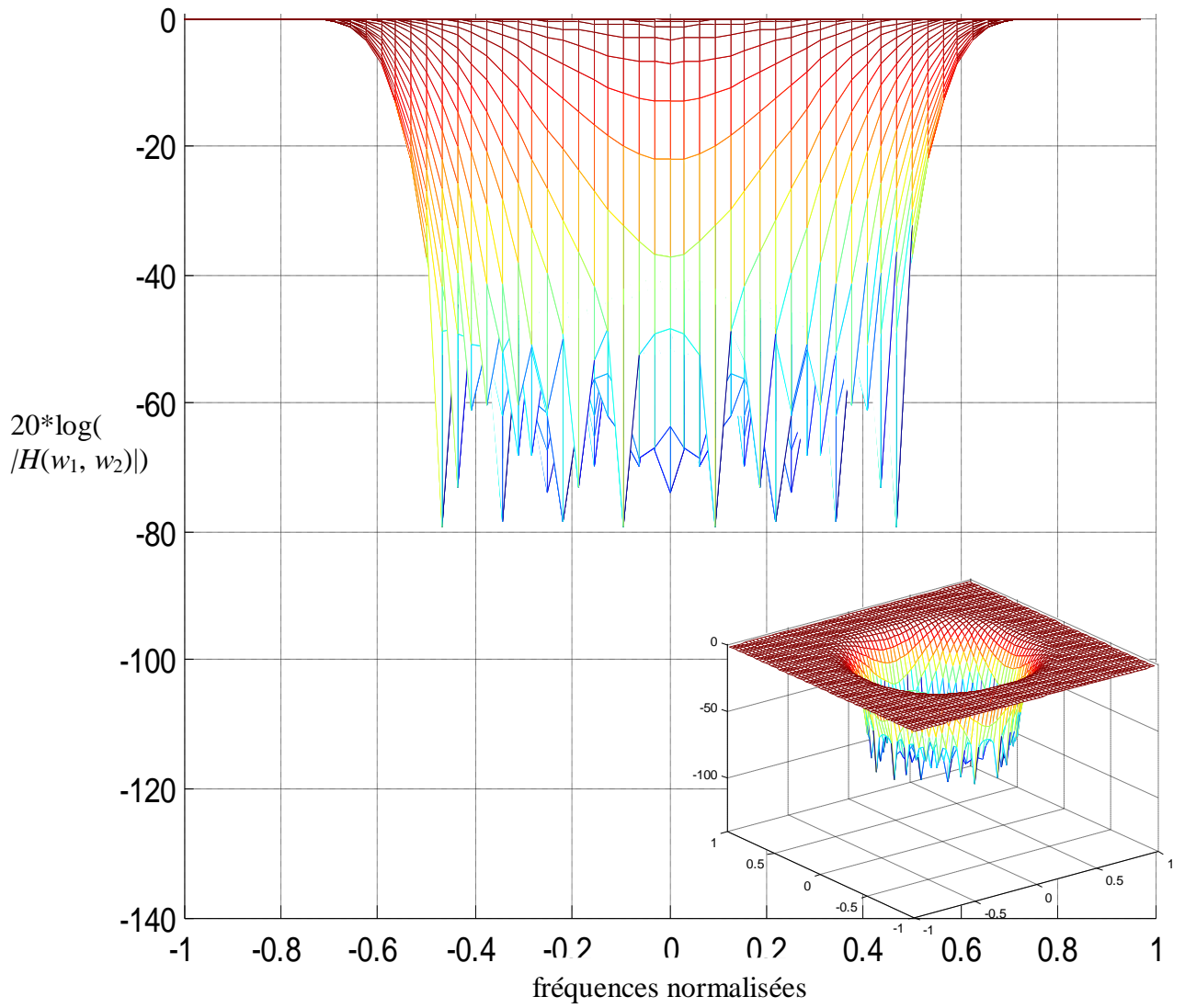


Figure (5.12) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-haut.

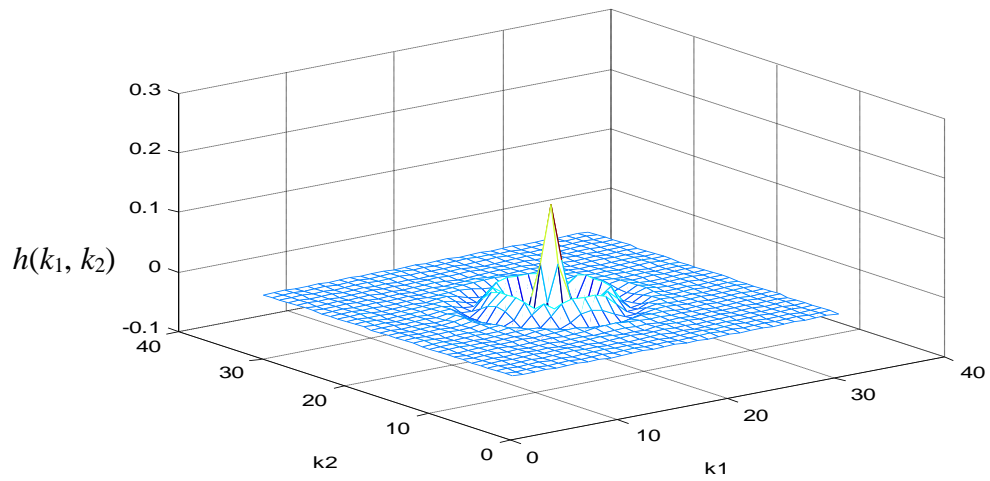


Figure (5.13) : réponse impulsionnelle du filtre RII passe-bande, méthode Itérative.

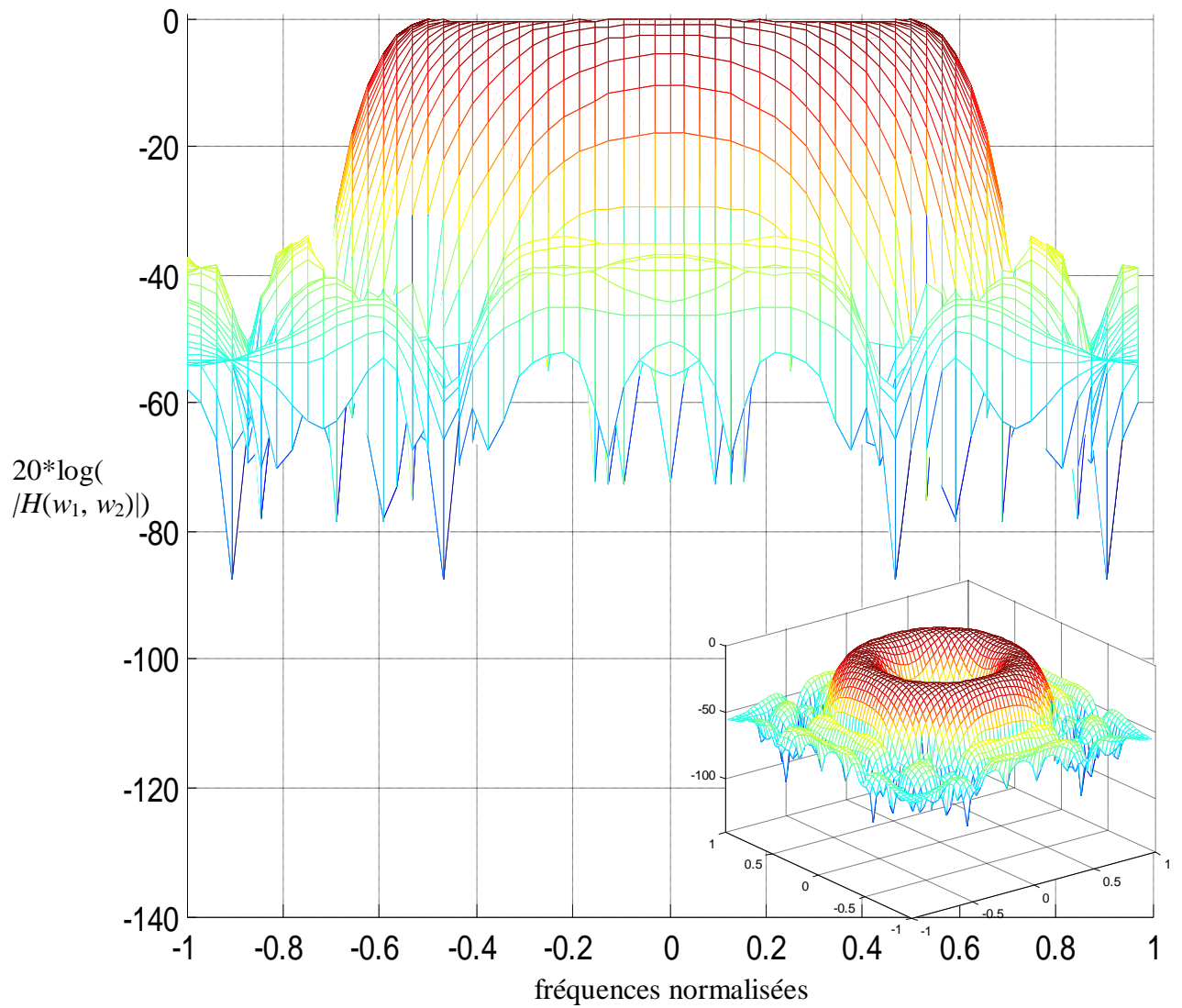


Figure (5.14) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-bande.

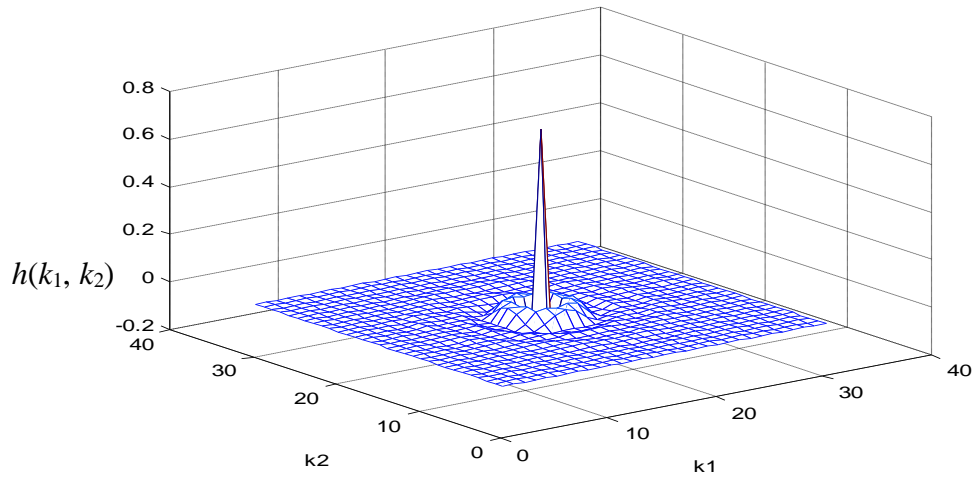


Figure (5.15) : réponse impulsionnelle du filtre RII coupe-bande, méthode Itérative.

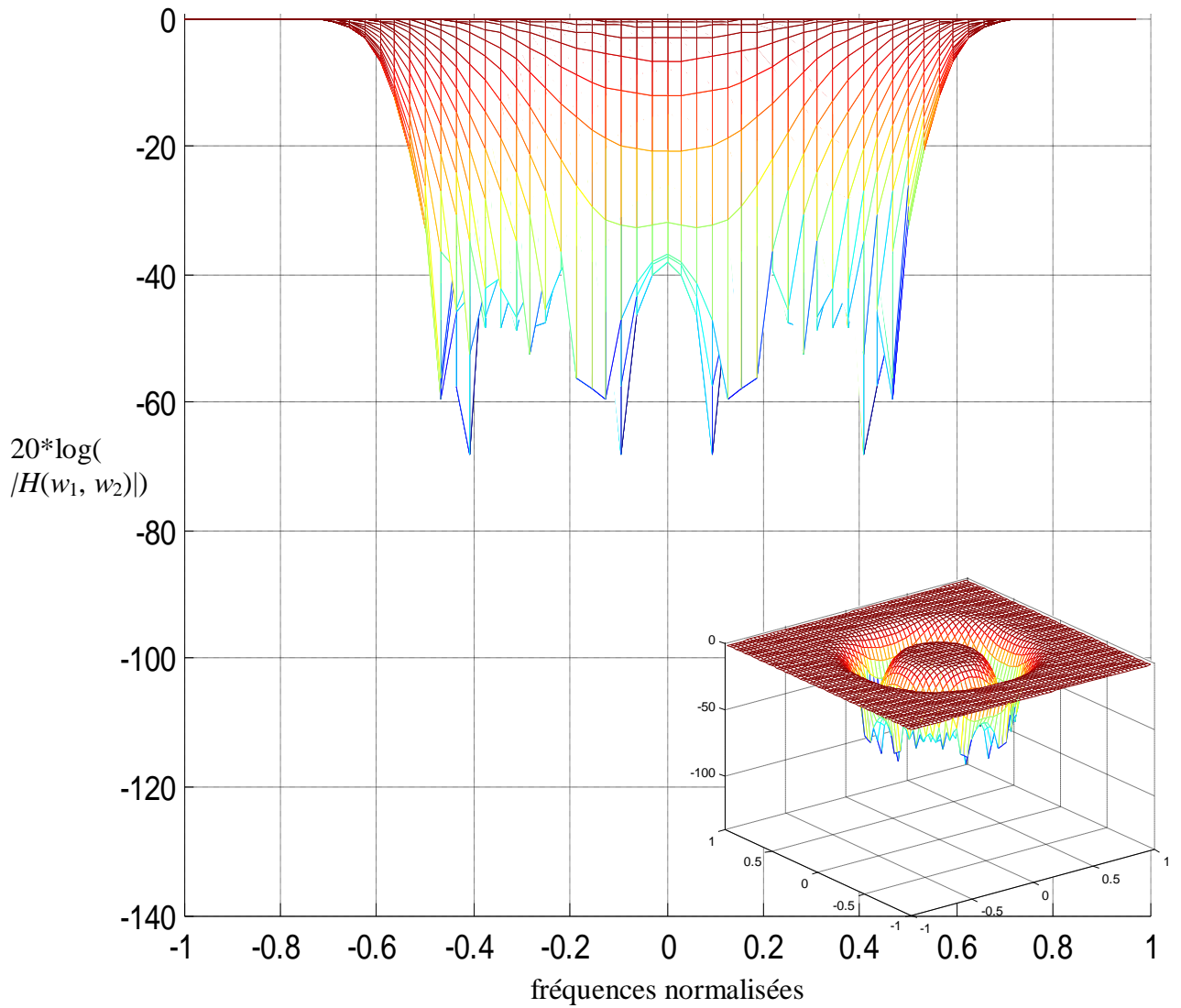


Figure (5.16) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII coupe-bande.

5.1.3 Synthèse : Méthode SDP

Soit les filtres numériques RII donnés par les gabarits suivants :

- l'ordre de filtre $n = 13$, $r = 13$ (dimension de la matrice a , et b).
- les régions normalisées de la bande passante et de la bande coupée correspondant pour chaque type de filtre égales :

Passe-bas : $R_p = [0 \ 0.4]$, et $R_s = [0.5 \ 1]$.

Passe-haut : $R_p = [0.6 \ 1]$, et $R_s = [0 \ 0.5]$.

Passe-bande : $R_p = [0.48 \ 0.55]$, $R_{s_1} = [0 \ 0.3]$ et $R_{s_2} = [0.7 \ 1]$.

Coupe-bande : $R_p = [0.3 \ 0.6]$, $R_{s_1} = [0 \ 0.1]$ et $R_{s_2} = [0.8 \ 1]$.

- nombre de itérations : $it=2$.

Remarque:

- le nombre totale de coefficients est égale à 13×13 (matrice b) + 13×13 (matrice a) = 338.

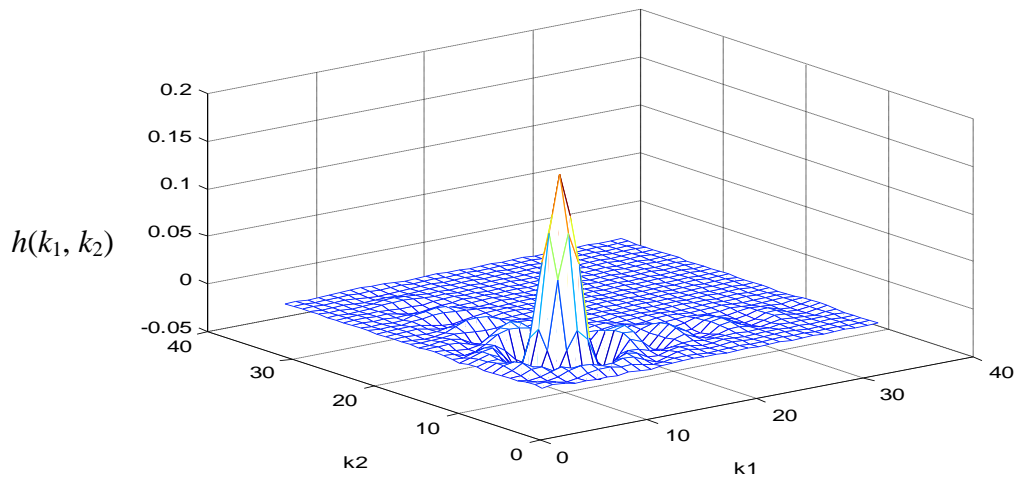


Figure (5.17) : réponse impulsionnelle du filtre RII passe-bas, méthode SDP.

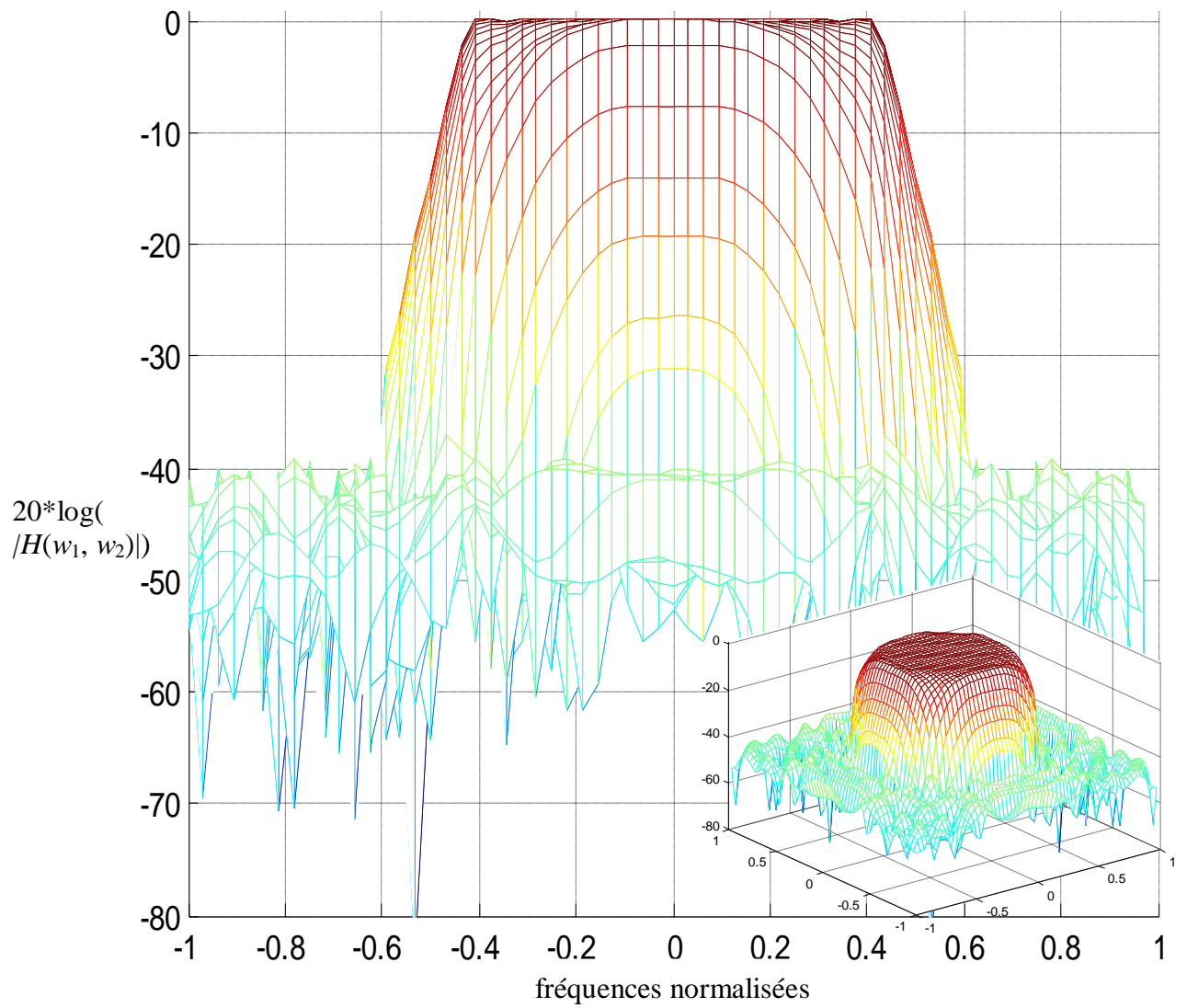


Figure (5.18) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-bas.

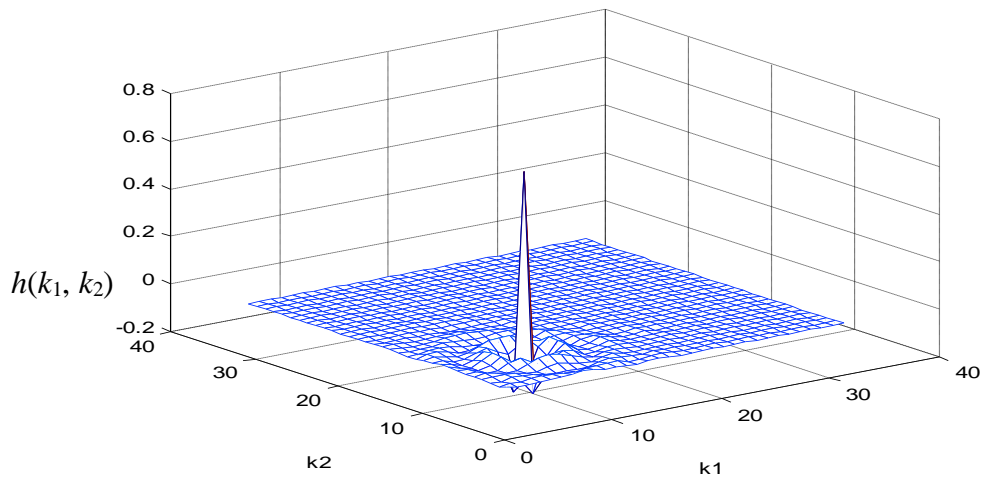


Figure (5.19) : réponse impulsionnelle du filtre RII passe-haut, méthode de SDP.

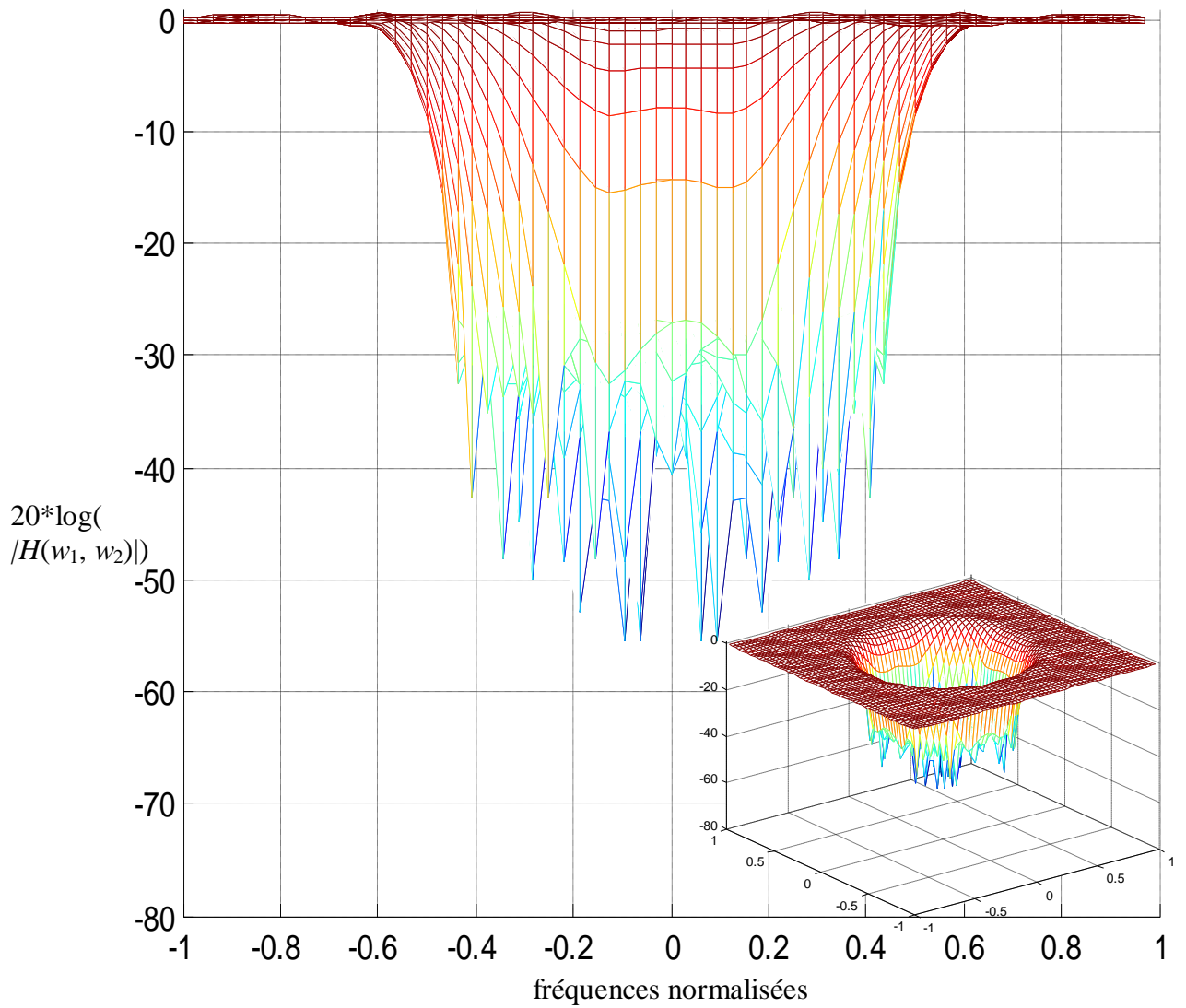


Figure (5.20) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-haut.

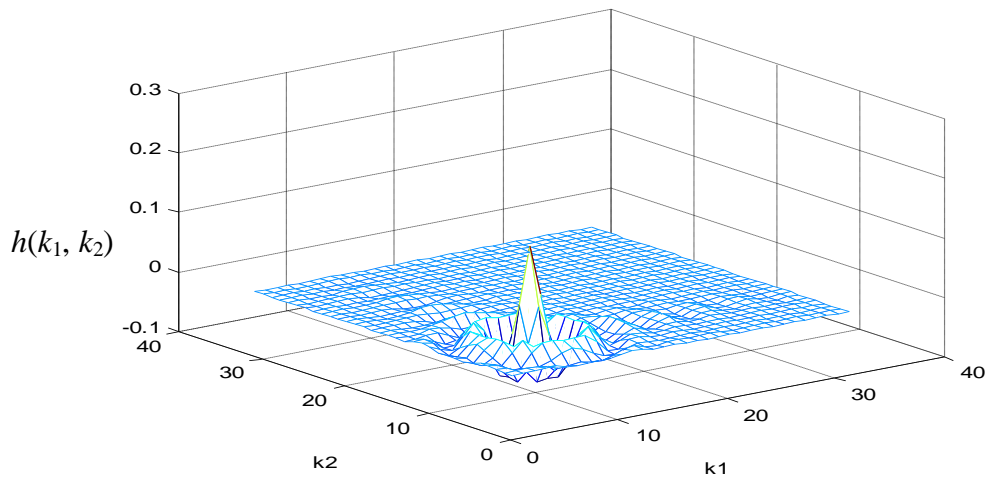


Figure (5.21) : réponse impulsionnelle du filtre RII passe-bande, méthode de SDP.

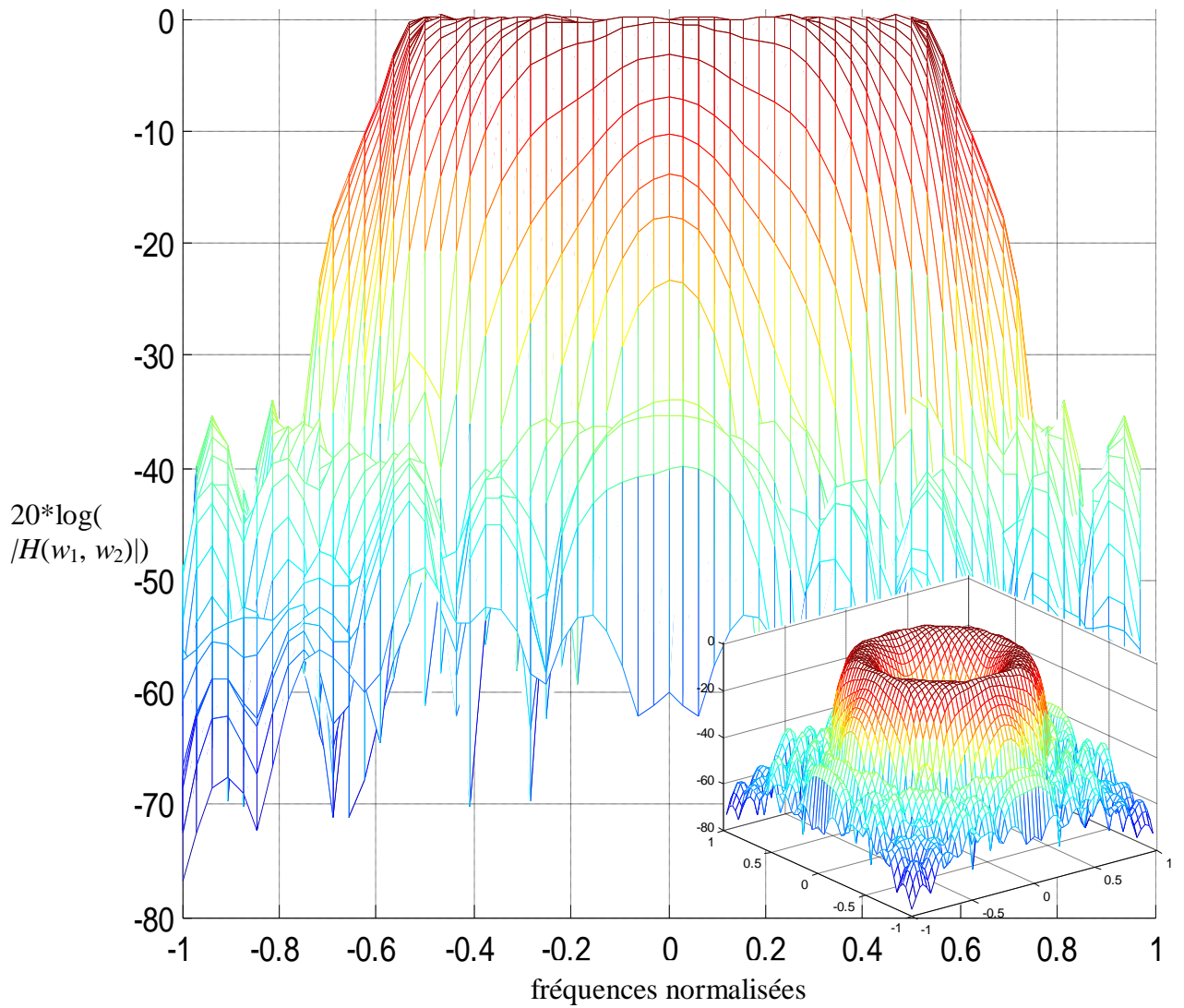


Figure (5.22) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII passe-bande.

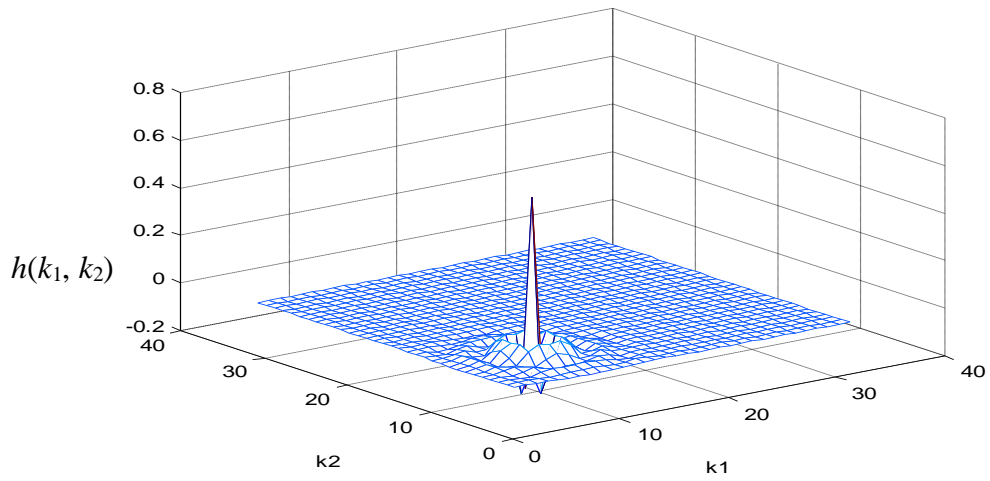


Figure (5.23) : réponse impulsionnelle du filtre RII coupe-bande, méthode de SDP.

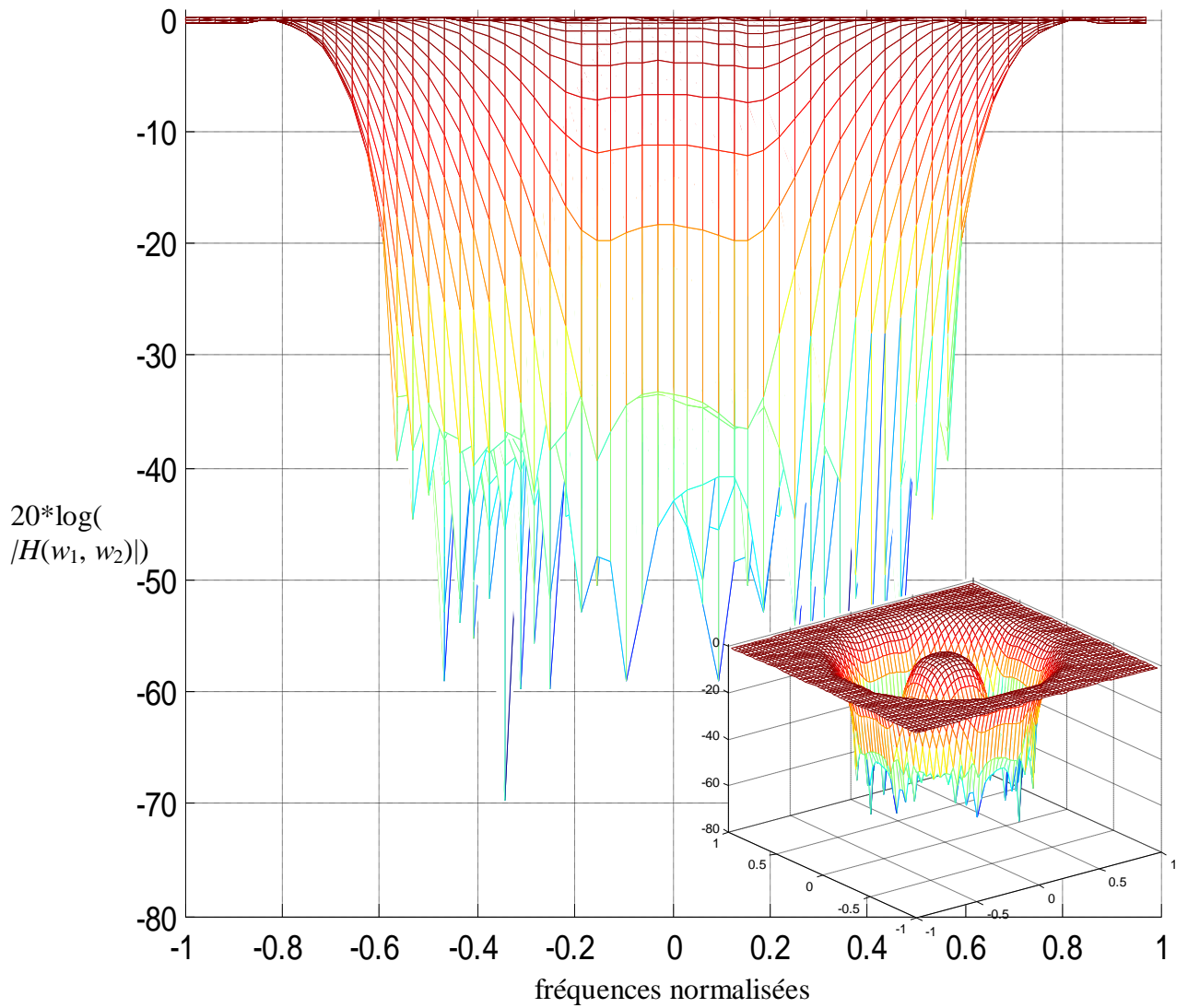


Figure (5.24) : spectre d'amplitude perspective "X-Z" (en dB) du filtre RII coupe-bande.

5.3 Réduction

Pour la réduction d'ordre, on applique la méthode de quasi-gramians au filtre synthétisé, on choisit deux types de filtres (passe-bas et passe-bande), et on utilise notre fonction `tf2ss2_2D.m` pour transformer les matrices a , et b au modèle dans l'espace d'état $\{A, B, C, d\}$. Ensuite, on applique la méthode quasi-gramians pour calculer un modèle d'ordre réduit $\{A_r, B_r, C_r, d\}$. Après la réduction on revient à la matrice a , et b , par la fonction `ss2tf_2D.m`.

5.3.1 Réduction : Méthode Itérative

- l'ordre de filtre original $n = 5, m = 5$, (dimension de la matrice a , et b).
- les dimensions de la matrice A, B et C sont $(n+2 \times m) \times (n+2 \times m), (n+2 \times m) \times 1, 1 \times (n+2 \times m)$, respectivement.
- nombre d'itérations égale $it = 10$.
- l'ordre de filtre (passe-bas) réduit $r_1 = 4, r_2 = 5$.
- l'ordre de filtre (passe-bande) réduit $r_1 = 4, r_2 = 4$.

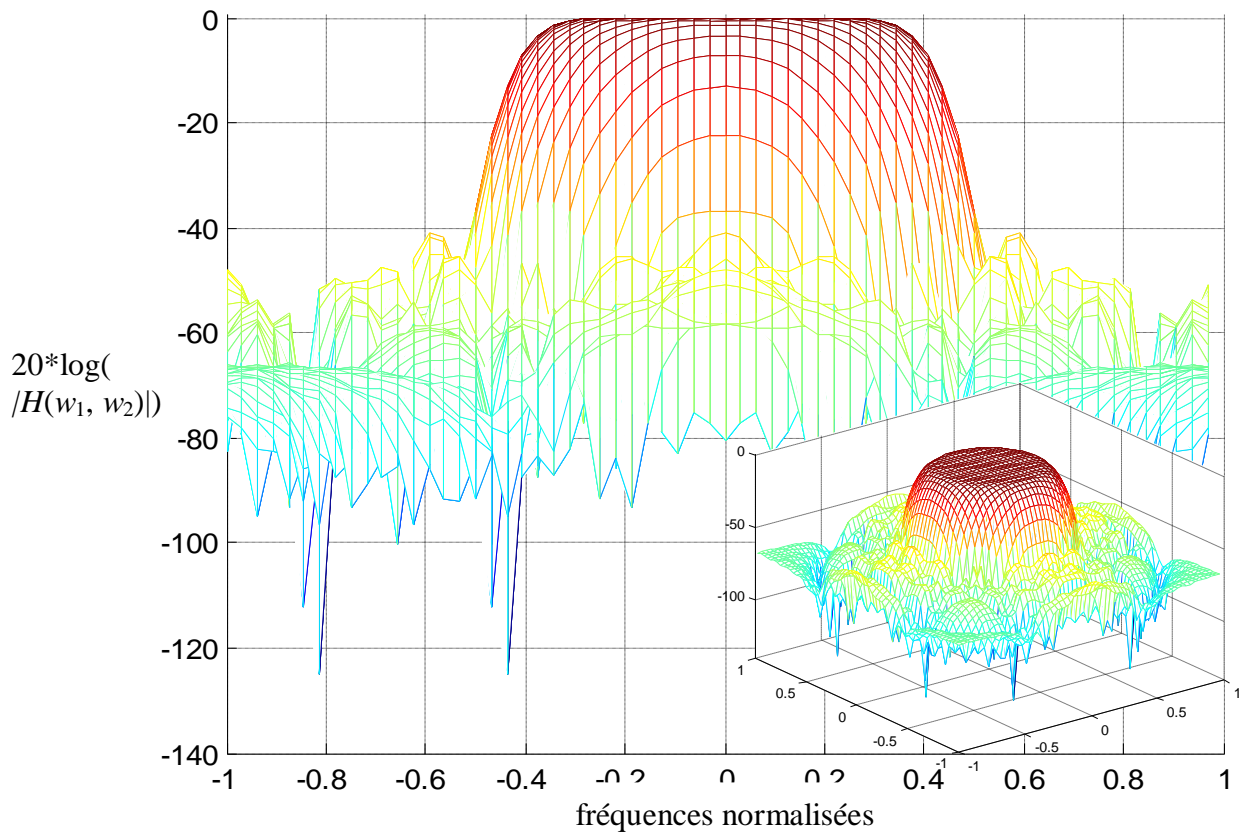


Figure (5.25) : spectre d'amplitude du filtre original, perspective "X-Z" (en dB) du filtre RII passe-bas (200 coefficients).

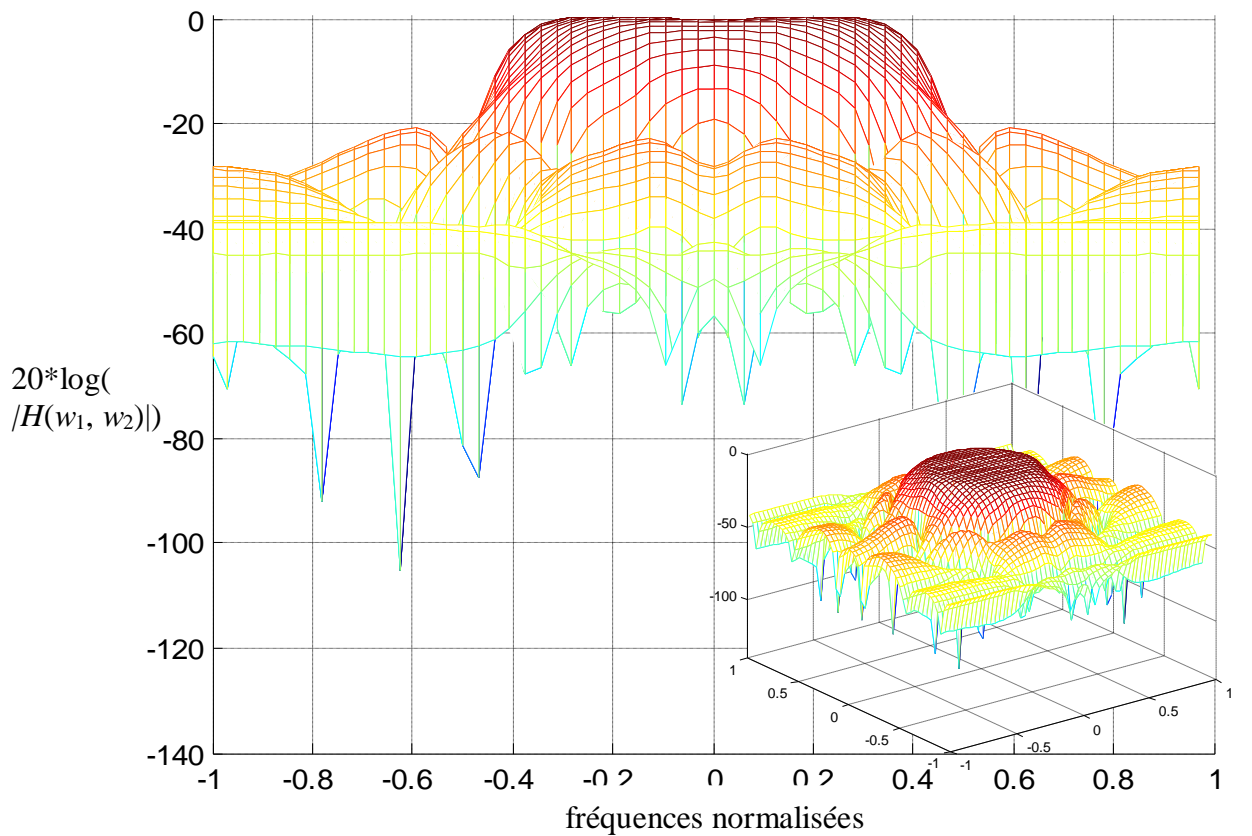


Figure (5.26) : spectre d'amplitude du filtre réduit d'ordre 4×5 , perspective "X-Z" (en dB) du filtre RII passe-bas (160 coefficients).

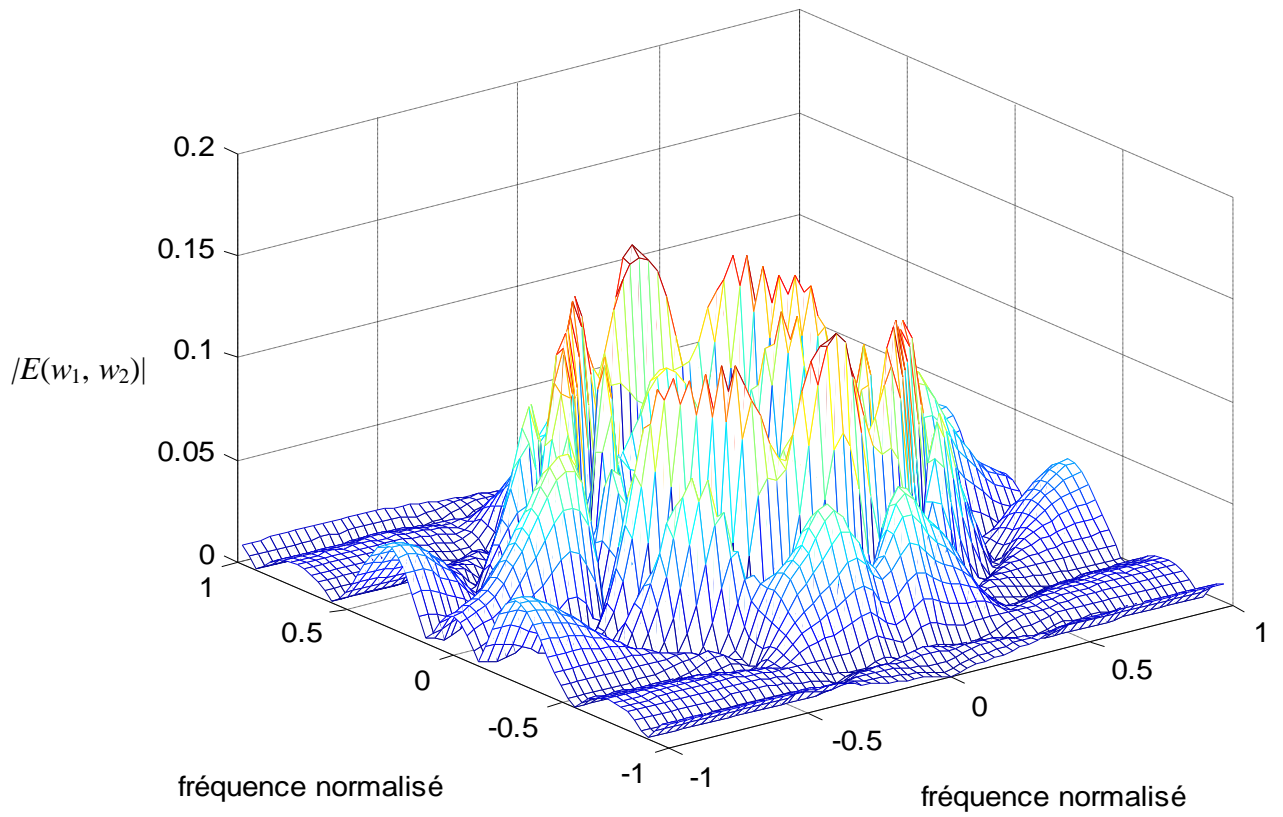


Figure (5.27) : *Erreur fréquentielle (échelle normal) entre filtre original et réduit*

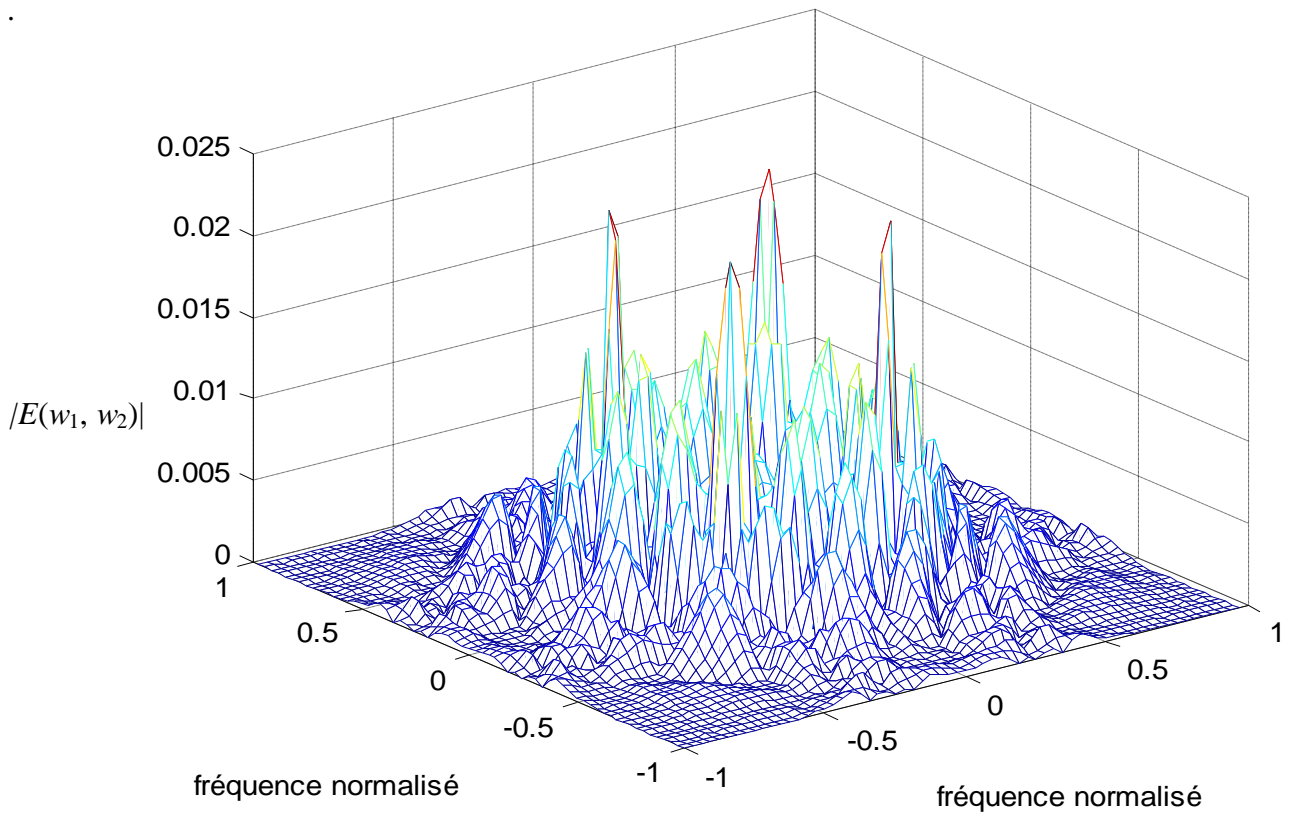


Figure (5.28) : *Erreur fréquentielle (échelle normal) entre filtre original et filtre synthétisée par la même méthode d'ordre 4×4 c.-à-d. 128 coefficients* .

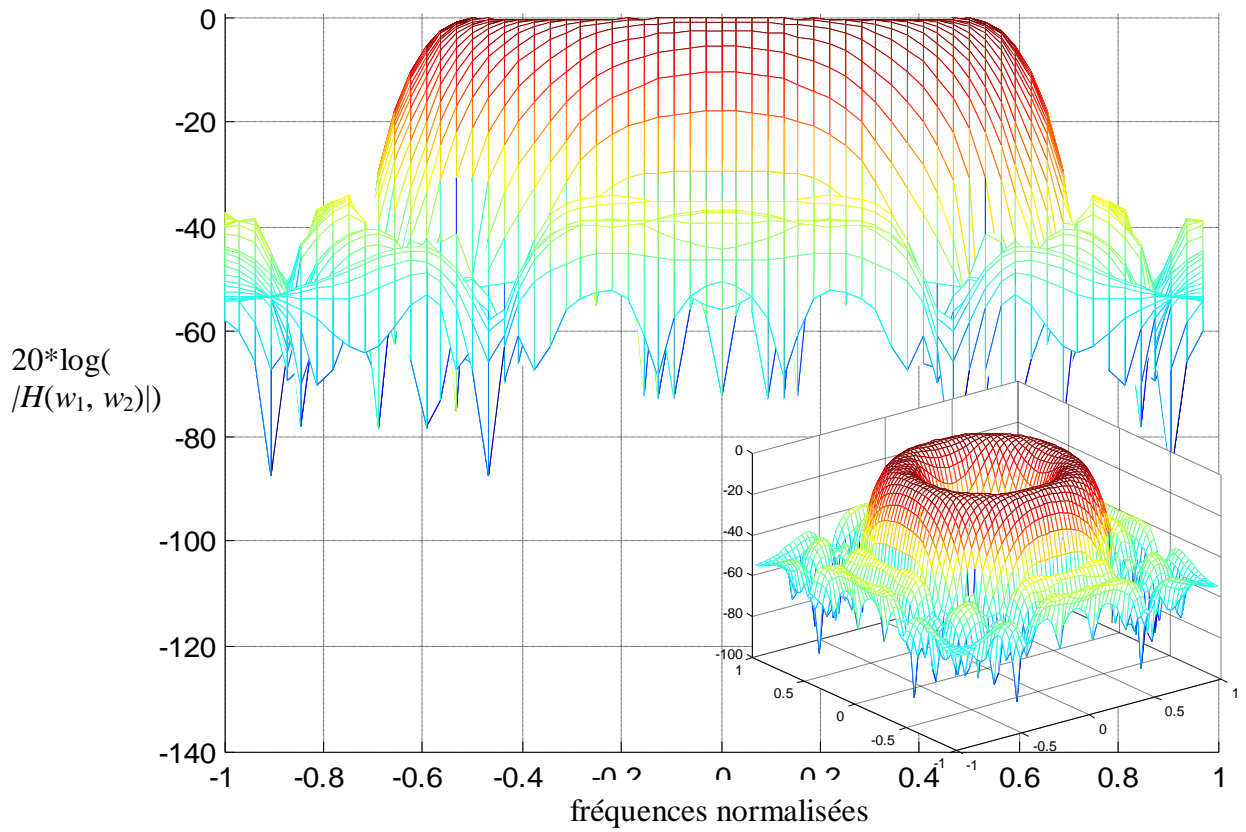


Figure (5.29) : *spectre d'amplitude du filtre original, perspective "X-Z" (en dB) du filtre RII passe-bande (200 coefficients).*

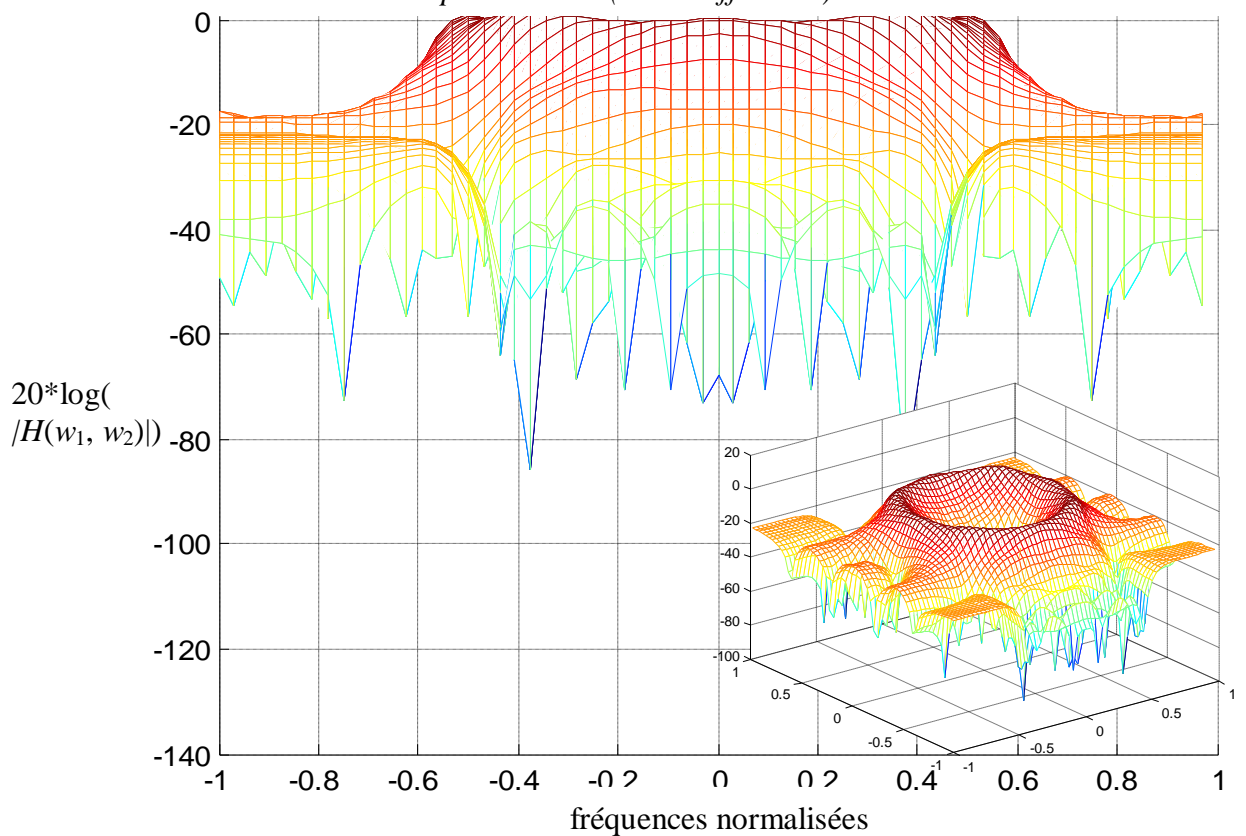


Figure (5.30) : *spectre d'amplitude du filtre réduit d'ordre 4×4 , perspective "X-Z" (en dB) du filtre RII passe-bande (128 coefficients).*

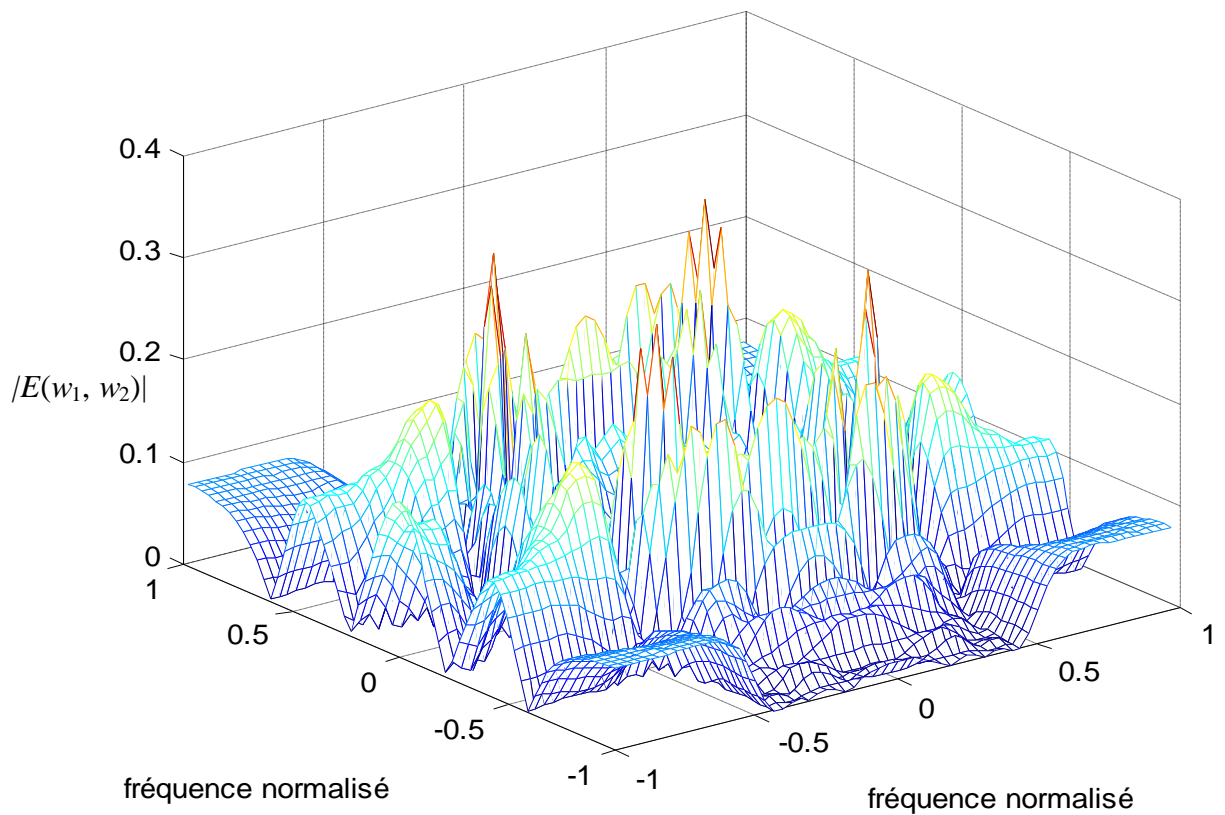


Figure (5.31) : Erreur fréquentielle (échelle normal) entre filtre original et réduit .

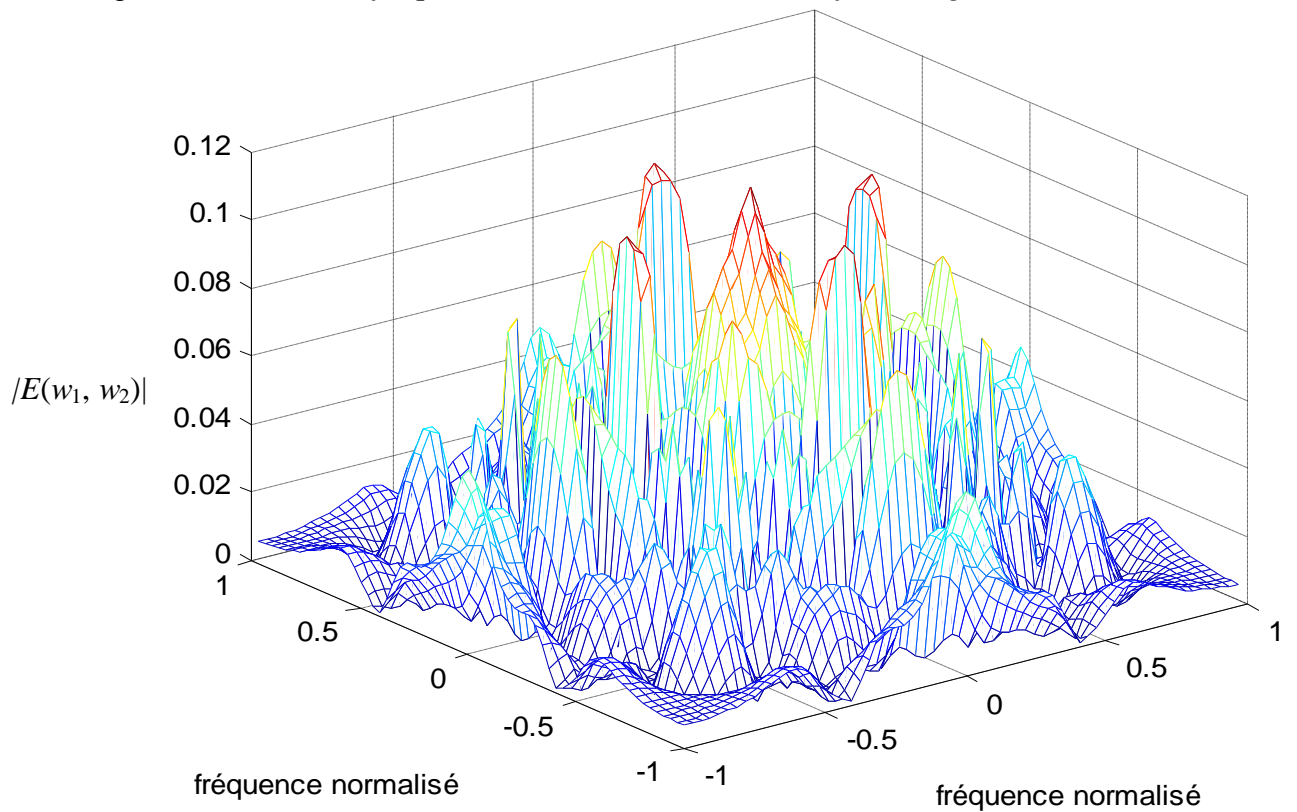


Figure (5.32) : Erreur fréquentielle (échelle normal) entre filtre original et filtre synthétisée par la même méthode d'ordre (4×4 c.-à-d. 128 coefficients) .

5.3.2 Réduction : Méthode SDP

- l'ordre de filtre original $n=13, m=13$, (dimension de la matrice a , et b).
- les dimensions de la matrice A, B et C sont $(n+m) \times (n+m), (n+m) \times 1, 1 \times (n+m)$, respectivement (*réalisation minimal*).

- nombre d'itérations égale $it=2$, (le nombre totale de coefficients est égale à 13×13 (matrice b) + 13×13 (matrice a) =338).
- l'ordre de filtre (passe-bas) réduit $r_1 = 8, r_2 = 8$, (le nombre totale de coefficients est égale à 8×8 (matrice b) + 8×8 (matrice a) =128).
- l'ordre de filtre (passe-bande) réduit $r_1 = 9, r_2 = 9$, (le nombre totale de coefficients est égale à 9×9 (matrice b) + 9×9 (matrice a) =162).

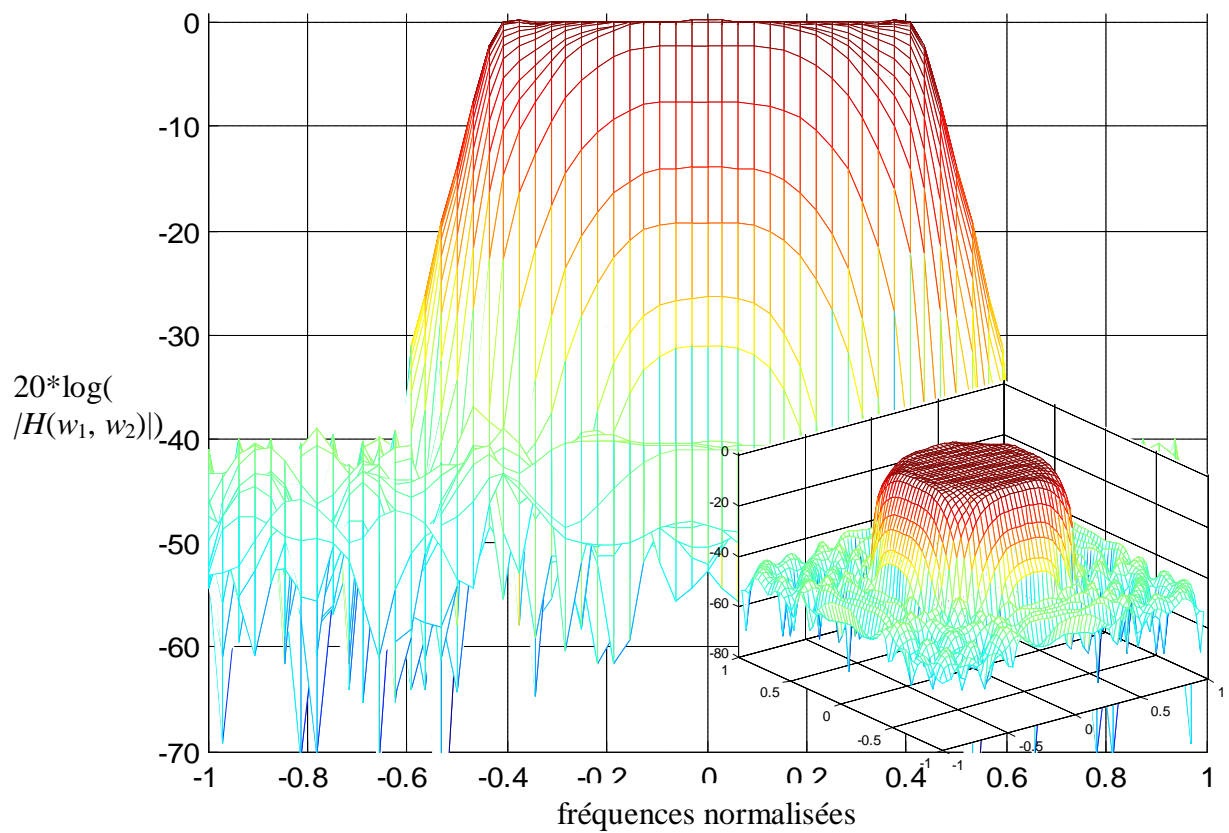


Figure (5.33) : *spectre d'amplitude du filtre original, perspective "X-Z" (en dB) du filtre RII passe-bas (338 coefficients).*

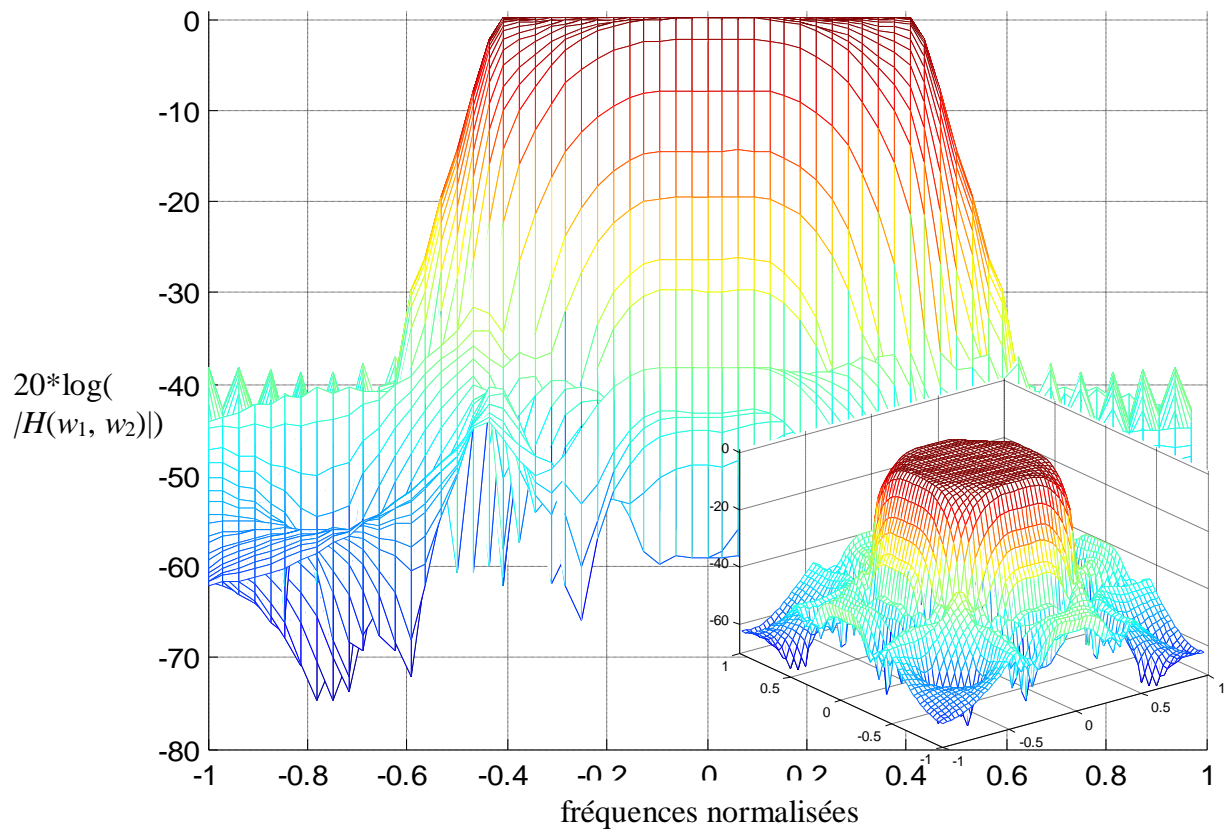


Figure (5.34) : *spectre d'amplitude du filtre réduit d'ordre 8x8, perspective "X-Z" (en dB) du filtre RII passe-bas (128 coefficients).*

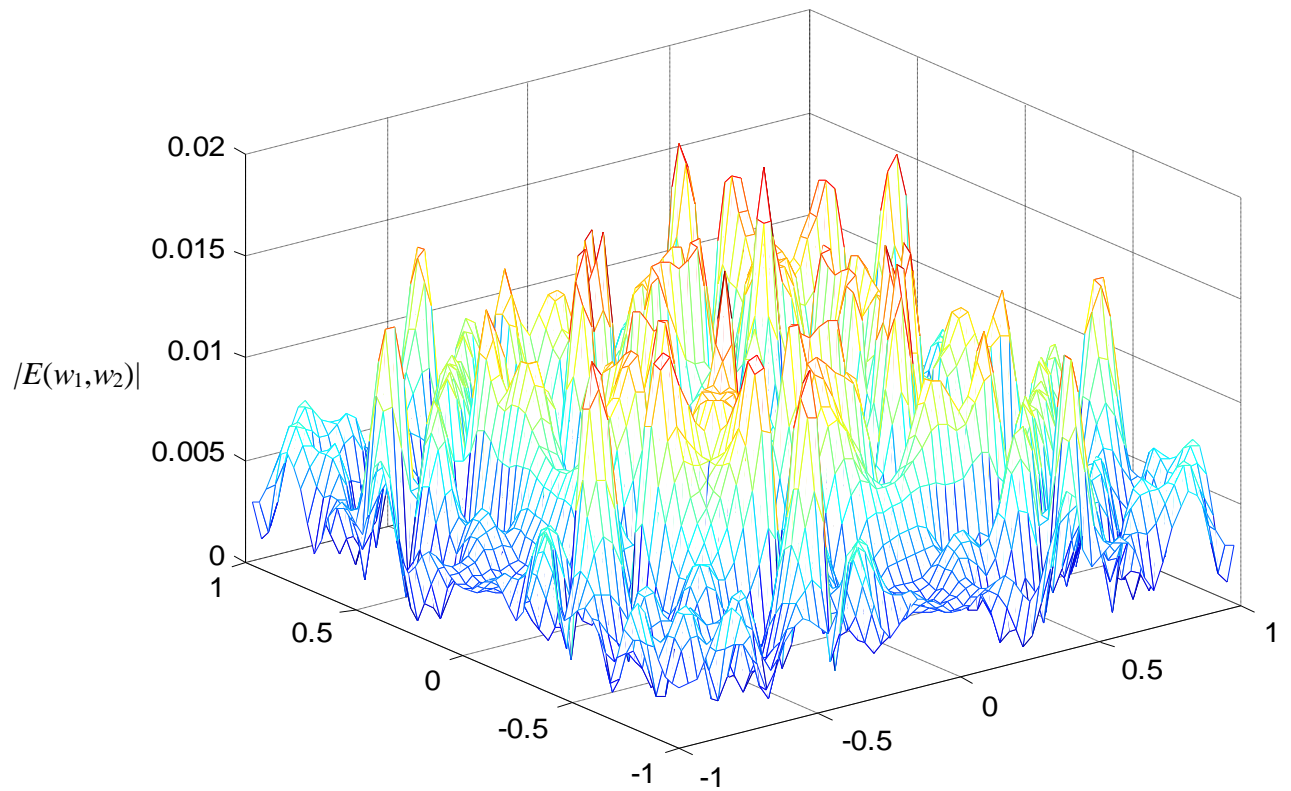


Figure (5.35) : Erreur fréquentielle (échelle normal) entre filtre original et réduit .

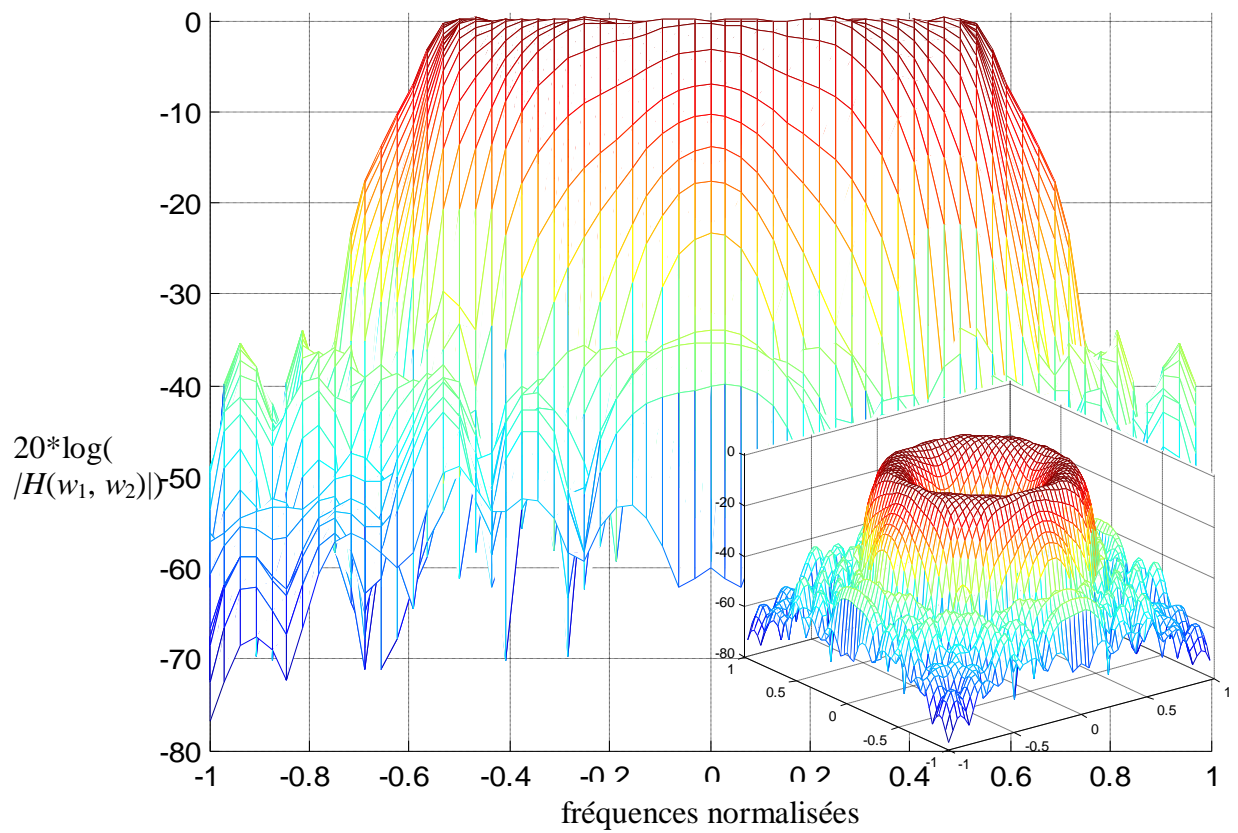


Figure (5.36) : *spectre d'amplitude du filtre original, perspective "X-Z" (en dB) du filtre RII passe-bande (338 coefficients).*

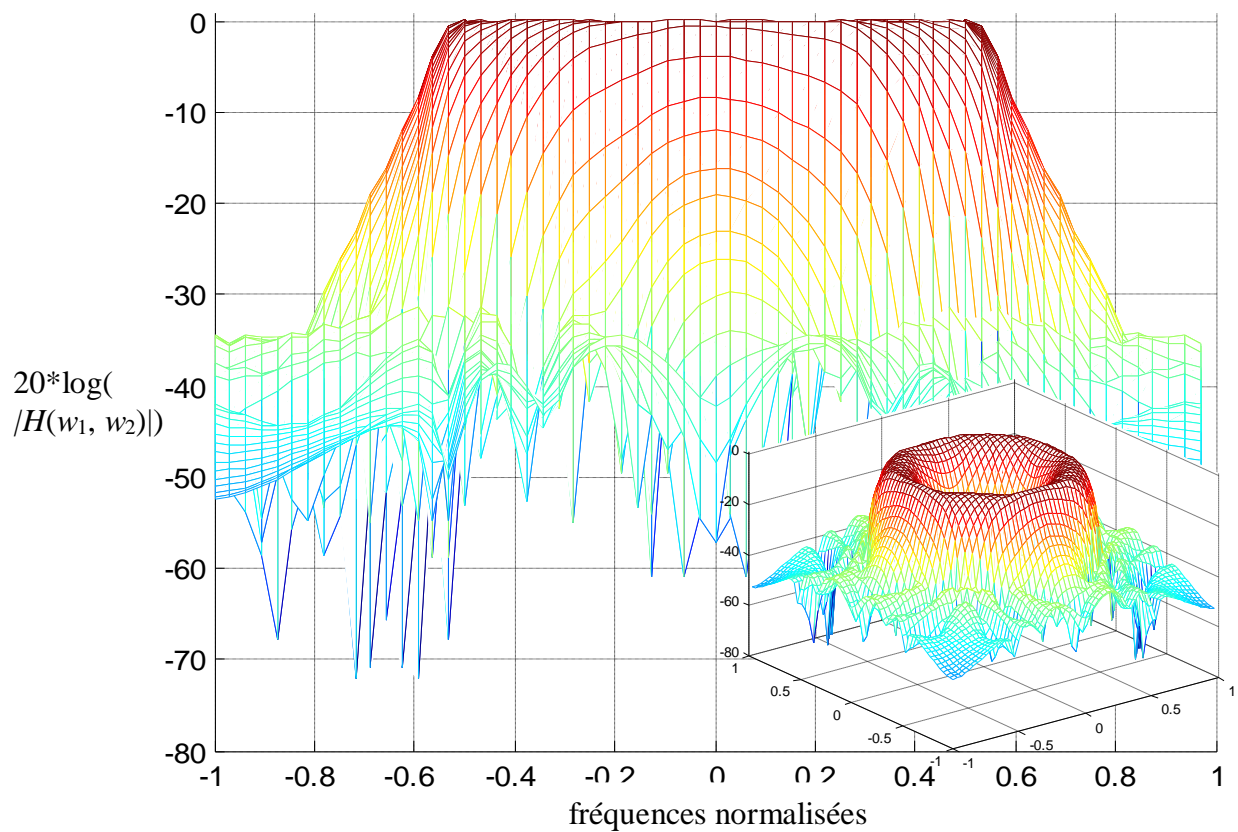


Figure (5.37) : *spectre d'amplitude du filtre réduit d'ordre 9x9, perspective "X-Z" (en dB) du filtre RII passe-bande (162 coefficients).*

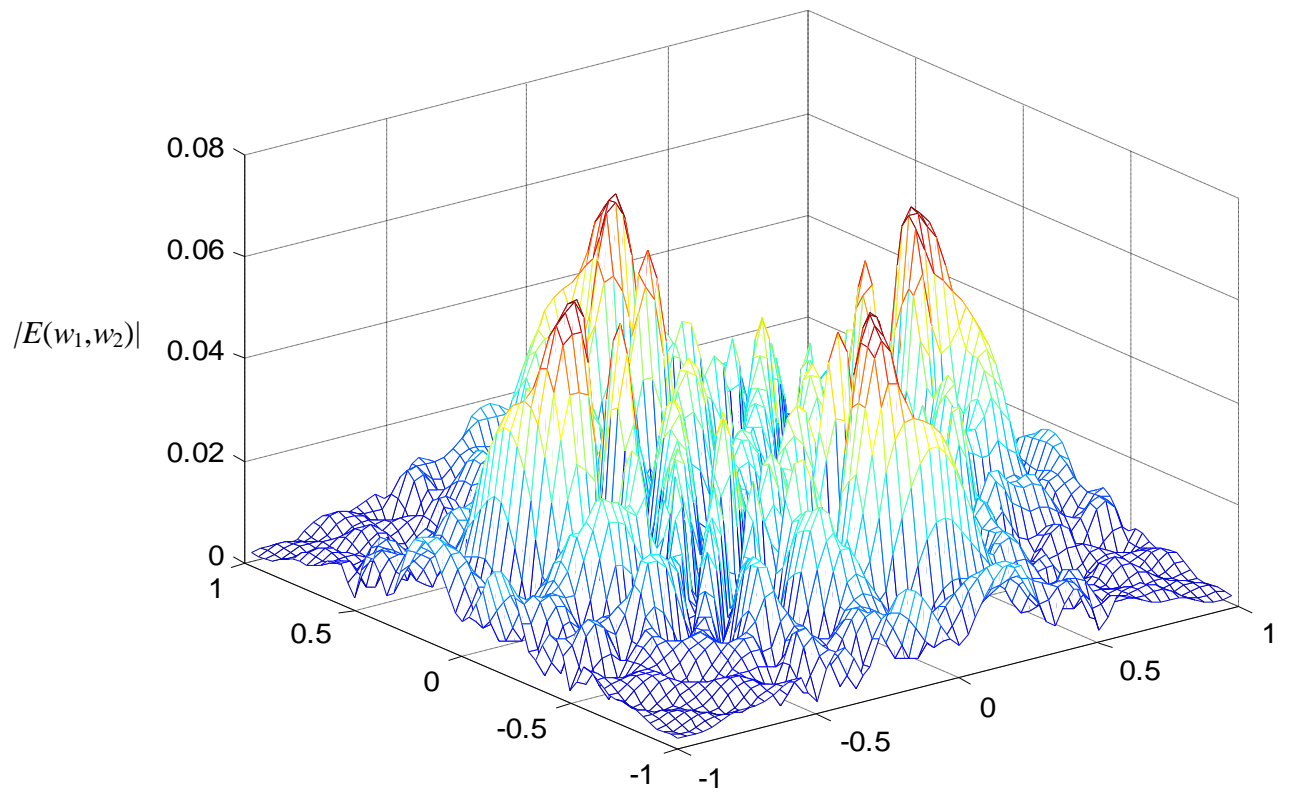


Figure (5.38) : *Erreur fréquentielle (échelle normal) entre filtre original et réduit .*

5.4 Interprétation et conclusion

Les résultats obtenus conduisent aux remarques suivant :

* La synthèse :

- Il n'existe pas une grande différence entre la méthode de Prony et la méthode Itérative, mais on peut remarquer une petite amélioration dans la méthode Itérative au niveau de la bande coupée (affaiblissements) tel que :

Prony / $\min(H(w_1, w_2)) = -49,09$ dB, (filtre passe-bas).

Itérative / $\min(H(w_1, w_2)) = -55,77$ dB, (filtre passe-bas).

- Les résultats obtenus dans la méthode SDP sont comparables aux deux autres méthodes, tel qu'on trouve un décroissement dans la performance dans les deux bandes (bande passant et bande coupée), il est possible d'améliorer les résultats par l'augmentation de nombre des coefficients ou de nombre d'itération. Il y a un bon aspect de cette méthode, c'est la stabilité, tel que la stabilité est mesurée par le grand module des racines de vecteur de colon ou ligne de la matrice de dénominateur 'a', on a trouvé les résultats suivant :

- $\max(\text{abs}(\text{roots}(a(:,1)))) = 0,8926 \Rightarrow$ (filtre passe-bas).

$= 0,7235 \Rightarrow$ (filtre passe-haut).

$= 0,8578 \Rightarrow$ (filtre passe-bande).

$= 0,7202 \Rightarrow$ (filtre coupe-bande).

où max, abs, et roots sont des fonctions dans logicielle Matlab qui représentant le valeur max d'un vecteur, le valeur absolue, et les racines d'une vecteur, respectivement.

* La réduction :

- A cause de la réalisation non minimal de filtre synthétiser par les méthodes de Prony et Itérative, et a cause que le filtre réduit peut être instable, et les résultats de réduction n'est pas bien (voir les figures 5.27, 5.31), on peut remarque que ces résultats de réduction comparable avec les filtres synthétiser par la méthode Itérative de même ordre (r_1, r_2) (filtre passe-bande) n'est pas mieux (voir les figures 5.28, 5.32).

- Mais dans la méthode de programmation semi-définie (SDP) la stabilité reste préserver :
 $\max(\text{abs}(\text{roots}(a(:,1)))) = 0,8926 \Rightarrow$ (filtre original) $\Rightarrow 0,9119 \Rightarrow$ (filtre réduit) (filtre passe-bas).

$\max(\text{abs}(\text{roots}(a(:,1)))) = 0,7202 \Rightarrow$ (filtre original) $\Rightarrow 0,6859 \Rightarrow$ (filtre réduit) (filtre coupe-bande).

Les résultats obtenus apparent que le filtre passe-bas réduit mieux que filtre coupe-bande tel que le nombre des coefficients diminuer de 338 \rightarrow 128, et l'erreur entre la réponse fréquentielle de filtre initial et réduit (figure 5.35 $\Rightarrow \max(E) = 0.02$) mais pour filtre coupe-bande le nombre des coefficients diminuer de 338 \rightarrow 162 avec un erreur entre la réponse fréquentielle de filtre initial et réduit (figure 5.38 $\Rightarrow \max(E) = 0.08$) .

CONCLUSION GENERALE

La stabilité et la linéarité de la phase d'un filtre RII 2-D sont peut être les grand problèmes dans la synthèse. Dans le cas 1-D on représente un pôle ou un zéro par un point dans repère 2-D (partie real et partie imaginaire), mais dans le cas 2-D les pôles rendent des surfaces dans repère 4-D, il est extrêmement difficile de tester la stabilité et stabiliser un filtre instable.

Dans le traitement d'image, l'image qui sortie d'un système (un filtre dans ce cas) peut être déformé à cause la non linéarité de la phase.

La réduction aussi endure quelques problèmes, la réduction dans l'espace d'état par les méthodes classique, par exemple, la réduction par la transformation équilibre exige la réalisation minimal d'une fonction transfert dans l'espace d'état, ensuite, trouver une méthode pour calculer la matrice de transformation à modèle équilibre, qui exige que le nombre de calculs soit le plus possible réduit et en même temps le maintien de la stabilité.

- Dans ce travail nous essayons de obtenir une méthode de synthèse qui traite ces problèmes avec les méthodes classiques tel que la méthode de Prony ou Itérative qui basent sur l'aide d'identification. Pour obtenir un filtre à phase linéaire, nous avons divisé la région de support de la réponse impulsionnelle désirée à quatre régions, ensuite, nous avons synthétisé les 4 filtres (un seul filtre dans la cas où $h_d(n_1, n_2)$ a symétrie quadruple) par la méthode de Prony ou méthode itérative. Le résultat un filtre à phase linéaire à ordre réduit (nombre total de coefficients égale 200) par ajout la troisième méthode. La méthode de programmation semi-définie (nombre total de coefficients égale 338) mais dans cette méthode la stabilité est assurée, avec une phase approximativement linéaire.

- Dans la réduction d'ordre nous avons choisie la méthode de quasi-équilibrées en raison de sa simplicité. Mais cette méthode a un inconvénient est ne préserve pas la stabilité du système réduit généralement, mais le cas où le système à un dénominateur séparable, le système réduit reste stable comme le filtre synthétisé par la méthode SDP.

Références Bibliographiques

- [1]: Jae S.Lim, '*Two-Dimensional Signal and Image Processing*', Prentice Hall PTR 1990.
- [2]: K. Premaratne, E. I. Jury, and M. Mansour, '*An algorithm for model reduction of 2-D discrete time systems*,' IEEE Trans. Circuits Syst, vol. 38, pp. 1116-1132, Sept. 1990.
- [3]: W.-S. Lu, '*Design of stable 2-D IIR digital filters using iterative semidefinite programming*,' in Proc. ISCAS'00, Geneva, Switzerland, May 2000.
- [4]: W.-S. Lu, '*A Unified Approach for the Design of 2-D Digital Filters via Semidefinite Programming*,' IEEE Transactions on Circuits and Systems, vol. 49, no. 6, JUNE 2002.
- [5]: T. Kailath, *Linear Systems*, Prentice-Hall, 1981.
- [6]: R. P. Roesser, '*A discrete state space model for linear image processing*,' IEEE Trans. Automat. Contr., Feb. 1975.
- [7] : S. Attasi, '*Modélisation et traitement des suites à deux indices*,' Rapport Laboria, Sept. 1975.
- [8]: E. Fornasini and G. Marchesini, '*State-space realization theory of two-dimensional filters*,' IEEE Trans Automat Contr., vol. AC-21, pp. 484-492, Aug.1976.
- [9]: D. D. Givone and R. P. Roesser, '*Minimization of multidimensional linear iterative circuits*,' IEEE Trans. Comput., vol. C-22, pp. 673-678, July 1973.
- [10]: S. Y. Kung, B. C. Levi, M. Morf, and T. Kailath, '*New results in 2-D systems theory, Part II: 2-D state-space models-Realization and the notions of controllability, observability, and minimality*,' Proc. IEEE, vol. 65, pp. 945-961, June 1977.
- [11] : S. K. Mitra, A. D. Sagar, and N. A. Pendergrass, '*Realizations of two-dimensional recursive digital filters*,' IEEE Trans. Circuits and Systems, vol. CAS-22, pp. 177-184, Mar. 1975.
- [12]: E. D. Sontag, '*On first order equations for multidimensional filters*,' IEEE Tram. Arount. Speech, Signal Processing, vol. ASSP-26, pp. 480-482, Oct. 1978.
- [13]: S. J. Varoufakis, P. N. Paraskevopoulos, and G. E. Antoniou, '*On the minimal state-space realizations of all-pole and all-zero 2-D systems*,' IEEE Trans. Circuits Syst., vol. CAS-34, pp. 289-292, Mar. 1987.
- [14]: T. Hinamoto and F. W. Fairman, '*Separable denominator state-space realization of two-dimensional filters using a canonic form*,' IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSP-29, pp. 846-853, Aug. 1981.

- [15]: H. Luo, W.-S. Lu, and A. Antoniou, 'New Algorithms for the Derivation of the Transfer-Function Matrices of 2-D State-Space Discrete Systems', IEEE Transactions on Circuits and Systems -1:Fundamental Theory and Applications, vol. 44, no. 2, February 1997.
- [16]: T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1981.
- [17]: W.-S. Lu and E. B. Lee, 'Stability analysis for two-dimensional analysis', IEEE Trans. Circuits Syst., vol. CAS-30, pp. 455–461, July 1983.
- [18]: B. C. Moore, 'Principal component analysis in linear systems: Controllability, observability and model reduction', IEEE Trans. Automat. Contr., vol. 26, pp. 17-32, Feb. 1981.
- [19]: K. Glover, 'All optimal Hankel norm approximation of linear multivariable systems and their L_2 -error bounds', Int. J. Contr., vol. 39, no. 6, pp. 1115-1193, 1984. 17-31, 1981.
- [20]: W.-S. Lu, H.-P. Wang, and A. Antoniou, 'Design of two-dimensional digital filters using the singular value decomposition and balanced approximation', IEEE Trans. Signal Processing, vol. 39, pp. 2253-2262, Oct. 1991.
- [21]: W.-S. Lu, H. Luo, and A. Antoniou, 'Recent Results on Model Reduction Methods for 2-D Discrete Systems', IEEE, 1996
- [22]: W.-S. Lu, E. B. Lee, and Q. T. Zhang, 'Balanced approximation of two-dimensional and delay differential systems', Int. J. Contr., vol. 47, pp. 2199-2218, 1987.
- [23]: W. Wang, K. Glover, J. C. Doyle, and C. Beck, 'Model reduction of LFT systems', Proc. 30th IEEE Conf. on Decision and Control, pp. 1233-1238, Brighton, UK, Dec. 1991.
- [24]: K. Zhou, J. L. Aravena, G. Gu, and D. Xiong, 'Two-dimensional system model reduction by quasi-balanced truncation and singular perturbation', IEEE Trans. Circuits Syst., vol. 41, pp. 593-602, Sept. 1994.
- [25]: A. J. Laub, 'On computing balancing transformations', Proc. JACC, FA 8-E, San Francisco, 1980.
- [26]: H. Luo, W.-S. Lu, and A. Antoniou, 'Numerical solution of the two-dimensional Lyapunov equations and application in order reduction of recursive digital filters', in Proc. IEEE Pacific Rim Con& Commun., Comput., Signal Process., Victoria, B.C., Canada, May 1993, pp. 232-235.