

PARCOURS MASTER 2

ATiAM

Parcours multi-mentions du Master Sciences et Technologies
Université Pierre et Marie Curie - Paris 6
en collaboration avec TELECOM ParisTech et l'Ircam

Mémoire de stage

*Mesure, Identification et Simulation de systèmes
non-linéaires à mémoire : Vers un sampler
d'effets*

Romain HENNEQUIN

< romain.hennequin @ polytechnique.org >

Directeur de stage : Rémy MULLER
Période : 17 Mars - 12 Septembre 2008
Organisme : Univers-Sons
159 rue Amelot
75011 Paris

Remerciements

Je tiens à remercier toute l'équipe d'Univers-sons pour m'avoir accueilli pendant près de 6 mois pour ce stage de recherche. En particulier, bien entendu, mon directeur de stage Rémy Muller qui m'a permis de faire ce stage et qui s'est particulièrement investi pour que celui-ci se déroule dans les meilleures conditions. Je souhaite également remercier Olivier Tristan pour ses conseils sur le développement en C++ et Alain Etchart à qui je peux garantir que mon stage a marché avec des piles. Je tiens aussi à remercier tout particulièrement Thomas Hélié de l'IRCAM pour le temps qu'il m'a consacré durant ce stage et pour ses conseils avisés qui ont permis d'orienter mes recherches. Enfin, je remercie Caroline Furois pour ses nombreuses et vaines tentatives de comprendre le sujet de mon stage.

Table des matières

Introduction	i
1 Mesure de systèmes linéaires invariants	1
1.1 Impulsion de Dirac	2
1.2 Méthode spectrale	3
1.3 Méthode d'intercorrélation	4
2 Mesure de systèmes linéaires variants	5
2.1 Modèle	5
2.2 Interpolation paramétrique	6
2.2.1 Cas général	6
2.2.2 Hypothèses supplémentaires : cas de la pédale Wah Wah	6
2.3 Mesure des systèmes périodiques	8
2.3.1 Mesure de la période	8
2.3.2 Mesures des systèmes périodiques	9
2.3.3 Systèmes multipériodiques et apériodiques	10
2.3.4 Mise en oeuvre et résultats	10
3 Cadre général des systèmes non-linéaires	13
3.1 Série de Volterra	13
3.1.1 Définition des séries de Volterra	13
3.1.2 Transformée de Laplace et transformée en z	14
3.1.3 Unicité de la description d'une série de Volterra	15
3.2 Systèmes simplifiés	15
3.2.1 Systèmes de type <i>schéma-bloc</i>	16
3.2.2 Convolution dynamique	16
4 Méthodes d'identification	21
4.1 Séparation des ordres	21
4.1.1 Méthodes	21
4.1.2 Améliorations	24
4.1.3 Utilisation	24
4.2 Signaux aléatoires	24
4.2.1 Cas général	24
4.2.2 Cas du modèle de Hammerstein généralisé	28

4.2.3	Utilisation et Résultats	29
4.3	Impulsions de Dirac et convolution dynamique	30
4.3.1	Réponse d'un système de Volterra à des impulsions de Dirac	30
4.3.2	Mesure de systèmes décrits par la convolution dynamique	31
4.3.3	Mise en oeuvre	31
4.4	Sinusoïdes et sweeps sinusoïdaux	33
4.4.1	Réponse d'un système de Volterra à une sinusoïde pure (régime permanent)	33
4.4.2	Propriétés des sweeps logarithmiques	34
4.4.3	Système de Hammerstein généralisé	36
4.4.4	Système S_M	39
5	Régression à l'aide de noyaux reproduisants	43
5.1	Moindres carrés	43
5.1.1	Cas général	43
5.1.2	Modèle de Hammerstein généralisé	44
5.2	Régression non-linéaire et noyaux reproduisants	45
5.2.1	Problèmes de régression	45
5.2.2	Dictionnaire de fonctions	46
5.2.3	Méthode à noyau reproduisant	46
5.3	Utilisation et résultats	51
	Conclusion et perspectives	55
	Annexes	I
A	Simulation numérique des systèmes non-linéaires	I
A.1	Repliement spectral	I
A.2	Méthodes générales de simulation	II
A.3	Système de Volterra	III
A.4	Simulation de systèmes par bloc	III
B	Equivalence <i>schéma-bloc</i>/Volterra	V
B.1	Système de Hammerstein	V
B.2	Système de Wiener	V
B.3	Système de Wiener-Hammerstein	VI
B.4	Système de Hammerstein généralisé	VII
B.5	Système de Wiener généralisé	VII
B.6	Système S_M	VII
C	Calcul pour l'identification de systèmes S_M	IX
C.1	Identification de systèmes S_M à l'aide de sweeps	IX
C.2	Identification de l'ordre 2	XIII
D	Calcul pour l'identification par intercorrélacion	XVII
	Bibliographie	XX

Introduction

L'objectif de ce stage a été de trouver des méthodes qui permettraient de réaliser un “sampler d'effets” : de façon analogue aux “samplers” de sons qui permettent de lire des échantillons enregistrés de sons pouvant avoir été produits par n'importe quelle source, l'idée de pouvoir “enregistrer” un effet audio est particulièrement séduisante bien qu'un peu utopique si on reste dans un cadre très général. Le sampler d'effets est d'ailleurs utilisé depuis un certain temps dans le cadre des systèmes audio linéaires invariants, notamment dans le cadre de la simulation d'acoustique de salles : La réponse impulsionnelle de la salle est mesurée (elle correspond à l'“enregistrement” de l'effet audio) puis est utilisée pour simuler l'acoustique de cette salle par simple convolution.

A l'heure où ces lignes sont écrites, plusieurs produits commerciaux proposent déjà des samplers d'effets un peu plus complets (on peut citer notamment Nebula de Acustica Audio et la technologie de Convolution Dynamique de Sintefex) qui permettent la simulation d'effets linéaires invariants ou variants et d'une catégorie restreinte d'effets non-linéaires. Si les simulations de systèmes linéaires variants ou invariants semblent maîtrisées, les simulations de systèmes non-linéaires donnent des résultats assez décevants et il semble qu'il y ait un sérieux manque dans ce domaine.

Qu'est-ce qu'un sampler d'effets

Un sampler d'effets doit pouvoir modéliser un ensemble le plus vaste possible de systèmes notamment :

- Des systèmes linéaires invariants (Reverb, Delay, filtres...)
- Des systèmes linéaires variants (Flanger, Phaser, Chorus, Tremolo...)
- Des systèmes non-linéaires invariants (tout type de distorsion) ou variants (cabine leslie avec non-linéarité...).

Ce rapport traite en majeure partie du cas non-linéaire invariant. Contrairement à la plupart des méthodes actuellement utilisées pour simuler des processeurs d'effets analogiques non-linéaires qui s'appuient le plus souvent sur une analyse du système, l'approche adoptée est la suivante : on cherche à identifier un système sans connaissance a priori sur le fonctionnement de ce système (si ce n'est

qu'il est non-linéaire et invariant). Il s'agit donc d'une identification de type boîte noire, qui va s'appuyer uniquement sur l'observation de l'entrée et de la sortie du système. Cette approche est bien plus complexe qu'une méthode d'analyse de système mais se veut beaucoup plus générale. De nombreux articles ont été écrits sur l'identification de systèmes non-linéaires mais seulement très peu dans le cadre de systèmes audio, or ceux-ci présentent des particularités (comme par exemple l'absence de composante continue dans les signaux¹) qui nécessitent souvent d'adapter les méthodes, quand cela est possible.

Ce rapport commence par présenter les méthodes existantes pour l'identification des systèmes linéaires invariants (chapitre 1) puis certaines méthodes pour l'identification des systèmes linéaires variants (chapitre 2). Il présente ensuite un cadre d'étude des systèmes non-linéaires invariants et notamment le formalisme de Volterra (chapitre 3). Plusieurs méthodes d'identification "classiques" sont alors présentées (chapitre 4). Le dernier chapitre est dédié à l'identification à l'aide de méthodes récentes utilisant notamment des espaces de Hilbert à noyaux reproduisants (chapitre 5). La première partie des annexes (annexe A) présente les techniques de simulation des systèmes non-linéaires, les parties suivantes contiennent le détail de calculs abordés au cours du rapport.

Ce rapport a été conçu en partie comme un catalogue de méthodes pour l'identification de systèmes non-linéaires. Toutes les méthodes présentées ont été implémentées en Matlab et testées sur des systèmes simulés voire lorsque cela était possible sur des systèmes réels, notamment un haut-parleur de mauvaise qualité et une pédale de distorsion de guitare. Les résultats de ces mesures sont présentés à la fin des sections correspondantes.

Tous les calculs présentés en annexe ont été réalisés par mes soins : ils ont permis à la fois de préciser les articles auxquels ils se rattachent et de mettre en lumière des points obscurs de ces articles souvent à peine abordés.

¹Il existe par exemple des méthodes d'identification qui linéarisent le système autour de points de fonctionnement ce qui impose une composante continue

Chapitre 1

Mesure de systèmes linéaires invariants

Système à temps continu : Un système linéaire invariant (on parle aussi de système temporellement homogène, c'est-à-dire qu'une translation temporelle de l'entrée se traduit par une translation temporelle de la sortie) à temps continu est entièrement décrit par sa réponse impulsionnelle. En notant $h(t)$ cette réponse impulsionnelle, la sortie d'un système linéaire invariant d'entrée $x(t)$ s'exprime alors par convolution de l'entrée avec la réponse impulsionnelle :

$$y(t) = h * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau$$

On peut alors définir la transformée de Laplace de la réponse impulsionnelle pour tout s complexe par :

$$H(s) = \int_{-\infty}^{+\infty} h(t)e^{-st} dt$$

La transformée de Laplace est une extension à tout le plan complexe de la transformée de Fourier qui est définie par :

$$H(\omega) = \int_{-\infty}^{+\infty} h(t)e^{-i\omega t} dt$$

Remarque : On note $H(\omega)$ la transformée de Fourier et non $H(i\omega)$ comme l'imposerait plutôt la transformée de Laplace.

Système à temps discret : Pour un système à temps discret, on a le même type de propriété. La sortie $y(n)$ d'un tel système de réponse impulsionnelle $h(n)$

pour une entrée $x(n)$ s'écrit :

$$y(n) = h * x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

On définit alors la transformée en z de la réponse impulsionnelle pour tout z complexe par :

$$H(z) = \sum_{k=-\infty}^{+\infty} h(k)z^{-k}$$

La transformée en z est une extension à tout le plan complexe de la transformée de Fourier discrète qui est définie par :

$$H(e^{i\omega}) = \sum_{k=-\infty}^{+\infty} h(k)e^{-i\omega k}$$

Remarque : On considère généralement des systèmes causaux, c'est-à-dire que les valeurs de la sortie ne dépendent que des valeurs passées de l'entrée. Cette propriété est équivalente à ce que la réponse impulsionnelle soit nulle pour $t < 0$ (système continu) ou bien $k < 0$ (système discret). On peut alors remplacer les bornes inférieures des intégrales et sommes ci-dessus par 0. Dans toute la suite on ne considérera que des systèmes causaux.

Il existe principalement trois méthodes de mesure des systèmes linéaires invariants :

- les méthodes “directes” qui utilisent des impulsions de Dirac
- les méthodes spectrales qui utilisent des sinusoides.
- les méthodes d'intercorrélation qui utilisent des signaux pseudo-aléatoires.

1.1 Impulsion de Dirac

L'idée de mesurer une réponse impulsionnelle à l'aide d'une impulsion de Dirac est la plus simple qui soit : la réponse impulsionnelle d'un système étant la réponse du système à une impulsion de Dirac, il suffit de nourrir le système avec une impulsion de Dirac pour récupérer en sortie sa réponse impulsionnelle.

Historiquement, c'est la première méthode qui a été utilisée pour la mesure de réponse impulsionnelle de salle : de la poudre à canon était utilisée pour générer des impulsions.

Cependant, cette méthode de mesure se heurte à de nombreux problèmes pratiques : d'une part il est impossible de générer une impulsion de Dirac parfaite (il faudrait être capable de générer une énergie positive sur un support temporel nul ce qui implique une puissance infinie). D'autre part, les systèmes physiques mesurés ne peuvent être considérés comme linéaires que dans une certaine limite d'amplitude de signaux et l'utilisation d'impulsion entraîne des niveaux élevés qui peuvent générer d'importantes non-linéarités dans le système mesuré ainsi que

dans les instruments de mesure (microphone, préamplificateur, convertisseurs...). Cette méthode est donc aujourd'hui assez peu utilisée.

1.2 Méthode spectrale

Les méthodes spectrales reposent sur le fait que les opérateurs linéaires invariants sont diagonalisés par la transformée de Fourier. Les signaux complexes $x(t) = e^{i\omega t}$ sont donc des vecteurs propres pour tout opérateur linéaire invariant L , la valeur propre associée étant la valeur de la transformée de Fourier H en ω :

$$L(e^{i\omega t}) = H(\omega)e^{i\omega t}$$

Pour un opérateur à réponse impulsionnelle réelle, la réponse fréquentielle est à symétrie hilbertienne et on obtient donc :

$$L(\sin(\omega t)) = |H(\omega)| \sin(\omega t + \angle H(\omega))$$

De même, pour un système à temps discret :

$$L(\sin(\omega n)) = |H(e^{i\omega})| \sin(\omega n + \angle H(e^{i\omega}))$$

En envoyant un sinus dans un système et en mesurant l'amplitude relative et la phase relative de la réponse, on peut donc calculer la réponse fréquentielle de ce système. Afin de couvrir toutes les fréquences, on utilise généralement des sweeps sinusoïdaux (cf. [16]) qui parcourent toute la bande de fréquence qu'on désire mesurer.

Ces méthodes sont cependant assez sensibles au bruit et ne sont adaptées qu'aux systèmes qui ont une réponse impulsionnelle courte (comme des filtres fréquentiels par exemple) : en effet, on utilise généralement des sweep sinusoïdaux qui sont des sinusoïdes dont la fréquence varie lentement dans le temps (cf. 4.4). Or, pour que le signal puisse être considéré localement comme une sinusoïde pure, le temps caractéristique de la variation de fréquence du signal doit être très inférieure à la taille de la réponse impulsionnelle du système (notée t_{RI}). Par exemple, si on utilise des sweeps logarithmiques comme ceux présentés à la section 4.4, la dérivée de la fréquence instantanée du sweep doit être, à tout moment, petite devant le rapport variation de fréquence considérée comme faible sur durée de la réponse impulsionnelle :

$$\omega'(t) \ll \frac{\Delta\omega}{t_{RI}}$$

En s'autorisant une variation de 50Hz par t_{RI} dans le haut du spectre, le calcul montre que le signal doit durer plus d'une heure pour pouvoir mesurer une réponse impulsionnelle de l'ordre de 10s dans le domaine du spectre audible. Cette méthode est donc difficilement utilisable pour mesurer des réponses impulsionnelles de salles.

1.3 Méthode d'intercorrélation

Les méthodes à base d'intercorrélation reposent sur des propriétés des bruits blancs. Un bruit blanc est un processus¹ stationnaire (au sens large) qui a pour autocorrélation une impulsion de Dirac (multiplié par une constante). Soit $X(t)$ un tel processus et soit L un opérateur linéaire stable de réponse impulsionnelle h . Le processus filtré s'écrit :

$$Y(t) = \int h(\tau)X(t - \tau)d\tau$$

L'intercorrélation du processus filtré et du processus d'origine peut donc s'écrire :

$$E(Y(t)X(t - t_0)) = E\left(\int h(\tau)X(t - \tau)X(t - t_0)d\tau\right) = \int h(\tau)E(X(t - \tau)X(t - t_0))d\tau$$

Comme $X(t)$ est un bruit blanc son autocorrélation vaut : $E(X(t - \tau)X(t)) = \sigma^2\delta(\tau)$, et donc :

$$E(Y(t)X(t - t_0)) = \sigma^2h(t_0)$$

On peut écrire de même pour un système à temps discret :

$$E(Y(n)X(n - n_0)) = \sigma^2h(n_0)$$

Si on dispose d'une réalisation d'un bruit blanc ainsi que de la sortie du système linéaire pour ce bruit, on peut alors estimer l'intercorrélation de l'entrée et de la sortie et par conséquent, on peut estimer la réponse impulsionnelle h du système.

Tout le problème réside donc dans l'obtention de réalisations de tels bruits blancs discrets dont l'autocorrélation soit le plus proche possible d'une impulsion de Dirac.

On utilise généralement des signaux pseudo-aléatoires, comme par exemple des séquences MLS (cf. [22]) qui possèdent la propriété d'avoir une autocorrélation très proche d'un Dirac et dont le résidu est majoré par l'inverse de la longueur de la séquence.

Plus récemment d'autres signaux ont été proposés comme les codes de Golay (cf. [9]). La méthode est légèrement différente et nécessite deux mesures : les codes de Golay se présentent sous la forme de fonctions complémentaires dont la somme des intercorrélations est une impulsion parfaite.

Ce type de méthode a l'avantage d'être particulièrement résistant au bruit et est adapté même pour des réponses impulsionnelles longues. Il se révèle cependant plus sensible aux non-linéarités du système que les méthodes spectrales.

¹On réserve parfois l'expression bruit blanc à une réalisation d'un tel processus plutôt qu'au processus lui-même

Chapitre 2

Mesure de systèmes linéaires variants

Ce chapitre présente une introduction à l'identification de systèmes linéaires variants dans le temps. Les méthodes présentées ici sont particulièrement simples et nécessiteraient une étude plus poussée mais l'objet principal du stage était l'identification des systèmes non-linéaires invariants. Les méthodes présentées ont cependant l'avantage de fonctionner correctement et dans certaines limites d'être simulables en temps réel.

2.1 Modèle

Intuitivement, on peut définir les systèmes linéaires variants comme des systèmes qui se comportent localement comme des systèmes linéaires : il s'agit en fait de systèmes linéaires dont la réponse impulsionnelle varie lentement dans le temps. Cette variation peut être de plusieurs natures :

- périodique (effet à oscillateur basse fréquence : flanger, phaser, chorus...)
- contrôlée par un utilisateur (par exemple une pédale Wah Wah).
- contrôlée par descripteur audio (comme par exemple l'enveloppe du signal pour un compresseur).

Dans le cas général, on peut supposer que la réponse impulsionnelle h dépend d'un paramètre $\theta(n) \in \Omega \subset \mathbb{R}^q$ qui dépend lui-même du temps. La sortie d'un tel système s'écrit alors :

$$y(n) = \sum_{k=0}^{\infty} h(\theta(n), k)x(n-k)$$

Il faut remarquer que la première variable de la fonction h est une variable continue. Or en règle générale, dans le cadre de système mesuré, on ne dispose pas d'expression explicite de h par rapport à la première variable mais d'un ensemble de valeurs discrètes : on dispose généralement d'un ensemble de réponses

impulsionnelles $(g_i(n))_{i \in \{1 \dots N\}}$ avec $h(\theta_i, k) = g_i$. Il est donc nécessaire d'avoir un moyen de reconstruire correctement les réponses impulsionnelles manquantes.

2.2 Interpolation paramétrique

Dans un premier temps, on se place dans le cas simple où le paramètre agissant sur la variation de la réponse impulsionnelle est contrôlée par un utilisateur (via un ensemble de potentiomètres par exemple) : on peut alors retirer la dépendance temporelle dans le modèle, puisqu'on peut fixer le paramètre autant qu'on veut.

On effectue alors des mesures du système à paramètre fixé : celui-ci se comporte donc comme un système linéaire et les réponses impulsionnelles peuvent donc être obtenues par une des méthodes présentées dans le chapitre 1. En notant $\eta \in \Omega \subset \mathbb{R}^q$ le paramètre (Ω est l'ensemble des paramètres possibles), on dispose alors d'un ensemble de mesures réalisées pour un ensemble η_1, \dots, η_N de valeurs du paramètre qui nous permettent de calculer les réponses impulsionnelles $h(\eta_1, n), \dots, h(\eta_N, n)$. À partir de cet ensemble de mesures, on voudrait être capable de bien approximer $h(\eta, n)$ pour tout $\eta \in \Omega$.

2.2.1 Cas général

Dans le cas général, on ne fait pas d'hypothèses supplémentaires sur le fonctionnement du système.

Une solution simple consiste à interpoler multilinéairement les réponses mesurées : cette solution fonctionne généralement assez bien quand on dispose d'un très grand nombre de réponses mesurées et que les réponses voisines sont très proches les unes des autres. Mais cela n'est généralement possible que lorsque la dimension du paramètre est très petite et dès que cette dimension dépasse 2 ou 3 le nombre de réponses à mesurer pour avoir un résultat correct devient beaucoup trop important et il est donc nécessaire d'utiliser des outils plus évolués.

Une approche envisageable (qui reste une piste à explorer) serait de considérer le problème comme un problème de régression générale entre les valeurs du paramètre η et la distribution de réponse impulsionnelle $h(\eta, n)$. Le problème pourrait alors être résolu en utilisant des méthodes de régressions non-linéaires telles que celles abordées dans un autre contexte dans la section 5.

2.2.2 Hypothèses supplémentaires : cas de la pédale Wah Wah

Si on s'autorise des hypothèses supplémentaires sur le fonctionnement du système on peut utiliser un modèle beaucoup plus approprié et l'interpolation entre les mesures peut se faire de manière beaucoup plus intelligente et beaucoup plus simplement. Dans le cadre de l'écoute binaurale utilisée avec des sources mobiles, par exemple, il est nécessaire d'interpoler intelligemment les HRIR mesurées afin de créer une sensation de continuité (cf. [15] et [14]).

On présente ici le cas simple de la pédale Wah Wah : il s'agit d'un effet très utilisé par les guitaristes, qui consiste en un filtre résonant, dont la fréquence de

résonance est contrôlée par une pédale. Le modèle utilisé s'inspire notamment des techniques d'interpolation de filtres présentées dans [27]

Cet effet se modélise assez bien comme un filtre ARMA à deux pôles (complexes conjugués) : La partie AR (notée A) correspond à la partie résonante et devra être interpolée intelligemment alors que la partie MA (notée B) qui varie assez peu pourra être simplement interpolée linéairement. Les coefficients du filtre ARMA peuvent être obtenus facilement à partir des réponses impulsionnelles en utilisant la méthode de Prony (cf. [17]).

Le paramètre η ici considéré est donc la position de la pédale. Il s'agit d'un paramètre réel (monodimensionnel) qu'on peut normaliser entre 0 et 1. On suppose qu'on a fait K mesures pour des valeurs du paramètre uniformément réparties sur $[0, 1]$, c'est-à-dire pour $\eta_1 = 0, \eta_2 = 1/(K-1), \dots, \eta_K = 1$.

On obtient alors un ensemble de filtres estimés : $(B_{\eta_1}, A_{\eta_1}), \dots, (B_{\eta_K}, A_{\eta_K})$

Pour la partie AR, comme on connaît la variable (la fréquence de résonance) sur laquelle agit le paramètre, on va chercher à ce que la variable varie linéairement par morceaux (c'est-à-dire linéairement entre deux mesures) avec le paramètre de contrôle : cette hypothèse simplificatrice est suffisante si K est assez grand. Or la fréquence de résonance est déterminée par l'argument des pôles du filtre AR. Ainsi, au lieu d'interpoler directement les coefficients, on va plutôt interpoler les pôles : comme il n'y a que deux pôles complexes conjugués par AR mesuré, il ne se pose pas de problème d'appariement des pôles. En notant r_{η_i} le pôle de A_{η_i} de partie imaginaire positive, on peut alors estimer la valeur du pôle interpolé en fonction du paramètre :

$$|\hat{r}_\eta| = (K\eta - E(K\eta)) |r_{E(K\eta)}| + (1 - K\eta + E(K\eta)) |r_{E(K\eta)+1}|$$

$$\angle \hat{r}_\eta = (K\eta - E(K\eta)) \angle r_{E(K\eta)} + (1 - K\eta + E(K\eta)) \angle r_{E(K\eta)+1}$$

La valeur de l'argument du pôle qui correspond à la fréquence de résonance est interpolée linéairement, ce qui assure une bonne cohérence entre la variable contrôlée et le paramètre de contrôle. Ainsi, on est capable de modéliser le système pour toutes les valeurs du paramètre de contrôle.

Remarque : Comme un modèle ARMA est utilisé, il se pose la question de la stabilité pour la partie AR. Comme l'opération d'interpolation se fait sur le module des pôles, il est évident que si les pôles estimés à l'aide de la méthode de Prony sont de module inférieur à 1 alors les pôles interpolés le seront aussi et donc la partie AR sera stable. Or, les réponses impulsionnelles d'effets comme la WahWah ont des comportements exponentiels décroissants, et par conséquent les pôles estimés sont de module inférieur à 1 (sauf erreur trop importante lors de la mesure) et par conséquent la stabilité est assurée.

Résultats : Nous avons implémenté la méthode présentée en temps réel sous forme de plugin VST. Les résultats audio sont plutôt convaincants. La méthode présentée qui ne s'applique qu'à une catégorie restreinte d'effets fonctionne bien du fait même qu'on utilise des connaissances importantes sur la structure de l'effet. Il serait cependant intéressant de pouvoir généraliser la méthode à un paramètre

de contrôle multidimensionnel, tout en extrayant de façon automatique la relation entre ce paramètre de contrôle et les réponses impulsionnelles.

2.3 Mesure des systèmes périodiques

Dans le cas de systèmes périodiques, on n'a plus un accès direct au paramètre de contrôle : on fait alors la seule supposition que le système est périodique. La sortie de tels systèmes peut alors s'écrire :

$$y(n) = \sum_{k=0}^{\infty} h(\theta(n), k)x(n - k)$$

où $\theta(n)$ est une fonction périodique du temps. On peut donc associer à l'instant n une réponse impulsionnelle $h(\theta(n), \cdot)$ qu'on appellera *réponse impulsionnelle locale*.

De tels systèmes sont très courants dans le domaine de l'audio. On peut citer notamment :

- le flanger : filtre en peigne harmonique dont la position des zéros varie périodiquement
- le phaser : analogue au flanger sauf que les zéros du filtre ne sont pas en disposition harmonique.
- les haut-parleurs rotatifs
- certains chorus simples (à un seul oscillateur) : fonctionnement proche du flanger avec des paramètres différents
- le tremolo : variation périodique du volume
- ...

Pour plus d'informations, tous ces effets sont présentés en détail dans [26].

2.3.1 Mesure de la période

La période des systèmes présentés plus haut est la plupart du temps réglable mais n'est généralement pas connue, ou en tout cas pas avec précision : Souvent (notamment sur les pédales d'effets destinées à la guitare), le potentiomètre de réglage de la période n'est gradué qu'avec une échelle arbitraire qui ne permet absolument pas de déterminer celle-ci.

Une première mesure est donc nécessaire pour déterminer cette période.

Une méthode simple consiste à utiliser un train d'impulsions de durée de l'ordre de quelques périodes. La réponse à ce type de signal est alors quasiment périodique et de même période que celle de variation du système. On peut alors déterminer la période en recherchant le maximum de l'autocorrélation de la réponse. Cette méthode permet d'estimer la période avec une précision égale à l'inverse de l'écart temporel entre deux impulsions dans le train d'impulsions en entrée : on peut améliorer la précision en utilisant plusieurs mesures et en utilisant une interpolation parabolique du point maximum de l'autocorrélation et de ses voisins et en calculant la position du maximum de la parabole.

2.3.2 Mesures des systèmes périodiques

On se limite aux systèmes périodiques dont le temps caractéristique de variation est petit devant la longueur des réponses impulsionnelles locales. Comme les systèmes périodiques dispose généralement d'un réglage de fréquence, on peut la plupart du temps se placer dans cette hypothèse en prenant une période très longue.

On fait également l'hypothèse que le comportement du système reste le même quelle que soit la valeur de réglage de la période, c'est-à-dire que le contrôleur qui règle la fréquence n'influe pas sur les réponses impulsionnelles locales.

Dans ce cadre, il est assez facile de mesurer un grand nombre de réponses impulsionnelles locales. On peut par exemple utiliser un nouveau train d'impulsions de longueur égale à une période de l'oscillateur de l'effet mesuré. La période de ce train d'impulsions doit être choisie plus longue que la plus grande des réponses impulsionnelles locales. Il est possible d'affiner cette mesure en effectuant une mesure sur plusieurs périodes de l'oscillateur, en décalant la phase du train d'impulsions pour chaque période.

On obtient alors directement les réponses impulsionnelles locales comme le montre la figure 2.1 qui est la réponse d'un flanger à un train d'impulsions.

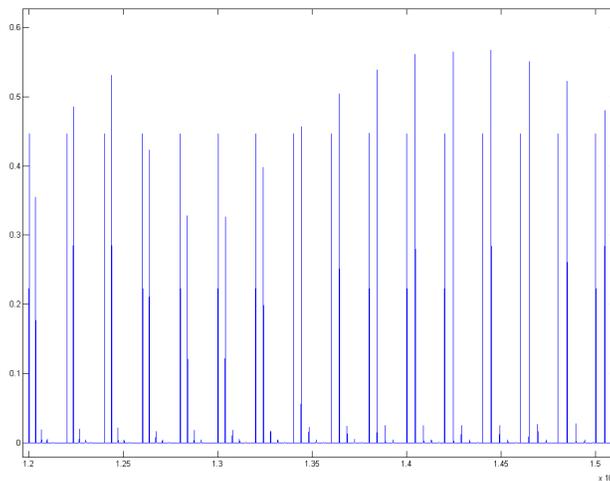


FIG. 2.1 – Réponse d'un flanger à un train d'impulsions

Comme on dispose alors d'un grand nombre de réponses impulsionnelles, très proches les unes des autres, une simple interpolation linéaire entre ces réponses suffit en général à donner un résultat très convaincant sur le plan perceptif.

2.3.3 Systèmes multipériodiques et apériodiques

Certains systèmes qui disposent de plusieurs oscillateurs de périodes différentes peuvent être qualifiés de multipériodiques. Lorsque les périodes de ces différents oscillateurs sont toutes en rapport rationnel les unes avec les autres, tous les oscillateurs ont une période commune, et le système est alors périodique et peut être traité comme tel (si cette période commune n'est pas trop grande). Sinon le système n'est pas périodique et il semble que les méthodes précédentes soient alors inefficaces.

Cependant, quand le nombre d'oscillateurs devient important (à partir de 4 oscillateurs), et que ces oscillateurs ne sont pas synchronisés, l'effet n'est pas perçu comme ayant des aspects se répétant périodiquement : on perçoit généralement une sensation d' "aléatoire", comme c'est le cas dans le chorus.

Ainsi on peut sensiblement appliquer la même méthode que pour les systèmes périodiques : on calcule une pseudo-période qui soit assez longue pour déterminer la plage dans laquelle on va mesurer nos réponses impulsionnelles locales, puis à la simulation, on fait comme si le système était périodique, mais en retournant le sens du temps une période sur deux afin d'assurer une certaine continuité à la rupture. Cette technique est directement issue des lecteurs d'échantillons sonores à boucle traditionnels. On utilise en effet un bouclage avant/arrière : un échantillon est lu alternativement dans le bon sens puis dans le sens inverse, ce qui assure une meilleure continuité temporelle et fréquentielle que de boucler l'échantillon toujours dans le même sens.

2.3.4 Mise en oeuvre et résultats

Les méthodes d'identification présentées plus haut ont été implémentées sous Matlab et un simulateur temps réel (un lecteur de samples d'effet) a été réalisé sous forme de plugin VST.

Plusieurs systèmes ont été mesurés : nous avons d'abord mesuré des effets logiciels périodiques dont nous pouvions contrôler précisément la phase de l'oscillateur afin de pouvoir synchroniser l'oscillateur de l'effet original et le pseudo-oscillateur de l'effet simulé, et donc de pouvoir comparer objectivement les signaux simulés des signaux originaux.

Nous avons donc mesuré le plugin VST *Blue Cat's Flanger* pour des fréquences faibles de l'oscillateur (de l'ordre de $0.2Hz$). Dans la figure 2.2 est comparé un signal simulé au signal original correspondant traité par l'effet. Comme la différence est peu visible, le résiduel (différence entre les deux signaux) a été également tracé (en vert). Il est apparu lors de nos mesures que lorsque la période de l'oscillateur du système mesuré (même inconnue) est très grande devant la longueur des réponses impulsionnelles locales, le résiduel reste faible et pour la plupart de nos mesures quasiment inaudibles (le nombre de personnes test reste cependant trop faible pour pouvoir affirmer qu'il soit réellement inaudible et il faudrait faire une étude perceptive plus poussée avec plus d'échantillons et de personnes).

Nous avons alors voulu tester l'efficacité de notre méthode d'identification en mesurant un système analogique. Ce type de mesure comporte en effet un certain

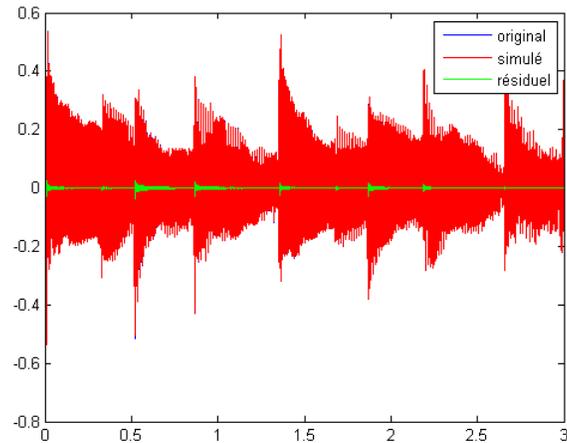


FIG. 2.2 – Comparaison du flanger simulé et de l'original

nombre de défauts : bruit de fond du système, imperfection des convertisseurs numérique/analogique.

Nous avons mesuré une pédale d'effet flanger de guitare (Ibanez FL7 cf. figure 2.3). Comme l'oscillateur de cette effet n'est pas synchronisable, il est difficile de comparer objectivement les signaux simulés et les signaux traités par le processeur d'effet. Les comparaisons subjectives à l'oreille ont cependant été très convaincantes

Nous avons ensuite mesuré un chorus logiciel (Chorus CH-2) à plusieurs oscillateurs en utilisant la technique du bouclage avant/arrière : sur les phases de lecture avant synchronisé, les résultats sont similaires à ceux du flanger (en terme de qualité à l'écoute). Le passage de la lecture avant à arrière ne peut être évalué que subjectivement à l'oreille vu qu'il s'agit d'une astuce perceptive. Les signaux de tests ont été présentés à plusieurs personnes (3 personnes) sans qu'aucune de celles-ci ne perçoive de rupture. Une étude perceptive plus détaillée pourrait cependant être envisagée.



FIG. 2.3 – Flanger analogique mesuré

Chapitre 3

Cadre général des systèmes non-linéaires

La classe des systèmes non-linéaires étudiés est l'ensemble des systèmes qui peuvent être décrits par une série de Volterra. Cette classe de systèmes permet de décrire un très large ensemble de systèmes physiques réels. Elle exclut cependant certains comportements produits par des non-linéarités moins régulières (qui incluent par exemple des discontinuités) : on peut citer notamment l'hystérésis ou les phénomènes chaotiques qui ne peuvent pas être modélisés par les séries de Volterra. On s'est limité dans cette étude aux systèmes non-linéaires invariants. Cependant, il existe aussi, dans la littérature, des modèles de systèmes non-linéaires variants comme par exemple dans [13].

3.1 Série de Volterra

3.1.1 Définition des séries de Volterra

Les séries de Volterra permettent d'étendre à la fois la convolution et le développement en série de Taylor des fonctions non-linéaires statiques et permettent donc de décrire une très large classe de systèmes non-linéaires à mémoire souvent nommée *systèmes faiblement non-linéaires*.

Dans le cas général, un système de Volterra à temps continu, d'entrée $x(t)$ a pour sortie :

$$y(t) = \sum_{k=0}^{\infty} \int_0^{\infty} \int_0^{\infty} \dots \int_0^{\infty} h_k(\tau_1, \tau_2, \dots, \tau_k) x(t - \tau_1) x(t - \tau_2) \dots x(t - \tau_k) d\tau_1 d\tau_2 \dots d\tau_k$$

Pour tout k , la fonction $h_k(\tau_1, \tau_2, \dots, \tau_k)$ est appelée noyau d'ordre k de la série de Volterra. On peut alors noter $\{h_k\}_{k \in \mathbb{N}}$ le système associé à la série.

On a introduit un terme constant (terme $k = 0$) dans l'expression ci-dessus : pourtant, dans le cadre de signaux audio, la quasi-totalité des systèmes n'est pas prévue pour recevoir ou générer une composante constante. Ils sont d'ailleurs le

plus souvent munis de filtres coupe-bas qui permettent de supprimer celle-ci afin d'éviter de les endommager. C'est pourquoi dans la suite de ce rapport, nous ne prendrons généralement pas en compte cette composante continue (la somme commence alors à $k = 1$).

On peut également définir la notion de séries de Volterra pour des systèmes à temps discret. Ainsi pour un système discret de Volterra d'entrée $x(n)$, la sortie s'écrit :

$$y(n) = \sum_{k=1}^{\infty} \sum_{q_1, q_2, \dots, q_k=0}^{\infty} h_k(q_1, q_2, \dots, q_k) x(n - q_1) x(n - q_2) \dots x(n - q_k)$$

Pour des systèmes discrets, on considère parfois des systèmes de Volterra à mémoire finie c'est-à-dire des systèmes pour lesquelles la sortie à un instant donné ne dépend que d'un nombre fini d'échantillons de l'entrée. On peut alors écrire :

$$y(n) = \sum_{k=1}^{\infty} \sum_{q_1, q_2, \dots, q_k=0}^{M-1} h_k(q_1, q_2, \dots, q_k) x(n - q_1) x(n - q_2) \dots x(n - q_k)$$

la plus petite valeur de M permettant la description précédente sera appelée mémoire du système (cependant par abus de langage on appellera parfois mémoire n'importe quelle valeur qui permet une telle description quand celle-ci existe). On peut également définir de manière analogue une mémoire pour les systèmes continus.

On considère aussi parfois des séries de Volterra tronquées : la première sommation est tronquée à un entier P . On écrit alors :

$$y(n) = \sum_{k=1}^P \sum_{q_1, q_2, \dots, q_k=0}^{M-1} h_k(q_1, q_2, \dots, q_k) x(n - q_1) x(n - q_2) \dots x(n - q_k)$$

Le plus petit entier P qui permet cette description pour un système donné est appelé ordre du système. Comme il s'agit le plus souvent d'une approximation, la valeur de P va dépendre d'un critère d'approximation.

3.1.2 Transformée de Laplace et transformée en z

Transformée de Laplace : Par analogie à la transformée de Laplace définie pour des fonctions unidimensionnelles (cf. chapitre 1), on peut définir une transformée de Laplace multidimensionnelle de la façon suivante :

$$H_p(s_1, \dots, s_p) = \int_0^{+\infty} h_p(t_1, \dots, t_p) e^{-s_1 t_1} \dots e^{-s_p t_p} dt_1 \dots dt_p$$

On a donc une définition de la transformée de Laplace pour chaque noyau d'un système de Volterra.

Transformée en z : De même, pour un système de Volterra discret, on peut définir pour chaque noyau une transformée en z multidimensionnelle par :

$$H_p(z_1, \dots, z_p) = \sum_{k_1 \dots k_p=0}^{+\infty} h_p(k_1, \dots, k_p) z_1^{-k_1} \dots z_p^{-k_p}$$

3.1.3 Unicité de la description d'une série de Volterra

La description d'un système par une série de Volterra n'est pas unique. En effet si $h_p(t_1, t_2, \dots, t_p)$ est le noyau d'ordre p d'une série de Volterra, on peut le remplacer par $h_p(t_{\sigma(1)}, t_{\sigma(2)}, \dots, t_{\sigma(p)})$ ou σ est une permutation de $\{1, 2, \dots, p\}$ et le système décrit reste le même. Par conséquent, il est souvent utile d'imposer une structure au noyau afin d'en garantir l'unicité. Il existe plusieurs types de telles structures, notamment les noyaux symétriques qui sont tels que $h_p(t_{\sigma(1)}, t_{\sigma(2)}, \dots, t_{\sigma(p)}) = h_p(t_1, t_2, \dots, t_p)$ pour toute permutation σ et les noyaux triangulaires qui sont par exemple tels que $h_p(t_1, t_2, \dots, t_p)$ est non nul si et seulement si $t_1 \leq t_2 \leq \dots \leq t_p$. Ces structures réduisent également le nombre de coefficients à calculer dans le cas d'une série de Volterra à temps discret tronquée : en effet, un noyau d'ordre p et de mémoire M sans structure imposée possède M^p coefficients, alors qu'un noyau triangulaire ne possède que $\binom{M-p+1}{p}$ coefficients non nuls. Le nombre de coefficients d'une série de Volterra tronquée est souvent beaucoup trop important pour que ceux-ci puissent être estimés. Pour un système d'ordre P et de mémoire M , le nombre de coefficients à estimer est en effet égal à :

$$L(P, M) = \sum_{p=1}^P \binom{M-p+1}{p}$$

Par exemple pour un système d'ordre 7 et de mémoire 100 échantillons, le nombre de coefficients à estimer est supérieur à 10^{10} , ce qui limite considérablement les méthodes d'estimation des noyaux de Volterra dans leur ensemble. C'est pourquoi de nombreuses méthodes d'identification des systèmes non-linéaires reposent sur des systèmes simplifiés qui ont beaucoup moins de paramètres à estimer.

3.2 Systèmes simplifiés

L'importante complexité des systèmes de Volterra dans un cadre général amène souvent à faire des hypothèses simplificatrices sur les systèmes considérés afin de réduire considérablement le nombre de coefficients à identifier. On présente dans cette section deux modèles simplifiés :

- les modèles *schéma-bloc*
- la convolution dynamique

3.2.1 Systèmes de type *schéma-bloc*

Un modèle courant qui reste assez général et surtout très intuitif est d'utiliser des systèmes de type *schéma-bloc*¹. Les systèmes de type *schéma-bloc* sont construits comme des produits, des sommes ou des cascades des systèmes élémentaires suivants :

- filtres linéaires (à mémoire).
- fonctions non-linéaires statiques (sans mémoire).

On se limite dans ce rapport aux systèmes sans bouclage : ceux-ci sont notamment présentés dans [3].

Il existe des méthodes d'identification de structure ([5]) qui permettent de déterminer de quel type est un système donné. Ce chapitre présente un ensemble de systèmes par bloc couramment rencontrés dans la littérature dans le cadre de l'identification de systèmes non-linéaires et donne la représentation de Volterra pour chacun de ces systèmes.

Les modèles les plus fréquemment utilisés à partir desquels nous avons travaillé sont présentés dans le tableau récapitulatif 3.2.1. Pour ce tableau, on note *g les blocs de type filtre linéaire et F[.] les blocs de type fonction non-linéaire statique. Pour le calcul de la série de Volterra équivalente, on suppose la fonction F développable en série entière :

$$F(x) = \sum_{p=1}^{\infty} \alpha_p x^p$$

La figure 3.2 représente les noyaux d'ordre 2 des différents systèmes.

3.2.2 Convolution dynamique

La société Sintefex a mis en place une méthode d'identification et de simulation d'une certaine catégorie de systèmes audio non-linéaires. Cette méthode repose sur le modèle de la convolution dynamique. Elle a été déposée et son utilisation commerciale est donc soumise à conditions. Il semblait néanmoins intéressant de faire le rapprochement entre la théorie de la convolution dynamique et le formalisme de Volterra et de tester cette méthode pour la comparer aux autres méthodes existantes.

Modèle

Les systèmes décrits par la convolution dynamique étendent la convolution linéaire en rendant la réponse impulsionnelle du système dépendante à la fois du temps (comme une réponse impulsionnelle classique) et de la valeur de l'entrée. La sortie d'un tel système est donc donnée par (cf. [12]) :

$$y(t) = \int_0^{\infty} h(\tau, x(t-\tau))x(t-\tau)d\tau$$

¹On rencontre généralement en anglais l'expression *block diagram*

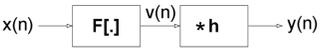
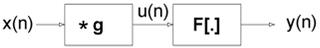
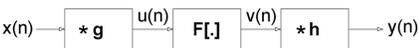
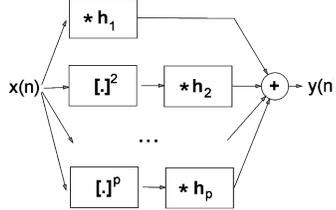
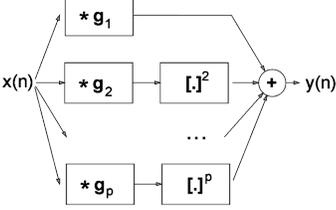
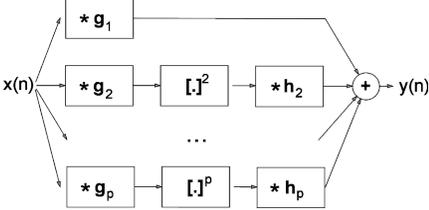
Nom du système	Diagramme	Noyaux de Volterra $f_p(n_1, \dots, n_p)$
Hammerstein		$\alpha_p h(n_1) \delta(n_1 - n_2) \dots \delta(n_1 - n_p)$
Wiener		$\alpha_p g(n_1) g(n_2) \dots g(n_p)$
Wiener-Hammerstein		$\alpha_p \sum_{m=0}^{\infty} h(m) g(n_1 - m) \dots g(n_p - m)$
Hammerstein généralisé		$h_p(n_1) \delta(n_1 - n_2) \dots \delta(n_1 - n_p)$
Wiener généralisé		$g_p(n_1) g_p(n_2) \dots g_p(n_p)$
S_M		$\alpha_p \sum_{m=0}^{\infty} h_p(m) g_p(n_1 - m) \dots g_p(n_p - m)$

FIG. 3.1 – Systèmes schéma-bloc les plus couramment utilisés

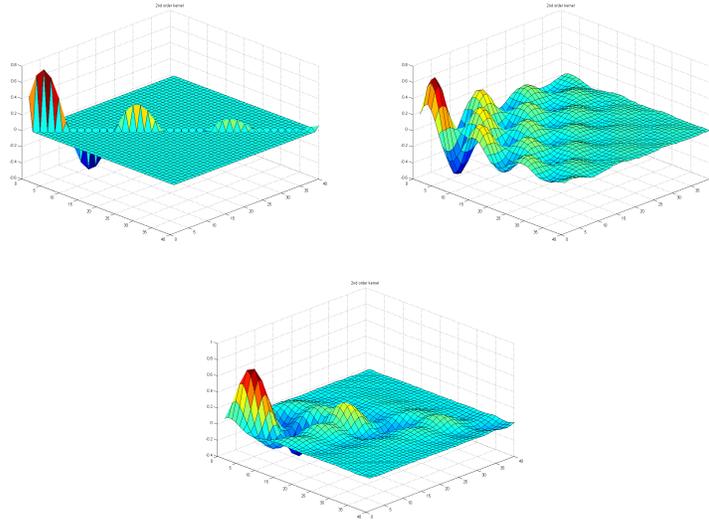


FIG. 3.2 – Noyaux d'ordre 2 des systèmes *schéma-bloc*. De gauche à droite et de haut en bas : Hammerstein, Wiener, Wiener-Hammerstein

Le fait d'utiliser l'entrée retardée $x(t-\tau)$ et non l'entrée instantanée $x(t)$ dans l'expression de $h(\tau, x(t-\tau))$ n'a pas de réel fondement scientifique : cela permet en fait par une astuce de calcul de garder un coût faible pour la simulation de tels systèmes (cf. [12]).

Equivalence avec les systèmes de Volterra

Dans le cas où $h(\tau, x)$ est tel que la fonction $f_\tau(x) = h(\tau, x)$ est développable en série entière pour tout τ , on peut écrire :

$$h(\tau, x) = \sum_{k=0}^{\infty} h_k(\tau) x^k$$

La sortie du système s'écrit alors :

$$y(t) = \int_0^{\infty} \sum_{k=0}^{\infty} h_k(\tau) x(t-\tau)^k x(t-\tau) d\tau$$

Sous de bonnes hypothèses :

$$y(t) = \sum_{k=0}^{\infty} \int_0^{\infty} h_k(\tau) x(t-\tau)^{k+1} d\tau$$

Il s'agit donc d'un système de type Hammerstein généralisé.

Dans le cas général, les systèmes définis par la convolution dynamique ne constituent cependant pas un sous-ensemble des systèmes de Volterra : la fonction $h(\tau, x)$ pourrait en effet avoir d'importantes discontinuités et créer des phénomènes complexes comme de l'hystérésis qui n'est pas modélisable par les séries de Volterra. Néanmoins, les méthodes de simulation utilisent une interpolation entre les points mesurés, ce qui implique qu'on suppose une certaine régularité sur $h(\tau, x)$.

Chapitre 4

Méthodes d'identification

Plusieurs types de signaux peuvent être utilisés pour mesurer les systèmes de Volterra. Les signaux les plus couramment utilisés sont :

- des séries d'impulsions de Dirac
- des sinusoïdes (et des sweeps sinusoïdaux)
- des signaux pseudo-aléatoires

La première section de ce chapitre présente une méthode de séparation des ordres. Ce n'est pas à proprement parler une méthode d'identification mais elle est utilisée comme première phase de plusieurs méthodes d'identification. Les parties suivantes présentent différentes méthodes d'identification pour chacun des types de signaux présentés plus haut.

4.1 Séparation des ordres

4.1.1 Méthodes

Un des problèmes fréquemment rencontrés dans les méthodes d'identification des noyaux de Volterra d'un système est le problème de la séparation des ordres : Pour estimer les noyaux à partir de la réponse d'un système à un signal donné on commence généralement par séparer les composantes dues aux différents ordres dans la réponse. Pour un système de noyaux $\{h_p\}$, la réponse $y(n)$ à un signal $x(n)$ est donnée par :

$$y(n) = \sum_{p=1}^{\infty} w_p(n) \quad (4.1)$$

avec :

$$w_p(n) = \sum_{k_1, k_2, \dots, k_p=0}^{\infty} h_p(k_1, k_2, \dots, k_p) x(n - k_1) x(n - k_2) \dots x(n - k_p) \quad (4.2)$$

L'équation (4.2) montre que l'application qui, à une entrée $x(n)$, associe la composante d'ordre p $w_p(n)$ de la réponse du système est homogène d'ordre p , i.e :

$$\begin{aligned} w_{p,\alpha}(n) &= \sum_{k_1, k_2, \dots, k_p=0}^{\infty} h_p(k_1, k_2, \dots, k_p) \alpha x(n - k_1) \alpha x(n - k_2) \dots \alpha x(n - k_p) \\ &= \alpha^p w_p(n) \end{aligned}$$

A partir de cette considération, on peut séparer les ordres par une méthode présentée dans [11] : l'idée est d'appliquer à l'entrée du système le même signal à des amplitudes différentes : l'homogénéité des composantes de chaque ordre étant d'ordre différent, l'influence de la variation d'amplitude sera différente pour tous les ordres. On applique donc les signaux $\alpha_i x(n)$ avec $i = 1..M$ et les α_i sont distincts les uns des autres et on note les réponses associées $y_i(n)$ et en utilisant la décomposition en composantes des différents ordres de la réponse $y(n)$ à $x(n)$ donnée par (4.1), on obtient :

$$y_i(n) = \sum_{p=1}^{\infty} \alpha_i^p w_p(n) + e_i(n) \quad (4.3)$$

$e_i(n)$ représente ici un bruit non issu de la modélisation en série de Volterra du système mais qu'on observe nécessairement lors d'une mesure (ou même à un niveau moindre dans le cas d'un système simulé numériquement pour lequel apparaît un bruit de quantification).

Les noyaux de la série de Volterra convergeant en norme vers 0, on peut considérer que la contribution des noyaux devient négligeable (notamment quand cette contribution est de l'ordre de celle du bruit de mesure pour les amplitudes maximales utilisées) à partir d'un certain ordre P . On peut alors inclure la contribution des ordres supérieurs à P dans le bruit e_i et écrire sous forme matricielle l'équation (4.3), qui fait apparaître une matrice de Vandermonde :

$$\begin{pmatrix} y_1(n) \\ y_2(n) \\ \vdots \\ y_M(n) \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_1^2 & \dots & \alpha_1^P \\ \alpha_2 & \alpha_2^2 & \dots & \alpha_2^P \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \alpha_M & \alpha_M^2 & \dots & \alpha_M^P \end{pmatrix} \begin{pmatrix} w_1(n) \\ w_2(n) \\ \vdots \\ w_P(n) \end{pmatrix} + \begin{pmatrix} e_1(n) \\ e_2(n) \\ \vdots \\ e_M(n) \end{pmatrix} \quad (4.4)$$

La séparation des composantes de chaque ordre de $y(n)$ revient donc à résoudre un problème du type moindres carrés. Si on prend un nombre d'amplitudes différentes suffisant ($M \geq P$), la solution minimisant la valeur du bruit e en norme quadratique est donnée par :

$$\begin{pmatrix} \hat{w}_1(n) \\ \hat{w}_2(n) \\ \dots \\ \hat{w}_P(n) \end{pmatrix} = A^\dagger \begin{pmatrix} y_1(n) \\ y_2(n) \\ \dots \\ y_M(n) \end{pmatrix}$$

où A est la matrice de Vandermonde de l'équation (4.4) et \dagger désigne la pseudo-inverse.

La méthode de la séparation des ordres est illustrée par les figures 4.1 et 4.2 : un sweep logarithmique est envoyé dans un système d'ordre 5 (pour plusieurs amplitudes différentes), le spectrogramme de la réponse est présenté dans la figure 4.1 et les spectrogrammes des différents ordres séparés dans la figure 4.2. On peut voir notamment que les ordres impairs génèrent des harmoniques impairs, et que les ordres pairs génèrent des harmoniques pairs : par exemple l'ordre 5 seul génère des harmoniques d'ordre 1, 3 et 5. Pour plus de détails sur les sweeps logarithmiques et le comportement des systèmes de Volterra en réponse à ce type de signal, se référer à la section 4.4.

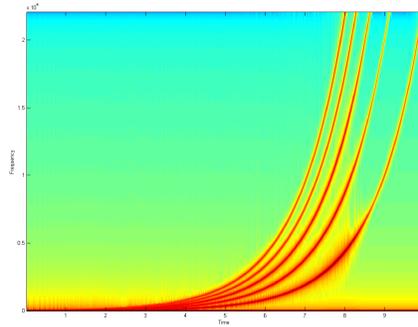


FIG. 4.1 – Spectrogramme de la réponse d'un système d'ordre 5 à un sweep logarithmique

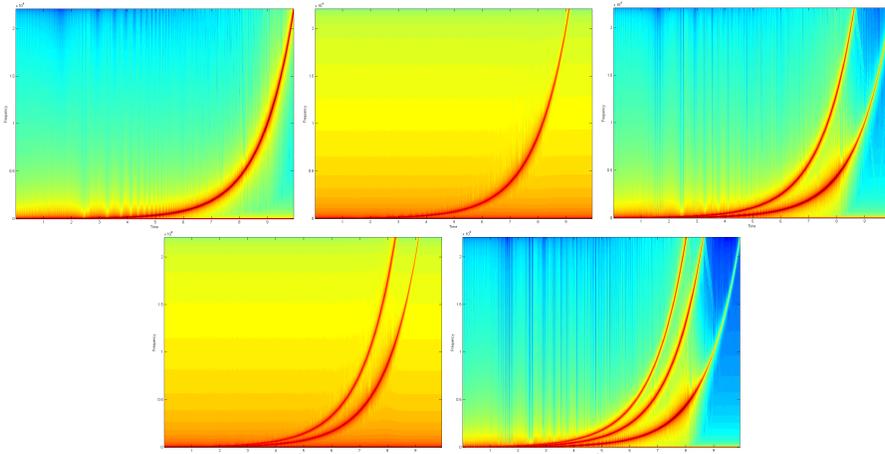


FIG. 4.2 – Spectrogrammes des différents ordres séparés (de 1 à 5)

4.1.2 Améliorations

La matrice à (pseudo-)inverser devient très vite mal conditionnée lorsque l'ordre augmente. Un premier moyen d'améliorer considérablement le conditionnement de cette matrice et par conséquent les résultats est d'utiliser à la fois des amplitudes positives et négatives, ce qui permet notamment de séparer parfaitement les ordres pairs et impairs (cf. [11]). Nous avons cherché, sans succès, à généraliser cette technique en utilisant d'autres phases que 0 et π . Nos calculs n'ont malheureusement pas abouti. Cette piste reste cependant à explorer et pourrait éventuellement améliorer les résultats à la séparation.

Un second moyen est de régulariser la solution, en utilisant une pénalisation de la fonction de coût. La pénalisation permet de limiter la norme de la solution et l'empêche ainsi d'"exploser". Au lieu de minimiser l'erreur quadratique, on minimise une fonction de coût pénalisée donnée par :

$$l = \|\mathbf{y} - \mathbf{A}\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$$

Le facteur λ permet de donner plus ou moins de poids à la pénalisation. On a alors comme solution :

$$\hat{\mathbf{w}} = (\mathbf{A}^t\mathbf{A} + \lambda I_n)^{-1} \mathbf{A}^t\mathbf{y}$$

4.1.3 Utilisation

Nous avons implémenté en Matlab la méthode de séparation des ordres afin de l'utiliser dans les méthodes d'identification qui s'en servent. Nous présentons comment on peut utiliser en pratique cette méthode : ce qu'elle permet de faire et ses limites.

Le principal défaut de la méthode est bien sur l'explosion du conditionnement avec l'ordre. On peut cependant réduire au maximum le conditionnement de la matrice en optimisant la valeur du rapport entre les amplitudes α : à P fixé, on observe un minimum du conditionnement fonction de ce rapport, comme le montre la figure 4.3. La valeur du rapport pour laquelle le minimum est atteint est d'ailleurs quasiment indépendante de l'ordre et vaut environ $1.2dB$

Il est alors préférable d'utiliser cette valeur pour maximiser le conditionnement. La figure 4.4 montre alors qu'il est difficile de dépasser l'ordre 10 de beaucoup sans avoir des problèmes importants dus au mauvais conditionnement de la matrice.

La méthode permet cependant même en présence de bruit important de parfaitement séparer les ordres pairs et impairs (le bruit n'est cependant pas supprimé) comme le montre la figure 4.5.

4.2 Signaux aléatoires

4.2.1 Cas général

Les méthodes d'identification à l'aide de signaux aléatoires décrites dans [19] reposent sur une extension de l'identification des systèmes linéaires par la méthode

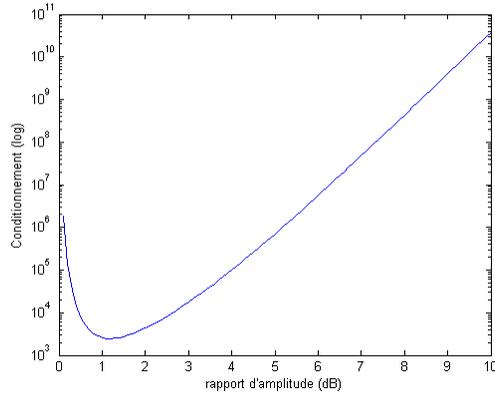


FIG. 4.3 – Conditionnement en fonction du rapport entre amplitudes consécutives pour $P = 10$ (échelle logarithmique)

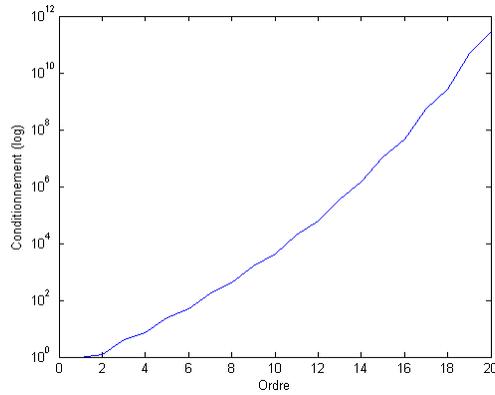


FIG. 4.4 – Conditionnement en fonction de l'ordre utilisé (échelle logarithmique)

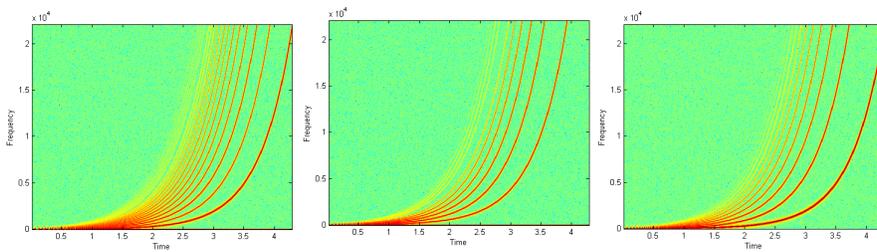


FIG. 4.5 – Séparation des ordres pairs/impairs : à gauche la réponse entière, au milieu les ordres pairs, à droite les ordres impairs

de la corrélation croisée. Dans toute cette section, l'entrée $x(n)$ est un bruit blanc identiquement distribué stationnaire et centré et on note $A_k = E(x^k)$ les moments de x . Si la densité de probabilité du bruit est paire, on a alors $A_k = 0$ pour tout k impair.

On rappelle que dans le cas linéaire, le calcul de la corrélation croisée de l'entrée et de la sortie permet d'obtenir directement la réponse impulsionnelle du système (cf. chapitre 1) : $h(T) = \frac{E(y(n)x(n-T))}{A_2}$. Cette méthode peut être étendue aux noyaux de Volterra homogènes.

Ordre 2

Commençons par nous intéresser au cas simple du noyau de Volterra homogène d'ordre 2. On considère la forme symétrique du noyau. La sortie du système s'écrit :

$$y(n) = \sum_{k_1, k_2=0}^{\infty} h_2(k_1, k_2)x(n-k_1)x(n-k_2)$$

Sous de bonnes hypothèses de convergence, pour $x(n)$ bruit blanc identiquement distribué stationnaire et centré :

$$E(y(n)x(n-T_1)x(n-T_2)) = \sum_{k_1, k_2=0}^{\infty} h_2(k_1, k_2)E(x(n-k_1)x(n-k_2)x(n-T_1)x(n-T_2))$$

Du fait de la stationnarité de l'entrée, l'espérance précédente est indépendante de n .

Dans le cas $T_1 \neq T_2$, on a :

$$E(x(n-k_1)x(n-k_2)x(n-T_1)x(n-T_2)) = A_2^2(\delta(k_1-T_1)\delta(k_2-T_2) + \delta(k_1-T_2)\delta(k_2-T_1))$$

Par conséquent, en utilisant la symétrie du noyau :

$$E(y(n)x(n-T_1)x(n-T_2)) = 2A_2^2h_2(T_1, T_2)$$

On peut donc obtenir directement les valeurs du noyau.

Dans le cas $T_1 = T_2 = T$, le calcul est un peu plus compliqué :

$$E(y(n)x(n-T)^2) = \sum_{k_1, k_2=0}^{\infty} h_2(k_1, k_2)E(x(n-k_1)x(n-k_2)x(n-T)^2)$$

or :

$$E(x(n-k_1)x(n-k_2)x(n-T)^2) = \delta(k_1-k_2) ((1-\delta(k_1-T))A_2^2 + \delta(k_1-T)A_4)$$

Par conséquent :

$$\begin{aligned}
E(y(n)x(n-T)^2) &= A_2^2 \sum_{k=0, k \neq T}^{\infty} h_2(k, k) + h_2(T, T)A_4 \\
&= A_2^2 \sum_{k=0}^{\infty} h_2(k, k) + h_2(T, T)(A_4 - A_2^2)
\end{aligned}$$

Dans ce cas, on n'obtient pas directement les valeurs du noyau : il faut résoudre un système linéaire. En pratique, on peut considérer que $h_2(T, T)$ est nul pour T grand et donc que $E(y(n)x(n-T)^2) = A_2^2 \sum_{k=0}^{\infty} h_2(k, k)$ pour T grand, ce qui permet de calculer $C = \sum_{k=0}^{\infty} h_2(k, k)$. On obtient alors :

$$h_2(T, T) = \frac{E(y(n)x(n-T)^2) - A_2^2 C}{A_4 - A_2^2}$$

Dans le cas général du noyau d'ordre P , le problème reste assez simple pour les coefficients $h_P(T_1, \dots, T_P)$ tels que tous les $T_1 \dots T_P$ sont distincts et on obtient la formule :

$$h_P(T_1, \dots, T_P) = \frac{1}{n! A_2^n} E(y(n)x(n-T_1) \dots x(n-T_P))$$

Quand les $T_1 \dots T_P$ ne sont pas distincts, le calcul se présente beaucoup moins bien et il semble qu'il soit difficile de le faire dans le cas général.

Ordre 3 :

Les résultats des cas particuliers sont présentés ici pour l'ordre 3 (le détail des calculs est donné en Annexe B), c'est-à-dire pour :

$$y(n) = \sum_{k_1, k_2, k_3=0}^{\infty} h_3(k_1, k_2, k_3) x(n-k_1) x(n-k_2) x(n-k_3)$$

Dans le cas $T_1 \neq T_2$, on obtient

$$\begin{aligned}
E(y(n)x(n-T_1)x(n-T_2)^2) &= (A_4 A_2 - 3A_2^3) h_3(T_1, T_1, T_1) \\
&+ 3(A_4 A_2 - A_2^3) h_3(T_1, T_2, T_2) \\
&+ 3A_2^3 \sum_{k=0}^{\infty} h_3(T_1, k, k)
\end{aligned}$$

Dans le cas $T_1 = T_2 = T$, on peut écrire :

$$E(y(n)x(n-T)^3) = (A_6 - 3A_4 A_2) h_3(T, T, T) + 3A_4 A_2 \sum_{k=0}^{\infty} h_3(T, k, k)$$

Les deux équations précédentes forment un système linéaire en les $h_3(T_1, T_2, T_2)$ (T_1 et T_2 pouvant être égaux). Dans le cas où on considère une série tronquée, les

sommes dans (D.1) et (D.4) deviennent des sommes finies et le système peut se résoudre facilement.

En pratique, on remplace dans les formules les espérances par leurs estimateurs empiriques.

4.2.2 Cas du modèle de Hammerstein généralisé

Dans le cas d'un système de Hammerstein généralisé, la méthode de séparation des ordres proposés dans la section 4.1 permet de séparer les sorties de l'ensemble des sous-systèmes homogènes qui ont une forme très simple. Les formules de corrélation présentées dans le paragraphe précédent sont alors considérablement simplifiées et les paramètres du modèle se calculent facilement.

Pour un tel système la sortie du sous-système homogène d'ordre p s'écrit :

$$y_p(n) = \sum_{m=0}^{\infty} h_p(m)x(n-m)^p$$

Ordres impairs

Dans le cas où p est impair, on peut écrire :

$$E(y_p(n)x(n-T)) = \sum_{m=0}^{\infty} h_p(m)E(x(n-m)^p x(n-T))$$

Or, comme x est un bruit blanc centré, on a $E(x(n-m)^p x(n-T)) = A_{p+1}\delta(m-T)$ et par conséquent :

$$h_p(T) = \frac{E(y_p(n)x(n-T))}{A_{p+1}}$$

Ordres pairs

Dans le cas où p est pair, on peut écrire :

$$E(y_p(n)x(n-T)^2) = \sum_{m=0}^{\infty} h_p(m)E(x(n-m)^p x(n-T)^2)$$

Or, comme x est un bruit blanc centré, on a $E(x(n-m)^p x(n-T)^2) = A_{p+2}\delta(m-T) + A_p A_2(1 - \delta(m-T))$ et par conséquent :

$$E(y_p(n)x(n-T)^2) = A_p A_2 \sum_{m=0}^{\infty} h_p(m) + (A_{p+2} - A_p A_2)h_k(T)$$

Comme $h_k(n)$ tend vers 0 pour n tendant vers ∞ , on peut considérer que $h_k(T)$ est nul pour T suffisamment grand, ce qui permet de calculer $C = \sum_{m=0}^{\infty} h_p(m)$ à partir de $E(y_p(n)x(n-T)^2)$ (afin d'améliorer l'estimation, on peut calculer une moyenne sur un ensemble de grandes valeurs de T). On en déduit alors :

$$h_p(T) = \frac{E(y_p(n)x(n-T)^2) - A_p A_2 C}{A_{p+2} - A_p A_2}$$

4.2.3 Utilisation et Résultats

La méthode par intercorrélation dans le cas général est, comme on l'a déjà précisé, difficilement applicable pour identifier intégralement un système de Volterra complet. Elle a cependant un intérêt majeur : une fois les ordres séparés, on peut calculer n'importe quel coefficient du noyau indépendamment des autres.

Nous nous en sommes donc servi pour identifier intégralement le noyau d'ordre 2 d'un système réel : la méthode a été implémentée sous Matlab et nous avons mesuré notamment un haut-parleur de mauvaise qualité (analogue à un haut-parleur d'ordinateur portable) qui génère de la distorsion dès que le niveau d'entrée est élevé. Le noyau d'ordre 2 issu d'une de nos mesures est représenté dans la figure 4.6.

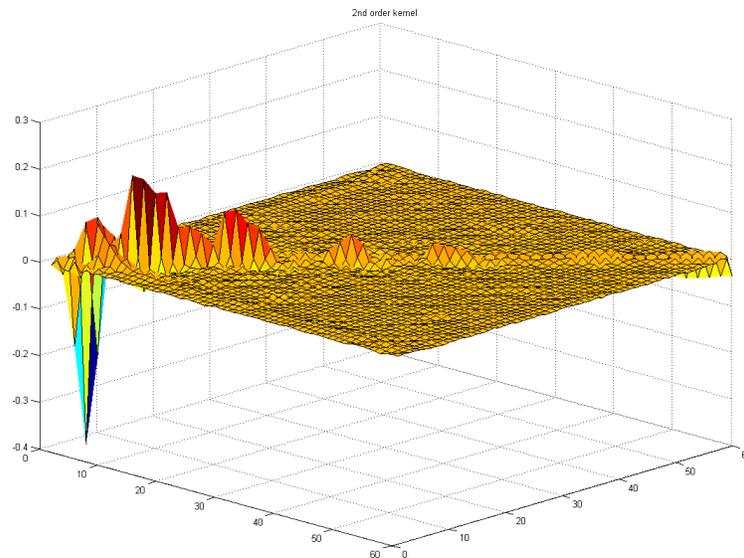


FIG. 4.6 – Noyau d'ordre 2

On peut remarquer que les coefficients non diagonaux sont tous quasi nuls : il est donc possible que ce système puisse être modélisé par un système de Hammerstein généralisé. Ainsi la méthode d'identification par intercorrélation peut être utilisée comme nous l'avons fait pour déterminer une structure que pourrait avoir le système.

Un point très important lorsqu'on utilise la méthode par intercorrélation est la génération des signaux aléatoires utilisés. En effet, l'autocorrélation empirique d'un signal pseudo-aléatoire n'est jamais exactement une impulsion de Dirac. Or pour les autocorrélations d'ordres supérieurs la différence avec une impulsion de

Dirac a tendance à devenir très importante : c'est notamment le cas pour les bruits blancs gaussiens générés par Matlab qui deviennent inutilisables au-dessus de l'ordre 5. Les séquences MLS ont, quant à elles, la propriété d'avoir le résiduel des autocorrélations de tous les ordres contrôlé uniformément par leur longueur. Ces signaux sont donc particulièrement adaptés pour utiliser cette méthode.

4.3 Impulsions de Dirac et convolution dynamique

4.3.1 Réponse d'un système de Volterra à des impulsions de Dirac

Dans cette section, on se limite au cas des systèmes homogènes. Comme on l'a déjà signalé, se limiter au calcul pour des systèmes homogènes n'est pas très restrictif puisqu'on est capable (en théorie du moins) de séparer les réponses de tous les sous-systèmes homogènes comme cela a été présenté dans la section 4.1.

Impulsion unique

Dans le cas d'un système homogène d'ordre p , la réponse d'un système de Volterra à une unique impulsion de Dirac se calcule très facilement :

$$\begin{aligned} y_p(n) &= \sum_{q_1, q_2, \dots, q_p=0}^{\infty} h_k(q_1, q_2, \dots, q_p) \delta(n - q_1) \delta(n - q_2) \dots \delta(n - q_p) \\ &= h_k(n, n, \dots, n) \end{aligned}$$

Une impulsion unique ne permet donc de déterminer que les coefficients diagonaux. Cette méthode très simple permet donc d'identifier intégralement un système de Hammerstein généralisé. Cependant, pour déterminer les coefficients non diagonaux, il faut utiliser plusieurs impulsions successives.

Impulsions multiples

L'utilisation d'une somme d'impulsions comme signal d'entrée s'avère plus complexe mais permet de calculer les coefficients hors de la diagonale. On prend comme signal d'entrée :

$$x(n) = \delta_0(n) + \delta_0(n - n_0)$$

La sortie s'obtient par un simple résultat de combinatoire, en utilisant la forme symétrique du noyau :

$$\begin{aligned} y_p(n) &= \sum_{q_1, \dots, q_p=0}^{\infty} h_k^{sym}(q_1, \dots, q_p) (\delta(n - q_1) + \delta(n - q_1 - n_0)) \dots (\delta(n - q_p) + \delta(n - q_1 - n_0)) \\ &= \sum_{m=0}^p \binom{p}{m} h_k^{sym}(\underbrace{n, \dots, n}_m, \underbrace{n - n_0, \dots, n - n_0}_{p-m}) \end{aligned}$$

On peut généraliser l'approche précédente en prenant pour signal d'entrée :

$$x(n) = \delta_0(n) + \delta_0(n - n_1) \dots \delta_0(n - n_{k-1})$$

On obtient alors comme sortie (pour plus de détail sur le calcul, se référer à [19]) :

$$y_p(n) = \sum_{m_1 + \dots + m_k = p} \frac{p!}{m_1! m_2! \dots m_k!} h_k^{sym}(n, \dots, n, \dots, \underbrace{n - n_{k-1}, \dots, n - n_{k-1}}_{m_k})$$

En utilisant des sommes de 1 à p impulsions, on obtient alors un système linéaire. Si on suppose que les valeurs du noyau deviennent nulles loin de l'origine, on peut alors calculer les valeurs du noyau.

4.3.2 Mesure de systèmes décrits par la convolution dynamique

Le modèle de la convolution dynamique a été présenté dans la section 3.2.2.

Méthode de mesure

Pour mesurer $h(\tau, x)$, il suffit de mesurer directement la réponse à des impulsions de Dirac d'amplitudes différentes : la réponse du système à une impulsion d'amplitude a est en effet $h(\tau, a)$. Les autres valeurs de $h(\tau, x)$ sont obtenues par simple interpolation linéaire entre $h(\tau, a_1)$ et $h(\tau, a_2)$ pour $x \in [a_1, a_2]$.

Remarque : la méthode présentée dans les articles de Sintefex ([12]) utilise en fait des signaux en marches d'escalier (multistep) à la place des impulsions pour mesurer $h(\tau, x)$. Le principe reste cependant le même.

Différence pratique avec Volterra

Les approximations ne sont pas les mêmes suivant qu'on utilise le formalisme de la convolution dynamique ou celui de Volterra : Dans le cas de la convolution dynamique, la "réponse impulsionnelle" $h(\tau, x)$ est discrétisée en x et interpolée entre les échantillons (dans [12], une interpolation linéaire entre les échantillons est utilisée, ce qui ne se justifie que par la simplicité de la mise en oeuvre). Dans le cas de Volterra, l'approximation se fait plutôt au niveau de la troncation des ordres. Ainsi cette méthode a l'avantage de ne pas tronquer les ordres ce qui la rend assez intéressante.

4.3.3 Mise en oeuvre

Nous avons mis en oeuvre l'algorithme d'identification de la convolution dynamique présenté ci-dessus. L'identification de systèmes simulés est assez performante. Bien qu'elle soit limitée aux systèmes de Hammerstein généralisés, elle permet en théorie d'identifier et de simuler des systèmes d'ordre éventuellement

infini. La figure 4.7 montre des réponses fréquentielles mesurées pour différentes valeurs de l'amplitude de l'entrée à partir d'un système simulé d'ordre élevé : les réponses à basse amplitude qui excitent peu les non-linéarités ressemblent à la réponse fréquentielle de l'ordre 1 seul du système, alors que les réponses pour les plus hautes amplitudes font apparaître les non-linéarités.

Un des problèmes majeurs de toutes les méthodes utilisant des impulsions de Dirac est qu'elles sont peu adaptées à la mesure de systèmes analogiques réels : le bruit est directement récupéré dans les réponses impulsionnelles mesurées (ce qui peut en partie être atténué en faisant une moyenne de plusieurs mesures) et surtout les impulsions génèrent de nombreux phénomènes problématiques. Notamment, les convertisseurs numérique/analogique ne peuvent pas générer de vraies impulsions (même à bande limitée puisque celles-ci ont un support temporel qui va de $-\infty$ à ∞). De plus lorsque l'énergie de ces impulsions devient importante, on génère souvent des non-linéarités dans les systèmes servant à la mesure (pré-amplificateur, microphone...). Ces artefacts sont difficilement contrôlables du fait même qu'ils traversent un système non-linéaire. Par conséquent, les méthodes à impulsions de Dirac sont à exclure dès que les non-linéarités deviennent trop importantes : ainsi l'avantage de pouvoir identifier des systèmes d'ordre infini est en pratique inexistant.

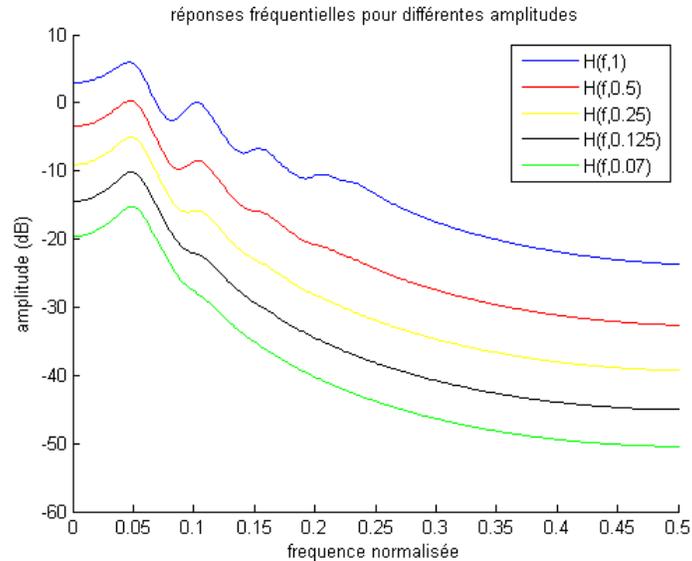


FIG. 4.7 – Réponses fréquentielles mesurées pour plusieurs amplitudes

4.4 Sinusoïdes et sweeps sinusoïdaux

4.4.1 Réponse d'un système de Volterra à une sinusoïde pure (régime permanent)

La réponse d'un système de Volterra à une sinusoïde pure est bien plus compliquée à exprimer que celle d'un système linéaire. Pour un système homogène d'ordre p nourri par une sinusoïde pure $x(n) = 2A \cos(\omega n)$, on obtient à partir de la forme symétrique du noyau la sortie :

$$y_p(n) = \sum_{q_1, \dots, q_p=0}^{\infty} h_p^{sym}(q_1, \dots, q_p) A^p (e^{i\omega(n-q_1)} + e^{-i\omega(n-q_1)}) \dots (e^{i\omega(n-q_p)} + e^{-i\omega(n-q_p)})$$

En développant l'expression précédente, on obtient :

$$y_p(n) = A^p \sum_{k_1=0}^1 \dots \sum_{k_p=0}^1 \left[\sum_{q_1, \dots, q_p=0}^{\infty} h_p^{sym}(q_1, \dots, q_p) \exp\left(-\sum_{j=1}^p (-1)^{k_j} i\omega q_j\right) \right] \exp\left(\sum_{j=1}^p (-1)^{k_j} i\omega n\right)$$

L'expression ci-dessus fait apparaître la transformée de Laplace du noyau, ce qui donne finalement :

$$y_p(n) = A^p \sum_{k_1=0}^1 \dots \sum_{k_p=0}^1 H_p^{sym}((-1)^{k_1} i\omega, \dots, (-1)^{k_p} i\omega) \exp\left(\sum_{j=1}^p (-1)^{k_j} i\omega n\right)$$

Cette expression fait apparaître qu'un système non-linéaire homogène d'ordre p nourri par une sinusoïde pure génère des harmoniques. Or on constate que $\sum_{j=1}^p (-1)^{k_j}$ est de la même parité que p pour tout k_j . Ainsi si p est pair, le système génère des harmoniques paires et si p est impair, le système génère des harmoniques impaires.

Pour un système de Volterra non homogène d'ordre P , en sommant les contributions homogènes calculées précédemment, on obtient une expression du type (pour plus de détail sur le calcul, se référer à [19]) :

$$y(n) = f_0(A, \omega) + 2 \sum_{p=1}^P |f_p(A, \omega)| \cos(p\omega n + \angle f_p(A, \omega))$$

Cette expression montre que la réponse d'un système non-linéaire à une sinusoïde est une somme d'harmoniques c'est-à-dire des sinusoïdes de fréquence un multiple entier de la sinusoïde en entrée. Cette propriété est fondamentale pour l'identification de système de Volterra à partir de sinusoïdes.

Il est également possible de façon analogue de calculer la réponse d'un système de Volterra à une somme de sinusoïdes (cf. [4] et [25]) : il apparaît alors également dans la réponse des fréquences dites d'intermodulation qui sont des combinaisons linéaires entières des fréquences des différentes composantes sinusoïdales du signal d'entrée.

4.4.2 Propriétés des sweeps logarithmiques

Définition des sweeps : Afin de parcourir toute la bande de fréquence pour laquelle on souhaite mesurer le système étudié, on utilise généralement des sweeps sinusoïdaux : ces signaux parcourent toute une bande de fréquence et dans certaines conditions peuvent être localement considérés comme des sinusoïdes. On utilise des sweeps logarithmiques de la forme suivante :

$$x(t) = \sin\left(\frac{\omega_1 T}{\ln\frac{\omega_2}{\omega_1}} \left(e^{\frac{t}{T} \ln\frac{\omega_2}{\omega_1}} - 1\right)\right) = \cos(\Phi_c(t))$$

avec :

$$\Phi_c(t) = \frac{\omega_1 T}{\ln\frac{\omega_2}{\omega_1}} \left(e^{\frac{t}{T} \ln\frac{\omega_2}{\omega_1}} - 1\right) - \frac{\pi}{2}$$

Ces sweeps décrivent l'intervalle de fréquences (pulsations) réduites $[\omega_1, \omega_2]$ pendant une durée T , comme illustré dans la figure 4.8.

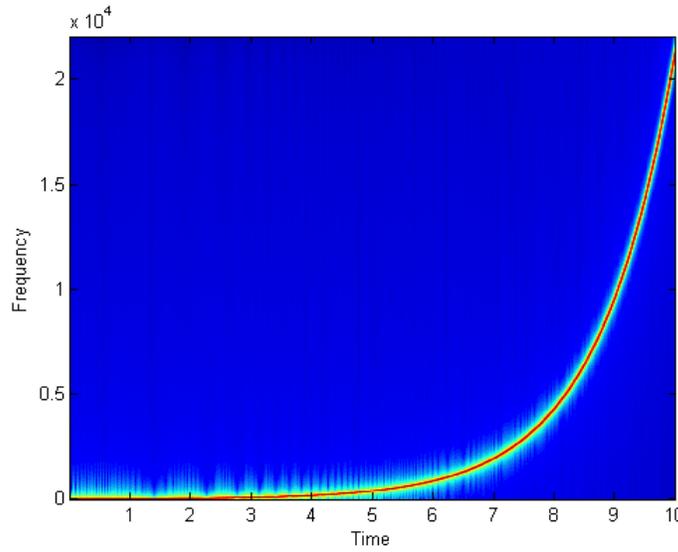


FIG. 4.8 – Spectrogramme d'un sweep logarithmique ($\omega_1 = 0.001$, $\omega_2 = \pi$, $T = 10s$)

Propriété fondamentale : Pour tout $k \in \mathbb{N}^*$ on a :

$$k\Phi_c(t) = \Phi_c\left(t + \frac{T \ln k}{\ln\frac{\omega_2}{\omega_1}}\right) - (k-1) \left(\frac{\pi}{2} + \frac{\omega_1 T}{\ln\frac{\omega_2}{\omega_1}}\right)$$

En prenant $T = (2p\pi - \frac{\pi}{2}) \frac{\ln \frac{\omega_2}{\omega_1}}{\omega_1}$ avec $p \in \mathbb{N}^*$ on obtient :

$$\cos(k\Phi_c(t)) = \cos(\Phi_c(t + \frac{T \ln k}{\ln \frac{\omega_2}{\omega_1}}))$$

Le gros avantage des sweeps logarithmiques réside dans la formule précédente : une multiplication de la phase par un entier peut être vue comme un décalage temporel. Or comme on l'a vu dans la section précédente, la sortie d'un système non-linéaire nourri par une sinusoïde pure est une somme d'harmoniques. Ainsi, lorsqu'on utilise en entrée un sweep on pourra interpréter la sortie comme une somme d'harmoniques ou bien comme une somme de sweeps retardés les uns par rapport aux autres, comme présenté dans la figure 4.9 pour un systèmes ne générant que des harmoniques impaires

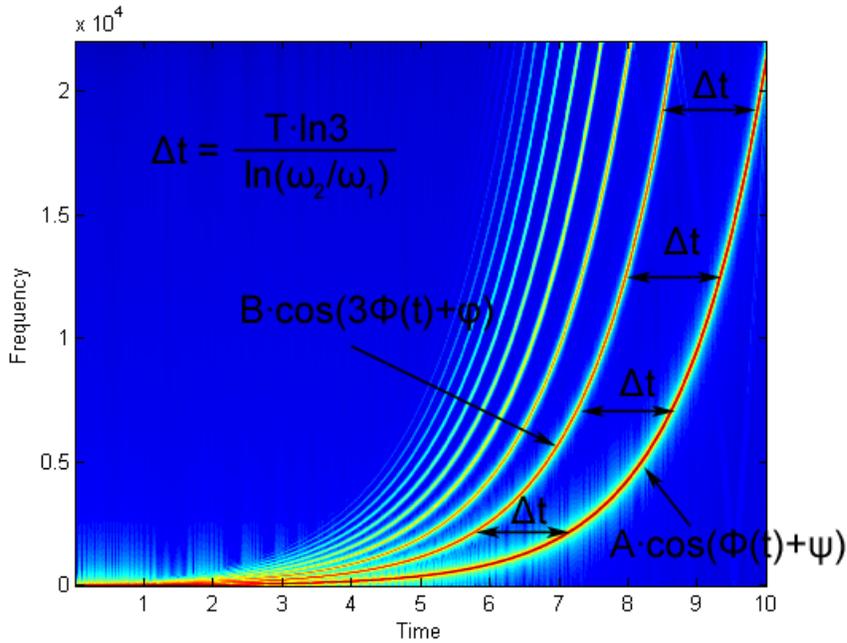


FIG. 4.9 – Spectrogramme de la réponse d'un système impair à un sweep logarithmique ($\omega_1 = 0.001$, $\omega_2 = \pi$, $T = 10s$)

Inverse au sens de la convolution : Une très bonne approximation de l'inverse au sens de la convolution d'un sweep logarithmique (cf. section 4.4.2) peut être obtenue très simplement à partir de ce sweep. Il suffit de lui appliquer les deux transformations suivantes :

- appliquer un miroir temporel (inverser le sens du temps)

- filtrer le signal obtenu à l'aide d'un filtre à 3dB/octave (Il s'agit d'un filtre de blanchiment : l'énergie du sweep logarithmique décroît en effet de 3dB/octave)

On note $x(t)$ le sweep utilisé et $x_r(t)$ le signal obtenu par la transformation précédente. On a donc :

$$x * x_r(t) \approx \delta(t) \quad (4.5)$$

Remarque : Des méthodes sont présentées dans [7], pour pallier le fait qu'il ne s'agisse que d'une approximation.

4.4.3 Système de Hammerstein généralisé

Méthode

A. Farina présente dans [6] et [8] une méthode d'identification des systèmes de Hammerstein généralisés à l'aide de sweeps logarithmiques. Cette méthode est présentée ici en détail.

Lorsqu'on applique un sweep logarithmique à l'entrée d'un système de Hammerstein généralisé d'ordre P , on obtient la sortie :

$$y(t) = \sum_{p=1}^P h_p * x^p(t) \quad (4.6)$$

À l'aide des polynômes de Tchebychev, on peut exprimer $x^p(t) = \cos^p(\Phi_c(t))$ en fonction des $\cos(k\Phi_c(t))$ pour $k = 0 \dots P$. Les polynômes de Tchebychev T_k vérifient en effet $T_k(\cos(x)) = \cos(kx)$. En écrivant cette relation pour tout $k = 0 \dots P$, on obtient alors un système linéaire :

$$\begin{pmatrix} 1 \\ \cos(x) \\ \cos(2x) \\ \vdots \\ \cos(Px) \end{pmatrix} = \begin{pmatrix} a_{0,0} & 0 & \cdot & \dots & 0 \\ a_{1,0} & a_{1,1} & 0 & \dots & 0 \\ a_{2,0} & a_{2,1} & a_{2,2} & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_{P,0} & a_{P,1} & \cdot & \dots & a_{P,P} \end{pmatrix} \begin{pmatrix} 1 \\ \cos(x) \\ \cdot \\ \cdot \\ \cdot \\ \cos^P(x) \end{pmatrix}$$

Où les $(a_{k,i})_{(i=0 \dots k)}$ sont les coefficients des polynômes T_k .

En inversant ce système on obtient donc la relation suivante :

$$x^p(t) = \sum_{k=0}^P a_{k,p} \cos(k\Phi_c(t))$$

En substituant cette expression dans l'équation (4.6), on obtient :

$$\begin{aligned} y(t) &= \sum_{p=1}^P h_p * \sum_{k=0}^P a_{k,p} \cos(k\Phi_c(t)) \\ &= \sum_{k=0}^P \left[\sum_{p=1}^P a_{k,p} h_p \right] * \cos(k\Phi_c(t)) \end{aligned}$$

En posant :

$$g_k = \sum_{p=1}^P a_{k,p} h_p \quad (4.7)$$

On obtient alors :

$$\begin{aligned} y(t) &= \sum_{k=0}^P g_k * \cos(k\Phi_c(t)) \\ &= \sum_{k=0}^P g_k * \cos\left(\Phi_c\left(t + \frac{Tlnk}{ln\frac{\omega_2}{\omega_1}}\right)\right) \end{aligned}$$

Cette nouvelle expression est assez pratique car elle fait apparaître directement les harmoniques séparément. En convoluant la sortie mesurée $y(t)$ avec $x_r(t)$, on obtient alors :

$$y * x_r(t) = \sum_{k=0}^P g_k * x_r * \cos\left(\Phi_c\left(t + \frac{Tlnk}{ln\frac{\omega_2}{\omega_1}}\right)\right)$$

En utilisant alors l'équation (4.5), on obtient :

$$x_r * \cos\left(\Phi_c\left(t + \frac{Tlnk}{ln\frac{\omega_2}{\omega_1}}\right)\right) \approx \delta\left(t + \frac{Tlnk}{ln\frac{\omega_2}{\omega_1}}\right)$$

Par conséquent :

$$y * x_r(t) = \sum_{k=0}^P g_k\left(t + \frac{Tlnk}{ln\frac{\omega_2}{\omega_1}}\right)$$

On obtient ainsi la somme d'un ensemble de pseudo-réponses impulsionnelles g_k retardées dans le temps les unes par rapport aux autres. Si le sweep d'entrée est assez long, ces réponses impulsionnelles ne se chevauchent pas et on est donc capable de les extraire simplement comme présenté dans la figure 4.10. Il suffit alors d'inverser l'expression (4.7) qui donne les g_k en fonction des h_p , afin d'obtenir les h_p (en réalité il n'est pas nécessaire de d'inverser de matrice puisque les $a_{k,p}$ dans l'équation (4.7) ont déjà été obtenus par une inversion).

Résultats

La méthode de Farina donne d'assez bons résultats sur tout système de Hammerstein généralisé d'ordre petit. Le gros défaut de la méthode est issu de la troncation de l'ordre du système : le calcul des h_p à partir des g_p utilise en effet les polynômes de Tchebychev T_p , dont les coefficients augmentent de façon conséquente avec p . Ainsi les composantes d'ordre important, qui généralement ont une contribution moindre et qui sont donc relativement beaucoup plus affectées par le bruit que les premiers ordres, prennent beaucoup d'importance, ce qui vient fausser la résolution. Vient s'ajouter à ce problème le fait que la fonction x_r n'est qu'une approximation de l'inverse au sens de la convolution du sweep (l'inverse

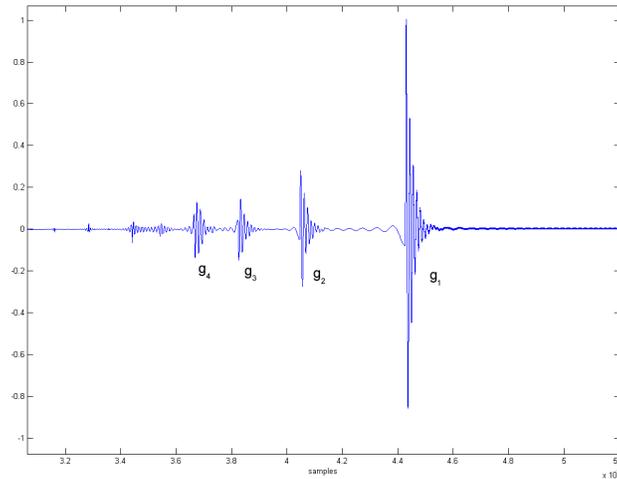


FIG. 4.10 – Réponse d'un système de Hammerstein généralisé à un sweep, convoluée par le sweep inverse : on peut extraire les g_k

exacte n'existe d'ailleurs pas) : ainsi les artefacts créés par cette approximation pour l'ordre 1 se font ressentir dans les ordres élevés d'amplitude beaucoup plus faible.

Pour illustrer ces problèmes, nous présentons en figure 4.11 les résultats de l'identification d'un système d'ordre 9 simulé. Un bruit blanc gaussien est ajouté à la sortie du système pour simuler des conditions réelles d'enregistrement. Ce bruit blanc a été pris d'amplitude $-60dB$ ce qui est comparable à ce qui peut être observé lors d'une mesure d'un système analogique. On remarque clairement que plus on monte dans les ordres, plus l'estimation est mauvaise : à partir de l'ordre 5, l'estimation devient trop mauvaise pour pouvoir être utilisée.

Nous avons d'ailleurs tenté de mesurer un système réel : il s'agissait d'un haut-parleur de mauvaise qualité générant de la distorsion, le même que celui mesuré dans la section 4.2.3. Cette dernière mesure tendait à légitimer l'hypothèse que ce système se comporte comme un système de Hammerstein généralisé (modèle qui est d'ailleurs souvent utilisé pour modéliser les non-linéarités d'un haut-parleur dans la littérature, notamment par Farina) ce qui n'est cependant pas garanti.

Nous avons ensuite simulé le système mesuré sur des signaux musicaux, et les avons comparés (subjectivement, à l'oreille) à ces mêmes signaux traités par le système original. Les signaux mesurés font bien apparaître une distorsion nettement audible mais la différence avec les signaux traités reste très nette.

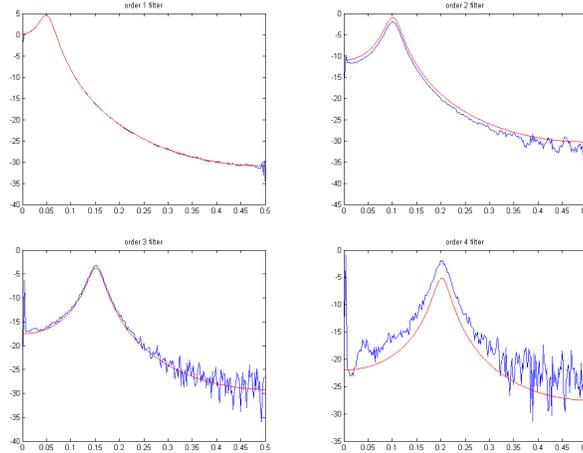


FIG. 4.11 – Mesure des filtres d'un système de Hammerstein généralisé par la méthode de Farina. En rouge : filtres utilisés, en bleu : filtres estimés (de gauche à droite : ordre 1 à 4)

4.4.4 Système S_M

Méthode

Abel et Berners présentent dans [1] une méthode d'identification des systèmes S_M . Le même type de sweeps logarithmiques que pour la méthode de Farina est utilisé. Cette méthode est assez proche de celle de Farina mais utilise en plus une séparation des ordres comme présentée dans 4.1. On obtient donc indépendamment les réponses de chaque sous-système homogène $r_p(t)$.

En convoluant chacune de ces réponses avec le sweep inverse $x_r(t)$ (cf. 4.4.2), on obtient de façon analogue à la méthode de Farina un ensemble de pseudo-réponses impulsionnelles pour lesquelles on note $H_{k,p}$ les réponses fréquentielles associées. Ces pseudo-réponses fréquentielles sont liées aux réponses des filtres d'entrée et de sortie g_p et h_p (voir le schéma du modèle S_M à la section 3.2.1) par les relations :

$$H_{p,k}(\omega) = 2^{1-p} \binom{p}{k} \hat{h}_p(\omega) \hat{g}_p^p(\omega/(p-2k)) \quad (4.8)$$

Intuitivement, la réponse $H_{p,k}$ correspond à la $2k$ -ème (respectivement $(2k+1)$ -ème) harmonique générée par l'ordre p du système pour p pair (respectivement p impair).

En passant au logarithme dans l'équation (4.8) et en discrétisant la fréquence ω , on obtient alors un système linéaire en les $\ln(\hat{h}_p(\omega))$ et les $\ln(\hat{g}_p(\omega))$ qui se résout facilement pour tous les ordres p sauf l'ordre 2 pour lequel le système est sous-déterminé. Pour l'ordre 2 une seconde mesure est nécessaire : on utilise alors un signal qui est la somme d'un sweep logarithmique et de ce même sweep pris à l'octave supérieure.

L'intégralité des calculs et le déroulement de la méthode sont présentés dans l'annexe C.

Résultats

La méthode d'Abel et Berners a l'avantage de pouvoir identifier des systèmes assez généraux puisque les systèmes S_M englobent les systèmes de Hammerstein généralisés et les systèmes de Wiener généralisés.

Cependant, les mesures que nous avons effectuées sur des systèmes simulés nous ont permis de mettre à jour des défauts importants qui ne sont pas évoqués dans l'article de présentation de la méthode et qui pourtant rendent son utilisation très difficile pour la mesure de systèmes audio réels.

En premier lieu, la méthode utilise une séparation des ordres dont les défauts ont déjà été présentés dans la section 4.1. De plus, le passage au logarithme à partir de l'équation (4.8) est problématique : si certains ordres intermédiaires s'avèrent nuls (ce qui est couramment le cas dans les systèmes réels) les calculs explosent. De même, lorsque le bruit devient prédominant dans les zones de fréquences de faible énergie, les solutions du système deviennent absurdes. Ainsi, la stabilité numérique de cette méthode est assez mauvaise.

Nous avons effectué plusieurs mesures de systèmes réels notamment d'un haut-parleur de mauvaise qualité ainsi que d'une pédale de distorsion de guitare (Electro Harmonix Bigg Muff) : les résultats étaient vraiment mauvais du fait de la mauvaise stabilité numérique de la méthode, tout particulièrement pour la pédale de distorsion qui générait trop de distorsion semble-t-il. Ainsi, la méthode est apparue comme très peu robuste aux conditions réelles de mesure des systèmes réels.

Conclusion sur l'identification

Nous avons mis en oeuvre un certain nombre de méthodes d'identification, mais toutes nous ont semblé avoir des défauts trop importants pour permettre une simulation convaincante des systèmes mesurés. Dans tous les cas il est apparu que les méthodes étaient très sensibles à l'augmentation de l'ordre du système dans le cas de mesures de systèmes réels dans lesquelles intervient notamment du bruit. Il s'agit là d'un des points les plus difficiles du problème d'identification des systèmes non-linéaires : le bruit qui peut exister en entrée du système génère des phénomènes (notamment d'intermodulation) qui sont très difficilement contrôlables.

En outre, il semble qu'un des principaux défauts de la plupart de ces méthodes est de considérer des systèmes tronqués en ordre : or cette troncation est toujours une approximation, qui peut générer de gros artefacts (par exemple dans la méthode de Farina) dans l'identification.

Ces importantes difficultés à identifier des systèmes complexes nous ont donc amené à nous pencher sur de puissantes méthodes de régression que sont les méthodes à noyaux reproduisants.

Chapitre 5

Régression à l'aide de noyaux reproduisants

Pour introduire les méthodes de régression non-linéaire à noyaux reproduisants, nous commençons dans la première section par utiliser la structure de Volterra pour transformer le problème en problème de régression linéaire paramétrique afin de pouvoir la résoudre par la méthode des moindres carrés classiques. Cette présentation peut paraître étrange au lecteur ayant déjà une bonne connaissance des méthodes à noyaux reproduisants, mais elle a l'avantage d'aborder de façon simple les méthodes de régression par une méthode linéaire et surtout de montrer le cheminement que nous avons suivi durant ce projet qui nous a amené à utiliser les méthodes à noyaux reproduisants pour identifier des systèmes non-linéaires à mémoire. Les méthodes à noyaux reproduisants sont ensuite présentées de façon aussi succincte que possible (le développement complet étant beaucoup trop long à exposer) et il est nécessaire de se référer à la bibliographie pour plus de détails.

5.1 Moindres carrés

5.1.1 Cas général

Nous commençons par présenter une méthode de régression simple prenant directement en compte la structure des séries de Volterra :

Cette méthode très simple sur le plan théorique est en pratique vouée à l'échec pour des systèmes dont les non-linéarités sont trop importantes. Elle consiste à utiliser directement la méthode des moindres carrés pour l'identification. Si on considère un système bien représenté par une série de Volterra tronquée en ordre avec une mémoire finie, le problème de l'identification est un problème de régression paramétrique linéaire. En utilisant la forme triangulaire supérieure des

noyaux de Volterra, on peut en effet écrire la sortie de la manière suivante :

$$y(n) = \sum_{p=1}^P \sum_{k_1=0}^{M-1} \sum_{k_2=k_1}^{M-1} \dots \sum_{k_p=k_{p-1}}^{M-1} h_p(k_1, \dots, k_p) x(n-k_1) \dots x(n-k_p)$$

En regroupant tous les monômes $x(n-k_1) \dots x(n-k_p)$ de la somme précédente dans un seul vecteur $X(n)$ et en regroupant tous les $h_p(k_1, \dots, k_p)$ dans un vecteur de paramètres θ associé au premier, on obtient l'expression de la sortie du système suivante :

$$y(n) = {}^t X(n) \theta$$

Par exemple, pour $M = 2$ et $P = 2$, on aurait :

$$X(n) = \begin{pmatrix} x(n) \\ x(n-1) \\ x(n)^2 \\ x(n)x(n-1) \\ x(n-1)^2 \end{pmatrix} \quad \theta = \begin{pmatrix} h_1(0) \\ h_1(1) \\ h_2(0,0) \\ h_2(0,1) \\ h_2(1,1) \end{pmatrix}$$

Une expression générale du vecteur $X(n)$ construit à partir de produits tensoriels est proposée dans [18].

Le problème peut donc être résolu par la méthode des moindres carrés : En notant :

$$\begin{aligned} \mathbf{X} &= {}^t [X(n) \ X(n-1) \ \dots \ X(n-N+1)] \\ Y &= {}^t [y(n) \ y(n-1) \ \dots \ y(n-N+1)] \end{aligned}$$

On obtient :

$$Y(n) = \mathbf{X} \theta$$

L'estimateur des moindres carrés du paramètre θ qui minimise l'erreur quadratique $e = \|y - \mathbf{X}\theta\|$ s'écrit alors :

$$\hat{\theta} = \mathbf{X}^\dagger Y(n)$$

Cette méthode n'est cependant utilisable que pour des systèmes d'ordre très petit et de mémoire courte. En effet la matrice \mathbf{X} est de taille $N \times L(P, M)$ ou N doit être pris au moins aussi grand que $L(P, M)$ pour être dans le cas (sur)déterminé. A l'heure actuelle, un ordinateur standard n'est capable d'inverser une matrice que jusqu'à une taille de l'ordre de 1000×1000 , ce qui pour une mémoire de l'ordre de $M = 100$, ne permet l'identification complète que pour un système d'ordre 2.

5.1.2 Modèle de Hammerstein généralisé

Si on se place dans le cadre d'un système simplifié tel que le modèle de Hammerstein généralisé, le nombre de paramètres à estimer est beaucoup moins im-

portant. La sortie du système s'écrit :

$$y(n) = \sum_{p=1}^P \sum_{k=0}^{M-1} h_p(k)x(n-k)^p$$

La matrice à inverser pour calculer l'estimateur des moindres carrés est alors de taille $N \times MP$. Par conséquent pour un système qui a une mémoire de l'ordre de $M = 100$, on peut largement aller jusqu'à l'ordre $P = 10$. La méthode semble alors tout à fait utilisable. Il faut cependant faire attention au conditionnement de la matrice \mathbf{X} qui devient très mauvais pour $P = 10$.

5.2 Régression non-linéaire et noyaux reproduisants

Le problème le plus souvent rencontré jusqu'ici dans le cadre de l'identification des systèmes était le nombre beaucoup trop important de coefficients à calculer pour bien décrire un système. Les méthodes de régression dans des espaces de Hilbert à noyau reproduisant permettent de limiter ce problème tout en permettant d'identifier des systèmes de Volterra d'ordre éventuellement infini. L'idée générale est de définir un mapping non-linéaire de l'espace des entrées du système Ω (qui est une partie de \mathbb{R}^M où M est la taille de la mémoire du système) dans un espace de dimension beaucoup plus importante (potentiellement infinie) qui contient toutes les fonctions entrées/sortie possibles. La régression est faite dans cet espace de grande dimension à l'aide du fameux *coup du noyau*¹ (cf. section 5.2.3) qui permet de manipuler des données qui ne sont que de la taille de l'espace d'entrée. Ces méthodes constituent une généralisation des méthodes de régression linéaire (comme les moindres carrés) à des problèmes de régression non-linéaire.

Remarque importante : Bien que le même terme soit utilisé pour désigner les noyaux de Volterra et les noyaux reproduisants, il s'agit de notions totalement différentes et il faudra donc faire attention à bien distinguer ces deux types de noyaux.

5.2.1 Problèmes de régression

La méthode des moindres carrés présentées plus haut est la résolution d'un problème de régression particulier. Nous allons ici présenter une classe de problèmes de régression plus large afin d'introduire les méthodes utilisant des noyaux reproduisants.

Nous allons commencer par redéfinir clairement notre problème comme un problème de régression. On se place dans le cadre d'un système (de Volterra) à mémoire finie. La sortie d'un tel système de mémoire de longueur M peut s'écrire en fonction de l'entrée :

$$y(n) = f(x(n), x(n-1), \dots, x(n-M+1))$$

¹plus souvent rencontré en anglais sous l'appellation *kernel trick*

On cherche alors la “meilleure” fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Il s’agit donc d’un problème de régression très général.

Afin de donner un sens à la “meilleure” fonction f , on note $x_n = [x(n), x(n-1), \dots, x(n-M+1)]$ et on suppose qu’on dispose de N couple entrée/sortie (x_n, y_n) .

On définit alors une fonction de coût $C(f(x_1), \dots, f(x_N), y_1, \dots, y_N)$ (généralement convexe). La fonction f recherchée est alors définie comme :

$$f = \operatorname{argmin}_{f \in \mathcal{H}} C(f(x_1), \dots, f(x_N), y_1, \dots, y_N)$$

Où \mathcal{H} est un espace de fonctions définies sur \mathbb{R}^n à valeurs dans \mathbb{R} . Cet espace peut-être explicité par exemple quand on utilise un dictionnaire de fonctions (cf. section 5.2.2) ou non comme c’est généralement le cas lorsqu’on utilise des méthodes à base de noyaux reproduisants (cf. 5.2.3).

5.2.2 Dictionnaire de fonctions

Si on dispose d’informations à priori sur le type de fonctions de f , il peut être intéressant d’utiliser une régression linéaire sur un ensemble de fonctions de base. On se donne un dictionnaire de K fonctions ϕ_j , et on cherche f de la forme :

$$f = \sum_{j=1}^K \alpha_j \phi_j$$

Ce qui revient à poser $\mathcal{H} = \operatorname{vect}(\phi_1, \phi_2, \dots, \phi_K)$

Il s’agit alors d’un problème de régression linéaire entre les vecteurs $[\phi_1(x_k), \phi_2(x_k), \dots, \phi_K(x_k)]$ qui peuvent être calculés et les y_k (pour k compris entre 1 et N).

Cependant, les méthodes de régression linéaire classiques telles que la méthode des moindres carrés deviennent très coûteuses en calcul, dès que la dimension de l’espace d’entrée devient importante, c’est-à-dire dans notre cas dès que le cardinal K du dictionnaire de fonctions devient importante : la méthode des moindres carrés notamment nécessite l’inversion d’une matrice de taille $K \times K$. Or lorsqu’on veut identifier une série de Volterra tronquée d’ordre P et de mémoire M , la base de fonctions la plus générale qui regroupe tous les monômes d’ordre inférieur ou égal à P est de l’ordre de M^P , ce qui rend très vite le problème posé incalculable.

5.2.3 Méthode à noyau reproduisant

L’idée sous-jacente des méthodes de régression dans les espace de Hilbert à noyau reproduisant est la suivante : on sait résoudre les problèmes de régression linéaire, donc en définissant un “mapping” qui transforme les entrées de telle sorte que les entrées transformées soit en relation linéaire avec les sorties, on sera capable de résoudre le problème de régression non-linéaire via une simple régression linéaire (cf. figure 5.1). Bien sur, une telle relation ne va généralement exister que quand la dimension de l’espace image de la transformation devient grande (voire infinie), mais la complexité des méthodes à noyau reproduisant ne dépend pas de la dimension de l’espace image mais uniquement du nombre de couples entrée/sortie (nommés exemples) dont on dispose, grâce au *couple du*

noyau, ce qui permet le calcul même lorsque l'espace transformé est de dimension infinie.

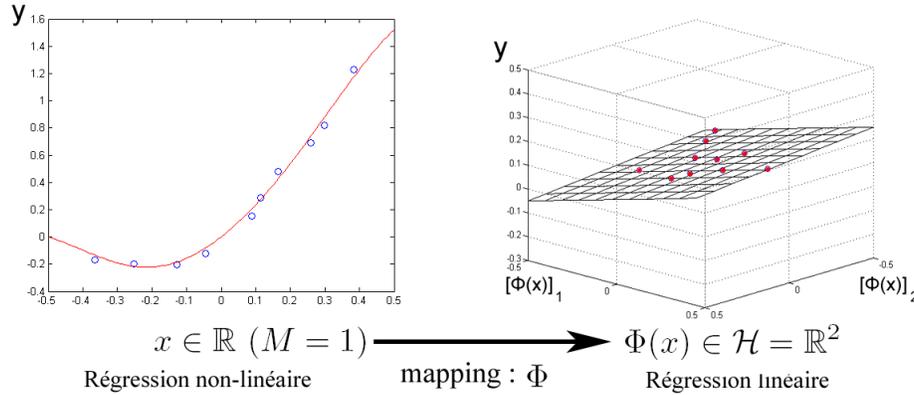


FIG. 5.1 – Mapping Φ

Noyaux reproduisants

Ce paragraphe constitue une très succincte introduction à la théorie des espaces de Hilbert à noyau reproduisant : il est nécessaire de se référer à des ouvrages bien plus complets comme [23] ou [21] pour de plus amples informations sur le sujet.

Définition 5.2.1 *On appelle noyau sur \mathbb{R}^M toute fonction continue symétrique de $\mathbb{R}^M \times \mathbb{R}^M$ dans \mathbb{R} :*

$$k : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R} \text{ avec } \forall(x, y) \ k(x, y) = k(y, x)$$

On dit que le noyau est défini positif si la matrice de Gram $K = (k(x_i, x_j))_{i,j}$ est définie positive pour tout n et pour tout (x_1, \dots, x_n) .

On peut trouver une liste de noyaux couramment utilisés dans [24].

Remarque : La théorie des espaces de Hilbert à noyau reproduisant définit les noyaux dans un cadre plus large : nous nous limiterons ici à des noyaux définis sur \mathbb{R}^M .

On notera $(\mathcal{H}, \langle, \rangle)$ un espace de Hilbert de fonctions définies sur \mathbb{R}^M et à valeur dans \mathbb{R} .

Définition 5.2.2 *On dit que k est un noyau reproduisant de \mathcal{H} s'il vérifie :*

- $\forall x \in \mathbb{R}^M$ l'application $y \mapsto k(x, y)$ (notée $k(x, \cdot)$ par commodité) appartient à \mathcal{H}

– $\forall x \in \mathbb{R}^M$ et $\forall \Psi \in \mathcal{H}$, on a : $\Psi(x) = \langle \Psi, k(x, \cdot) \rangle$

Théorème 5.2.1 *On peut démontrer que tout noyau défini positif est le noyau reproduisant d'un espace de Hilbert. Ce résultat découle directement du théorème de Mercer dont on peut trouver une démonstration dans [21].*

Coup du noyau et théorème de représentation Les propriétés utilisées dans les méthodes de régression à base de noyaux reproduisants découlent pour la plupart du *coup du noyau* couplé au théorème de représentation.

Le *coup du noyau* correspond à la constatation suivante :

$$k(x, y) = \langle k(x, \cdot), k(y, \cdot) \rangle$$

Cette formule s'obtient immédiatement à partir de la définition 5.2.2. On peut donc, à partir des valeurs du noyau, calculer le produit scalaire dans \mathcal{H} de toutes les fonctions $k(x, \cdot)$ et par conséquent de toutes les combinaisons linéaires de ces applications. Ce résultat est particulièrement intéressant car les produits scalaires restent calculables dans \mathcal{H} même si celui-ci est de dimension infinie.

Le *coup du noyau* est généralement utilisé conjointement au théorème de représentation :

Théorème 5.2.2 *Soit $C(f(x_1), \dots, f(x_N), y_1, \dots, y_N) + \lambda \|f\|^2$ une fonction de coût régularisée (C est une fonction convexe). L'élément f_0 de \mathcal{H} qui minimise la fonction de coût précédente peut s'écrire comme une combinaison linéaire des $k(x_i, \cdot)$:*

$$f_0 = \sum_{i=1}^N \alpha_i k(x_i, \cdot)$$

Série de Volterra et noyaux reproduisants

Les noyaux utilisés seront dans un premier temps issus d'un mapping explicite Φ de \mathbb{R}^M vers \mathcal{H} , c'est-à-dire qu'ils seront définis par :

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

On commence donc par définir un mapping de l'espace d'entrée dans un espace \mathcal{H} de dimension généralement beaucoup plus importante muni d'un produit scalaire (cf. figure 5.1) :

$$\Phi : x \in \mathbb{R}^M \mapsto \Phi \in \mathcal{H}$$

où \mathcal{H} est un espace de Hilbert, et on cherche une relation linéaire sur le nouvel ensemble d'exemples :

$$\{(\Phi(x_1), y_1), \dots, (\Phi(x_N), y_N)\}$$

On peut alors définir une fonction sur $\mathbb{R}^M \times \mathbb{R}^M$ à valeurs dans \mathbb{R} grâce au produit scalaire dans l'espace \mathcal{H} :

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$$

On peut démontrer que tout noyau défini positif s'exprime comme un produit scalaire dans un espace (cf. [21] ou [23]). On peut alors calculer les valeurs du noyau sans avoir à calculer les valeurs du mapping Φ .

L'expression de ce mapping est d'ailleurs généralement inconnue.

Un mapping explicite qui correspond aux série de Volterra est présenté dans [10]. Pour un système de Volterra homogène d'ordre p , le mapping suivant est utilisé :

$$\Phi_p(x) = (x(1)^p, x(1)^{p-1}x(2), \dots, x(1)^{p-1}x(M), x(1)^{p-2}x(2)^2, \dots, x(M)^p)$$

Le noyau correspondant est alors défini par :

$$k_p(x, y) = {}^t\Phi_p(x)\Phi_p(y) = ({}^txy)^p$$

Pour un système de Volterra global d'ordre P , on regroupe les mappings de tous les systèmes homogènes d'ordre de 1 à P en un seul mapping :

$$\Phi(x) = (a_0\Phi_0(x), a_1\Phi_1(x), \dots, a_P\Phi_P(x))$$

On obtient alors le noyau :

$$k(x, y) = {}^t\Phi(x)\Phi(y) = \sum_{p=0}^P a_p^2 ({}^txy)^p$$

Il existe des cas particuliers de ce noyau qui se calculent rapidement à partir du produit scalaire des entrées, notamment le noyau polynomial, qui s'obtient en prenant $a_k = \sqrt{\binom{P}{k}}$. On a alors :

$$k(x, y) = \sum_{p=0}^P \binom{P}{p} ({}^txy)^p = (1 + {}^txy)^P$$

Le mapping précédent se généralise facilement dans le cas d'un système de Volterra d'ordre infini. On peut obtenir alors, en utilisant par exemple $a_k = \sqrt{\frac{1}{k!}}$, le noyau suivant :

$$k(x, y) = \sum_{p=0}^{\infty} \frac{({}^txy)^p}{p!} = e^{{}^txy}$$

Il n'est bien entendu pas imposé d'utiliser un tel noyau : ces deux noyaux (polynomial et exponentiel) ne sont présentés qu'à titre indicatif car on peut les construire à partir d'un mapping explicite qui correspond aux séries de Volterra.

Méthodes de résolution

Les méthodes de résolution du problème de minimisation posé en introduction de cette section, sont des méthodes classiques d'optimisation convexe présentées notamment dans [21] et dans [2].

Il existe plusieurs méthodes qui dépendent directement de la fonction de coût utilisée :

- pour une fonction de coût quadratique : moindres carrés duaux
- pour une fonction de coût ϵ -sensible : machines à vecteurs de support

Moindres carrés duaux On utilise ici une fonction de coût quadratique. la fonction de coût est définie comme :

$$C(f(x_1), \dots, f(x_N), y_1, \dots, y_N) = \sum_{k=1}^N |f(x_k) - y_k|^2$$

On est amené à utiliser la méthode des moindres carrés duaux : il s'agit en fait de la méthode des moindres carrés calculées dans l'espace \mathcal{H} (espace dual). Grâce au théorème 5.2.2, on sait que la solution f qui minimise le coût sur nos N exemples s'écrit :

$$f = \sum_{i=1}^N \alpha_i k(x_i, \cdot)$$

Par conséquent :

$$\begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} = \begin{pmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \dots & \dots & \dots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \dots \\ \alpha_N \end{pmatrix}$$

Ainsi les α_i s'obtiennent par :

$$\begin{pmatrix} \alpha_1 \\ \dots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \dots & \dots & \dots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{pmatrix}^{-1} \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix}$$

Cependant quand le nombre d'exemples devient important, l'inversion de la matrice devient très coûteuse et il est nécessaire d'utiliser des méthodes de sélection des exemples ou de choix des directions principales (comme par exemple, kernel PCA, cf. [20]). Les méthodes à vecteurs de support qui utilisent une fonction de coût ϵ -sensible permettent de générer des ensembles réduits d'exemples (les vecteurs de support) ce qui est particulièrement intéressant dans notre cas où les exemples sont très nombreux.

Régression à vecteurs de support Dans le cas des machines à vecteur de support, on utilise des fonctions de coût ϵ -insensible comme par exemple :

$$C(f(x_1), \dots, f(x_N), y_1, \dots, y_N) = \sum_{k=1}^N |f(x_k) - y_k|_\epsilon$$

où $| \cdot |_\epsilon$ est défini par : $|y|_\epsilon = \max(0, y - \epsilon)$

Le problème s'écrit alors comme un problème d'optimisation sous contrainte (le détail n'est pas donné car son exposition nécessite beaucoup de place, se référer à [21]). La méthode de résolution s'appelle ϵ -SVR (pour régression à vecteurs de support en anglais). La résolution sépare les exemples en deux types : ceux pour lesquels l'erreur va être nulle (saturation de la contrainte) parce qu'ils appartiennent au tube de largeur ϵ et les autres, sur lesquels va s'exprimer la solution (les vecteurs de support). Pour illustrer la régression à vecteur de support, la

figure 5.2 montre la solution d'un problème de régression non-linéaire unidimensionnel en séparant les vecteurs à l'intérieur du tube des vecteurs de support. Une fois le problème résolu, la solution s'écrit sous la forme :

$$f(x) = \sum_{j \in SV} \lambda_j k(x_j, x) \quad (5.1)$$

Où SV est l'ensemble des vecteurs de support. La phase de calcul de SV et des λ_i à partir des exemples s'appelle entraînement. La phase de calcul des valeurs de f pour différent x s'appelle alors prédiction et permet de simuler un système mesuré.

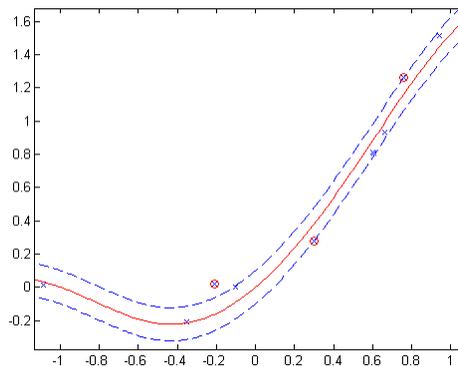


FIG. 5.2 – Régression à vecteurs de support avec noyau polynomial d'ordre 5 avec $\epsilon = 0.1$

Il existe des méthodes pour optimiser le choix de ϵ , notamment la ν -SVR

5.3 Utilisation et résultats

Pour utiliser la régression à noyaux reproduisants, nous avons implémenté la méthode des moindres carrés duaux en Matlab sans optimisation. Cela nous a permis d'identifier des systèmes simulés simples (dont l'ordre ne dépassait pas 10 et la mémoire 30) avec une assez bonne précision, mais ne nous a pas permis d'aller plus loin : en effet, l'inversion de la matrice sans optimisation ne permet pas de dépasser quelques milliers d'exemples (le nombre d'exemples nécessaires pour avoir des résultats corrects croît très vite avec l'ordre et la mémoire).

Nous avons donc utilisé des toolbox de régression à vecteurs de support optimisées disponibles gratuitement sur Internet qui permettent le calcul sur plusieurs dizaines de milliers d'exemples.

Après avoir testé plusieurs toolbox, nous avons fini par opter pour LibSVM qui semblait donner les meilleures performances tout en permettant d'effectuer une validation croisée pour optimiser les paramètres de la régression.

Choix du noyau

Dans ce chapitre nous avons présenté deux noyaux qui permettent de faire explicitement le lien avec le formalisme de Volterra (cf. section 5.2.3) : le noyau polynomial et le noyau exponentiel. Nous n'avons pu tester que le premier de ces deux noyaux, le second n'étant pas implémenté dans les toolbox que nous avons utilisées (il serait cependant intéressant de voir les résultats que celui-ci pourrait donner).

Comme on la vu à la section 5.2.3, le noyau polynomial ne peut identifier que des systèmes non-linéaires d'ordre fini inférieur au degré du noyau. Nous avons donc testé ce noyau avec des systèmes de Volterra simulés d'ordre inférieur au degré du noyau utilisé : les résultats ont été tout à fait corrects quand le degré du noyau était proche de l'ordre du système. Ainsi le noyau polynomial semble permettre d'identifier de tels systèmes. Cependant, nous avons pu constater que lorsque l'ordre du système et le degré du noyau devenaient très différents l'un de l'autre, le nombre de vecteurs de support augmentait considérablement (ce qui reflète que le noyau est "mal adapté" au système) et les résultats devenaient très mauvais : lors de la phase de prédiction effectuée sur des signaux musicaux, nous avons observé des explosions d'amplitudes des signaux, ce qui les rendait très difficile d'écoute. Ce phénomène d'explosion des amplitudes est présenté dans la figure 5.3 : le signal simulé (en bleu) voit son amplitude exploser en certains points. Ces résultats ont donc montré qu'il était impératif de pouvoir estimer assez précisément l'ordre du système pour pouvoir utiliser ce noyau et qu'il était tout à fait inadapté à la mesure de systèmes d'ordre élevé voire infini. Ce noyau était donc particulièrement inadapté à la mesure de systèmes réels, c'est pourquoi nous l'avons vite abandonné.

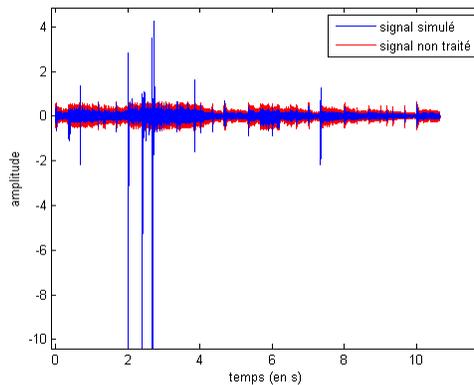


FIG. 5.3 – Défaut du noyau polynomial

Nous avons alors utilisé un noyau gaussien défini par :

$$k(x, y) = \exp(-\gamma \|x - y\|^2)$$

Ce noyau est très utilisé par les méthodes de classification à vecteurs de support,

et c'est pourquoi nous avons pensé à l'utiliser bien que le mapping qu'il définit soit implicite et que l'espace \mathcal{H} dans lequel on cherche nos solutions n'est pas bien connu. Il est très vite apparu que ce noyau n'avait pas les mêmes défauts de stabilité que le noyau polynomial. Nous avons effectué des mesures de différents systèmes simulés en utilisant ce noyau : les résultats ont été assez bons quel que soit l'ordre du système, même infini (le nombre d'exemples devenait cependant vite très important).

Nous avons ensuite essayé de mesurer un système réel logiciel qui simule une pédale de distorsion de guitare (Il s'agit du Tube Screamer de Simulanalog). Cet effet génère une distorsion nettement audible d'ordre important. Nous nous sommes limité à des réglages de distorsion assez faible : en effet pour un tel système le nombre d'exemples nécessaires était déjà de l'ordre de 50000 et la phase d'entraînement a nécessité plusieurs jours de calcul. Les résultats sont particulièrement encourageants. Nous avons en effet comparé à l'oreille des signaux simulés et des signaux traités par l'effet original : la différence est audible mais le grain produit par la distorsion semble très proche dans les deux signaux. Vu l'importance de la distorsion utilisée (celle-ci reste modérée mais est très nettement audible), la méthode apparaît comme très puissante par rapport aux méthodes présentées dans le chapitre 4.

Choix et nombre d'exemples

Une question importante qui s'est posé lorsque nous nous sommes servi de la régression à vecteurs de support a été le choix des exemples à utiliser pour la phase d'entraînement. Comme les exemples sont ici des morceaux de signal, il est assez difficile d'envisager de trouver un ensemble d'exemples représentatif de tous les signaux audio. Nous avons donc utilisé principalement deux types d'exemples : des morceaux de signaux musicaux et des signaux pseudo-aléatoires. Les morceaux de signaux musicaux n'ont en fait que peu d'intérêt : ils donnent généralement des résultats corrects uniquement sur les signaux dont ils ont été extraits mais de moins bons sur d'autres. C'est pourquoi nous avons également utilisé des signaux pseudo-aléatoires. Ces signaux pseudo-aléatoires ont été construits de sorte qu'ils aient des caractéristiques proches de signaux audio réels (même type de densité de probabilité, même type de densité spectrale).

Dans tous les cas, un même problème a été rencontré : Dans l'état actuel de nos méthodes, celles-ci nécessitent un nombre très important d'exemples et gardent un nombre également très important de vecteurs de support. En effet, on peut considérer qu'un effet audio est convenablement reproduit lorsqu'il est difficile de reconnaître un signal traité par ce système d'un signal simulé : la régression nécessite donc une très grande précision et l'erreur doit être extrêmement faible pour que la simulation soit bonne. Le principal problème engendré par ce nombre important d'exemples est que la phase d'entraînement peut nécessiter plusieurs jours de calcul (ce qui n'est pas si grave que ça puisque celle-ci n'est réalisée qu'une fois) et surtout que la phase de simulation (prédiction) peut nécessiter jusqu'à plusieurs heures par seconde de signal traité (la durée de la simulation dépend directement du nombre de vecteurs de support comme le montre l'équation (5.1)) :

on est donc très loin d'une simulation de l'effet en temps réel qui n'est donc pour l'instant pas du tout envisageable. Il pourrait donc être intéressant de chercher des méthodes qui permettent de "factoriser" la structure du système à partir des coefficients de régression (et des exemples) : l'idée serait par exemple de pouvoir extraire une structure simplifiée (du type des structures présentées dans 3.2.1) des résultats de la régression afin de pouvoir simuler en temps réel les systèmes mesurés. Cette idée est tout à fait réalisable quand on connaît explicitement l'espace \mathcal{H} (par exemple quand on utilise un noyau polynomial ou exponentiel : cf. section 5.2.3) mais s'avère bien plus complexe pour le noyau gaussien.

Conclusion et perspectives

Nous avons cherché tout au long de ce stage à trouver et évaluer des méthodes d'identification de système en vue de la réalisation future d'un sampler d'effets le plus général possible : nous avons en partie travaillé sur les systèmes linéaires variants ce qui nous a très vite donné des résultats satisfaisants et nous avons principalement travaillé sur les systèmes non-linéaires invariants qui semblent beaucoup plus difficiles à identifier de façon générale.

Nos études des systèmes linéaires variants sont restées succinctes mais très intéressantes : il semble qu'il s'agisse d'un secteur plus facile d'accès que les systèmes non-linéaires et qui peut donner des résultats plus facilement. Cette partie de notre étude pourrait donc être développée dans l'idée de faire un sampler d'effets paramétrique : l'idée serait de pouvoir extraire d'un nombre fini de mesures une fonction qui à partir de n'importe quelle valeur du paramètre (éventuellement multidimensionnel) de contrôle donne une réponse impulsionnelle qui soit très proche de celle du système réel pour cette valeur du paramètre.

Pour réaliser cette idée, il pourrait être envisageable d'utiliser des méthodes de régression à vecteurs de support pour établir la fonction non-linéaire entre le paramètre et les réponses impulsionnelles.

Il pourrait également être intéressant de pouvoir décrire un modèle intermédiaire entre les valeurs du paramètre et les réponses impulsionnelles qui permettrait de rendre compte intelligemment de la structure de l'effet et par conséquent de calculer simplement les réponses impulsionnelles pour chaque valeur du paramètre.

Un autre aspect qui pourrait être abordé est le problème des variations rapides (quand on ne peut plus faire la supposition que le temps caractéristique de variation de la réponse impulsionnelle est court devant la taille de ces réponses) et éventuellement non contrôlées (c'est à dire quand l'utilisateur n'est pas maître du paramètre mais que le temps ou un descripteur audio entre en jeu). Les méthodes de mesure devraient alors être adaptées pour pouvoir tenir compte de cette variation.

Lors de nos recherches et de nos tests de méthodes d'identification de systèmes non-linéaires, nous nous sommes heurté à de nombreux problèmes que nous avons cherchés à expliquer dans ce rapport en vu d'études futures : un des premiers problèmes était la grande complexité des systèmes de Volterra globaux qui

possèdent un nombre souvent beaucoup trop important de coefficients pour être traités directement. Les méthodes d'identification de systèmes non-linéaires que nous avons mises en oeuvre ont révélé pour la plupart d'importantes faiblesses qui les rendaient souvent inutilisables dans le domaine de l'audio. Les faiblesses de ces méthodes venaient le plus souvent de la troncation des séries de Volterra : l'identification polynomiale pose en effet d'importants problèmes notamment d'instabilité numérique qui ne la rend utilisable que sur des systèmes très faiblement non-linéaires. Il serait donc intéressant de s'orienter plutôt vers des méthodes qui ne tronquent pas en ordre.

La régression à vecteurs de support semble prometteuse dans cette voie mais de nombreux aspects sont encore à étudier : le choix du noyau utilisé pour la SVR par exemple qui pourrait mieux rendre compte de la structure interne du système et ainsi permettrait d'avoir moins de vecteurs de supports. De plus, il faudrait pouvoir trouver des méthodes qui permettent une simulation en temps réel par exemple en extrayant une structure simplifiée à partir des résultats de la régression.

Annexe A

Simulation numérique des systèmes non-linéaires

Si la simulation discrète de système linéaire continu ne pose pas de problèmes théoriques particuliers dès que la fréquence d'échantillonnage utilisée permet de respecter le théorème de Nyquist/Shannon, la simulation de systèmes non-linéaires s'avère plus complexe tant sur le plan théorique que pratique.

A.1 Repliement spectral

Contrairement aux systèmes linéaires pour lesquels le signal de sortie ne peut contenir que des fréquences contenues dans le signal d'entrée, les systèmes non-linéaires peuvent générer de nouvelles fréquences. Voici un exemple très simple. Si on prend le signal d'entrée suivant :

$$x(t) = \cos(\omega t)$$

La sortie d'un système linéaire quelconque ne pourra contenir que la fréquence ω . En revanche, si on prend le système non-linéaire sans mémoire qui a pour sortie le carré de l'entrée, alors la sortie de ce système s'écrit :

$$y(t) = \cos(\omega t)^2 = \frac{\cos(2\omega t)}{2} + \frac{1}{2}$$

La sortie contient donc de nouvelles fréquences qui n'étaient pas contenues dans le signal d'entrée : la fréquence nulle et 2ω .

On peut démontrer assez facilement qu'un système de Volterra d'ordre P peut générer à la sortie des fréquences jusqu'à P fois plus grande que la plus grande des fréquences contenues dans l'entrée.

Lorsqu'on veut simuler des systèmes continus en discret, on se heurte alors à un problème important : pour des systèmes d'ordre élevé, les fréquences générées par les non-linéarités dépassent vite la fréquence de Nyquist et par conséquent, si aucune précaution n'est prise le phénomène de repliement spectral apparaît.

A.2 Méthodes générales de simulation

Il est généralement impossible de supprimer totalement le repliement spectral, notamment dans le cas de système d'ordre infini. Il est cependant souvent possible de le limiter afin de le rendre inaudible.

Pour diminuer le repliement, on peut augmenter la bande passante en sur-échantillonnant le signal d'entrée, on applique alors la partie non-linéaire, puis on sous-échantillonne la sortie de la partie non linéaire. En sur-échantillonnant d'un facteur assez grand, le repliement devient assez faible pour être masqué par le reste du signal et donc inaudible.

Le facteur minimum de sur-échantillonnage nécessaire à la suppression du repliement pour un système d'ordre P est donné par $f = E(P/2) + 1$ ou E désigne la partie entière. Ce résultat est assez simple à démontrer et est illustré par la figure A.1 : le système considéré est un système d'ordre 5 qui génère donc pour une entrée sinusoïdale 5 composantes. En sur-échantillonnant d'un facteur 5, on est assuré de ne pas générer de repliement. Mais en sur-échantillonnant seulement d'un facteur 3, on ne génère du repliement qu'à l'extérieur de la bande utile (qui est la bande passante du signal d'entrée). Par conséquent il suffit de sur-échantillonner d'un facteur 3 l'entrée pour ne pas obtenir de repliement dans le signal de sortie.

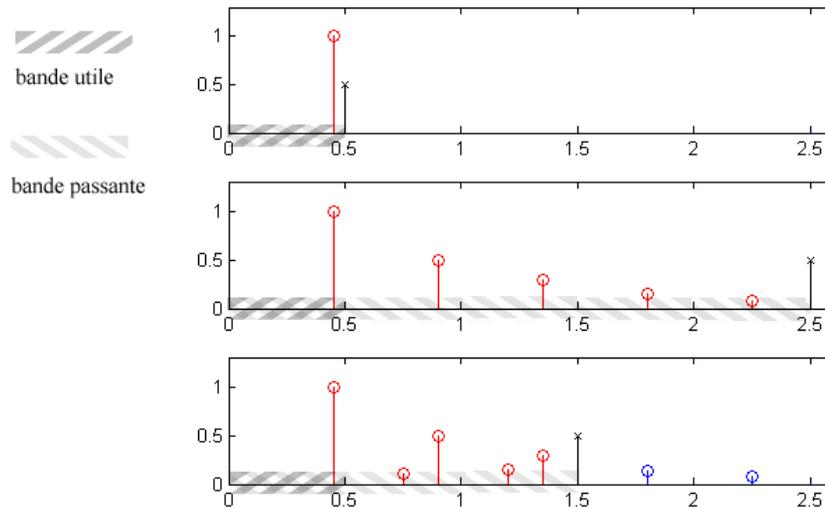


FIG. A.1 – facteur de rééchantillonnage (domaine fréquentiel). De haut en bas : entrée, sortie avec sur-échantillonnage fois 5, sortie avec sur-échantillonnage fois 3

Le sur-échantillonnage a cependant le gros défaut de nécessiter de très importantes ressources en calcul : lorsque le facteur de sur-échantillonnage devient élevé, les coûts de calcul deviennent très importants.

Dans le cadre de simulation en temps réel, il est donc parfois nécessaire de faire des compromis importants sur le son pour que la simulation reste calculable : une méthode couramment utilisée (décrite dans [26]) consiste à filtrer les différents ordres afin de supprimer les fréquences du signal d'entrée qui pourrait générer du repliement dans le signal de sortie. Ainsi, en filtrant à la moitié de la bande passante l'entrée, on est sûr d'avoir supprimé le repliement dans le sous-système homogène d'ordre 2.

A.3 Système de Volterra

Il a déjà été mentionné plusieurs fois qu'un système de Volterra nécessite généralement beaucoup trop de calcul pour être identifié intégralement. Il en est évidemment de même pour la simulation une fois le système identifié, d'autant plus qu'on veut souvent pouvoir simuler un système en temps réel, ce qui impose de très importantes contraintes sur le temps de calcul : ainsi, la puissance des processeurs actuels ne permet une simulation complète que pour des systèmes de Volterra d'ordre et de mémoire très réduite : il est difficile d'envisager la simulation en temps réel d'un système complet d'ordre supérieur à 2 dès que la mémoire du système dépasse 100 échantillons.

A.4 Simulation de systèmes par bloc

Dans cette section est présentée une méthode de simulation de systèmes par bloc. Cette méthode est présentée pour le système par bloc présenté dans le chapitre 3.2.1 le plus complexe - le système S_M - mais peut s'appliquer à tous les autres systèmes présentés. La chaîne de simulation complète est présentée dans la figure A.2 : le système est décomposé en sous-systèmes homogènes et l'algorithme est le même pour chaque branche (sauf pour l'ordre 1 qui se traite comme un système linéaire classique). On note M la taille des filtres d'entrée et de sortie (on les suppose de même taille) de la branche d'ordre p , N la taille du signal d'entrée et $K = E(p/2) + 1$.

La simulation s'effectue de la manière suivante :

1. convolution rapide (bloc 1) pour le filtre d'entrée
2. sur-échantillonnage d'un facteur K (bloc 2 et 3), cf. section A.1 pour le choix de K
3. mise à l'exposant p du signal sur-échantillonné (bloc 4)
4. sous-échantillonnage d'un facteur K (bloc 5 et 6)
5. convolution rapide (bloc 7) pour le filtre de sortie

Complexité

Les complexités des différents blocs sont représentées dans la figure A.2 :

IV ANNEXE A. SIMULATION NUMÉRIQUE DES SYSTÈMES NON-LINÉAIRES

- on suppose pour simplifier que M est une puissance de 2 et que M divise N , ce qui permet d'obtenir des convolutions rapides en $O(N \log(M))$ en découpant le signal d'entrée en N/M blocs de taille M
- le sur-échantillonnage et le sous-échantillonnage ont une complexité linéaire en la taille du signal filtré (le filtre utilisé est indépendant des paramètres du problème et on ne fait ici pas apparaître sa longueur dans la complexité, on peut également utiliser une décomposition polyphase pour réduire ce facteur)
- la mise à l'exposant peut s'effectuer de manière rapide en $O(\log_2(p))$ par échantillon.

La simulation complète de la branche d'ordre p a donc pour complexité : $O(N \log M + Np \log(p))$. Cette complexité reste d'ailleurs valable pour l'ordre 1. La complexité globale de la simulation d'un système d'ordre P a donc pour complexité (en considérant par souci de simplicité que tous les filtres ont la même taille) :

$$O\left(\sum_{p=1}^P \{N \log_2 M + Np \log_2(p)\}\right) \leq O(NP(\log_2 M + P \log_2 P))$$

La dépendance en l'ordre P est donc importante, ce qui est dû au sur-échantillonnage.

En pratique, lorsqu'on veut simuler un tel système en temps réel on est très vite limité par la puissance des processeurs lorsque la taille des filtres M et l'ordre P deviennent importants. En général on est obligé de se limiter à un sur-échantillonnage d'un facteur 8 (à une fréquence d'échantillonnage de 44100Hz) quitte à avoir du repliement : on suppose qu'il est assez faible pour être masqué par les autres fréquences et être inaudible.

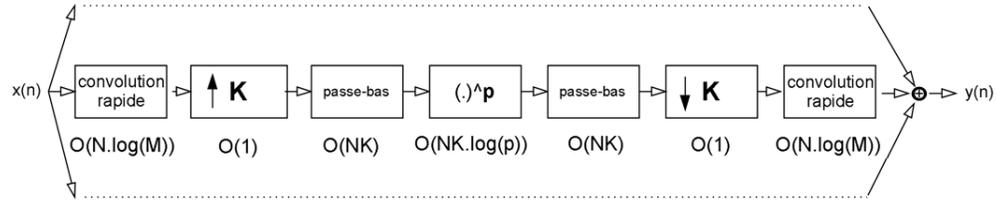


FIG. A.2 – chaîne de simulation de l'ordre p d'un système S_M .

Annexe B

Equivalence *schéma-bloc* / Volterra

Cette annexe présente les calculs permettant d'obtenir les noyaux de Volterra des systèmes simplifiés de type *schéma-bloc* présentés dans la section 3.2.1.

B.1 Système de Hammerstein

Un système de Hammerstein est constitué d'une non-linéarité statique F , suivie d'un filtre linéaire h .

Lorsque la non-linéarité est développable en série entière, on peut écrire :

$$v(n) = F(x(n)) = \sum_{p=1}^{\infty} \alpha_p x(n)^p$$

Dans de bonnes hypothèses de convergence, la sortie du système peut s'écrire :

$$y(n) = h * F(x(n)) = \sum_{m=0}^{\infty} h(m) \sum_{p=1}^{\infty} \alpha_p x(n-m)^p = \sum_{p=1}^{\infty} \sum_{m=0}^{\infty} \alpha_p h(m) x(n-m)^p$$

Le système est donc équivalent au système de Volterra de noyaux :

$$f_p(n_1, \dots, n_p) = \alpha_p h(n_1) \delta(n_1 - n_2) \delta(n_1 - n_3) \dots \delta(n_1 - n_p)$$

où δ est la fonction de Kronecker.

B.2 Système de Wiener

Un système de Wiener est constitué d'un filtre linéaire g , suivie d'une non-linéarité statique F .

Lorsque la non-linéarité est développable en série entière, on peut écrire :

$$y(n) = F(x(n)) = \sum_{p=1}^{\infty} \alpha_p u(n)^p$$

Dans de bonnes hypothèses de convergence, la sortie du système peut s'écrire :

$$\begin{aligned} y(n) &= F(g * x(n)) = \sum_{p=1}^{\infty} \alpha_p \left(\sum_{m=0}^{\infty} g(m)x(n-m) \right)^p \\ &= \sum_{p=1}^{\infty} \sum_{n_1, n_2, \dots, n_p=0}^{\infty} \alpha_p g(n_1) \dots g(n_p) x(n-n_1) \dots x(n-n_p) \end{aligned}$$

Le système est donc équivalent au système de Volterra de noyaux :

$$f_p(n_1, \dots, n_p) = \alpha_p g(n_1)g(n_2) \dots g(n_p)$$

B.3 Système de Wiener-Hammerstein

Un système de Wiener-Hammerstein est la mise en cascade d'un système de Wiener et d'un système de Hammerstein.

Lorsque la non-linéarité est développable en série entière, on peut écrire :

$$v(n) = F(x(n)) = \sum_{p=1}^{\infty} \alpha_p u(n)^p$$

Dans de bonnes hypothèses de convergence, la sortie du système peut s'écrire :

$$y(n) = h * F(g * x(n)) = \sum_{m_1=-\infty}^{\infty} h(m_1) \sum_{p=1}^{\infty} \alpha_p \left(\sum_{m_2=-\infty}^{\infty} g(m_2)x(n-m_1-m_2) \right)^p$$

D'où :

$$y(n) = \sum_{p=1}^{\infty} \alpha_p \sum_{m_1=-\infty}^{\infty} h(m_1) \sum_{n_1 \dots n_p=-\infty}^{\infty} g(n_1) \dots g(n_p) x(n-m_1-n_1) \dots x(n-m_1-n_p)$$

Par un changement de variable on obtient :

$$y(n) = \sum_{p=1}^{\infty} \sum_{n_1 \dots n_p=-\infty}^{\infty} \left(\alpha_p \sum_{m_1=-\infty}^{\infty} h(m_1) g(n_1-m_1) \dots g(n_p-m_1) \right) x(n-n_1) \dots x(n-n_p)$$

En supposant h causal, le système est donc équivalent au système de Volterra de noyaux :

$$f_p(n_1, \dots, n_p) = \alpha_p \sum_{m=0}^{\infty} h(m) g(n_1-m) \dots g(n_p-m)$$

La représentation des systèmes de Hammerstein, Wiener et Wiener-Hammerstein n'est pas unique : on peut par exemple pour un système de Hammerstein remplacer le filtre h et la fonction F par λh et $\frac{1}{\lambda}F$ et le système reste inchangé. Il peut donc être nécessaire d'imposer des conditions supplémentaires sur F , g et h afin de garantir l'unicité de la représentation du système.

B.4 Système de Hammerstein généralisé

On définit les systèmes de Hammerstein généralisés (on trouve aussi dans la littérature l'expression Série de Hammerstein) comme un ensemble de branches homogènes mises en parallèle constituées chacune d'une mise à l'exposant suivie d'un filtre linéaire.

Par un calcul analogue a celui du système de Hammerstein simple (cf. B.1), on montre que le système est équivalent au système de Volterra de noyaux :

$$f_p(n_1, \dots, n_p) = h_p(n_1)\delta(n_1 - n_2)\delta(n_1 - n_3)\dots\delta(n_1 - n_p)$$

Il s'agit donc des systèmes dont les noyaux de Volterra sont diagonaux.

B.5 Système de Wiener généralisé

On définit les systèmes de Wiener généralisé comme un ensemble de branches homogènes mises en parallèle constituées chacune d'un filtre linéaire suivi d'une mise à l'exposant.

Par un calcul analogue a celui du système de Wiener simple (cf B.2), on montre que le système est équivalent au système de Volterra de noyaux :

$$f_p(n_1, \dots, n_p) = g_p(n_1)g_p(n_2)\dots g_p(n_p)$$

B.6 Système S_M

On définit les systèmes S_M comme un ensemble de branches homogènes mises en parallèle constituées chacune d'un filtre linéaire suivi d'une mise à l'exposant suivie d'un filtre linéaire.

La représentation d'un système S_M donnée par le tableau de synthèse de la section 3.2.1 n'est pas unique : on peut, par exemple, remplacer les filtres g_k et h_k par $\frac{1}{\lambda}g_k$ et $\lambda^k h_k$ et le système reste inchangé. Il peut donc être nécessaire d'imposer des conditions supplémentaires sur ces filtres afin de garantir l'unicité de la représentation du système.

Par un calcul analogue a celui du système de Wiener-Hammerstein simple (cf B.3), on montre que le système est équivalent au système de Volterra de noyaux :

$$f_p(n_1, \dots, n_p) = \alpha_p \sum_{m=0}^{\infty} h_p(m)g_p(n_1 - m)\dots g_p(n_p - m)$$

Annexe C

Calcul pour l'identification de systèmes S_M

C.1 Identification de systèmes S_M à l'aide de sweeps

Dans cette section, on présente le calcul détaillé de l'identification de système S_M à l'aide de sweeps, dont les résultats sont présentés dans la section 4.4.4.

Dans le cas général, la sortie d'un système S_M alimenté par un signal $s(t)$ est donnée par ;

$$r(t) = g_1(t) * s(t) + \sum_{p=2}^P h_p(t) * [g_p(t) * s(t)]^p$$

Remarque Comme déjà évoqué dans la section 3.2.1, cette description n'est pas unique et il faut donc imposer une condition supplémentaire pour avoir l'unicité de h_p et g_p . En pratique, on impose que la transformée de Fourier de g_p soit nulle en 0.

Si on prend comme signal d'entrée un sweep logarithmique (cf. 4.4.2), on obtient en sortie :

$$r(t) = \sum_{p=1}^P h_p(t) * [g_p(t) * \cos(\Phi_c(t))]^p$$

On note $r_p(t)$ alors chaque terme de cette somme :

$$r_p(t) = h_p(t) * [g_p(t) * \cos(\Phi_c(t))]^p$$

On peut estimer facilement ces termes en utilisant plusieurs mesures à différentes amplitudes grâce à la méthode de séparation des ordres présentée dans la section 4.1.

En utilisant la formule d'Euler :

$$r_p(t) = h_p(t) * \left[g_p(t) * \left(\frac{e^{i\Phi_c(t)} + e^{-i\Phi_c(t)}}{2} \right) \right]^p$$

$$r_p(t) = 2^{-p} \cdot h_p(t) * \left[g_p(t) * \left(e^{i\Phi_c(t)} + e^{-i\Phi_c(t)} \right) \right]^p$$

En utilisant la formule du binôme de Newton :

$$r_p(t) = 2^{-p} \cdot h_p(t) * \sum_{k=0}^p \binom{p}{k} (g_p(t) * e^{-i\Phi_c(t)})^k (g_p(t) * e^{i\Phi_c(t)})^{p-k}$$

On peut considérer que le signal d'entrée se comporte localement comme une sinusoïde pure de fréquence $\omega(t)$, et on peut donc écrire :

$$g_p(t) * e^{i\Phi_c(t)} = \hat{g}_p(\omega(t)) e^{i\Phi_c(t)}$$

$$g_p(t) * e^{-i\Phi_c(t)} = \hat{g}_p(-\omega(t)) e^{-i\Phi_c(t)}$$

où : $\hat{g}_p(\omega)$ est la transformée de Fourier de $g_p(t)$

Donc :

$$r_p(t) = 2^{-p} \cdot h_p(t) * \sum_{k=0}^p \binom{p}{k} \hat{g}_p(-\omega(t))^k e^{-ik\Phi_c(t)} \hat{g}_p(\omega(t))^{p-k} e^{i(p-k)\Phi_c(t)}$$

$$r_p(t) = \left(\frac{\alpha}{2}\right)^p \sum_{k=0}^p \binom{p}{k} \hat{g}_p(-\omega(t))^k \hat{g}_p(\omega(t))^{p-k} h_p(t) * e^{i(p-2k)\Phi_c(t)}$$

Or :

$$h_p(t) * e^{i(p-2k)\Phi_c(t)} = \hat{h}_p((p-2k)\omega(t)) e^{i(p-2k)\Phi_c(t)}$$

où : $\hat{h}_p(\omega)$ est la transformée de Fourier de $h_p(t)$

Donc

$$r_p(t) = 2^{-p} \sum_{k=0}^p \binom{p}{k} \hat{g}_p(-\omega(t))^k \hat{g}_p(\omega(t))^{p-k} \hat{h}_p((p-2k)\omega(t)) e^{i(p-2k)\Phi_c(t)}$$

Comme $\hat{g}_p(\omega) = \hat{g}_p^*(-\omega)$ et $\hat{h}_p(\omega) = \hat{h}_p^*(-\omega)$, on a :

$$r_p(t) = 2^{-p} \sum_{k=0}^p \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]}$$

Où $\angle \hat{g}_p(\omega(t))$ et $\angle \hat{h}_p(\omega(t))$ représentent respectivement les arguments de $\hat{g}_p(\omega(t))$ et de $\hat{h}_p(\omega(t))$.

cas p pair On peut écrire :

$$\begin{aligned} r_p(t) &= 2^{-p} \binom{p}{p/2} |\hat{g}_p(\omega(t))|^p \hat{h}_p(0) \\ &+ 2^{-p} \sum_{k=0}^{p/2-1} \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]} \\ &+ 2^{-p} \sum_{k=p/2+1}^p \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]} \end{aligned}$$

Par un changement de variable dans la deuxième somme, on obtient :

$$\begin{aligned}
 r_p(t) &= 2^{-p} \binom{p}{p/2} |\hat{g}_p(\omega(t))|^p \hat{h}_p(0) \\
 &+ 2^{-p} \sum_{k=0}^{p/2-1} \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]} \\
 &+ 2^{-p} \sum_{k=0}^{p/2-1} \binom{p}{p-k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((2k-p)\omega(t)) \right| e^{i[(2k-p)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((2k-p)\omega(t))]}
 \end{aligned}$$

c'est-à-dire :

$$\begin{aligned}
 r_p(t) &= 2^{-p} \binom{p}{p/2} |\hat{g}_p(\omega(t))|^p \hat{h}_p(0) \\
 &+ 2^{-p} \sum_{k=0}^{p/2-1} \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]} \\
 &+ 2^{-p} \sum_{k=0}^{p/2-1} \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{-i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]}
 \end{aligned}$$

Par conséquent :

$$r_p(t) = 2^{-p} \binom{p}{p/2} |\hat{g}_p(\omega(t))|^p \hat{h}_p(0) + 2^{-p} \sum_{k=0}^{p/2-1} 2 \binom{p}{k} \psi(\omega(t)) \cos(\varphi(\omega(t)))$$

avec :

$$\begin{aligned}
 \psi(\omega(t)) &= \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| \\
 \varphi(\omega(t)) &= (p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t)).
 \end{aligned}$$

cas p impair Le calcul est similaire au cas p pair sans terme constant :

$$\begin{aligned}
 r_p(t) &= 2^{-p} \sum_{k=0}^{(p-1)/2} \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]} \\
 &+ 2^{-p} \sum_{k=(p+1)/2}^p \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]}
 \end{aligned}$$

Par un changement de variable dans la deuxième somme :

$$\begin{aligned}
 r_p(t) &= 2^{-p} \sum_{k=0}^{(p-1)/2} \binom{p}{k} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| e^{i[(p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t))]} \\
 &+ 2^{-p} \sum_{k=0}^{(p-1)/2} \binom{p}{k-p} \left| \hat{g}_p(\omega(t))^p \hat{h}_p((2k-p)\omega(t)) \right| e^{i[(2k-p)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((2k-p)\omega(t))]}
 \end{aligned}$$

XII ANNEXE C. CALCUL POUR L'IDENTIFICATION DE SYSTÈMES S_M

Par conséquent :

$$r_p(t) = 2^{-p} \sum_{k=0}^{(p-1)/2} 2 \binom{p}{k} \psi(\omega(t)) \cos(\varphi(\omega(t)))$$

avec :

$$\begin{aligned} \psi(\omega(t)) &= \left| \hat{g}_p(\omega(t))^p \hat{h}_p((p-2k)\omega(t)) \right| \\ \varphi(\omega(t)) &= (p-2k)(\Phi_c(t) + \angle \hat{g}_p(\omega(t))) + \angle \hat{h}_p((p-2k)\omega(t)). \end{aligned}$$

En convoluant $r_p(t)$ par le signal $x_r(t)$ inverse au sens de la convolution du sweep d'entrée, on obtient alors des pseudo-réponses impulsionnelles correspondant à chaque harmonique décalées dans le temps les unes par rapport aux autres (phase analogue à la méthode de Farina, cf. 4.4.3). Les pseudo-réponses fréquentielles associées sont obtenues à partir de :

$$H_{p,k}((p-2k)\omega) = 2^{1-p} \binom{p}{k} \hat{h}_p((p-2k)\omega) \hat{g}_p^p(\omega)$$

Ce qui donne, pour $k = 1 \dots E((p-1)/2)$:

$$H_{p,k}(\omega) = 2^{1-p} \binom{p}{k} \hat{h}_p(\omega) \hat{g}_p^p(\omega/(p-2k))$$

En passant au logarithme, on obtient alors :

$$\ln(H_{p,k}(\omega)) = (p-1)\ln 2 + \ln \binom{p}{k} + \ln(\hat{h}_p(\omega)) + p \ln(\hat{g}_p(\omega)/(p-2k))$$

Remarque : La formule précédente reste valable en prenant le logarithme complexe. Il peut cependant être préférable de considérer les valeurs absolues des différentes réponses fréquentielles et d'utiliser le logarithme réel : on perdra cependant des informations sur la phase.

Or on connaît les $H_{p,k}$. On dispose donc d'un système linéaire en les $\ln(\hat{h}_p(\omega))$ et les $\ln(\hat{g}_p(\omega))$. On peut donc obtenir une estimation de $\hat{h}_p(\omega)$ et de $\hat{g}_p(\omega)$. La résolution du système est détaillée ici pour l'ordre 3. Pour l'ordre 3, on dispose pour toute fréquence ω du système :

$$\begin{cases} \ln(H_{3,0}(\omega)) - 2\ln 2 = \ln(\hat{h}_3(\omega)) + 3\ln(\hat{g}_3(\omega/3)) \\ \ln(H_{3,1}(\omega)) - 2\ln 2 - \ln(3) = \ln(\hat{h}_3(\omega)) + 3\ln(\hat{g}_3(\omega)) \end{cases}$$

Le problème réside dans le terme $3\ln(\hat{g}_3(\omega/3))$ qui met en jeu une homothétie sur les fréquences. On commence par discrétiser les fréquences en $\omega = \omega_0 \dots \omega_{K-1}$, avec $\omega_i = \frac{i\pi}{K}$. Les fréquences $\omega_i/3$ seront obtenues par rééchantillonnage. On peut alors écrire le système global contenant les équations pour chaque ω_i de la façon suivante :

$$\begin{pmatrix} \ln(H_{3,0}(\omega_0)) - 2\ln 2 \\ \dots \\ \ln(H_{3,0}(\omega_{K-1})) - 2\ln 2 \\ \ln(H_{3,1}(\omega_0)) - 2\ln 2 - \ln 3 \\ \dots \\ \ln(H_{3,1}(\omega_{K-1})) - 2\ln 2 - \ln 3 \end{pmatrix} = \begin{pmatrix} I_K & 3R_{K,3} \\ I_K & 3I_K \end{pmatrix} \begin{pmatrix} \ln(\hat{h}_3(\omega_0)) \\ \dots \\ \ln(\hat{h}_3(\omega_{K-1})) \\ \ln(\hat{g}_3(\omega_0)) \\ \dots \\ \ln(\hat{g}_3(\omega_{K-1})) \end{pmatrix}$$

Où I_K est la matrice identité de taille K et où $R_{K,3}$ est une matrice de rééchantillonnage d'ordre 3 de taille $K \times K$, c'est-à-dire telle que :

$$\begin{pmatrix} \ln(\hat{g}_3(\omega_0/3)) \\ \dots \\ \ln(\hat{g}_3(\omega_{K-1}/3)) \end{pmatrix} = R_{K,3} \begin{pmatrix} \ln(\hat{g}_3(\omega_0)) \\ \dots \\ \ln(\hat{g}_3(\omega_{K-1})) \end{pmatrix}$$

On peut alors inverser le système précédent :

$$\begin{pmatrix} \ln(\hat{h}_3(\omega_0)) \\ \dots \\ \ln(\hat{h}_3(\omega_{K-1})) \\ \ln(\hat{g}_3(\omega_0)) \\ \dots \\ \ln(\hat{g}_3(\omega_{K-1})) \end{pmatrix} = \begin{pmatrix} I_K & 3R_{K,3} \\ I_K & 3I_K \end{pmatrix}^{-1} \begin{pmatrix} \ln(H_{3,0}(\omega_0)) - 2\ln 2 \\ \dots \\ \ln(H_{3,0}(\omega_{K-1})) - 2\ln 2 \\ \ln(H_{3,1}(\omega_0)) - 2\ln 2 - \ln 3 \\ \dots \\ \ln(H_{3,1}(\omega_{K-1})) - 2\ln 2 - \ln 3 \end{pmatrix} \quad (\text{C.1})$$

On a donc identifié par cette méthode les coefficients des filtres h_3 et g_3 . Pour les autres ordres la méthode est analogue à cela près :

- pour les ordres supérieurs à 4 on dispose d'un système surdéterminé qui se résout en utilisant la pseudo-inverse à la place de l'inverse dans l'équation (C.1).
- l'ordre 2 est sous déterminé et il faut une mesure complémentaire afin de pouvoir identifier h_2 et g_2 .
- Pour tous les ordres pairs, on ne peut déterminer la réponse fréquentielle du filtre d'entrée que jusqu'à la moitié de la bande passante.

C.2 Identification de l'ordre 2

Afin d'estimer les filtres d'entrée et de sortie de l'ordre 2 (modèle S_M), on utilise en signal d'entrée la somme d'un sweep logarithmique et d'un sweep à l'octave du premier :

$$x(t) = \cos(\Phi_c(t)) + \cos(2\Phi_c(t))$$

La phase $\Phi_c(t)$ utilisée est explicitée dans 4.4.2.

On isole alors l'ordre 2 grâce à la méthode proposée en 4.1. La composante d'ordre 2 peut également être calculée dans le cas du modèle S_M :

$$w_2(t) = h_2 * (g_2 * x(t))^2$$

XIV ANNEXE C. CALCUL POUR L'IDENTIFICATION DE SYSTÈMES S_M

On commence par développer les cosinus à l'aide de la formule d'Euler :

$$x(t) = \frac{1}{2} \left(e^{i\Phi_c(t)} + e^{-i\Phi_c(t)} + e^{2i\Phi_c(t)} + e^{-2i\Phi_c(t)} \right)$$

Comme la fréquence de chaque exponentielle complexe varie lentement, on peut considérer qu'elle est localement constante et on peut écrire :

$$g_2(t) * e^{i\Phi_c(t)} = \hat{g}_2(\omega(t)) e^{i\Phi_c(t)} \quad (\text{C.2})$$

où : $\hat{g}_2(\omega)$ est la transformée de Fourier de $g_2(t)$. Par conséquent :

$$g_2 * x(t) = \frac{1}{2} \left(\hat{g}_2(\omega(t)) e^{i\Phi_c(t)} + \hat{g}_2(-\omega(t)) e^{-i\Phi_c(t)} + \hat{g}_2(2\omega(t)) e^{i2\Phi_c(t)} + \hat{g}_2(-2\omega(t)) e^{-i2\Phi_c(t)} \right)$$

On en déduit :

$$\begin{aligned} (g_2 * x(t))^2 &= \frac{1}{4} \left(\hat{g}_2^2(\omega(t)) e^{i2\Phi_c(t)} + \hat{g}_2^2(-\omega(t)) e^{-i2\Phi_c(t)} \right) \\ &+ \frac{1}{4} \left(\hat{g}_2^2(2\omega(t)) e^{4i\Phi_c(t)} + \hat{g}_2^2(-4\omega(t)) e^{-4i\Phi_c(t)} \right) \\ &+ \frac{1}{2} \left(\hat{g}_2(-\omega(t)) \hat{g}_2(2\omega(t)) e^{i\Phi_c(t)} + \hat{g}_2(\omega(t)) \hat{g}_2(-2\omega(t)) e^{-i\Phi_c(t)} \right) \\ &+ \frac{1}{2} \left(\hat{g}_2(\omega(t)) \hat{g}_2(2\omega(t)) e^{i3\Phi_c(t)} + \hat{g}_2(-\omega(t)) \hat{g}_2(-2\omega(t)) e^{-i3\Phi_c(t)} \right) \\ &+ \frac{1}{2} \left(\hat{g}_2(\omega(t)) \hat{g}_2(-\omega(t)) + \hat{g}_2(2\omega(t)) \hat{g}_2(-2\omega(t)) \right) \end{aligned}$$

En appliquant à h_2 la propriété de l'équation C.2, on obtient :

$$\begin{aligned} h_2 * (g_2 * x(t))^2 &= \frac{1}{4} \left(\hat{h}_2(2\omega(t)) \hat{g}_2^2(\omega(t)) e^{i2\Phi_c(t)} + \hat{h}_2(-2\omega(t)) \hat{g}_2^2(-\omega(t)) e^{-i2\Phi_c(t)} \right) \\ &+ \frac{1}{4} \left(\hat{h}_2(4\omega(t)) \hat{g}_2^2(2\omega(t)) e^{4i\Phi_c(t)} + \hat{h}_2(-4\omega(t)) \hat{g}_2^2(-4\omega(t)) e^{-4i\Phi_c(t)} \right) \\ &+ \frac{1}{2} \left(\hat{h}_2(\omega(t)) \hat{g}_2(-\omega(t)) \hat{g}_2(2\omega(t)) e^{i\Phi_c(t)} + \hat{h}_2(-\omega(t)) \hat{g}_2(\omega(t)) \hat{g}_2(-2\omega(t)) e^{-i\Phi_c(t)} \right) \\ &+ \frac{1}{2} \left(\hat{h}_2(3\omega(t)) \hat{g}_2(\omega(t)) \hat{g}_2(2\omega(t)) e^{i3\Phi_c(t)} + \hat{h}_2(-3\omega(t)) \hat{g}_2(-\omega(t)) \hat{g}_2(-2\omega(t)) e^{-i3\Phi_c(t)} \right) \\ &+ \frac{1}{2} \left(\hat{h}_2(0) \hat{g}_2(\omega(t)) \hat{g}_2(-\omega(t)) + \hat{h}_2(0) \hat{g}_2(2\omega(t)) \hat{g}_2(-2\omega(t)) \right) \end{aligned}$$

On en déduit la relation entre les pseudo-réponses fréquentielles $H_2, k(\omega)$ (obtenues dans le domaine temporel par convolution de la composante d'ordre 2 avec un sweep inversé) et les filtres h_2 et g_2 :

$$\begin{aligned} H_{2,0}(\omega) &= \frac{1}{2} \left[\hat{h}_2(\omega) \hat{g}_2(-\omega) \hat{g}_2(2\omega) \right] \\ H_{2,1}(2\omega) &= \frac{1}{4} \left[\hat{h}_2(2\omega) \hat{g}_2^2(\omega) \right] \\ H_{2,2}(3\omega) &= \frac{1}{2} \left[\hat{h}_2(3\omega) \hat{g}_2(\omega) \hat{g}_2(2\omega) \right] \\ H_{2,3}(4\omega) &= \frac{1}{4} \left[\hat{h}_2(4\omega) \hat{g}_2^2(2\omega) \right] \end{aligned}$$

c'est-à-dire :

$$\begin{aligned} H_{2,0}(\omega) &= \frac{1}{2} \left[\hat{h}_2(\omega) \hat{g}_2(-\omega) \hat{g}_2(2\omega) \right] \\ H_{2,1}(\omega) &= \frac{1}{4} \left[\hat{h}_2(\omega) \hat{g}_2^2(\omega/2) \right] \\ H_{2,2}(\omega) &= \frac{1}{2} \left[\hat{h}_2(\omega) \hat{g}_2(\omega/3) \hat{g}_2(2\omega/3) \right] \\ H_{2,3}(\omega) &= \frac{1}{4} \left[\hat{h}_2(\omega) \hat{g}_2^2(\omega/2) \right] \end{aligned}$$

Ce qui donne en passant au logarithme :

$$\begin{aligned} \ln(H_{2,0}(\omega)) &= -\ln(2) + \ln(\hat{h}_2(\omega)) + \ln(\hat{g}_2(-\omega)) + \ln(\hat{g}_2(2\omega)) \\ \ln(H_{2,1}(\omega)) &= -\ln(4) + \ln(\hat{h}_2(\omega)) + 2\ln(\hat{g}_2(\omega/2)) \\ \ln(H_{2,2}(\omega)) &= -\ln(2) + \ln(\hat{h}_2(\omega)) + \ln(\hat{g}_2(\omega/3)) + \ln(\hat{g}_2(2\omega/3)) \\ \ln(H_{2,3}(\omega)) &= -\ln(4) + \ln(\hat{h}_2(\omega)) + 2\ln(\hat{g}_2(\omega/2)) \end{aligned}$$

Le système devient alors déterminé et on peut résoudre le système comme présenté dans la section précédente.

XVI ANNEXE C. CALCUL POUR L'IDENTIFICATION DE SYSTÈMES S_M

Annexe D

Calcul pour l'identification par intercorrélation

Le calcul présenté ici permet d'identifier les coefficients du noyau d'un système de Volterra homogène d'ordre 3 à partir de l'intercorrélation entrée/sortie d'ordre 3.

Le système considéré est donc un système de Volterra homogène d'ordre 3 :

$$y(n) = \sum_{k_1, k_2, k_3=0}^{\infty} h_3(k_1, k_2, k_3) x(n - k_1) x(n - k_2) x(n - k_3)$$

Dans le cas $T_1 \neq T_2$, dans de bonnes hypothèses de convergence :

$$E(y(n)x(n-T_1)x(n-T_2)^2) = \sum_{k_1, k_2, k_3=0}^{\infty} h_3(k_1, k_2, k_3) E(x(n-k_1)x(n-k_2)x(n-k_3)x(n-T_1)x(n-T_2)^2)$$

Pour le calcul de $e(k_1, k_2, k_3, T_1, T_2) = E(x(n - k_1)x(n - k_2)x(n - k_3)x(n - T_1)x(n - T_2)^2)$, on peut distinguer plusieurs cas dans la somme sur k_1, k_2 et k_3 :

- $k_1 = k_2 = k_3 = T_1$: $e(k_1, k_2, k_3, T_1, T_2) = A_4 A_2$
- $k_1 = T_1$ et $k_2 = k_3 = T_2$ et les 3 permutations de ce cas : $e(k_1, k_2, k_3, T_1, T_2) = A_4 A_2$
- $k_1 = T_1$ et $T_1 \neq k_2 = k_3 \neq T_2$ et les 3 permutations de ce cas : $e(k_1, k_2, k_3, T_1, T_2) = A_2^3$
- dans tous les autres cas : $e(k_1, k_2, k_3, T_1, T_2) = 0$

Par conséquent :

$$\begin{aligned} E(y(n)x(n - T_1)x(n - T_2)^2) &= A_4 A_2 h_3(T_1, T_1, T_1) \\ &+ 3A_4 A_2 h_3(T_1, T_2, T_2) \\ &+ 3A_2^3 \sum_{\substack{k=0 \\ k \neq T_1, k \neq T_2}}^{\infty} h_3(T_1, k, k) \end{aligned}$$

XVIII ANNEXE D. CALCUL POUR L'IDENTIFICATION PAR INTERCORRÉLATION

Ce qui peut aussi s'écrire :

$$E(y(n)x(n-T_1)x(n-T_2)^2) = (A_4A_2 - 3A_2^3)h_3(T_1, T_1, T_1) \quad (\text{D.1})$$

$$+ 3(A_4A_2 - A_2^3)h_3(T_1, T_2, T_2) \quad (\text{D.2})$$

$$+ 3A_2^3 \sum_{k=0}^{\infty} h_3(T_1, k, k) \quad (\text{D.3})$$

Dans le cas $T_1 = T_2 = T$, dans de bonnes hypothèses de convergence :

$$E(y(n)x(n-T)^3) = \sum_{k_1, k_2, k_3=0}^{\infty} h_3(k_1, k_2, k_3) E(x(n-k_1)x(n-k_2)x(n-k_3)x(n-T)^3)$$

Pour le calcul de $e(k_1, k_2, k_3, T) = E(x(n-k_1)x(n-k_2)x(n-k_3)x(n-T)^3)$, on peut distinguer plusieurs cas dans la somme sur k_1, k_2 et k_3 :

- $k_1 = k_2 = k_3 = T$: $e(k_1, k_2, k_3, T) = A_6$

- $k_1 = T$ et $k_2 = k_3 \neq T$ et les 3 permutations de ce cas : $e(k_1, k_2, k_3, T) = A_4A_2$

- dans tous les autres cas : $e(k_1, k_2, k_3, T) = 0$

Par conséquent :

$$E(y(n)x(n-T)^3) = A_6h_3(T, T, T) + 3A_4A_2 \sum_{\substack{k=0 \\ k \neq T}}^{\infty} h_3(T, k, k)$$

Ce qui peut aussi s'écrire :

$$E(y(n)x(n-T)^3) = (A_6 - 3A_4A_2)h_3(T, T, T) + 3A_4A_2 \sum_{k=0}^{\infty} h_3(T, k, k) \quad (\text{D.4})$$

Les équations (D.1) et (D.4) forment un système linéaire en les $h_3(T_1, T_2, T_2)$ (T_1 et T_2 pouvant être égaux). Dans le cas où on considère une série tronquée, les sommes dans (D.1) et (D.4) deviennent des sommes finies et le système peut se résoudre facilement.

Bibliographie

- [1] Jonathan S. Abel and David P. Berners. A technique for nonlinear system measurement. *121st Convention Audio Engineering Society*, Octobre 2006.
- [2] Gregoire Allaire. *Analyse numérique et optimisation*. Editions de l'Ecole Polytechnique, 2006.
- [3] S. A. Billings and S. Y. Fakhouri. Identification of non-linear unity feedback systems. *IEEE Decision and Control*, janvier 1978.
- [4] Stephen Boyd, Y. S. Tang, and Leon O. Chua. Measuring Volterra kernels. *IEEE Transactions on circuits and systems*, CAS-30, août 1983.
- [5] Hosam E. Emara-Shabaik, Kamal A. F. Moustafa, and Jaleel H. S. Talaq. Structure identification of a class of non-linear systems using correlation and bispectrum approaches. *UKACC International Conference on CONTROL, IEE*, Septembre 1996.
- [6] Angelo Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. *108th Convention Audio Engineering Society*, Février 2000.
- [7] Angelo Farina. Advancements in impulse response measurements by sine sweeps. *122nd Convention Audio Engineering Society*, mai 2007.
- [8] Angelo Farina and Adriano Farina. Realtime auralization employing a non-linear, not-time-invariant convolver. *123rd Convention Audio Engineering Society*, Octobre 2007.
- [9] S. Foster. Impulse response measurements using golay codes. *Proc. ICASSP*, Avril 1986.
- [10] Matthias O. Franz and Bernhard Schölkopf. A unifying view of Wiener and Volterra theory and polynomial kernel regression. *Neural Computation*, 18, Avril 2006.
- [11] Robert Haber and Laszlo Keviczky. *Nonlinear system identification - input-output modeling approach*. Kluwer academic publishers, 1999.
- [12] Michael J. Kemp. Analysis and simulation of non-linear audio processes using finite impulse responses derived at multiple impulse amplitudes. *106th Convention, Audio Engineering Society*, Octobre 1999.
- [13] Françoise Lamnabhi-Lagarrigue. *Analyse des systèmes non linéaires*. Hermes, 1994.

- [14] Véronique Larcher. *Techniques de spatialisation des sons pour la réalité virtuelle*. PhD thesis, Université de Paris VI, 2001.
- [15] Véronique Larcher, Jean-Marc Jot, and Guillaume Vandernoot. Equalization methods in binaural technology. *105th Convention AES*, septembre 1998.
- [16] S. Müller and P Massarani. Transfer-function measurement with sweeps. *JAES*, 49, Juin 2001.
- [17] T.W. Parks and C.S. Burrus. *Digital Filter Design*. John Wiley and Sons, 1987.
- [18] M.J. Reed and M.O.J. Hawksford. Identification of discrete Volterra series using maximum length sequences. *IEE Proc.-Circuits Devices Syst.*, 143, Octobre 1996.
- [19] Wilson J. Rugh. *Nonlinear System Theory - The Volterra/Wiener Approach*. The Johns Hopkins University Press, 1981.
- [20] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Max-Planck-Institut für biologische Kybernetik, Technical Report No.44*, decembre 1996.
- [21] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. The MIT Press, 2002.
- [22] M.R. Schroeder. Integrated impulse method for measuring sound decay without using impulses. *JASA*, 66, Aout 1979.
- [23] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.
- [24] V. Vapnik. *The Nature of Statistical Learning Theory*. New York : Springer-Verlag, 1995.
- [25] Hua-Liang Wei, Zi-Qiang Lang, and Stephen A. Billings. An algorithm for determining the output frequency range of Volterra models with multiple inputs. *IEEE Transactions on circuits and systems*, Juin 2007.
- [26] Udo Zolzer and al. *DAFX - Digital Audio Effects*. John Wiley and Sons, 2002.
- [27] Udo Zolzer, Bernd Redmer, and Jochen Bucholtz. Strategies for switching digital audio filters. *95th convention AES*, octobre 1993.