

IMPLEMENTATION EN TEMPS REEL DES RETARDS FRACTIONNAIRES VARIABLES PAR INTERPOLATION DE LAGRANGE

Rapport de stage présenté par

Geoffroy PEETERS

DEA A.T.I.A.M. 1995-1996

Université Paris VI

réalisé au Département Analyse/Synthèse de l'I.R.C.A.M.
sous la direction de Ph. Depalle et S. Tassart

Année universitaire 1995-1996

Remerciements

*Je tiens à remercier Stéphan Tassart et Philippe Depalle,
pour l'aide et les nombreux conseils qu'ils m'ont fournis pendant mon stage
ainsi que Xavier Rodet,
pour les mêmes raisons
merci également à Maurizio De Cecco, François Déchelle,
et Norbert Schnell de l'équipe temps réel*

INTRODUCTION

De nombreux modèles physiques en synthèse sonore repose sur la théorie des guides d'ondes numériques ([9] [19]). Ces méthodes consistent à simuler la propagation d'ondes acoustiques dans le guide d'onde à l'aide de ligne à retard numérique. Dans ces conditions la longueur du guide d'onde se retrouve quantifiée par la fréquence d'échantillonnage. Le réglage fin de la longueur des guides d'ondes s'obtient par l'adjonction au modèle de filtres réalisants des retards fractionnaires. En particulier, les retards fractionnaires sont les seuls recours disponibles pour modifier continûment la longueur des guides d'ondes.

Ce travail aura pour but de fournir une implémentation en temps réel des retards fractionnaires variables. Il s'agira d'évaluer les différentes stratégies d'implémentation et de contrôle du filtre afin d'éliminer toute distorsion possible du signal.

Dans la première partie de ce travail, nous définissons les retards fractionnaires numériques idéaux et expliquons pourquoi ceux-ci ne peuvent être réalisés. Nous montrons différentes méthodes d'approximations pour la réalisation de filtres R.I.F. et I.I.F. approximant les retards fractionnaires. Parmi celles-ci, le critère de platitude maximale à basse fréquence équivalent à l'interpolation de Lagrange semble tout indiqué pour une utilisation dans le cas de retards fractionnaires variables.

Dés lors dans la deuxième partie nous étudions deux implémentations différentes des filtres interpolateurs de Lagrange : une implémentation sous forme de structure de Farrow (ou de structure de Farrow modifiée) et une nouvelle implémentation sous forme de structure modulaire du filtre. Nous comparons alors les deux implémentations du point de vue du coût de calcul et de leur comportement dans un contexte de retards fractionnaires variables.

Dans la troisième partie, nous utilisons la nouvelle implémentation des F.I.L. sous forme de structure modulaire dans la réalisation d'un modèle physique simplifié d'instrument à tube. Ce modèle sera réalisé sur le logiciel temps réel de l'IRCAM Max/FTS, et nous conduira à la réalisation de nouveaux objets externes pour la réalisation du modèle.

1. THEORIE DES RETARDS FRACTIONNAIRES

Dans cette première partie, nous définissons les retards fractionnaires idéaux. Ceci sera fait par référence au temps continu, bien que d'autres définitions soient possibles. Nous donnons également la forme idéale que devraient revêtir les filtres les réalisant ainsi que les approximations que nous devons faire pour les réaliser.

1.1. Définition des retards fractionnaires idéaux

En temps continu, le retard sur un signal $x_c(t)$ s'exprime :

$$y_c(t) = x_c(t - \tau), \text{ où } \tau \text{ constitue le retard désiré.}$$

La fonction de transfert de cet élément de retard peut s'écrire à l'aide de la transformée de Fourier :

$$H_{c,id}(j\omega_c) = e^{-j\omega_c \tau}$$

En temps discret, le même retard sur un signal qui se doit maintenant d'être à largeur de bande finie, s'exprimera :

$$y(n) = x(n - D), \text{ où } D = \tau / T \text{ entier, } T \text{ constituant la période d'échantillonnage.}$$

La transformée en z étant uniquement définie pour des puissances entières de z , la fonction de transfert en temps discret $H_{id}(e^{j\omega})$ de cette élément de retard s'écrira :

$$H_{id}(e^{j\omega}) = e^{-j\omega D} \text{ avec } |\omega| \leq \pi \text{ et } D \in \mathbb{N}$$

Celle-ci constitue la fonction de transfert d'un retard idéal en temps discret, qui est donc de période 2π .

Produire un retard fractionnaire en temps discret, consiste à calculer l'amplitude du signal continu sous-jacent au signal discret (reconstruction) entre deux échantillons. La définition des retards fractionnaires par référence au temps continu [2] se fait donc de la manière suivante : à partir d'une séquence discrète x_k , nous reconstruisons le signal continu correspondant $x_c(t)$ en utilisant la fonction de Whittaker, nous décalons ensuite ce signal continu du retard D désiré et ré-échantillonons le signal continu décalé. Nous obtenons ainsi le retard fractionnaire désiré D sur la séquence discrète x_k [21].

$$\begin{array}{ccc}
 (x_n) & \xrightarrow{h^d} & (x_{n-D}) = (x_n) \otimes \text{sinc}(n-D) \\
 \downarrow \text{reconstruction} & & \uparrow \text{échantillonnage} \\
 x_c(t) = \sum_{n \in \mathbb{Z}} x_n \text{sinc}(t-n) & \xrightarrow{\text{retard}} & x_c(t-D) = \sum_{n \in \mathbb{Z}} x_n \text{sinc}(t-D-n)
 \end{array}$$

Remarquons que nous devons limiter cette définition aux signaux discrets à bande limitée, i.e. aux signaux dont la composante à la fréquence de Nyquist est nulle [21]. Prenons comme exemple la séquence discrète $\{-1\}^k$. Cette séquence présentant de l'énergie uniquement à la fréquence de Nyquist, tous les signaux continus de type $\frac{\cos(\pi t + \Phi)}{\cos(\Phi)}$ sont candidats à sa reconstruction puisque donnant lieu après échantillonnage à la même séquence. Comment retarder cette séquence de D échantillons puisque a priori pour toute valeur K il est possible de trouver une valeur Φ tel que $\{(-1)^k\}$ puisse être interprétée comme la retardée de $\{-1\}^k$. Pour résoudre ce problème nous limiterons notre domaine de définition aux signaux discrets à bande limitée. $H_{id}(e^{j\omega})$ n'est donc ni définie ni continue en $\omega = \pm\pi$.

L'interpolateur idéal à bande limitée ainsi défini a comme fonction de transfert : $H_{id}(\omega) = e^{-j\omega D}$ avec $|\omega| < \pi$ et D non-réel, il possède les caractéristiques suivantes :

- réponse en amplitude : $|H_{id}(e^{j\omega})| = 1$
- réponse en phase : $\arg\{H_{id}(e^{j\omega})\} = -D\omega$

Il s'agit donc d'un système passe-tout à phase linéaire. De plus

- retard de phase : $\tau_{p,id}(\omega) = -\frac{\arg\{H_{id}(e^{j\omega})\}}{\omega} = D$
- retard de groupe : $\tau_{g,id}(\omega) = -\frac{\partial \arg\{H_{id}(e^{j\omega})\}}{\partial \omega} = D$

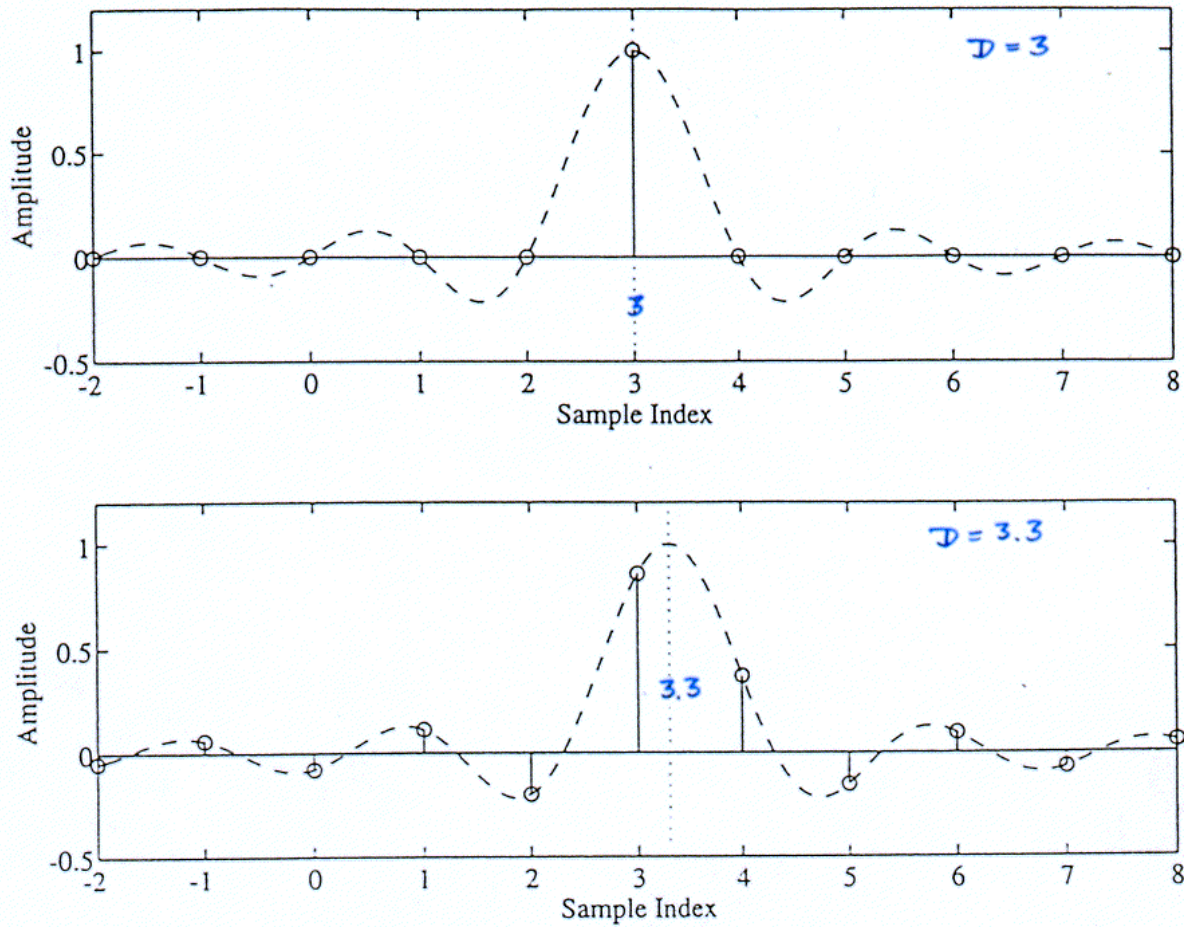
Le retard idéal décale donc toutes les composantes fréquentielles du même retard D .

La réponse impulsionnelle (R.I.) du retard idéal s'écrit

$$h_{id}(n) = \text{sinc}(n - D)$$

Cette fonction est une version décalée et échantillonnée du sinus cardinal. Elle s'annule pour toutes les valeurs entières de D exceptée celles où $D=n$, dans ce cas elle vaut exactement 1.

- Quand D est entier, la réponse impulsionnelle correspond exactement à la fonction de Dirac décalée. La R.I. d'un retard entier se réduit donc à un simple retard au sens traditionnel numérique (voir figure 1-1a.).
- Par contre si D est réel et non entier, la R.I. ne s'annule pour aucune valeur de n . La R.I. est dans ce cas infiniment longue (voir figure 1-1b.). $h_{id}(n)$ correspond alors à un filtre non-causal, qui ne peut pas être rendu causal par aucun décalage fini dans le temps. De plus le filtre n'est pas BIBO (Bounded Input Bouded Output) puisqu'il n'est pas absolument-sommable. En conclusion pour D réel et non entier, le filtre n'est pas réalisable et des approximations devront être utilisées.



Figures 1-1a. 1-1b La fonction sinus cardinal décalée de $D=3$ échantillons et décalée de $D=3.3$ échantillons

source : [24]

Autres approches de la définition des retards fractionnaires

Nous mentionnons l'existence d'autres approches théoriques qui permettent de définir les retards fractionnaires.

En particulier quand le retard est rationnel : par sur-échantillonnage puis sous-échantillonnage (ce qui revient à résoudre $H_{id} \left[(e^{j\omega})^p \right] \stackrel{F}{=} e^{-j\omega q}$ et par cumulation de retards $H_{id} \left(e^{j\omega} \right)^p = e^{-j\omega q}$).

Evidemment toutes ces approches aboutissent au même résultat en ce qui concerne la TFTD du retard idéal.

1.2. Approximation des retards fractionnaires par filtres R.I.F.

Dans le paragraphe 1.1. nous avons défini les retards fractionnaires idéaux numériques, nous avons également vu que la réalisation de ceux-ci nécessite des approximations puisque le filtre réalisant la réponse impulsionnelle idéale n'est pas réalisable. Dans ce paragraphe-ci et dans le suivant nous présentons quelques approches aboutissant à différentes approximations permettant la réalisation de filtres R.I.F. et R.I.I. pour les retards fractionnaires.

1.2.1. Filtres R.I.F. par fenêtrage de la réponse impulsionnelle idéale des retards fractionnaires

Nous présentons rapidement ici différentes méthodes de réalisations de filtres R.I.F. par fenêtrage de la réponse impulsionnelle idéale du retard fractionnaire (voir aussi [1] [11]) :

$$h_{id}(n) = \text{sinc}(n - D)$$

La transformée en z du filtre R.I.F. d'ordre N que nous cherchons est de la forme

$$H(z) = \sum_{n=0}^N h(n) z^{-n}$$

Nous définissons la fonction d'erreur $E(e^{j\omega})$, différence entre l'approximation que nous cherchons et la fonction de transfert idéale :

$$E(e^{j\omega}) = H(e^{j\omega}) - H_{id}(e^{j\omega})$$

1.2.1.1. Méthode des moindres carrés

Dans cette méthode (voir [1], [18]) nous cherchons à minimiser la fonction d'erreur des moindres carrés :

$$E_{LS} = \frac{1}{\pi} \int_0^{\pi} |E(e^{j\omega})|^2 d\omega .$$

Pour un filtre R.I.F. d'ordre N , la solution optimale est celle consistant à tronquer les $N+1$ coefficients de $h_{id}(n)$ autour de sa valeur maximale (l'erreur d'approximation diminuant évidemment avec l'ordre N). La réponse impulsionnelle du filtre R.I.F. ainsi créé est :

$$h(n) = \begin{cases} \text{sinc}(n - D) & 0 \leq n \leq N \\ 0 & \text{dans les autres cas} \end{cases}$$

Afin de minimiser l'erreur, N doit être choisi de façon à ce que [24]:

$$\frac{N-1}{2} \leq D \leq \frac{N+1}{2} \quad \text{pour } N \text{ impair,}$$

i.e. D doit être localisé entre les deux pics centraux du sinus cardinal

$$\frac{N-1}{2} \leq D \leq \frac{N}{2} + 1 \quad \text{pour } N \text{ pair,}$$

i.e. D doit être localisé à moins d'un demi-échantillon du pic central de sinus cardinal

Le problème de cette méthode de construction de filtres R.I.F. réside dans l'existence d'un phénomène de Gibbs, i.e. l'oscillation de la réponse en fréquence tant en amplitude qu'en phase et ce indépendamment de l'ordre du filtre.

1.2.1.2. Méthode des moindres carrés avec bande passante réduite

Cette méthode est une variante de la précédente. Cette fois un interpolateur passe-bas est utilisé comme filtre prototype à la place d'un filtre interpolateur toute bande (voir [11]). La solution idéale est celle qui minimise l'erreur dans l'intervalle de fréquence $\left[0, \frac{\alpha}{2}\right]$ ($0 < \alpha < 1$).

$$h(n) = \begin{cases} \frac{\sin[\alpha\Pi(n-D)]}{\Pi(n-D)} & N \leq n \leq M+N \\ 0 & \text{dans les autres cas} \end{cases}$$

Le phénomène de Gibbs est réduit mais la bande passante aussi (d'un facteur α).

Autres méthodes : méthode de fenêtrage de la réponse impulsionnelle [15], méthode général des moindres carrés pour réponse fréquentielle compliquée [11], méthode Minimax ou de Chebyshev [11] [15], méthode stochastique [14], critère de platitude maximale à basse fréquence, ...

1.2.2. Filtres R.I.F. - Interpolateur de Lagrange (F.I.L.) : Critère de platitude maximale

Une autre approche consiste à maximiser la platitude de la fonction d'erreur $E(e^{j\omega})$ à la fréquence nulle [11]. Ceci revient à annuler la fonction d'erreur $E(e^{j\omega})$, ainsi que ses N premières dérivées à fréquence nulle.

Nous allons voir que les coefficients du filtre R.I.F. d'ordre N obtenu de cette manière sont les mêmes que ceux utilisés dans un interpolateur de Lagrange d'ordre N passant par $N+1$ points équidistants.

1.2.2.1. Equivalence entre critère de platitude maximale et interpolation de Lagrange

Le critère de platitude maximale à fréquence nulle s'écrit :

$$\left. \frac{d^k E(e^{j\omega})}{d\omega^k} \right|_{\omega=0} = 0 \quad \text{pour } k = 0, 1, \dots, N$$

ou en explicitant $E(e^{j\omega})$:

$$\left. \frac{d^k}{d\omega^k} \left[\sum_{n=0}^N h(n, D) e^{-j\omega n} - e^{-j\omega D} \right] \right|_{\omega=0} = 0 \quad \text{pour } k = 0, 1, \dots, N$$

- pour $k=0$, il se réécrit $\sum_{n=0}^N h(n, D) - 1 = 0$ ou encore $\sum_{n=0}^N h(n, D) = 1$. La somme des coefficients du filtre doit donc être égale à 1, ou exprimé d'une autre manière l'amplitude de la réponse doit valoir 1 à la fréquence nulle.
- pour $k \neq 0$, nous obtenons N équations linéaires

$$\sum_{n=0}^N n^k h(n, D) = D^k \quad \text{pour } k = 1, \dots, N$$

Nous avons donc un système de $N+1$ équations à $N+1$ inconnues qui se résout simplement sous forme matricielle (inversion d'une matrice de Vandermonde). La solution de ce système donne les $N+1$ coefficients du filtre R.I.F. d'ordre N vérifiant le critère de platitude maximale :

$$h(n, D) = \prod_{\substack{k=0 \\ k \neq n}}^N \frac{D-k}{n-k} \quad \text{pour } n = 0, 1, \dots, N$$

Il est remarquable de noter que les coefficients ainsi obtenus sont identiques à ceux d'un interpolateur de Lagrange d'ordre N . L'interpolation de Lagrange (rappelons le) consiste à faire passer un polynôme d'ordre N par $N+1$ points équidistants et d'en déduire les valeurs intermédiaires (voir figure 1-2).

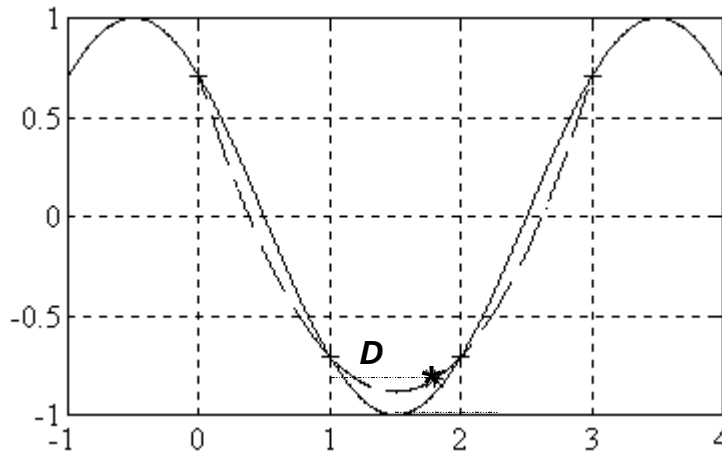


Figure 1-2. Interpolateur de Lagrange d'ordre 3 passant par 4 points équidistants, évalué en D

Autrement dit, le filtre interpolateur de Lagrange (F.I.L.) d'ordre N et de retard D correspond au filtre R.I.F. d'ordre N dont la fonction de transfert vérifie le critère de platitude maximale [7].

Remarque : Le filtre vérifiant le critère de platitude maximale s'interprète également comme le filtre à réponse impulsionnelle finie qui approxime le mieux possible la transformée de Fourier de la réponse impulsionnelle idéale à la fréquence nulle; il ne donne donc pas d'indications sur le comportement en dehors du voisinage de la fréquence nulle, seul le comportement local est garanti [21]. Pour le domaine d'application qui nous intéresse, les modèles de synthèse sonore par guide d'onde, ceci ne nous pose pas de problème puisque le domaine de fréquence utilisé est justement celui des basses fréquences (les instruments de musique ayant essentiellement un contenu basse fréquence).

Autre approche : interpolateur de Lagrange par fenêtrage binomial de la réponse impulsionnelle idéale [10]

1.2.2.2. Propriétés des filtres interpolateurs de Lagrange

- Symétrie des coefficients des filtres interpolateur de Lagrange

Une propriété importante de l'interpolateur de Lagrange est la symétrie de ses coefficients $h(n, D)$. Les coefficients pour le retard fractionnaire $N-D$ sont les mêmes que ceux pour le retard fractionnaire D mais dans l'ordre inverse.

$$\boxed{h(n, D) = h(N - n, N - D)}$$

Nous verrons plus loin (voir § 2.2.2.) que l'intérêt de cette propriété réside dans le fait que lorsque le retard passe de D à $N-D$, du fait de la symétrie des coefficients (et donc de la réponse impulsionnelle), la réponse en amplitude de filtre réalisant l'interpolation de Lagrange reste la même :

$$\left| \frac{H_N^D}{H_N^{N-D}} \right| = 1$$

Dans le cadre de retards fractionnaires variables cette propriété présente un intérêt décisif poussant à l'utilisation des filtres interpolateurs de Lagrange pour leur approximation.

- Calcul de l'erreur d'approximation des filtres interpolateur de Lagrange (voir aussi [1] [15])

L'erreur d'approximation dépend de la partie fractionnaire du retard D :

- si D est entier, la réponse impulsionnelle du filtre interpolateur de Lagrange se réduit à une impulsion unitaire décalée et l'erreur est nulle.
- si D est non-entier, l'erreur maximale se présente quand la partie fractionnaire s'approche de 0.5. (voir [1]).

L'erreur d'approximation dépend aussi fortement de la parité de l'ordre du filtre [11] :

- les filtres d'ordre pair permettent de diminuer l'erreur en amplitude (voir figures 1-3a 1-3b),
- les filtres d'ordre impairs permettent de diminuer l'erreur de phase (voir figures 1-4a 1-4b).

Pour une formulation de ces propriétés, nous renvoyons le lecteur au paragraphe 2.3.2..

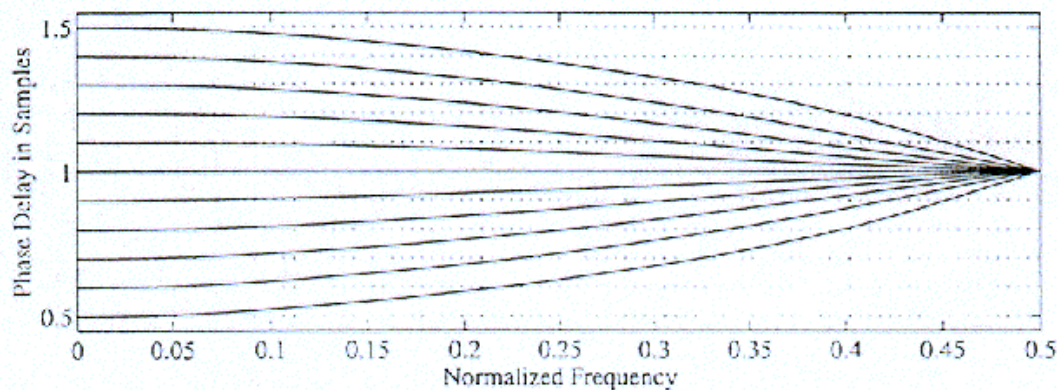
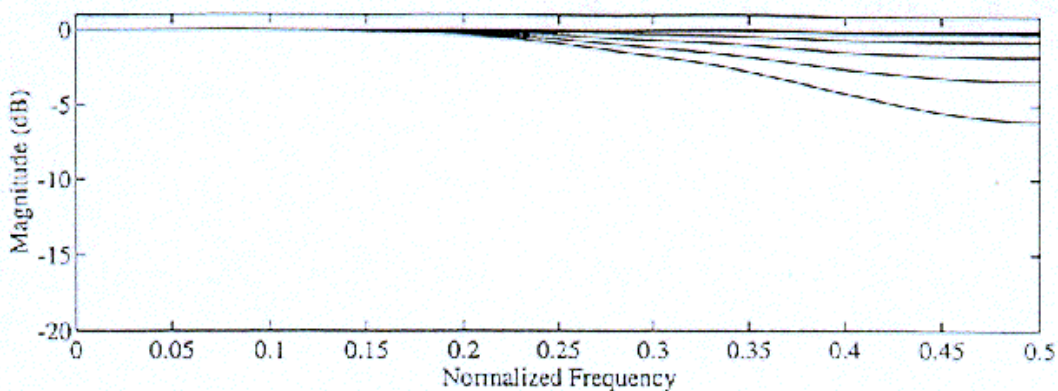
Page suivante :

Figures 1-3a. 1-3b Réponse en amplitude et retard de phase d'un filtre interpolateur de Lagrange d'ordre 2 ($N=2$) pour 11 valeurs équidistantes ($D=0.5, 0.6, 0.7, \dots, 1.5$)

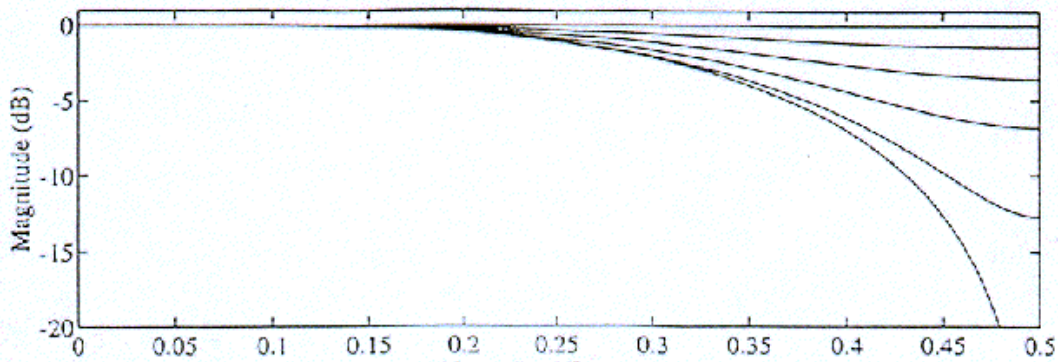
Figures 1-4a. 1-4b Réponse en amplitude et retard de phase d'un filtre interpolateur de Lagrange d'ordre 3 ($N=3$) pour 11 valeurs équidistantes ($D=1.0, 1.1, 1.2, \dots, 2.0$)

source : [24]

N=2



N=3



- Passivité des filtres interpolateurs de Lagrange

Un constat expérimental [11] [24] est que la réponse en amplitude des filtres interpolateurs de Lagrange ne dépasse jamais l'unité pour peu que la condition suivante soit réalisée

$$\frac{N-1}{2} \leq D \leq \frac{N+1}{2} \quad \text{pour } N \text{ impair}$$

$$\frac{N}{2} - 1 \leq D \leq \frac{N}{2} + 1 \quad \text{pour } N \text{ pair}$$

La passivité des filtres interpolateurs de Lagrange est particulièrement importante dans notre contexte de modèle physique où les filtres sont incorporés dans une boucle de rétroaction. C'est une condition nécessaire pour la stabilité de la boucle.

1.3. Approximation des retards fractionnaires par filtres R.I.I. Passe-Tout

Dans ce paragraphe nous présentons brièvement l'utilisation de filtres R.I.I. ou récursifs passe-tout pour la réalisation de retards fractionnaires.

Ces filtres ont une réponse en amplitude constante unitaire et ce quelque soit l'ordre du filtre. Leur réponse en phase est de plus paramétrable. Ils semblent donc tout indiqués pour la réalisation de retards fractionnaires qui est avant tout un problème d'approximation de phase.

La fonction de transfert du filtre R.I.I. passe-tout d'ordre N s'exprime :

$$H(z) = \frac{z^{-N}A(z^{-1})}{A(z)} = \frac{a_N + a_{N-1}z^{-1} + \dots + a_1z^{-(N-1)} + z^{-N}}{1 + a_1z^{-1} + \dots + a_{N-1}z^{-(N-1)} + a_Nz^{-N}}$$

Les pôles et les zéros d'un filtre R.I.I. passe-tout étant inverse l'un de l'autre, la réponse en amplitude est constante et égale à un :

$$|H(e^{j\omega})| = 1$$

Rappelons que pour qu'un filtre passe-tout soit stable les pôles de la fonction de transfert doivent se trouver à l'intérieur du cercle unité.

La réponse en phase s'exprime :

$$\arg\{H(e^{j\omega})\} = -N\omega + 2\Theta_D(\omega)$$

où $\Theta_D(\omega)$ est la réponse en phase de son dénominateur :

$$\Theta_D(\omega) = \arctan \left\{ \frac{\sum_{k=0}^N a_k \sin(k\omega)}{\sum_{k=0}^N a_k \cos(k\omega)} \right\}$$

La réalisation de retards fractionnaires sous forme de filtres R.I.I. passe-tout peut se faire en appliquant le critère de platitude maximale sur le retard de phase (voir [11]). Thiran [23] propose une solution analytique pour retrouver les coefficients d'un filtre passe-tout d'ordre N approximant un retard fractionnaire D :

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^N \frac{D - N + n}{D - N + k + n} \quad \text{pour } k=0, 1, \dots, N$$

$$\text{où } \binom{N}{k} = \frac{N!}{k!(N-k)!} \text{ est le produit binomial}$$

Le filtre passe-tout est stable lorsque $N - 1 < D$ et la plage de meilleur approximation du retard fractionnaire correspond à $N - 1 < D < N$ (voir [24]). Remarquons que dans le cas de l'utilisation de filtres passe-tout, l'ordre du filtre N est de l'ordre de D , le retard approximé ($2D$ pour les filtres interpolateurs de Lagrange d'ordre N).

Les filtres passe-tout réalisés selon ces critères sont généralement très précis à basse fréquence, mais contrairement aux filtres R.I.F., ils souffrent de problèmes de transitoires de réponse lorsque le retard varie [24]. Remarquons cependant l'existence de techniques permettant d'éliminer plus ou moins bien ces transitoires (voir [13], [16], [25], [26]).

2. IMPLEMENTATION DES FILTRES INTERPOLATEURS DE LAGRANGE POUR L'APPROXIMATION DE RETARDS FRACTIONNAIRES

Jusqu'à présent, nous avons défini les retards fractionnaires idéaux, et présenté différentes formes d'approximations de ceux-ci sous forme de filtres R.I.F. : par fenêtrage de la réponse impulsionnelle idéale ou par application du critère de platitude maximale équivalent à l'interpolation de Lagrange; ou sous forme de filtres R.I.I. passe-tout.

Le problème qui nous intéresse étant celui de la réalisation de retards fractionnaires variables, nous étudions dans ce paragraphe les implémentations les plus appropriées des filtres interpolateurs de Lagrange (voir § 1.2.2).

Nous exposons deux implémentations différentes de ces filtres : l'une sous forme de structure de Farrow (ou de structure de Farrow modifiée) et l'autre sous forme d'une nouvelle structure modulaire utilisant le développement en série de puissance. Nous étudions leur coût de calcul, ainsi que leur comportement dans le contexte de retards fractionnaires variables. Dans ce cas, l'adjonction au F.I.L. d'une ligne de retard entier et dès lors d'une remise à jour de l'état du système est nécessaire.

Ces deux implémentations ont fait l'objet d'une réalisation utilisable en temps réel sur Max/FTS. L'implémentation sous forme de structure de Farrow modifiée est utilisée dans l'objet externe "vd~" écrit par Miller-Puckette. L'implémentation sous forme de structure modulaire est utilisée dans l'objet "fd~" écrit au cours de ce stage.

2.1. Implémentation des F.I.L. sous forme directe de filtre R.I.F.

Nous ne parlons de cette implémentation directe des F.I.L. que dans la mesure où elle nous donne un point de référence concernant la complexité de calcul des filtres.

Dans le cas d'une implémentation directe le coût du calcul des coefficients est de $N(N+1)$ additions et $N(N+1)$ multiplications.

Välimäki [24] propose une méthode de calcul plus performante évitant les redondances de calcul. La complexité est alors en N pour les additions et moins de $N^2 + N$ pour les multiplications.

2.2. Implémentation des F.I.L. sous forme de structure de Farrow

2.2.1. Structure de Farrow directe

Farrow [5] propose d'exprimer tous les coefficients d'un filtre interpolateur R.I.F. sous la forme d'un polynôme d'ordre N en D (la valeur du retard). Ce qui conduit à la réalisation de $N+1$ filtres R.I.F. à coefficients constants.

La fonction de transfert $\Pi(z)$ du filtre R.I.F. s'exprime maintenant comme un polynôme en D au lieu de z^{-1}

$$H(z) = \sum_{k=0}^N C_k(z) D^k$$

$\Pi(z)$ doit évidemment prendre les valeurs entières z^{-D} lorsque D est entier, ce qui s'écrit

$$\sum_{k=0}^N C_k(z) D^k = z^{-k} \quad \text{pour } D=0, 1, \dots, N$$

Nous sommes donc en présence d'un système de $N+1$ équations à $N+1$ inconnues $\mathcal{C}_k(z)$. Nous pouvons le résoudre sous forme matricielle. En notant

$$\bar{U} = \begin{bmatrix} 0^0 & 0^1 & \dots & 0^N \\ 1^0 & 1^2 & & 1^N \\ \vdots & & \ddots & \vdots \\ N^0 & N^1 & \dots & N^N \end{bmatrix}, \bar{c} = [C_0(z) \ C_1(z) \ \dots \ C_N(z)]^T, \text{ et } z = [1 \ z^{-1} \ \dots \ z^{-N}]^T$$

Nous pouvons écrire le système d'équations sous la forme

$$\bar{J}\bar{c} = \bar{z}$$

Ce système se résout par inversion de la matrice \bar{J} qui est une matrice de Vandermonde inversible. D'où

$$\bar{c} = \bar{Q}\bar{z} \text{ où } \bar{Q} = \bar{U}^{-1}.$$

Les fonctions de transfert résultantes sont des polynômes d'ordre N en z^{-1} .

$$C_k(z) = \sum_{n=0}^N q_{k,n} z^{-n} \text{ pour } k=1, 2, \dots, N$$

Remarquons que la fonction $C_0(z)$ prend toujours la valeur constante 1 quelque soit l'ordre du filtre

$$\sum_{k=0}^N C_k(z) 0^k = 1 \Rightarrow C_0(z) = 1$$

La structure de Farrow est généralement implémentée sous forme de schéma de Horner (voir [8]) pour des raisons de facilité de réalisation évidente :

$$\sum_{k=0}^N C_k(z) D^k = C_0(z) + [C_1(z) + [C_2(z) + \dots + [C_{N-1}(z) + C_N(z) D] D] D] D$$

Le principal avantage de la structure de Farrow réside dans le fait que les coefficients $\mathcal{C}_k(z)$ sont indépendants de D , donc constants pour une valeur fixe de N . Pour un filtre d'ordre fixe, il est donc possible de stocker tout les calculs intermédiaires du filtre dans des tables, ce qui est un sérieux gain de temps de calcul lors de l'utilisation en temps réel de retards fractionnaires variables. Le filtre est alors directement commandé par D .

Par comparaison avec l'implémentation des F.I.L. sous forme directe de filtre R.I.F., la structure de Farrow est plus coûteuse puisqu'elle demande le calcul de N filtres d'ordre $N+1$, soit $N(N+1)$ multiplications et N^2 additions, auquel il faut rajouter N multiplications par D et N additions. Soit au total $N^2 + 2N$ multiplications et $N^2 + N$ additions par échantillons. C'est la raison pour laquelle une autre forme de la structure de Farrow est généralement utilisée.

2.2.2. Structure de Farrow modifiée

Dans la structure de Farrow modifiée [24], la partie entière du retard fractionnaire (s'il y en a une) est soustraite. De ce fait, la plage de variation du retard fractionnaire restant, notons le d , se situe maintenant entre $0 < d \leq 1$ (pour un filtre d'ordre impair) et $-0.5 \leq d \leq 0.5$ (pour un filtre d'ordre pair) : $d = D - \frac{N-1}{2}$.

Notons cependant que la structure de Farrow modifiée produit un retard supplémentaire inhérent à sa structure (pour $N=3$ un retard supplémentaire de 1 échantillon, pour $N=4$ un retard supplémentaire de 2 échantillons, ...)

Les nouvelle matrice des coefficients $\overline{\overline{Q}}$ s'obtient en multipliant la matrice des coefficients précédente \overline{Q} par une matrice de transformation qui tient compte du glissement de D en d .

Le coût opératoire global de la structure de Farrow modifiée est ramené à une valeur équivalente à celle de l'implémentation des F.I.L. sous forme directe de R.I.F., le nombre de multiplication est même diminué.

Nous donnons en exemple ci-après la structure de Farrow modifiée d'ordre 3. Celle-ci approxime le mieux les retards compris dans l'intervalle $0 < d \leq 1$ (plus le retard supplémentaire d'un échantillon inhérent à la structure).

Pour $N=3$, la structure de Farrow directe est :

$$\overline{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -11/6 & 3 & -3/2 & 1/3 \\ 1 & -5/2 & 2 & -1/2 \\ -1/6 & 1/2 & -1/2 & 1/6 \end{bmatrix}$$

alors que la structure de Farrow modifiée est :

$$\overline{\overline{Q}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1/3 & -1/2 & 1 & -1/6 \\ 1/2 & -1 & 1/2 & 0 \\ -1/6 & 1/2 & -1/2 & 1/6 \end{bmatrix}$$

Le coût opératoire de la structure de Farrow modifié est alors de 10 additions et 9 multiplications (voir figure 2-1).

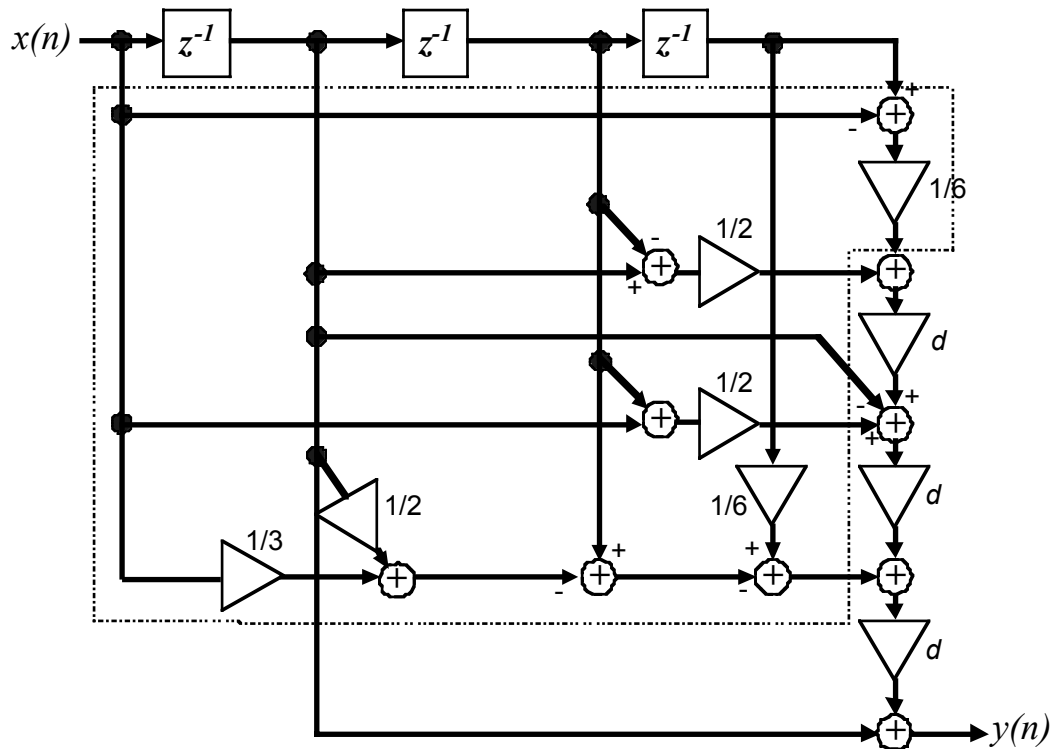


Figure 2-1. Schéma-bloc du F.I.L. sous forme de structure de Farrow modifiée d'ordre 3, la partie en encadré-pointillé renferme les coefficients \mathcal{A}_{kn} du filtre pouvant être mémorisés dans une table

La structure de Farrow modifiée a fait l'objet d'une première implémentation des retards fractionnaires variables en temps réel sur **Max-FTS**. Une présentation de ce logiciel sera faite dans la troisième partie de ce rapport. L'objet-externe "vd~" écrit par Miller Puckette contient un buffer circulaire implémentant une ligne de retard entier et une structure de Farrow modifiée d'ordre 4 implémentant la partie $-0.5 \leq d \leq 0.5$ du retard. Les états internes de la structure n'étant autre chose que des échantillons retardés, ils sont lus directement sur le buffer. La structure de Farrow "se déplace" donc le long du buffer (glissement des pointeurs d'états internes le long du buffer). Dans l'objet externe écrit par Miller-Pucket, les coefficients intermédiaires de la structure de Farrow sont stockés dans une table afin d'alléger le coût opératoire puisqu'il s'agit de temps réel.

2.2. Implémentation des F.I.L. sous forme de structure modulaire : expression récursive par développement en série de puissance

2.2.1. Nouvelle structure modulaire pour les F.I.L.

Cette nouvelle formulation des retards fractionnaires par filtres interpolateur de Lagrange utilise le développement en série de puissance de la réponse impulsionnelle idéale $\gamma_{id}(n)$ [22]. Elle permet l'expression des F.I.L. sous une forme récursive et une implémentation modulaire ce qui présente un sérieux avantage pour les retards fractionnaires variables (changement de l'ordre du filtre en fonction du retard à approximer).

Considérons l'extension analytique sur le plan complexe de la TF idéal $\mathfrak{z}^{-j\omega D} : z^{-D}$. Cette fonction est définie sur tout le plan complexe sauf sur le demi-axe des réels négatifs $]-\infty, 0]$ qui est un axe de discontinuité [22]. Cette fonction n'admet donc aucun développement en série de Laurent car elle devrait alors être au moins continue sur un anneau centré autour de zéro. Mais nous pouvons effectuer son développement en série entière au voisinage de 1 qui est un point absolument régulier pour z^{-D} . Le point 1 correspondant à la fréquence nulle, le développement constitue donc une approximation basse fréquence des retards fractionnaires.

Pour ce faire développons la fonction $(1+x)^D$, qui est définie sur tout le plan complexe excepté le demi-axe des réels négatifs $]-\infty, -1]$, en série entière :

$$(1+x)^D = \sum_{k=0}^{+\infty} \frac{D(D-1)\dots(D-k+1)}{k!} x^k, \quad \forall |x| < 1$$

Substituant x par $(z^{-1}-1)$, nous obtenons le développement en série de puissance de z^{-D} au voisinage de 1. On en déduit la famille de filtres R.I.F. suivant, dont les fonctions de transfert correspondent aux troncatures successives des développements de z^{-D} au voisinage de 1.

$$H_N^D(z) = \sum_{k=0}^N \frac{D(D-1)\dots(D-k+1)}{k!} (z^{-1}-1)^k$$

On constate que la famille de filtres R.I.F. $H_N^D(z)$ ainsi définie correspond aux filtres interpolateurs de Lagrange puisque leur fonction de transfert vérifie le critère de platitude maximale.

L'expression précédente se réécrit de manière récursive :

$$H_0^D(z) = 1 \text{ et } H_N^D(z) = H_{N-1}^D(z) + \frac{D(D-1)\dots(D-N+1)}{N!} (z^{-1}-1)^N$$

ce qui nous conduit à une forme modulaire des F.I.L.. Nous pouvons également la réécrire sous forme de schéma de Horner :

$$H_{N+1}^D(z) = 1 + D(z^{-1}-1) \left(1 + \frac{D-1}{2}(z^{-1}-1) \left(\dots \left(1 + \frac{D-N}{N+1}(z^{-1}-1) \right) \dots \right) \right)$$

En utilisant la notation $w^{-1} = (z^{-1}-1)$, le filtre F.I.L. peut donc maintenant s'implémenter sous la forme du schéma suivant (figures 2-2. et 2-3.):

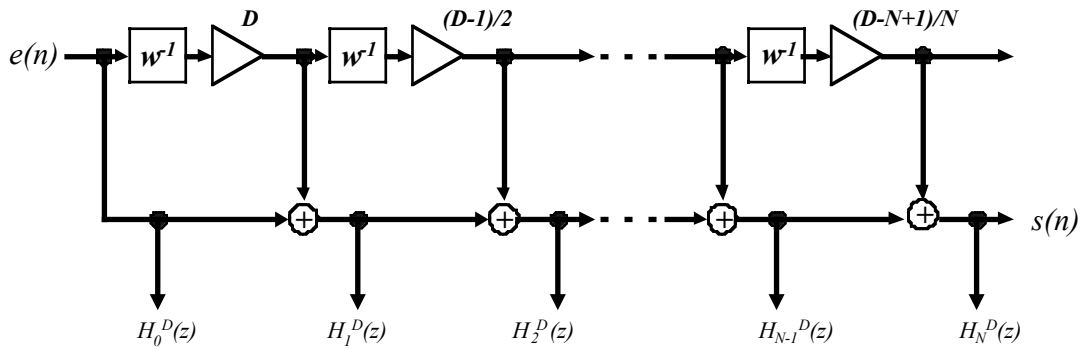


Figure 2-2. Schéma-bloc de la structure modulaire du F.I.L.

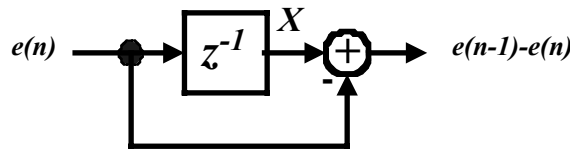


Figure 2-3. Détail de la fonction de transfert w^{-1}

Cette nouvelle implémentation des F.I.L. nécessite seulement N multiplications et $2N$ additions pour la réalisation du filtre, plus N additions entières et N multiplications pour la mise à jour des coefficients. Soit pour un filtre d'ordre 3 : 5 multiplications et 8 additions. Par comparaison avec la structure de Farrow modifiée d'ordre 3 (9 multiplications et 10 additions), ceci représente donc un gain de 4 multiplications et de 2 additions.

- Domaine de convergence des F.I.L. :

Le disque de convergence du développement en série effectué plus haut est le cercle unité centré en 1. Les F.I.L. convergent donc absolument à l'intérieur d'un cercle unité centré en 1, ce qui s'exprime en disant que les F.I.L. convergent absolument à basse fréquence $\phi \ll \frac{\pi}{6}$.

Utilisant une autre démarche [22], nous trouvons même que la convergence est garantie dans nos applications. En effet, si nous utilisons les F.I.L. dans la plage $\frac{N-1}{2} < D < \frac{N+1}{2}$, l'augmentation de l'ordre améliore l'approximation de D (symétrie par rapport à D) :

$$\forall z, \frac{1}{2} \left| \frac{z+z^{-1}}{2} - 1 \right| < 1, D \in]0,1[\lim_{N \rightarrow +\infty} z^N \cdot H_{2N+1}^{N+D}(z) = z^{-D}$$

$$\forall z, \frac{1}{2} \left| \frac{z+z^{-1}}{2} - 1 \right| < 1, D \in]0,1[\lim_{N \rightarrow +\infty} z^N \cdot H_{2N+2}^{N+D+1}(z) = z^{-D-1}$$

et comme le domaine de convergence contient le cercle unité (à l'exception du point -1), les F.I.L. convergent absolument vers la limite idéal) sauf à la fréquence de Nyquist qui reste un point singulier).

- Influence de la parité de l'ordre de F.I.L. :

Comme signalé précédemment (1.2.2.), l'erreur d'approximation des F.I.L. dépend fortement de la parité de l'ordre du filtre [11]. Les filtres d'ordre pair permettent de diminuer l'erreur en amplitude, et les filtres d'ordre impairs celle en phase. Nous montrons ci-après [22] une formulation de ces propriétés obtenue par développement de Taylor du gain en amplitude et en phase des F.I.L. :

$$\left| H_N^D(e^{j\omega}) \right| - 1 = -\alpha \omega^{N+1} \sin \frac{N\pi}{2} + \beta \omega^{N+2} \cos \frac{N\pi}{2} + o(\omega^{N+2})$$

$$\arg(H_N^D(e^{j\omega})) + D\omega = \alpha \omega^{N+1} \cos \frac{N\pi}{2} + \beta \omega^{N+2} \sin \frac{N\pi}{2} + o(\omega^{N+2})$$

$$\text{avec } \alpha = \prod_{k=0}^N \frac{D-k}{k+1}, \quad \text{et } \beta = \alpha \frac{N+1}{N+2} \left(D - \frac{N}{2} \right), \quad \text{et } D \in \left[\frac{N-1}{2}, \frac{N+1}{2} \right]$$

- Choix de l'ordre du F.I.L. :

L'ordre du filtre doit être choisi de manière à ce que $D \in \left] \frac{N-1}{2}, \frac{N+1}{2} \right[$. N est donc choisi égale à $\lfloor D/2 \rfloor$ où $\lceil D/2 \rceil$. Le pire des cas (plus mauvaise approximation) se présente lorsque $D = \frac{N}{2}$ (pour un F.I.L. d'ordre impair), pour $D = \frac{N \pm 1}{2}$ (pour un F.I.L. d'ordre pair) [4].

notation : $\lfloor x \rfloor$: représente le plus grand nombre entier inférieur à x

$\lceil x \rceil$: représente le plus petit nombre entier supérieur à x

- Bruit produit par l'erreur d'arrondi en arithmétique virgule flottante :

Le bruit apparaît pendant l'opération de stockage des valeurs du filtre, et pendant l'opération de multiplication par le gain [22]. Ce bruit peut-être évalué empiriquement par la loi expérimentale :

$$SNR(dB) = 10 - \frac{9}{2}N + 20g$$

où N est l'ordre du filtre et g le nombre de décimales utilisées pour effectuer les calculs. Il est facile de constater (voir [4] [22]) que pour un calcul en simple précision ce bruit est inférieur à -80dB lorsque le filtre est d'ordre inférieur à 20.

2.2.2. Utilisation de la structure modulaire dans le cas de retards fractionnaires variables

L'implémentation des filtres interpolateurs de Lagrange sous forme modulaire permet une utilisation aisée dans le cadre de retards fractionnaires variables. Le filtre d'ordre supérieur $N+1$ se déduit en effet simplement du filtre d'ordre N par l'adjonction d'un nouveau module à la suite des précédents. Remarquons que pour chaque nouvelle valeur du retard D nous pouvons choisir entre deux ordres de filtre différents $\lfloor D/2 \rfloor$ ou $\lceil D/2 \rceil$ et donc deux parités différentes. Néanmoins il est préférable de garder la parité du filtre constante lors du changement du retard car les parités paires et impaires optimisent deux critères différents (respectivement platitude en amplitude et platitude en phase). Dans notre implémentation nous garderons la parité de l'ordre du filtre constante si bien que les nouveaux modules seront ajoutés (ou soustraits) par paire.

- Algorithme de la structure modulaire

Nous proposons l'algorithme suivant [20] pour l'implémentation des F.I.L. sous forme modulaire.

où $X[i]$: représente l'état interne du système qui est gardé en mémoire d'un passage à l'autre;

e, b, d , et i : sont des variables intermédiaires;

$entrée$ et $sortie$: représentent l'entrée et la sortie du filtre.

Pour un filtre d'ordre N , l'algorithme parcouru à chaque échantillon s'écrit :

1) Initialisation

$sortie \leftarrow entrée$

$e \leftarrow entrée$

$b \leftarrow entrée$

$i \leftarrow 1$

$D \leftarrow -retard$

2) Boucle à parcourir tant que $i \leq N$

$e \leftarrow \frac{D}{T}(e - X[i])$

$X[i] \leftarrow b$

$e \leftarrow e$

$i \leftarrow i + 1$

$D \leftarrow D + 1$

$sortie \leftarrow sortie + e$

3) Dernière boucle

$e \leftarrow \frac{D}{T}(e - X[i])$

$X[i] \leftarrow b$

$e \leftarrow e$

$i \leftarrow i + 1$

$$X[i] \leftarrow b$$

La dernière boucle de l'algorithme sert à initialiser les états internes $X[N+1]$ et $X[N+2]$ de manière à anticiper une augmentation de l'ordre du filtre à $N+2$. L'augmentation ou la diminution de l'ordre du filtre se faisant quand la condition $D \in \left] \frac{N-1}{2}, \frac{N+1}{2} \right[$ n'est plus remplie. L'ordre du filtre est alors modifié de manière à maintenir D dans le bon intervalle. Cette algorithme permet donc l'augmentation de l'ordre du filtre de deux modules par passage.

Nous pourrions permettre une augmentation d'un nombre de modules supérieur à deux par passage, mais signalons que ceci correspond à augmenter le retard de plus d'un échantillon par passage d'un échantillon. Autrement dit ceci reviendrait à franchir le mur du son ! Un tromboniste poussant sa coulisse à plus de 300 mètres par seconde rattraperait son son !

Malgré cela, lors de notre implémentation temps réel (voir § 3.2.2.) nous serons obligé d'effectuer des variations supérieures à un échantillon. Ceci parce que notre implémentation se fera sur un logiciel traitant les signaux vectoriellement. Une variation du retard ne pourra s'appliquer que tout les 64 échantillons. Et faire varier le retard seulement d'un échantillon par passage de 64 échantillons nous conduirait à des variations trop lentes. Nous permettrons donc des variations de deux échantillons, ceci au détriment de la perte du sens physique d'une coulisse.

Signalons que notre but est de fournir une implémentation approximant les retards fractionnaires continûment variables. Pour les retards variant discrètement, le F.I.L. sera ré-initialisé. Aussi, dans la suite de ce texte, nous nous placerons dans le contexte d'un retard fractionnaire variant continûment.

- Comparaison de l'évolution de la réponse en amplitude de la structure modulaire et de la structure de Farrow modifiée

Dans le paragraphe 2.2.1., nous comparions le coût de calcul de la structure modulaire et celui de la structure de Farrow modifiée. Nous comparons maintenant l'évolution de leur réponse en amplitude lors d'une variation du retard. Les deux filtres ayant une fonction de transfert identique (mais implémentée différemment), lors d'un changement de retard l'évolution de leur fonction de transfert sera identique, sauf lorsque le retard total D sort de la plage de meilleur approximation ($D < \frac{N-1}{2}$ ou $D > \frac{N+1}{2}$). Dans ce dernier cas, les deux structures se comportent en effet différemment. Montrons cela en considérant le passage d'un retard $D=2.4$ à un retard $D=2.6$.

- Dans le cas de l'utilisation de la structure modulaire, le retard $D=2.4$ sera implémenté par un filtre d'ordre 4 et le retard $D=2.6$ s'obtiendra simplement en augmentant la valeur de d à 2.6 et en rajoutant deux modules au filtre (ordre 6) de manière à satisfaire la condition $d \in \left] \frac{N-1}{2}, \frac{N+1}{2} \right[$. Le paramètre d du filtre passe donc de $D=2.4$ à $D=2.6$.
- Dans le cas de l'utilisation d'une structure de Farrow modifiée d'ordre 4, le retard $D=2.4$ sera réalisé par le filtre seul : $d=0.4$ (plus le retard de deux échantillons inhérent à la structure de Farrow modifiée d'ordre 4). Lorsque D passe à la valeur 2.6, une partie du retard sera réalisée par une ligne de retard entier : $k=1$, le reste du retard par le filtre : $d=-0.4$. Ceci de manière à toujours satisfaire la condition $-0.5 \leq d \leq 0.5$. Le paramètre d du filtre saute donc brusquement de la valeur 0.4 à -0.4.

Les filtres interpolateur de Lagrange bénéficiant de la propriété de symétrie, i.e $h(n, \nu) = h(N-n, N-\nu)$ (voir § 1.2.2.), les coefficients du F.I.L. approximant le retard $D=2.6$

(respectivement $d=-0.4$) sont les coefficients du filtre implémentant $D=2.4$ (respectivement $d=0.4$) en ordre inverse :

$$h(4, 1.6) = h(0, 2.4), h(3, 1.6) = h(1, 2.4), \text{ et } h(2, 1.6) = h(2, 2.4).$$

Les réponses impulsionnelles sont donc symétriques l'une de l'autre et le rapport de leur réponse en amplitude est égale à l'unité :

$$\left| \frac{H_4^{1.6}(e^{j\omega})}{H_4^{2.4}(e^{j\omega})} \right| = 1.$$

Cependant, dans le cas de la structure de Farrow modifiée, le système que nous devons analyser n'est pas celui du filtre seul, mais bien celui du F.I.L. *et* de la ligne de retard entier (puisque le retard se répartit entre les deux). Lorsque D passe de 2.4 à 2.6, ce système passe de la fonction de transfert $H_4^{2.4}(z)$ à la fonction de transfert $z^{-1} \cdot H_4^{1.6}(z)$. Dans le cas de la structure modulaire seul l'ordre du filtre change et donc la fonction de transfert passe de $H_4^{2.4}(z)$ à $H_6^{2.6}(z)$.

A la figure 2-4., nous comparons la réponse en fréquence de la norme de la différence des fonctions de transfert avant/après dans le cas de la structure de Farrow $\left| H_4^{2.4}(z) - z^{-1} \cdot H_4^{1.6}(z) \right|$ et dans la cas de la structure modulaire $\left| H_4^{2.4}(z) - H_6^{2.6}(z) \right|$.

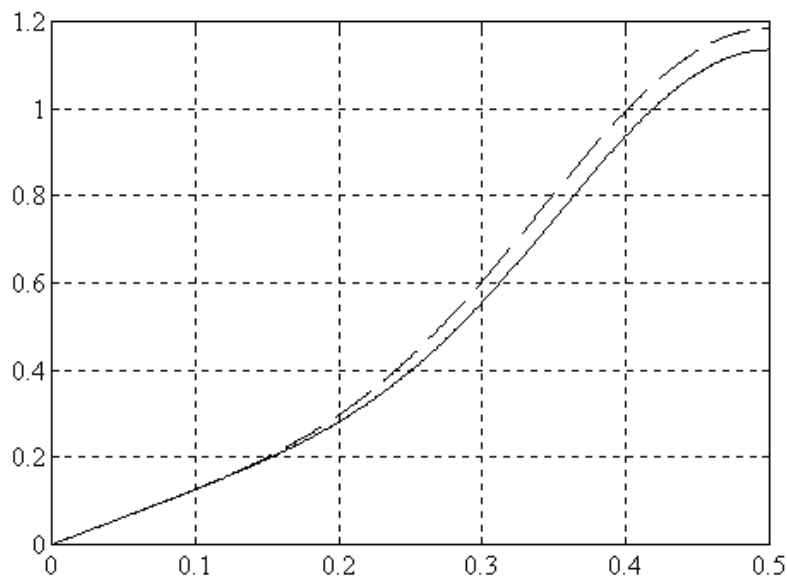


Figure 2-4. Réponse en fréquence de $H_4^{2.4}(z) - H_6^{2.6}(z)$ (ligne continue) et de $H_4^{2.4}(z) - z^{-1} \cdot H_4^{1.6}(z)$ (ligne en pointillé)

Nous constatons que les deux courbes sont très proches et que donc les deux systèmes se comportent de manière quasi-similaire lors d'un changement de retard. Néanmoins, à haute fréquence la fonction de transfert de la structure modulaire semble présenter une variation plus faible que celle du système incluant la structure de Farrow. Rappelons cependant que l'approximation des retards fractionnaires par filtres interpolateurs de Lagrange n'est valable qu'à basse fréquence (§ 1.2.2.1). Et en basse fréquence, les deux systèmes présentent une variation de fonction de transfert quasi-identique. Les deux structures sont donc équivalentes de ce point de vue.

2.2.2.1. Ajout d'une ligne de retard entier au système

Dans l'algorithme présenté ci-dessus, nous ne pouvons limiter la variation de l'ordre du F.I.L. sans limiter la variation du retard D (N étant de l'ordre de $2D$). Afin de pouvoir limiter l'ordre de variation du F.I.L., nous séparons le retard d en

$$D = d + k, \text{ où } k \in \mathbb{N}$$

où le retard implémenté par le F.I.L. est maintenant d , k est une valeur entière de retard implémentée sous forme d'une ligne de retard entier (buffer circulaire). Insistons sur le fait que d n'est pas la partie fractionnaire de D .

De cette manière, la limitation de l'ordre de variation du F.I.L. ne limite plus maintenant que la valeur de d , et non plus le retard total D .

Le schéma du système réalisant le retard total D se présente donc maintenant sous la forme d'une ligne de retard entier (buffer circulaire) réalisant le retard entier k dont la sortie est raccordée à un F.I.L. implémenté sous forme modulaire, réalisant le retard d (figure 2-5.).

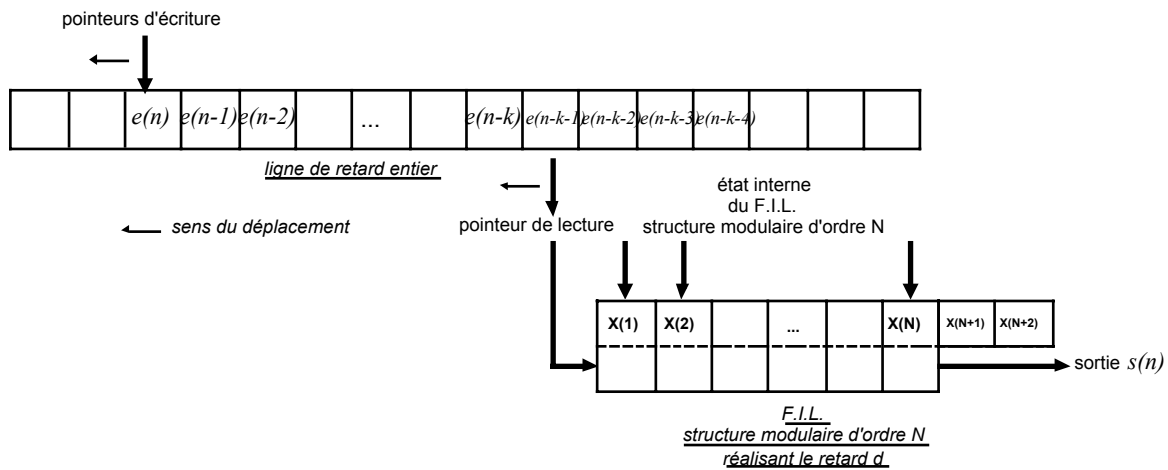


Figure 2-5. Schéma du système global réalisant le retard D : ligne de retard entier k suivie d'un F.I.L. sous forme de structure modulaire d'ordre N et de variables internes $X[i]$, réalisant le retard d

Dans la comparaison de l'évolution de la réponse en amplitude que nous avons effectué plus haut, nous n'avons pas tenu compte du fait que dans l'utilisation de la structure modulaire nous devons parfois aussi transférer une partie du retard total vers la ligne de retard entier. Cependant ce transfert n'est pas systématique car, grâce à la modularité de notre filtre, une variation de l'ordre du filtre est très aisée. Inversement le transfert d'une partie de retard du F.I.L. vers la ligne de retard entier ou inversement est une opération plus délicate comme nous allons le voir (§ 2.2.2.2.). Lorsque le retard D varie, nous préférons donc faire varier d (et donc changer l'ordre du filtre) que de systématiquement répartir D entre d et k (cas de la structure de Farrow modifiée). Ceci est vrai pour autant que le filtre reste de taille raisonnable. Lorsque l'ordre du filtre devient trop important, une partie du retard sera calculé par la ligne de retard entier ou, inversement, lorsque l'ordre du filtre devient trop petit, il récupérera une partie du retard implémenté par la ligne de retard entier. En pratique nous limiterons l'ordre du F.I.L. entre 5 et 13 pour les filtres d'ordre impair, et entre 6 et 14 pour les filtres d'ordre pair.

2.2.2.2. Mise à jour de l'état du système lors de transferts entre ligne de retard entier et F.I.L.

Pour commencer rappelons le processus de transfert de retard entre ligne de retard entier et F.I.L. dans le cas de l'utilisation de la structure de Farrow modifiée.

Lors de l'implémentation des retards fractionnaires sous forme de structure de Farrow modifiée, la partie du retard implémentée par le filtre est calculée de manière à respecter le critère $0 < d \leq 1$ (pour un filtre d'ordre impair) ou $-0.5 \leq d \leq 0.5$ (pour un filtre d'ordre pair). La partie du retard restante k_1 (entier) étant calculée par la ligne de retard entier. Si le retard que doit implémenter le filtre dépasse la limite supérieure (respectivement inférieure), c'est-à-dire si $d \geq 1$ ou $d \geq 0.5$ selon la parité du filtre (respectivement $\mathcal{I} \leq 0$ ou $\mathcal{I} \leq -0.5$), un nouveau retard entier k_2 est calculé de manière à vérifier le critère sur d . L'augmentation du retard entier de k_1 à k_2 se fait par simple glissement arrière (sur le buffer circulaire) des 5 pointeurs de lecture (de la ligne de retard entier) de $k_2 - k_1$ échantillons.

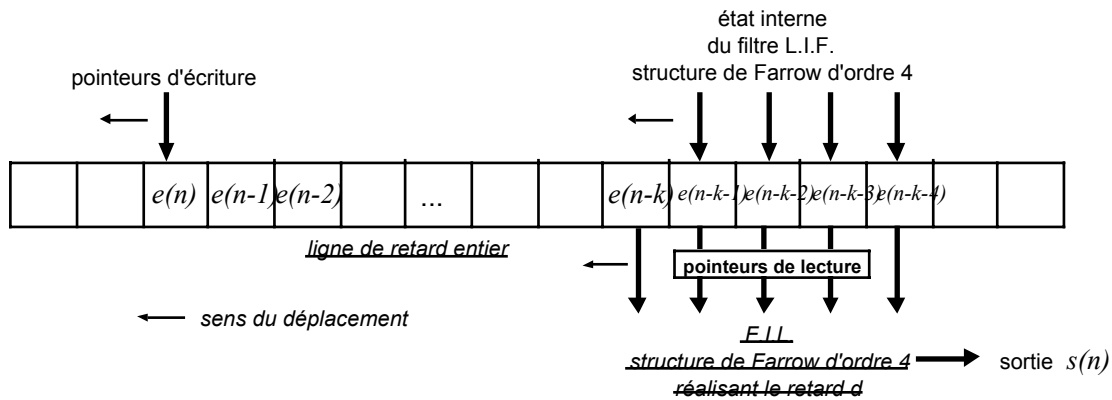


Figure 2-6. Schéma du système global réalisant le retard D : ligne de retard entier k suivie d'un F.I.L. sous forme de structure de Farrow modifiée d'ordre 4 et d'état interne $x(n-k-1)$, $x(n-k-2)$, $x(n-k-3)$, $x(n-k-4)$, réalisant le retard D

Dans notre nouvelle implémentation sous forme de structure modulaire, la mise à jour des coefficients du F.I.L. est plus délicate puisque les états internes du filtre X^L ne sont plus simplement des échantillons du passé gardés en mémoire dans le buffer circulaire. Nous ne pouvons donc plus simplement déplacer le pointeur de lecture et ceux de l'état interne d'un certain nombre d'échantillons à gauche ou à droite.

Procédure de mise à jour de l'état du système par lecture des entrées passées dans le buffer circulaire

Une première idée serait de recalculer l'état interne du F.I.L. sous forme de structure modulaire à partir des échantillons contenus dans le buffer circulaire (figure 2-5.).

Définissons $H_i(z)$ comme la fonction de transfert qui permet de reconstruire l'état interne $X[i]$ du F.I.L. de retard d à partir de ses échantillons d'entrées passées $\vartheta(n)$. Ces fonctions de transfert peuvent s'écrire (voir figure 2-2 et 2-3) :

$$\begin{aligned} H_1[z] &= z^{-1}, \\ H_2[z] &= z^{-1}d[(z^{-1} - 1)], \\ H_3[z] &= z^{-1} \frac{d-1}{2} [(z^{-1} - 1)](z^{-1} - 1)d, \\ &\dots \end{aligned}$$

Lors d'un transfert de retard, le nouveau signal que doit traiter le F.I.L. à changer, et donc les états internes $X[i]$ que le F.I.L. contient ne correspondent plus au nouveau signal à traiter. Les états internes que le F.I.L. devrait contenir pour traiter le nouveau signal peuvent se calculer directement en utilisant les échantillons contenu dans le buffer circulaire.

En effet, lorsque le signal d'entrée du F.I.L. était $\vartheta(n)$, ses états internes correspondaient à des entrées passées :

$$\begin{aligned} X[1] &= \vartheta(n-1), \\ X[2] &= [\vartheta(n-2) - \vartheta(n-1)] \cdot d \\ X[3] &= [\vartheta(n-3) - 2 \cdot \vartheta(n-2) + \vartheta(n-1)] \cdot d \cdot \frac{d-1}{2}, \\ &\dots \end{aligned}$$

Lorsque le nouveau signal d'entrée passe à $\vartheta(n-k)$, ses nouveaux états internes doivent correspondre à des entrées passées $\vartheta(n-k-1)$, $\vartheta(n-k-2)$, $\vartheta(n-k-3)$. Il suffit donc de lire ces entrées passées dans le buffer circulaire (voir figure 2-5.) et de calculer les nouveaux états internes à l'aide des fonctions de transfert $H_i(z)$:

$$\begin{aligned} X[1] &= \vartheta(n-k-1), \\ X[2] &= [\vartheta(n-k-2) - \vartheta(n-k-1)] \cdot d \\ X[3] &= [\vartheta(n-k-3) - 2 \cdot \vartheta(n-k-2) + \vartheta(n-k-1)] \cdot d \cdot \frac{d-1}{2}, \\ &\dots \end{aligned}$$

Toutefois il n'est pas forcément possible d'allouer la mémoire nécessaire au stockage des valeurs $\vartheta(n-k-1)$, $\vartheta(n-k-2)$, ... (cas d'un petit buffer circulaire). Dans ce cas, il reste possible de recalculer les valeurs d'entrées uniquement à partir de d et le vecteur d'état interne du système. Nous présentons cette procédure de remise à jour de l'état du système dans la suite. Remarquons que cette procédure englobe la précédente mais est plus générale.

Procédure de mise à jour de l'état du système par recomposition des entrées passées (uniquement) à partir de l'état interne actuelle

- $k \rightarrow k+1$ $N \rightarrow N-2$

Prenons comme exemple le transfert d'un échantillon de la ligne fractionnaire vers la ligne entière. Un nouvel élément de la ligne de retard entier doit être créé et deux modules du F.I.L. doivent être supprimés.

Le dernier échantillon de la ligne de retard entier est créé par copie de $X[k]$, première valeur du vecteur d'état du F.I.L.. Les autres valeurs du vecteur d'état du F.I.L. sont décalées "de manière appropriée" i.e. de manière à correspondre effectivement à la nouvelle portion du signal à traiter (puisque le signal a été "glissé") et aux bonnes valeurs des coefficients multiplicatifs (d , $(d-1)/2$, ...).

Ceci se produit lorsque l'ordre du filtre devient trop important. (>13 pour les filtres impairs, >14 pour les filtres pairs). Cette procédure une fois entamée sera répétée échantillon après échantillon jusqu'à ce que l'ordre du filtre reviennent à la valeur centrale (9 pour les filtres impairs, 10 pour les filtres pairs), ceci afin de retarder le plus possible un nouvel appel à cette procédure de transfert (l'ordre du filtre ramené à 9 peut alors varier de 4 sans nouveau besoin de transfert).

- $k \rightarrow k-1$ $N \rightarrow N+2$

L'autre cas est celui du transfert d'un échantillon de la ligne entière vers la ligne fractionnaire. Un élément de la ligne de retard entier est supprimé et deux modules supplémentaires doivent être ajoutés au F.I.L..

La première valeur du vecteur d'état du F.I.L. $X[k]$ est créé par copie du dernier échantillon de la ligne de retard entier. Les autres valeurs du vecteur d'état du F.I.L. sont décalées "de manière appropriée". La valeur supplémentaire $X[N+2]$ nécessitée par le passage de l'ordre du F.I.L. de N à $N+2$ est fournie par la dernière boucle de l'algorithme (voir 2.2.2.), cet état devant être décalé "de manière appropriée" en $X[N+2]$.

Ceci se produit lorsque l'ordre du filtre devient trop petit et cette procédure une fois entamée sera, de la même manière que précédemment, répétée jusqu'à ce que l'ordre du filtre atteigne la valeur centrale, ceci afin de retarder le plus possible un nouvel appel à cette procédure de transfert (puisque l'ordre du filtre ramené à 9 peut alors être diminuer de 4 sans nouveau besoin de transfert).

Pour réaliser la procédure de changement d'état nous devons :

- (1) dans un premier temps, calculer ou récupérer dans le buffer circulaire les valeurs d'entrée passées du F.I.L., i.e. les échantillons qui ont permis la création des états internes actuelles
- (2) dans un deuxième temps, à partir de ces valeurs et de celles de la ligne de retard entier actuelle, ré affecter les échantillons entre ligne de retard entier et F.I.L.
- (3) dans un dernier temps, reconstruire les états internes du F.I.L. à partir de ces échantillons

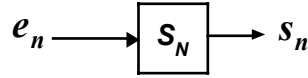
Nous présentons maintenant les différentes étapes du calcul permettant la mise à jour des états du système :

1) Passage de l'état du système au passé du système

Soit le système \mathfrak{S}_N correspondant à l'implémentation d'un filtre d'ordre N décrit formellement par l'équation d'état :

$$\begin{cases} \overline{X}[n+1] = A\overline{X}[n] + Be_n \\ s_n = g(\overline{X}[n], e_n) \end{cases}$$

où e_n est l'entrée du filtre, s_n sa sortie, et $\overline{X}[n]$ son vecteur d'état interne.



Ce qui peut se réécrire de manière récursive :

$$\forall n, \overline{X}[n] = A^n \overline{X}[0] + A^{n-1}Be_0 + A^{n-2}Be_1 + \dots + ABe_{n-2} + Be_{n-1}$$

On suppose le filtre au repos à l'état initial donc $\overline{X}[0] = \overline{0}$.

Si nous nous restreignons aux systèmes qui vérifient $A^N = 0$ (ce qui correspond au cas des filtres R.I.F.), alors

$$\overline{X}[n] = A^{N-1}Be_{n-N} + \dots + ABe_{n-2} + Be_{n-1}$$

en notant $\overline{\Omega} = (B, AB, A^2B, \dots, A^{N-1}B)$ [eq. 1]

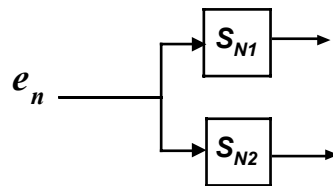
et $\overline{e}_n^N = (e_{n-1}, e_{n-2}, \dots, e_{n-N})^T$

Nous obtenons l'équation permettant de remonter au passé du système à partir de ses états actuels :

$$\overline{X}[n] = \overline{\Omega} \cdot \overline{e}_n^N \quad \text{[eq. 2]}$$

2) Changement d'état d'un système

Considérons le passage d'un système S_{N_1} d'état interne $\overline{X}_1[n]$ à un système S_{N_2} d'état interne $\overline{X}_2[n]$ d'ordre supérieur $N_2 \geq N_1$.



Grâce à l'équation [2] nous avons :

$$\forall n > N, \overline{X}_1[n] = \overline{\Omega}_1 \overline{e}_n^{N_1} \quad \text{et} \quad \overline{X}_2[n] = \overline{\Omega}_2 \overline{e}_n^{N_2}$$

Comme les entrées des deux systèmes sont liées par la relation suivante :

$$\overline{e_n^{N_1}} = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ddots & 1 \end{pmatrix} \overline{e_n^{N_2}}$$

$$\begin{matrix} \leftrightarrow & \leftrightarrow \\ N_2 & N_1 - N_2 \end{matrix}$$

les états internes du système 2 sont liés à ceux du système 1 par :

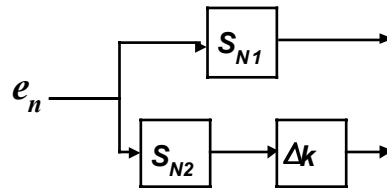
$$\overline{X_2[n]} = \overline{\Omega_2} \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ddots & 1 \end{pmatrix} \overline{\Omega_1^{-1}} \overline{X_1[n]}$$

[eq. 3]

$$\begin{matrix} \leftrightarrow & \leftrightarrow \\ N_2 & N_1 - N_2 \end{matrix}$$

3) Système rallongé par une ligne de retard entier

Considérons maintenant le déplacement d'état d'un système d'ordre N_1 à un système d'ordre N_2 incluant un retard entier k ($N_2 \leq N_1 \leq N_2 + k$).



$\overline{X_{\Delta_k + N_2}}$ correspond à l'état du système connectant en série la ligne de retard entier k et le filtre

d'ordre N_2 . La convention de rangement est la suivante $\overline{X_{\Delta_k + N_2}} = \begin{bmatrix} \overline{X_{\Delta_k}^T} \\ \overline{X_{N_2}^T} \end{bmatrix}$.

L'équation d'état du système $S_{\Delta_k + N_2}$ peut s'écrire :

$$\overline{X_{\Delta_k + N_2}[n+1]} = \begin{pmatrix} \sigma & 0 & 0 \\ \overline{Id} & \sigma^T & \sigma^T \\ \sigma & B & A \end{pmatrix} \overline{X_{\Delta_k + N_2}[n]} + \begin{pmatrix} 1 \\ \sigma^T \\ \sigma^T \end{pmatrix} e_n$$

$$\begin{matrix} \leftrightarrow & \leftrightarrow & \leftrightarrow \\ p-1 & 1 & N_2 \end{matrix}$$

En faisant intervenir $\overline{A_{\Delta_k + N_2}}$ et $\overline{B_{\Delta_k + N_2}}$ qui représentent respectivement la matrice d'état et la matrice de contrôle du système, nous calculons comme précédemment la matrice $\overline{\Omega_2}$ [eq. 1]:

$$\overline{\Omega_2} = (\overline{B_{\Delta_k + N_2}}, \overline{A_{\Delta_k + N_2}} \overline{B_{\Delta_k + N_2}}, \overline{A_{\Delta_k + N_2}}^2 \overline{B_{\Delta_k + N_2}}, \dots, \overline{A_{\Delta_k + N_2}}^{N_2-1} \overline{B_{\Delta_k + N_2}})$$

ce qui donne après développement le résultat intéressant suivant :

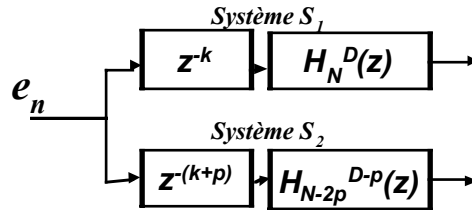
$$\overline{\Omega_2} = \begin{pmatrix} \overline{Id} & \sigma \\ \sigma & \overline{\Omega_{S_2}} \end{pmatrix} = \begin{pmatrix} \overline{Id} & \sigma \\ \sigma & \overline{\Omega_{S_2}} \end{pmatrix}$$

[eq. 4]

Son inverse s'obtient aisément puisque : $\overline{\Omega_{\Delta_k+N_2}^{-1}} = \begin{pmatrix} \overline{Id} & \overline{0} \\ \overline{0} & \overline{\Omega_{S_2}^{-1}} \end{pmatrix}$ [eq. 5]

4) Rattachement au processus de mise à jour des F.I.L. implémentés sous forme modulaire

Dans notre cas, nous désirons passer d'un système S_1 comprenant une ligne de retard entier k et un F.I.L. d'ordre N et de retard d , à un système S_2 comprenant une ligne de retard $k+p$ et un F.I.L. d'ordre $N-2p$ et de retard $D-p$.



Dans le cas des retards fractionnaires, les matrices du système s'écrivent (voir figures 2-2. et 2-3.) :

$$A_{(N,d)} = \begin{pmatrix} 0 & & & & 0 \\ \frac{d-1}{2} & \ddots & & & \\ & \ddots & & & \\ 0 & & \frac{d-N+1}{N} & & 0 \end{pmatrix} \text{ et } B_{(N,d)} = \begin{pmatrix} d \\ \sigma^T \end{pmatrix} \updownarrow N$$

\Leftrightarrow
 N

Si nous définissons \tilde{X} comme l'état interne normalisé du F.I.L., i.e.

$$\tilde{X} = \begin{pmatrix} d & & & \sigma \\ & \frac{d-1}{2} & & \\ & & \ddots & \\ \sigma & & & \frac{d-N+1}{N} \end{pmatrix} \tilde{X} \quad [\text{eq. 6}]$$

Celui-ci peut s'écrire (voir [20]) :

$$\tilde{X}[n+1] = \left[Id + A_{(N,d)} \right]^{-1} A_{(N,d)} \tilde{X}[n] + \left[Id + A_{(N,d)} \right]^{-1} B_{(N,d)} e_n$$

En utilisant la notation $A_{(N,d)} = \left(Id + A_{(N,d)} \right)^{-1} \cdot A_{(N,d)}$ et $B_{(N,d)} = \left(Id + A_{(N,d)} \right)^{-1} \cdot B_{(N,d)}$, nous pouvons trouver comme précédemment la matrice $\overline{\Omega_{(N,d)}}$ [eq. 1] correspondant au système S_1 ; nous faisons de même pour le système S_2 . Utilisant alors les équations [3] [4] [5] et [6], nous trouvons les états internes du système S_2 à partir de ceux du système S_1 .

Pour cela, nous avons donc effectué la suite d'opérations suivante :

- (1) normalisation de l'état interne du système S_1 ,
- (2) calcul des entrées passées correspondant aux états internes actuels du système S_1 ,
- (3) réaffectation (selon la valeur de k) de ces entrées passées entre la ligne λ_{k+p} de retard entier $k+p$ et le F.I.L. d'ordre $N - 2p$ implémentant le retard fractionnaire $d-p$,
- (4) calcul de l'état interne du système S_2
- (5) dénormalisation de l'état interne du système S_2 .

Les transformations appliquées aux états internes du système S_1 en vue de retrouver les états internes du système S_2 sont donc :

$$X[n]^2 = \begin{pmatrix} Id & \dots & 0 \\ \frac{1}{d-p} & & \\ \vdots & \frac{2}{d-p-1} & \vdots \\ 0 & \dots & \frac{N}{d-p-N+1} \end{pmatrix} \begin{pmatrix} Id & 0 \\ 0 & \Omega_{(N-2p,d-p)} \end{pmatrix} (Id) \begin{pmatrix} Id & 0 \\ 0 & \Omega_{(N,d)} \end{pmatrix} \begin{pmatrix} Id & \dots & 0 \\ d & & \\ \vdots & \frac{d-1}{2} & \vdots \\ 0 & \dots & \frac{d-N+1}{N} \end{pmatrix} X[n]^1$$

$\overline{\Omega_{(N-2p,d-p)}}$ est une matrice triangulaire inférieur, dont les valeurs sont

$$\text{pour } l \geq j, \overline{\Omega_{(N-2p,d-p)}}(i, j) = \frac{(-1)^{(i+j)}}{i(j-1)!(i-j)!} [d-p)(d-p-1)\dots(d-p-i+1)],$$

de même que $\overline{\Omega_{(N,d)}^{-1}}$:

$$\text{pour } l \geq j, \overline{\Omega_{(N-2p,d-p)}^{-1}}(i, j) = \frac{j(l-1)!}{(l-j)!} \frac{1}{d(d-1)\dots(d-j+1)}$$

Dans le cas où p est négatif (cas du transfert d'une partie du retard, de la ligne de retard entier vers le F.I.L.), le nombre d'états internes du F.I.L. que nous devons recomposer est supérieur aux échantillons reçus de la ligne de retard entier (car $k \rightarrow k+p$ et $N \rightarrow N-2p$). Dans ce cas, nous utiliserons aussi pour la reconstruction, les états internes d'ordre supérieur à N (i.e. ceux calculés dans la dernière boucle de l'algorithme du § 2.2.2. qui servent à anticiper une augmentation de l'ordre du filtre).

Les deux procédures que nous venons de présenter pour la mise à jour de l'état du système (par lecture des entrées passées dans le buffer circulaire ou par recombinaison des entrées passées à partir de l'état interne actuelle) ne sont pas équivalente lorsque d varie. La deuxième tenant compte des variations de d que le signal a subi dans le passé pour la reconstruction du signal.

C'est cette deuxième procédure que nous utiliserons dans la suite pour la mise à jour de l'état de notre système.

2.2.2.3. Simulation sur Matlab

Afin de clarifier les idées, nous montrons ci-après le résultat d'une simulation sur Matlab d'un système bouclé incluant la ligne de retard entier, le filtre interpolateur de Lagrange sous forme modulaire ainsi que la procédure de remise à jour de l'état interne du système par recombinaison des entrées passées.

Le système bouclé est une version simplifiée du modèle qui sera présenté dans la troisième partie. Disons simplement pour l'instant qu'il produit une forme d'onde périodique pseudo-

rectangulaire de fréquence variant en fonction de la longueur du retard. Nous l'utilisons ici afin

- de montrer dans quels cas la procédure de transfert entre ligne de retard entier et F.I.L. est utilisée,
- de montrer comment se répartit le retard total entre la ligne de retard entier et le F.I.L. sous forme modulaire,
- de pouvoir observer l'influence sur le signal de la procédure de remise à jour.

Le schéma-bloc du système bouclé utilisé est donné à la figure 2-7.

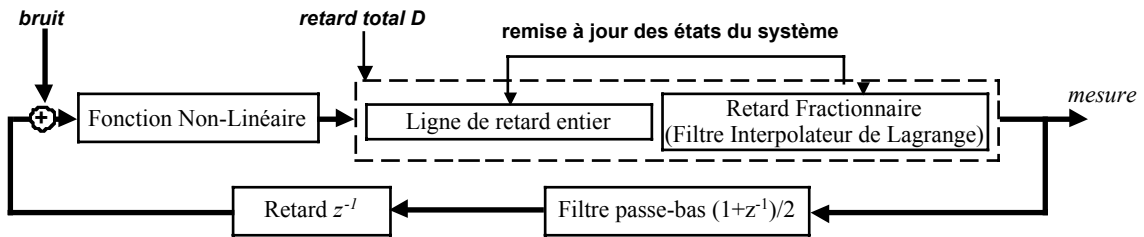


Figure 2-7. Système bouclé produisant une forme d'onde périodique pseudo-rectangulaire incluant la ligne de retard entier, le F.I.L. sous forme modulaire, et la procédure de remise à jour des états du système par recomposition des entrées passées

A la figure 2-8., nous montrons l'évolution des différents paramètres du système ainsi que le signal mesuré à la sortie du F.I.L. en fonction du temps (pour un signal de 1500 échantillons).

D est la valeur du retard total variant continûment

d est la valeur du retard implémenté par le F.I.L.

N est l'ordre du F.I.L.

" $d \leftarrow k$ " prend la valeur **1** lors d'un transfert d'une partie du retard du F.I.L. vers la ligne de retard entier ($k \rightarrow k+1, d \rightarrow d-1$), **-1** dans le cas contraire ($k \rightarrow k-1, d \rightarrow d+1$), et **0** lorsqu'il n'y a pas de transfert.

k est la longueur de la ligne de retard entier

$y(n)$ est le signal mesuré en sortie du F.I.L.

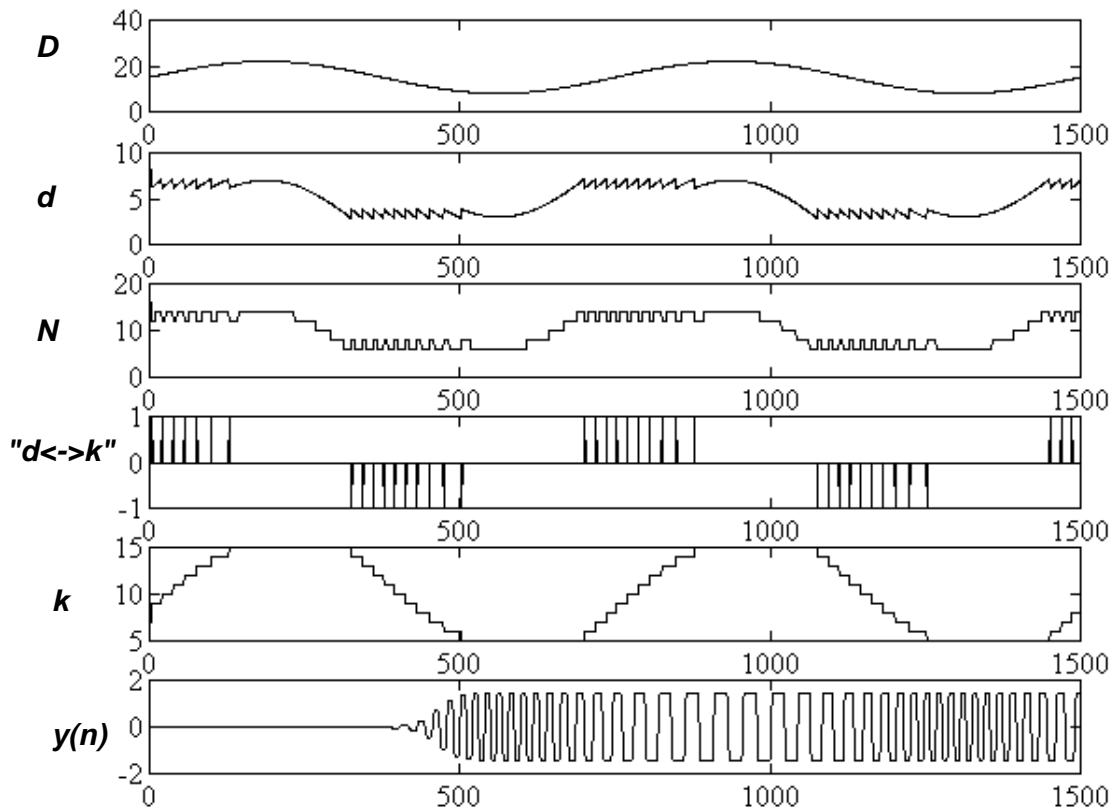


Figure 2-8. Retard total D , retard F.I.L. d , ordre du F.I.L. N , transfert $d \rightarrow k$ (1) ou $k \rightarrow d$ (-1), retard entier k , signal de sortie $y(n)$ (simulation sur Matlab du modèle de la figure 2-7. pour un signal de 1500 échantillon)

A partir de ces graphes, nous pouvons déduire que la procédure de remise à jour de l'état interne du système produit une discontinuité sur le signal mesuré $y(n)$ quasi-inexistante en tout cas très faible.

C'est cette implémentation de "ligne de retard entier + F.I.L.. sous forme modulaire + procédure de remise à jour des états du système par recombinaison des entrées passées" que nous avons porté sous forme vectorielle sur Max/FTS pour la réalisation d'un retard variable en temps réel.

3. IMPLEMENTATION EN TEMPS REEL DE LA STRUCTURE MODULAIRE DES F.I.L. DANS UN MODELE PHYSIQUE SIMPLIFIE D'INSTRUMENT A TUBE

Dans cette troisième partie, nous utilisons la nouvelle implémentation des F.I.L. sous forme de structure modulaire pour la réalisation d'un modèle physique simplifié d'instrument à tube. Cette réalisation est effectuée grâce au logiciel temps-réel FTS de l'IRCAM par la connexion dans Max d'objets externes réalisant les différentes parties du modèle. Pour cela, de nouveaux objets externes sont créés. Il s'agit des objets "xct~", "fd~" et "miditofreq".

Les paramètres du modèle peuvent être changés en temps réel via une interface Midi (un clavier Midi dans notre réalisation) ou via l'interface graphique de Max.

Après une présentation du logiciel FTS et de son interface graphique actuelle Max, nous présentons le modèle physique simplifié d'instrument à tube, et détaillons ses différentes parties.

3.1. Présentation de Max/FTS

FTS ("Faster Than Sound") est la plate-forme temps-réel de l'IRCAM pour le traitement numérique du son. FTS est un serveur de contrôle et de traitement du signal qui peut être connecté à une quelconque application. L'application communique avec FTS localement ou via un réseau sur un modèle client/serveur. L'interface utilisateur est donc dissociée de la gestion temps-réel de bas niveau. L'interface utilisateur que nous utiliserons est Max. Il s'agit d'un environnement de programmation graphique reproduisant l'ensemble des fonctions de FTS sous formes de modules (objets représentant des programmes) interconnectés. L'ensemble des modules interconnectés forment alors un "patch" contrôlable via l'interface graphique ou l'interface Midi. FTS exécute en temps réel un "patch" de Max en fournissant les services suivants : ordonnancement des traitements (selon les connections entre objets effectuées, cet ordonnancement s'effectue à l'initialisation du "patch"), communications entre les différents processeurs de l'application (lorsqu'un patch est réparti sur différents processeurs), gestion de la mémoire (allocation de tampons de calcul des échantillons, ...), gestion des entrées-sorties (synchronisation avec les convertisseurs numérique-analogique et analogique-numérique, ...).

L'utilisateur peut également créer ses propres modules, ce que l'on appelle les *objets externes*. Ceux-ci sont programmés en C (ANSI) et une fois compilés sont joints aux objets préexistant des bibliothèques FTS. Certaines de ces bibliothèques fournissent des services pour le développeur d'applications : objets de contrôle et de traitement du signal tel que processeurs arithmétiques et logiques, manipulation des données en nombre entiers ou flottants, contrôle par envoi et réception de message, gestion du temps, gestion du MIDI, synthèse additive, filtrage, ... D'autres bibliothèques permettent directement la manipulation des ressources du système, et donc de modifier le noyau de FTS (FTS kernel).

FTS inclut donc à la fois les fonctions propres à un système temps-réel de noyau distribué et un système d'objets basé sur une communication par messages.

Objets (Externes) de FTS

Nous présentons brièvement ici le concept d'objets externes dans FTS. Remarquons qu'actuellement sur FTS, il n'y a plus de différences entre objets externes et objets internes. Nous parlerons dès lors d'objets.

Un objet de FTS est un état (une valeur gardée en mémoire) et un jeu de connections. Ces connections sont des liaisons entre *inlets* (entrées d'une boîte Max) et *outlets* (sortie d'une boîte Max). Un *inlet* prend en entrée un message ou un signal DSP (s'il s'agit d'un objet DSP). Ce message est composé d'un sélecteur (un symbole) et d'une liste d'arguments. Lorsqu'un *inlet* reçoit un message, la méthode (une fonction en C) correspondante au sélecteur du message est appelée. La définition de l'ensemble des méthodes d'une classe associe donc une méthode à un sélecteur et à un numéro d'*inlet*. Une classe définit un ensemble d'objets ayant même nombre d'*inlets* et d'*outlets*, mêmes associations entre méthodes, sélecteurs et numéro d'*inlet*.

La définition d'un objet comprend les étapes suivantes :

- définition de l'objet :


```
typedef struct
    {   fts_object_t o
        ...
    } mon_objet_t
```
- définition des méthodes : `void ma_méthode (...)`
- envois de messages par *outlets* : `fts_outlet_send (...)`
- définition de la classe :


```
fts_status_t   mon_objet_instantiate (... )
{
    fts_class_init      initialisation de la classe, définition du nombre
                        d'inlets et d'outlets des objets de la classe

    fts_s_int (... )   méthode propre au système
    fts_s_delete (... ) méthode propre au système
    fts_method_define (... ) associations entre messages reçu en inlets et
                        méthodes appelées
    fts_outlet_type_define (... ) définition du type de message émis
                        par les outlets
}
```

Un objet DSP comporte en plus :

- définition des *inlets* et *outlets* recevants et sortants des signaux DSP :


```
dsp_sig_inlet ( ... )
dsp_sig_outlet ( ... )
```
- méthodes d'insertion et de retrait de l'objet dans la chaîne DSP :


```
dsp_list_insert
dsp_list_remove
```

- méthode donnant à la chaîne DSP les informations nécessaires concernant l'objet :

```
void mon_objet_put ( ... )
```

- méthode de traitement DSP appelée à chaque paquet d'échantillon à traiter :

```
void mon_objet_dsp_function ( ... )
```

A l'initialisation d'un "patch" dans Max, chaque objet DSP déclare ses propriétés à la chaîne DSP, et celle-ci se construit par ordonnancement des traitements à effectuer (dépendant de la manière dont les objets sont raccordés dans Max). Lorsque le "patch" tourne, la chaîne DSP effectue une suite d'appels aux fonctions de calculs vectoriels de FTS. Le moteur de calcul vectoriel DSP de FTS s'appelle FTL ("Faster Than Light"). Il est optimisé à l'heure actuelle pour le traitement par paquets de 64 échantillons. Dans les versions prochaines de FTS cette taille devrait être définissable par l'utilisateur.

Développement d'objets externes sur Max/FTS

Pour la réalisation temps-réel de notre modèle, nous avons travaillé sur la version 1.3 (patch level 7) de FTS pour station Silicon Graphic (une station Indigo de SGI pour être exact). Il s'agit de la première version distribuée de FTS pour ce type de station. Cette version est encore en développement ce qui explique les quelques problèmes auxquels nous nous sommes heurtés. Les versions prochaines devraient sans doute permettre un développement plus aisé.

Tous nos calculs sont effectués en *virgule flottante* (du moins pour le calcul des coefficients du filtre) : `type fts_float`.

La *fréquence d'échantillonnage* que nous avons utilisé est de 22050 Hz. Ceci pour des raisons de puissance de calcul de la machine. Notons que Max/FTS consomme près de 90% de la puissance de calcul de la machine Indigo Silicon Graphic 100 MHz (machine sur laquelle nous avons travaillé) lorsque le patch complet tourne (Max/FTS en consomme déjà près de 55% lorsqu'aucun patch ne tourne). Remarquons que cette puissance reste plus ou moins constante quelque soit la variation des paramètres en entrée.

Un autre paramètre important de la configuration est le *réglage de la taille des buffers d'entrée et de sortie audio*. Cette taille dépend de l'architecture de la station. Sur une Silicon Graphic, la taille minimum de ces buffers est de 511 échantillons. Pour une taille de 511 échantillons, nous obtenons une réponse quasi-immédiate du patch aux contrôles extérieurs. Cependant, dans ce cas nous avons observé des problèmes de puissance de calcul lors de variations rapides des paramètres du modèle. Cela peut se traduire par l'apparition de distorsions dans le signal. Pour une taille de 2044 échantillons (la valeur choisie lors de la réalisation de nos exemples sonores), la réponse est en léger décalage mais le signal est absent de distorsions.

Truc et Astuces pour la programmation d'objets externes permettant une meilleure optimisation :

Lors d'un transfert du contenu d'un pointeur dans un autre :

- mieux vaut effectuer le passage par des variables locales
- mieux vaut effectuer le transfert vers la variable locale 8 éléments par 8 éléments

Lors du transfert de variables de `dsp_put` vers `dsp_function` :

- obligation de passer les paramètres par adresses, l'appel de `dsp_function` par `dsp_put` n'étant effectué qu'à l'initialisation de la chaîne DSP (i.e. une seule fois).
- mieux vaut ne transférer qu'une seule adresse (celle d'une structure pointant sur d'autres variables)

Il est conseillé de faire l'allocation de mémoire dans la boucle `mon_objet_put` et d'utiliser un "flag" dans la boucle `mon_objet_init` (appelé uniquement à l'initialisation de l'objet) pour signaler au système (lorsque le DAC est relancé par exemple) si l'allocation a déjà été faite ou si il faut la faire. Surtout ne pas oublier de désallouer la mémoire (dans `mon_objet_delete`).

Faire attention à l'ordre de sortie des messages par les *outlets* d'un objet. Dans FTS il se fait dans l'ordre suivant : d'abord par l'*outlet* de droite puis par l'*outlet* de gauche

Les principales difficultés rencontrés lors du développement ont été les suivantes :

- L'impossibilité actuellement de réduire la taille du buffer FTS (64 échantillons pour l'instant). Ce qui a comme conséquence lors de l'implémentation de systèmes bouclés, de limiter la fréquence fondamentale à des valeurs inférieure à $\frac{f_{\text{échantillonnage}}}{64}$, et d'empêcher la variation des paramètres du modèle à intervalles inférieurs à 64 échantillons. Remarquons que tous nos objets ont été développés de manière à les rendre compatibles lors d'un changement de la taille des buffer FTS.
- L'absence actuellement de débogueur FTS, nous obligeant à déboguer les objets par essais/erreurs. Le système ne donne aucune explication lors du non-fonctionnement ou de la non-acceptation d'un nouvel objet. Cette opération de débogage se résumant la plupart du temps par un "crash" pur et simple du logiciel.
- L'impossibilité actuellement d'implémenter directement des matrices.
- L'absence actuellement d'un objet permettant de convertir le signal DSP de sortie du "patch" en fichier audio. Ceci afin de pouvoir analyser le signal ultérieurement. Les seules solutions pour parer à ce problème sont pour l'instant :
 - soit de ré-enregistrer le signal audio sortant des convertisseurs DAC
 - soit de sonder le signal dans FTS à l'aide de l'objet "probe~" et de connecter Max/FTS à Xspect par l'intermédiaire d'un démon (dans ce cas l'échantillon fournit par "probe~" est malheureusement de faible taille). Nous expliquons cette dernière opération :

lancement du démon qui va scruter les sockets de la machine

`ftsd -d 1 -f -e`

`-d 1:debug de niveau 1`

`-f:foreground -e: envoi du log`

du démon

lancement de Max-FTS

`Max -l socket -h "nom de la machine sur lequel le démon a été lancé"`

`-l socket : ouverture d'une connexion vers les sockets de la`

machine

lancement de Xspect

`Xspect`

`askFTS`

`Hostname : "nom de la machine sur lequel le démon a été lancé"`

remarque : ne surtout pas relancer Max/FTS pendant que Xspect tourne !

3.2. Présentation du modèle global

Le schéma du modèle physique simplifié d'instrument à tube est le suivant (figure 3-1.)

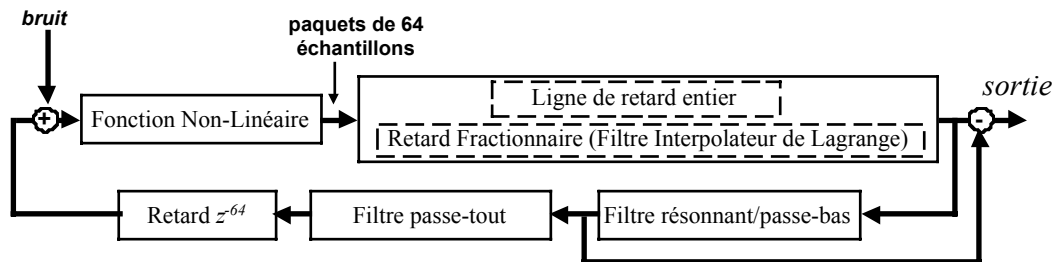


Figure 3-1. Schéma-bloc du modèle physique simplifié d'instrument à tube

Le modèle s'interprète selon l'analogie acoustique d'un instrument à vent. La vibration du système exciteur (par exemple l'anche ou l'embouchure) émet une onde progressive plane le long du tube de longueur variable (cas d'un trombone à coulisse par exemple). Cette onde se propage jusqu'au pavillon, où une partie de l'onde est réfléchi à l'intérieur du tube en sens inverse, l'autre partie est rayonnée vers le milieu extérieur. L'onde réfléchi et l'onde incidente contribue ensemble, avec le jeu de l'instrumentiste, à entretenir la vibration de l'anche par l'action d'effets fortement non-linéaires. Le tube contient bien évidemment des pertes visco-thermiques et autres.

Dans notre modèle simplifié, la ligne de retard entier et le F.I.L. constituent le tube de longueur variable (dans notre modèle le retard correspond au double de la longueur du tube, de cette manière notre retard symbolise à la fois l'aller et le retour de l'onde dans le tube). A la sortie du retard, le signal est passé à travers un filtre passe-bas (en pratique un filtre résonnant configuré en passe-bas), la différence entre le signal entrant et sortant du filtre passe-bas constitue le signal de sortie (rayonnement passe-haut). Le signal est modifié par l'action d'une fonction non-linéaire correspondant à l'effet non-linéaire de l'embouchure sur les ondes réfléchies et incidentes.

Les rôles exactes du filtre résonnant et du filtre passe-tout seront explicités dans la suite. Signalons simplement pour l'instant que le filtre passe-tout introduit dans la boucle n'est aucunement une deuxième implémentation des retards fractionnaires variables. Le pôle et le zéro de celui-ci sont gardés constants. D'un point de vue musical, la présence du filtre passe-tout se traduit par l'apparition de variations dans le contenu du spectre du son, rendant le son plus "vivant"

Dans tout système bouclé, la présence d'un retard minimal z^{-1} est nécessaire. Comme nous l'avons expliqué dans le paragraphe 3.1., FTS traite les signaux DSP sous forme vectorielle, à l'heure actuelle par paquets de 64 échantillons. Lors d'une implémentation dans FTS, la condition nécessaire de tout système bouclé se traduit alors par un retard minimal de z^{-64} (un retard z^{-1} appliqué à un paquet entier de 64 échantillons).

Une première conséquence de ce traitement vectoriel des signaux DSP par FTS est de limiter vers le haut la fréquence fondamentale du son obtenu. Une autre conséquence réside dans l'impossibilité de changer les valeurs des paramètres de notre modèle à l'intérieur d'un paquet de 64 échantillons. Les paramètres (en particulier un changement de longueur de retard) ne peuvent être changés que tout les 64 échantillons.

Nous sommes conscient de l'aspect extrêmement simplifié de notre modèle. Mais rappelons que celui-ci n'a pas la prétention d'imiter un quelconque instrument existant. Notre but dans cette réalisation est de tester en temps réel la nouvelle implémentation des interpolateurs de Lagrange. D'autres études devraient être entreprises (mesures d'impédances, de fonctions de réflexion, ...) afin de prétendre pouvoir se rapprocher du comportement d'un instrument existant. Disons simplement que le son produit par notre

modèle pourrait s'apparenter à celui produit par une clarinette basse ou un tuyau d'orgue.

Dans la suite nous détaillons le comportement des modules pris individuellement.

3.2.1. Fonction rationnelle non linéaire ("xct~")

La fonction non-linéaire que nous utilisons est la fonction rationnelle suivante (voir [17]) :

$$\gamma(x) = \frac{(x^3 + \alpha \cdot x)}{(1 + \beta \cdot x^2)}$$

où α est la pente à l'origine de $\gamma(x)$ et $\frac{1}{\beta}$ la pente de ses asymptotes à l'infini. A la figure 3-2., nous dessinons la fonction $\gamma(x)$ pour les valeurs $\alpha = -1.8$ et $\beta = 2$.

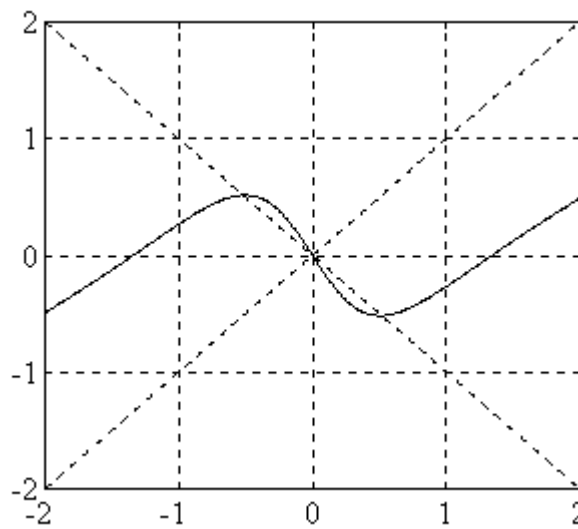


Figure 3-2. Graphe de la fonction $\gamma(x)$ pour $\alpha = -1.8$ et $\beta = 2$

Cette fonction placée dans une boucle sans filtre, a comme points de fonctionnement ceux où $\gamma(x) = -x$, i.e. $x_{1,2} = \pm \sqrt{\frac{-1-\alpha}{1+\beta}}$. Le système bouclé se comporte différemment selon la valeur de α . Par exemple pour $\beta = 1.32$:

- si $-1 \leq \alpha \leq 0$, le système n'oscille pas
- si $-4 \leq \alpha \leq -1$, nous obtenons une oscillation périodique stable
- si $-6 \leq \alpha \leq -4$, nous obtenons une oscillation périodique dont la période est double de la précédente.
- si $\alpha \leq -6.2$ environ, nous obtenons des solutions chaotiques

Discussion sur le fonctionnement de la boucle avec $\gamma(x)$:

- utilité du filtre résonnant pour l'obtention d'une solution oscillante unique

Telle quelle, la fonction $\gamma(x)$ introduite dans une boucle sans filtre permet l'obtention d'une forme d'onde rectangulaire. Dans FTS, les échantillons se déplaçant d'un objet externe à l'autre par paquets de 64, la périodicité de la forme d'onde sera de 128, soit pour une fréquence d'échantillonnage de 22050 Hz, une fréquence fondamentale de $\frac{22050}{128} = 172\text{Hz}$.

Si 64 échantillons de valeur $x = -\sqrt{-1-\alpha}/(1+\beta)$ sont introduits dans la fonction, 64 échantillons de valeurs $y = +\sqrt{-1-\alpha}/(1+\beta)$ sortiront (si la boucle n'introduit pas de perturbation). Lors du passage suivant dans la fonction, ils passeront de la valeur $x = +\sqrt{-1-\alpha}/(1+\beta)$ à $y = -\sqrt{-1-\alpha}/(1+\beta)$. Dans ce cas, la fonction produit donc une forme d'onde rigoureusement rectangulaire-symétrique de période égale à 128 échantillons.

Mais cette fonction non-linéaire admet aussi un grand nombre d'autres solutions, en fait toutes celles dont la première partie est composée avec les valeurs $x_{1,2}$ et dont la deuxième partie se déduit par symétrie.

Afin de limiter le nombre de solutions du système bouclé, nous introduisons dans la boucle un filtre résonnant fonctionnant en passe-bas (voir aussi § 3.2.4.). Les échantillons qui étaient jusqu'à présent décorrélés entre eux, sont corrélés ce qui réduit le nombre de solutions possibles du système. Vu d'une autre manière, l'introduction d'un filtre passe-bas limite les solutions du système à celles dont le contenu harmonique se situe en basse fréquence, ce qui est le cas des instruments de musique (le contenu fréquentiel d'un instrument décroît souvent en $1/f$ ou $1/f^2$). Le filtre passe-bas peut aussi être interprété comme modélisant le fait que dans un tube les modes basse fréquence se propagent mieux (pertes visco-thermiques, ...). En rajoutant un filtre passe-bas dans la boucle, les échantillons ne peuvent plus varier instantanément d'un point de fonctionnement à l'autre, mais sont contraints de passer par les valeurs intermédiaires de la fonction. Dès lors, la forme d'onde rectangulaire s'arrondit, ce qui est une autre manière d'exprimer l'évanescence des hautes fréquences.

- rajout d'une composante continue d'amplitude quasi-négligeable

Afin d'éviter que le système ne se fixe en zéro (même s'il s'agit d'un point fixe instable), nous rajoutons au signal entrant dans la fonction γ une partie aléatoire d'amplitude quasi-négligeable. Une autre possibilité est de rajouter au signal entrant une composante continue d'amplitude négligeable telle $1e-6$.

- fonction γ dissymétrique

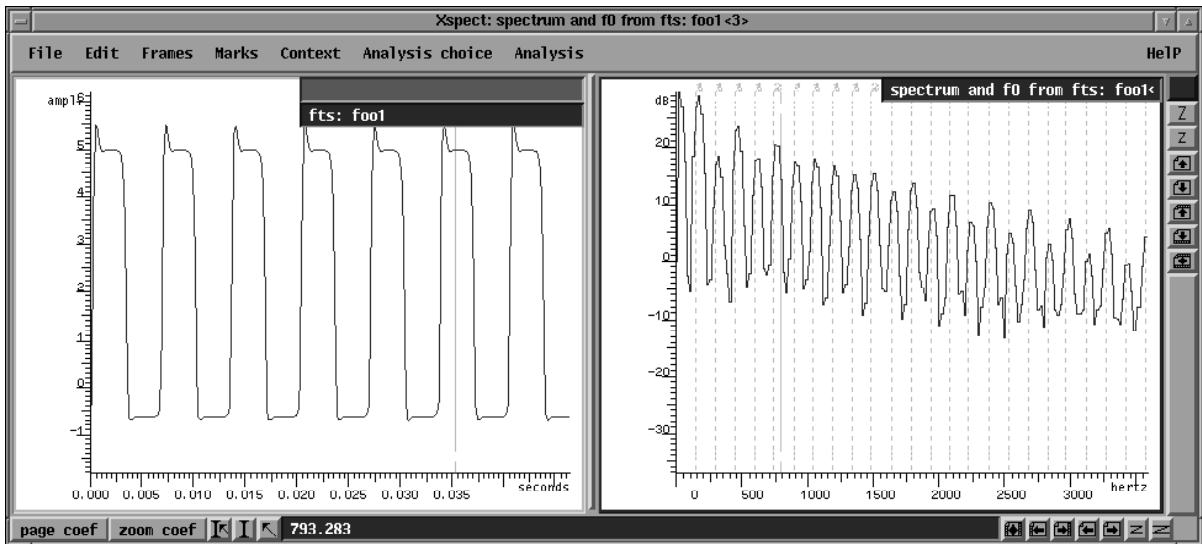
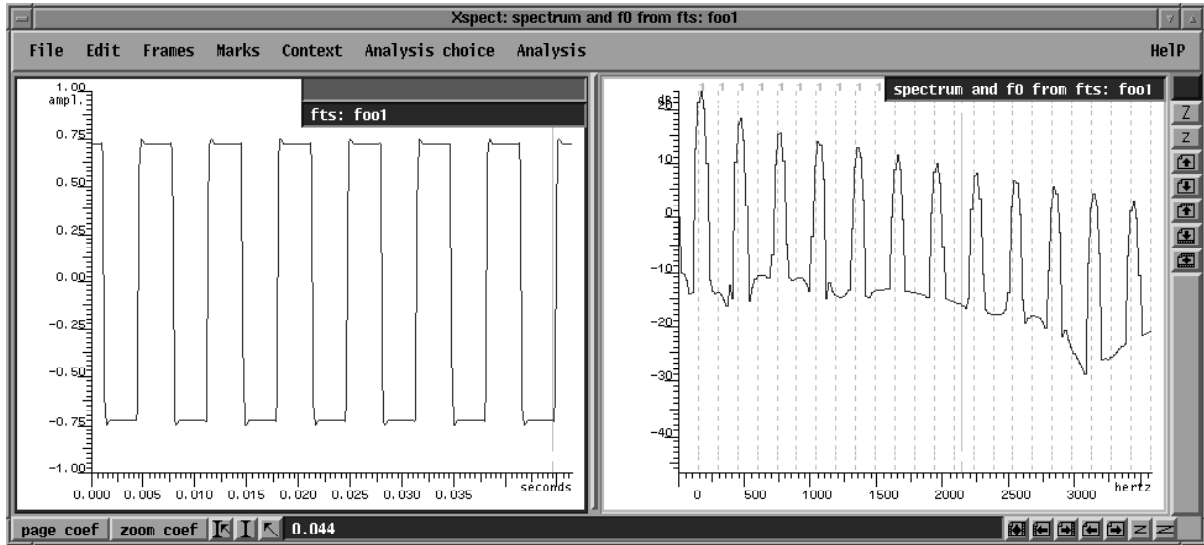
L'introduction d'une fonction γ impaire dans un tel système bouclé permet l'obtention d'un signal (plus ou moins arrondi selon le filtrage appliqué) composé d'harmoniques impaires seulement. Afin de pouvoir engendrer des harmoniques paires également, nous introduisons une dissymétrie dans la fonction γ . Notant le coefficient de dissymétrie σ ($0 < \sigma \leq 1$), la transformation que nous appliquons s'exprime par :

$$\forall x < 0, \gamma\left(\frac{x}{\sigma}\right)$$

$$\forall x > 0, \gamma(x \cdot \sigma)$$

Notre transformation a comme conséquence l'apparition d'harmoniques paires si la boucle comporte un filtrage.

Nous montrons aux figures 3-3. et 3-4. l'application de cette dissymétrie de γ dans notre modèle. Le signal a été mesuré par une sonde FTS (objet "**probe~**") et connecté au logiciel d'analyse de signaux Xspect par l'intermédiaire d'un démon. Les mesures ont été faites pour le cas où $\sigma = 1$ (figure 3-3) et $\sigma = 0.4$ (figure 3-4).



Figures 3-3. et 3-4. Signal et son contenu fréquentiel mesuré dans le cas d'une fonction $\gamma(x)$ symétrique ($\sigma = 1$) et dissymétrique ($\sigma = 0.4$).

Pour implémenter les fonctionnalités dont nous avons parlé précédemment, nous avons créé l'objet externe "xct~" pour la librairie Max/FTS. Celui-ci outre l'implémentation de $\gamma(x)$, renferme une série de fonctionnalités permettant l'utilisation de cette fonction dans un contexte temps réel. Les principales fonctionnalités sont les suivantes :

- Le contrôle du paramètre α (la pente de $\gamma(x)$ à l'origine) via l'interface Midi. Le contrôle s'effectue par la vitesse de frappe au clavier ("Midi velocity"). Le paramètre α prend la valeur -3.5 pour une vitesse de frappe maximale (ce qui permet au système d'atteindre son point de fonctionnement rapidement ou dit autrement permet une attaque nette du son) et -0.95 pour une vitesse de frappe nulle ou un message de fin de note (la pente à l'origine de $\gamma(x)$ étant inférieur à 1 le système n'est plus entretenu, i.e. le son s'éteint).
- L'incrémentant d'un compteur d'événements pour les événements Midi de début de note et sa décrementant pour ceux de fin de note. Ceci permet de bloquer le paramètre α , et donc de lier deux sons lorsque deux notes sont jouées de manière liée (glissando) sur un clavier.

L'objet "xct~" permet également le contrôle du paramètre β par la pression de toucher ("Midi aftertouch"), le contrôle du paramètre de dissymétrie ν par message de contrôle Midi ("Midi

message control"), de même que l'enclenchement d'une fonction de blocage du paramètre α (permettant un jeu en legato) par message de contrôle Midi.

3.2.2. Ligne de retard entier / Retard Fractionnaire (Filtre Interpolateur de Lagrange) ("fd~")

Dans le paragraphe 2.2.2. nous avons présenté une nouvelle implémentation des filtres interpolateurs de Lagrange pour l'approximation des retards fractionnaires sous forme de structure modulaire. Nous avons montré que l'adjonction à ce filtre d'une ligne de retard entier est nécessaire afin de pouvoir limiter la variation de l'ordre du filtre, et nous avons explicité une procédure de remise à jour des états du système (par calcul des entrées antérieures du système), applicable lors du transfert d'une partie du retard du F.I.L. vers la ligne de retard entier ou l'inverse.

L'objet externe "fd~" que nous avons réalisé permet de faire l'ensemble de ces opérations. Nous donnons son diagramme de fonctionnement à la figure 3-5..

Rappelons qu'il existe une autre implémentation des retards fractionnaires dans la librairie Max/FTS. Comme expliqué précédemment, l'objet "vd~" écrit par Miller-Puckette renferme une ligne de retard entier et un filtre R.I.F. sous forme de structure de Farrow d'ordre 4.

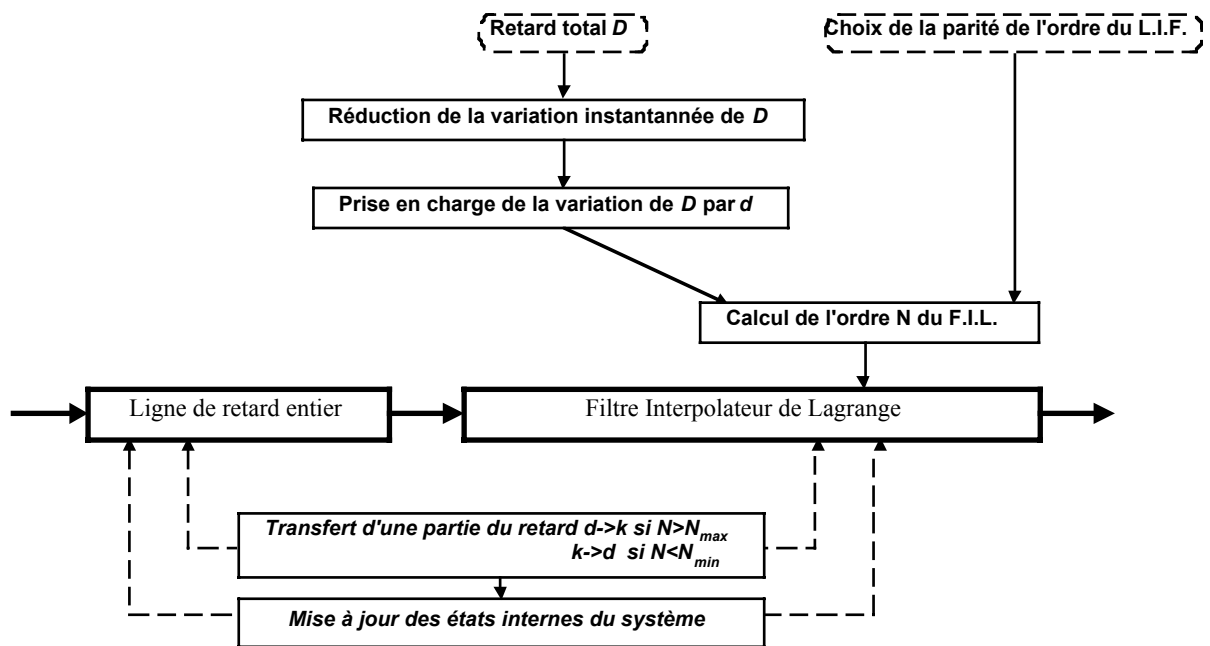


Figure 3-5. Diagramme de fonctionnement de l'objet externe "fd~"

Son fonctionnement est le suivant :

L'objet prend en entrée une valeur de retard exprimée en *ms* (voir objet externe "miditofreq") et la convertit en nombre d'échantillons de retard (fonction de la fréquence d'échantillonnage du système). Appelons cette valeur de retard convertie *D*.

Réduction de la variation instantannée de *D* : $\Delta D \leq \Delta D_{max}$

Comme nous l'avons signalé au paragraphe 2.2.2., une variation de retard supérieure à un échantillon par passage ce fait au détriment d'une perte de sens physique. Notre implémentation sous forme modulaire est d'ailleurs avant tout destinée aux variations continues du retard. Nous devrions donc théoriquement limiter la variation de *D* à une valeur inférieure à un échantillon par passage, et étaler les variations plus grandes sur plusieurs passages.

D'un autre côté, nous voudrions permettre le "jeu" du modèle à partir d'un clavier Midi. Ce qui implique de pouvoir produire des variations instantanées de D importantes. Nous ne pouvons donc nous limiter à varier D de seulement un échantillon par passage (dans FTS, un "passage" est effectué chaque fois qu'un nouveau paquet de 64 échantillons doit être traité). Car le fait de varier D de seulement un échantillon tous les 64 échantillons produirait des "glissandi" trop longs lorsque D doit passer d'une valeur à une autre éloignée.

De ce fait, dans l'objet "fd~" nous permettons des variations instantanées du retard D de $\Delta D_{\max} = 2$.

Ceci est un choix, puisque nous pourrions également décider de contourner le problème en étalant les variations de D supérieures à un échantillon, non pas sur les passages suivants de paquets de 64 échantillons, mais à l'intérieur même d'un paquet de 64 échantillons. Mais dans ce dernier cas, nous devrions traiter les échantillons un par un (à cause d'éventuelles variations de l'ordre du filtre d'un échantillon à l'autre, ou pire encore, à cause d'un besoin de transfert de retard et de mise à jour de l'état du système), ce qui empêcherait toutes implémentations vectorielles efficaces des algorithmes. Dans un contexte de temps réel, nous privilégions donc les implémentations vectorielles ceci afin d'éviter des temps de calcul prohibitifs. D'où notre choix $\Delta D_{\max} = 2$.

Prise en charge de la variation de D par d : $\Delta d = \Delta D \rightarrow d = d + \Delta d$

Comme vu précédemment (§ 2.2.2.), il est avantageux de privilégier les variations de d (la partie du retard implémentée par le F.I.L.) plutôt que de répartir en permanence le retard entre d et κ (la longueur de la ligne de retard entier) et de devoir en permanence remettre à jour l'état du système. Dés lors, dans notre implémentation, la variation de D sera prise en charge prioritairement par d , le retard implémenté sous forme de F.I.L..

Transfert d'une partie du retard $d \rightarrow k$ si $N > N_{\max}$ $k \rightarrow d$ si $N < N_{\min}$

Toutefois si l'ordre du filtre devient trop important (>13 pour les filtres impairs, >14 pour les filtres pairs), une partie du retard sera transférée vers la ligne de retard entier. Cette procédure une fois entamée sera poursuivie lors des appels suivants de manière à ramener l'ordre du filtre à une valeur centrale calculée de manière à permettre aussi facilement une augmentation de l'ordre qu'une diminution (valeur centrale : 9 pour les filtres impairs, 10 pour les filtres pairs). Lorsque l'ordre du filtre devient trop petit (<5 pour les filtres impairs, <6 pour les filtres pairs) la procédure inverse se produit, et le transfert d'une partie du retard vers le filtre se poursuivra aussi jusqu'à ce que l'ordre du filtre soit ramené à sa valeur centrale.

La procédure de remise à jour de l'état du système par recombinaison des entrées passées a été explicitée dans le paragraphe 2.2.2.2.. Signalons simplement que dans notre implémentation les transferts de retard s'effectuent 2 échantillons par 2 échantillons, afin d'accélérer les transferts. La remise à jour du système nécessite donc, dans un cas la reconstruction de 2 échantillons ($d \rightarrow k$), dans l'autre la construction de 4 modules supplémentaires du F.I.L. ($k \rightarrow d$).

Calcul de l'ordre N du F.I.L.

L'ordre du filtre interpolateur de Lagrange est alors calculé : $N = \lfloor z^\alpha \rfloor$ ou $\lceil z^\alpha \rceil$, en tenant compte du choix de la parité de l'ordre souhaitée.

Ligne de retard entier

La ligne de retard entier est implémentée sous forme d'un buffer circulaire de taille 512. FTS travaillant par paquets de 64 échantillons, à chaque passage le pointeur d'écriture se déplace de 64 échantillons vers la gauche et écrit les 64 nouveaux échantillons vers la droite. Le pointeur de lecture (qui est alors à $k_{\text{précédent}}$ échantillons à droite du pointeur d'écriture) se déplace lui de $64 - (k_{\text{nouveau}} - k_{\text{précédent}})$ vers la gauche et lit les 64 échantillons vers la droite. La sortie de la ligne de retard entier est raccordée à l'entrée du filtre interpolateur de Lagrange (figure 3-6.).

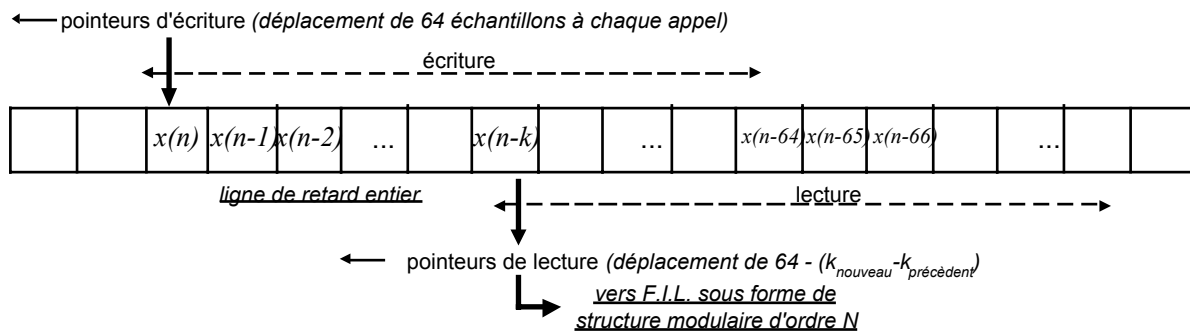


Figure 3-6. Fonctionnement de la ligne de retard entier

Retard fractionnaire (Filtre Interpolateur de Lagrange)

Le Filtre Interpolateur de Lagrange est implémenté sous forme de structure modulaire d'ordre N et de coefficients d (voir § 2.2.2.).

Pour le temps réel, l'algorithme de calcul du filtre est implémenté sous forme vectorielle.

Remarquons que du fait du traitement vectoriel, seuls les états internes $(X[n])$ pour $1 \leq n \leq N$ pour un F.I.L. d'ordre N correspondants au traitement du dernier échantillon du paquet ($i=64$) doivent être stockés en mémoire. Ceux-ci serviront pour le traitement du premier échantillon du paquet suivant ($i=1$).

L'algorithme vectoriel pour le calcul du F.I.L. d'ordre N peut alors s'écrire de la manière suivante :

où n : indice du $n^{\text{ème}}$ étage du filtre d'ordre N

i : indice correspondant au traitement du $i^{\text{ème}}$ échantillon du paquet de taille 64

$sortie[i]$: $i^{\text{ème}}$ échantillon du paquet de taille 64 sorti de la ligne de retard entier auquel s'additionne les valeurs obtenues par les différents étages du F.I.L..

1) Initialisation

$$n \leftarrow 1$$

Boucle à parcourir pour $i = 1$ à 64

$$b[i] \leftarrow sortie[i]$$

2) Boucle à parcourir tant que $n \leq N$

$$e[1] \leftarrow \frac{d}{n} (X[n] - b[1])$$

$$sortie[i] \leftarrow sortie[i] + e[1]$$

Boucle à parcourir pour $i = 2$ à 64

$$e[i] \leftarrow \frac{d}{n} (b[i-1] - b[i])$$

$$sortie[i] \leftarrow sortie[i] + e[i]$$

$$X[n] \leftarrow b[i]$$

Boucle à parcourir pour $i = 1$ à (taille buffer FTS)

$$b[i] \leftarrow e[i] \dots$$

$$d \leftarrow d - 1$$

$$n \leftarrow n + 1$$

3) Dernière boucle

$$e[64 - 2] \leftarrow \frac{d}{n}(b[64 - 3] - b[64 - 2])$$

$$e[64 - 1] \leftarrow \frac{d}{n}(b[64 - 2] - b[64 - 1])$$

$$e[64] \leftarrow \frac{d}{n}(b[64 - 1] - b[64])$$

$$X[n] \leftarrow b[64]$$

$$b[64 - 2] \leftarrow e[64 - 2]$$

$$b[64 - 1] \leftarrow e[64 - 1]$$

$$b[64] \leftarrow e[64]$$

$$d \leftarrow d - 1$$

$$n \leftarrow n + 1$$

$$e[64 - 1] \leftarrow \frac{d}{n}(b[64 - 2] - b[64 - 1])$$

$$e[64] \leftarrow \frac{d}{n}(b[64 - 1] - b[64])$$

$$X[n] \leftarrow b[64]$$

$$b[64 - 1] \leftarrow e[64 - 1]$$

$$b[64] \leftarrow e[64]$$

$$d \leftarrow d - 1$$

$$n \leftarrow n + 1$$

$$e[64] \leftarrow \frac{d}{n}(b[64 - 1] - b[64])$$

$$X[n] \leftarrow b[64]$$

$$b[64] \leftarrow e[64]$$

$$n \leftarrow n + 1$$

$$X[n] \leftarrow b[64]$$

Dans la dernière boucle de l'algorithme nous anticipons les 4 états internes du F.I.L.. d'ordre supérieurs $X[N+1]$, $X[N+2]$, $X[N+3]$, $X[N+4]$, de façon à pouvoir si le besoin est, augmenter l'ordre du filtre de 4 modules l'ordre du passage suivant ($\Delta d_{\max} = \Delta D_{\max} = 2$), et de façon aussi à pouvoir effectuer les transferts entre ligne de retard entier et F.I.L. (impliquant une remise à jour de l'état du système) par valeur de 2 échantillons (1 échantillon dans le § 2.2.2.).

Les échantillons de sortie du filtre ($sortie[i]$ pour $1 \leq i \leq 64$) sont envoyés dans la suite du modèle.

3.2.3. Contrôle de la longueur du retard via l'interface Midi ("miditofreq")

Ce petit objet tout simple permet le contrôle de l'objet "fd~" via un clavier Midi. Il prend en entrée le numéro de la note Midi jouée ainsi que la valeur du Pitch-Bend, et donne en sortie la valeur de la fréquence (en Hz) et la longueur de retard (en ms) correspondantes.

L'objet permet également de choisir :

- la correspondance entre le déplacement du Pitch-Bend et les valeurs en demi-tons,
- la transposition du clavier (par demi-tons).

Les valeurs en sortie sont obtenues à l'aide des équations suivantes (69 est le numéro de la note Midi correspondant au "la" 440 Hz et 64 est la valeur du Pitch-Bend à l'état de repos) :

$$fréquence(Hz) = 440.0 \cdot 2^{\frac{entrée midi du clavier - (69 - transposition du clavier)}{12} + \frac{entrée midi du pitch-bend - 64}{64 \cdot 12 / course du pitch-bend}}$$

$$retard (ms) = \frac{taille buffer FTS + retards de phase du filtre deux-pôles / un-zéro + retards de phase du filtre passe-tout}{fréquence + \frac{fréquence \cdot échantillonnage FTS}{}}$$

3.2.4. Filtres résonnant et filtre passe-tout ("2p2z~")

L'objet "2p2z~" de la librairie FTS implémente un filtre deux-pôles/deux-zéros. Nous nous en servons pour réaliser le filtre résonnant et le filtre passe-tout d'ordre 1 de notre modèle. La fonction de transfert du filtre réalisé par l'objet "2p2z~" est la suivante :

$$H(z) = c_0 \frac{d_0 + d_1 z^{-1} + d_2 z^{-2}}{1 - c_1 z^{-1} - c_2 z^{-2}}$$

Filtre résonnant

Comme signalé en 3.2.1., l'insertion d'un filtre résonnant configuré en passe-bas dans la boucle est nécessaire afin de forcer le système vers la solution présentant un contenu essentiellement en basse-fréquence.

Le filtre résonnant que nous avons utilisé possède les caractéristiques suivantes :

- fréquence de résonance : 750 Hz
- bande passants à -3dB : 1500 Hz
- fréquence de coupure : 11025 Hz

A la figure 3-7., nous en donnons la réponse en amplitude et en phase.

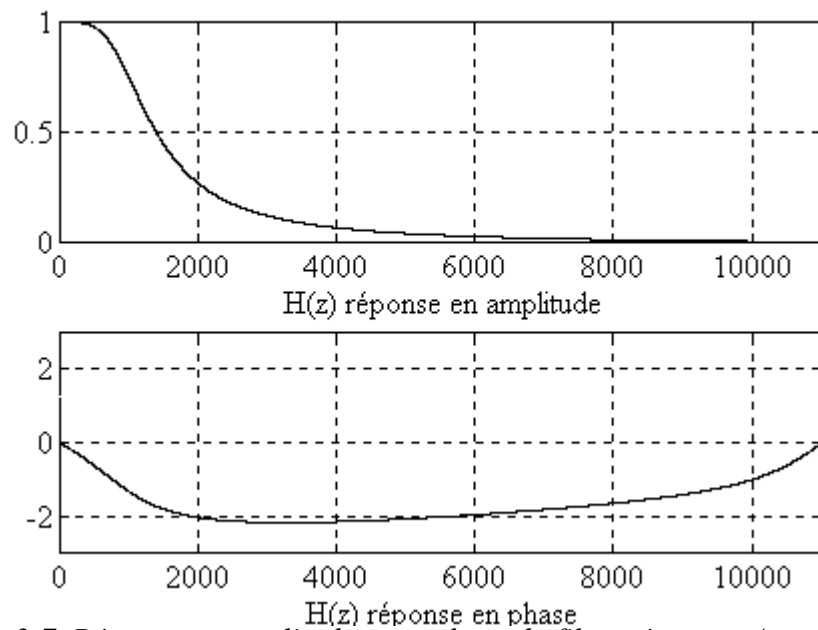


Figure 3-7. Réponse en amplitude et en phase du filtre résonnant/passe-bas de caractéristiques $f_r = 750 \text{ Hz}$ $BW=1500 \text{ Hz}$, $f_c=11025 \text{ Hz}$

Une étude plus approfondie des caractéristiques fréquentielles d'un instrument (une clarinette basse par exemple) devrait permettre de définir les paramètres idéaux de ce filtre d'une manière plus précise.

Le filtre résonnant est réalisé par un filtre deux-pôles/un-zéro :

$$H(z) = \frac{c + dz^{-1}}{1 + az^{-1} + bz^{-2}}$$

Ses coefficients s'obtiennent à partir des caractéristiques voulues par :

$$r = e^{-\frac{BW}{Fe}}$$

$$a = -2r \cos(2\pi \frac{f_r}{Fe})$$

$$b = r^2$$

$$c = \sin(\varphi)$$

$$d = r \sin(2\pi \frac{f_r}{Fe} - \varphi)$$

où BW est la bande passante à -3db et $\varphi = \frac{\pi}{2}$ est la phase initiale

Nous normalisons le filtre de manière à le rendre passif. Le filtre s'écrit alors :

$$H(z) = 0.0413 \frac{1 + 0.7892 z^{-1}}{1 - 1.5784 z^{-1} + 0.6522 z^{-2}}$$

Filtre passe-tout

Comme signalé dans l'introduction du paragraphe 3.2., le filtre passe-tout que nous insérons dans la boucle n'est nullement une autre implémentation des retards fractionnaires variables. Il s'agit d'un filtre passe-tout d'ordre 1 et de pôle constant a . D'ailleurs, lorsqu'on utilise un filtre passe-tout pour la réalisation de retards fractionnaires, il est utilisé pour ses propriétés de linéarité de phase (donc de retard de phase constant). Ce qui se passe lorsque le pôle est proche de zéro (pour un filtre d'ordre 1). Le passe-tout que nous utilisons a un pôle proche de un et possède une réponse en phase non-linéaire (donc un retard en phase non-constant) (voir figures 3-8. et 3-9.)

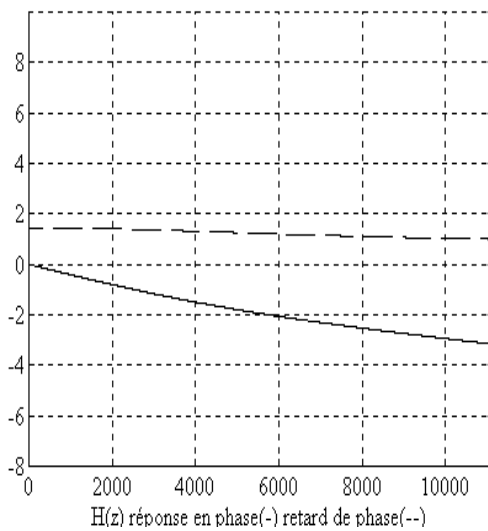


Figure 3-8. Réponse en phase (-) et retard de phase (--) d'un filtre passe-tout d'ordre 1 de pôle $a=0.18$

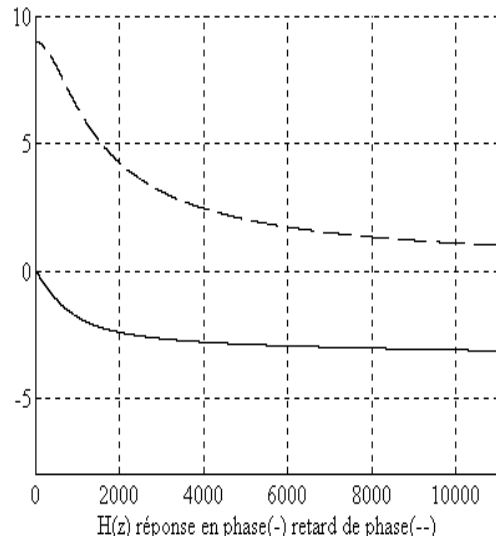


Figure 3-9. Réponse en phase (-) et retard de phase (--) d'un filtre passe-tout d'ordre 1 de pôle $a=0.8$

C'est cette dernière propriété de retard de phase non-constant (figure 3-9) qui nous intéresse. Elle nous permet de retarder les différentes composantes fréquentielles du signal d'un retard variant en fréquence $\tau(\omega)$. On peut noter que cette situation est aussi le cas d'instruments réels (exemple de la fonction de réflexion du saxophone).

Les répercussions sur le spectre du signal de l'introduction d'un filtre passe-tout avec un pôle proche de un sont difficiles à expliquer dans un contexte de boucle renfermant également une fonction non-linéaire. Ceci n'est d'ailleurs pas le propos de ce travail. Nous pouvons simplement dire qu'expérimentalement nous avons constaté une modification du spectre telle que les partiels principaux restent. Il se produit également un phénomène de modulation en amplitude des partiels à une fréquence inférieur, la raison pour laquelle sans doute le son paraît plus "vivant".

La fonction de transfert du filtre passe-tout d'ordre 1 que nous utilisons est la suivante :

$$H(z) = -a \frac{1 - \frac{1}{a}z^{-1}}{1 - az^{-1}} \text{ où } |a| < 1$$

Dans notre modèle, le pôle peut être réglé entre $]0,1[$. Dans les exemples sonores que nous montrerons lors de la soutenance de stage, nous avons utilisé les valeurs suivantes de pôles et de zéro :

$$\begin{aligned} \text{pôle en } a = 0.18 \text{ et donc zéro en } \frac{1}{a} = 5.55 \\ \text{pôle en } a = 0.8 \text{ et donc zéro en } \frac{1}{a} = 1.25 \end{aligned}$$

3.2.5. Schéma global du modèle physique d'instrument à tube et du contrôle de ses paramètres via Midi

Pour terminer nous donnons le schéma global de notre modèle physique simplifié d'instrument à tube et les contrôles de ses paramètres via Midi (figure 3-10) correspondant à notre patch "trombinette.pat" ("clarinette" + "trombone à coulisse" = trombinette) sur Max (figure 3-11).

Nous aurions aimé pouvoir analyser la forme de l'onde lors d'une variation de la longueur de retard (comme nous l'avons fait lors de la simulation sur Matlab). Mais le fait que les paramètres du système ne peuvent être changés que tous les 64 échantillons, nous forcerait à analyser le signal sur une longue période. Hors comme expliqué en 3.1., actuellement il n'existe pas d'objet sur FTS permettant de convertir le signal DSP en fichier audio. Seule une sonde nous permettrait de réaliser une analyse, mais la longueur du signal obtenue serait alors trop petite.

Les exemples sonores présentés lors de la soutenance de ce travail serviront donc à valider l'implémentation.

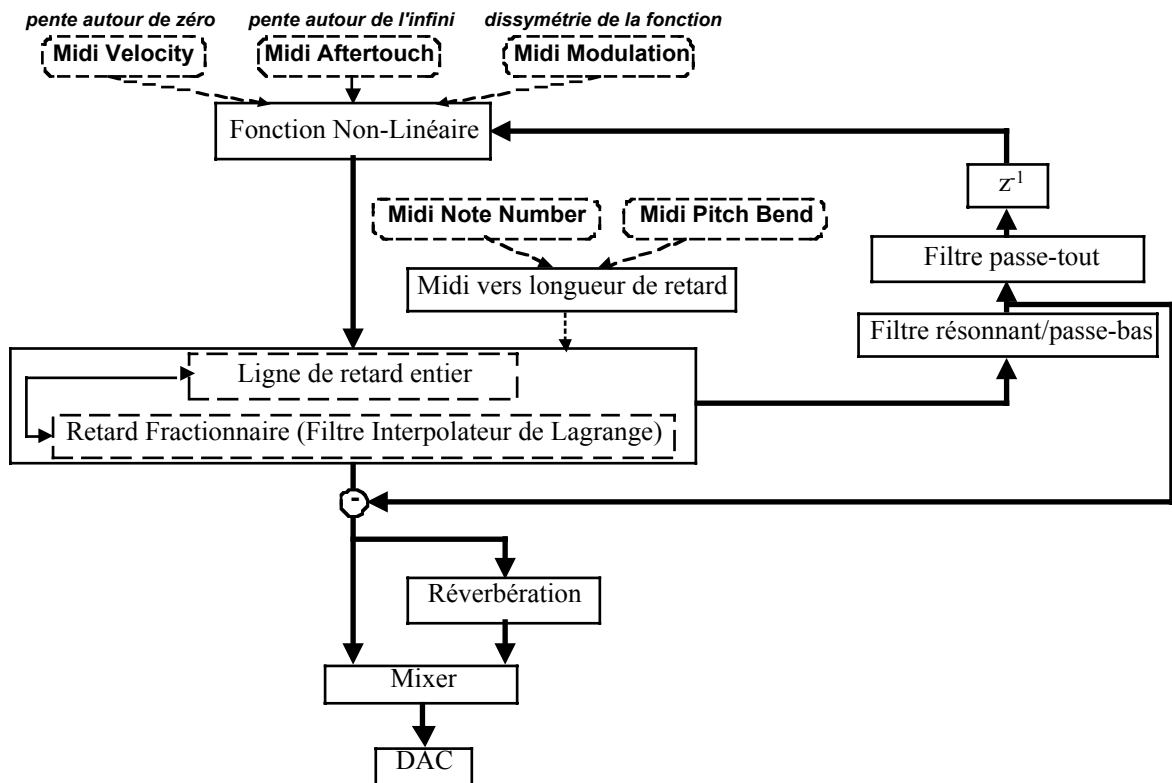


Figure 3-10. Schéma global du modèle physique simplifié d'instrument à tube et du contrôle de ses paramètres via Midi tel que réalisé sur le logiciel Max/FTS

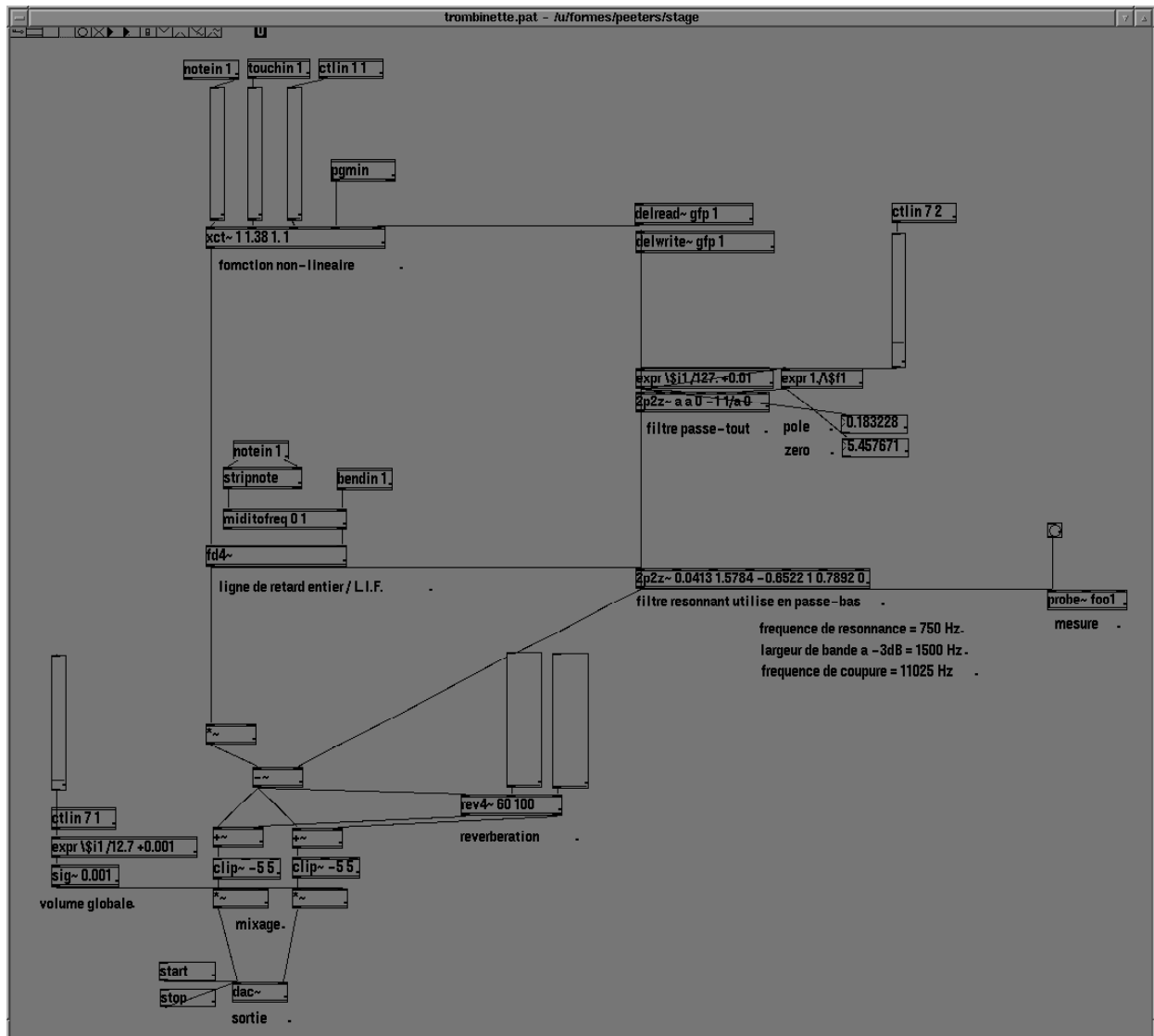


Figure 3-11. Notre patch "trombinette.pat" de Max correspondant au schéma de la figure 3-10

CONCLUSION

Ce travail avait pour but l'étude de différentes stratégies d'implémentation des filtres interpolateurs de Lagrange pour l'approximation réel des retards fractionnaires variables en temps.

Nous avons étudié l'implémentation des filtres interpolateurs de Lagrange sous formes de structure de Farrow modifiée et sous forme d'une nouvelle structure modulaire. Cette dernière est innovante sur de nombreux points : factorisation et vectorisation des calculs, très grande plage de variation de l'ordre du retard fractionnaire, évolution continue de la structure et de l'ordre du filtre, contrôle paramétrique de l'ordre du filtre.

La comparaison des deux structures à été faite tant du point de vue du coût de calcul que du point de vue du comportement du système dans un contexte de retards fractionnaires variables. La nouvelle structure modulaire nécessite un coup de calcul inférieur à la structure de Farrow modifiée mais nécessite une procédure de remise à jour de l'état du système dont le coût de calcul peut s'avérer lourd dans un contexte temps réel. Nous avons également montré que, lors de changement de retard, l'évolution de leur réponse en amplitude est équivalente.

Cette nouvelle implémentation des F.I.L. sous forme de structure modulaire a été implémentée sur le logiciel temps réel de l'IRCAM Max/FTS. et a été utilisée dans la réalisation d'un modèle physique simplifié d'instrument à tube. Pour cela, de nouveaux objets externes ont été créés. Ce modèle, contrôlable à partir d'un clavier Midi, a permis la réalisation d'exemples sonores montrant l'utilisabilité de la nouvelle implémentation dans un contexte temps réel. Les exemples sonores seront montrés lors de la soutenance de ce travail.

BIBLIOGRAPHIE

- [1] Cain, G. D., and Yardim, A., "The tunable fractionnal delay filter : passport to fine grain delay estimation", *Electrotechnical Review*, Slovenia, 1994
- [2] Crochiere, R. E., and Rabiner, K. R., "Multirate Digital Signal Processing", Prentice-Hall, Englewood Cliffs, NJ, 1983
- [3] Déchelle, F., De Cecco, M., "FTS, The Ircam Real-Time Platform", I.C.M.C. '95, Canada, 1995
- [4] Depalle, Ph., Tassart, S., "Fractional Delay Lines using Lagrange Interpolators", I.C.M.C. '96, Honk-Kong, 1996
- [5] Farrow, C. W., "A continuously variable digital delay element", *Proc. 1988 IEEE Int. Symp. on Circuit and Systems (ISCAS '88)*, vol. 3, 1988
- [6] Forsythe, G. E., Malcom, M. A., and Moler, C. B., "Computer Methods for Mathematical Computations", Prentice-Hall, Englewood Cliffs, NJ, 1977
- [7] Hermanowicz, E., "Explicit fomulas for weighting coefficients of maximally flat tunable F.I.R. delayers", *Electronic Letters*, vol. 28, 1992
- [8] Hildebrand, F. B., "Introduction to Numerical Analysis (Second Edition)", McGraw-Hill, New-York, 1974
- [9] Jaffe, and Smith, J.O., "Extensions of the Karplus-Strong plucked string algorithm", *Computer Music Journal*, vol. 7, Cambridge, MIT Press, 1983
- [10] Kootsookos, P. J., and Williamson, R. C., "FIR approximation of fractionnal sample delay systems", *IEEE Trans. Circuit and Systems*, pas encore publié
- [11] Laakso, T. I., Välimäki, V., Karjalainen, M., and Laine, U. K., "Splitting the Unit Delay", *IEEE Signal Processing magazine*, vol. 13, n°1, 1996
- [12] Lang, M., and Laakso, T. I., "Simple and robust method for the design of allpass filters using least-square phase error criterion", *IEEE Trans. Circuits and Systems*, vol. 41, n°1, 1994
- [13] Mourjopoulos, J. N., Kyriakis-Bitzaros, E. D., and Goutis, C. E., "Theory and real-time implementation of time-varying digital audio filters", *Journal of Audio Engineering Society*, vol. 38, n°7/8, 1990
- [14] Oetken, G. , Parks, T. W., and Schübler, H. W., "New results in the design of digital interpolators", *IEEE Trans. Acous. Speech, Signal Processing*, vol. 23, n°3, 1975
- [15] Parks, T. W., and Burrus, C. S., "Didital Filter Design", John Wiley and Sons, New-York, 1987
- [16] Ranbenstein, R., "Minimization of transient signal in recursive time-varying digital filters", *Circuits, Systems and Signal Processing*, vol. 7, n°3, 1988
- [17] Rodet, X., "Stability/Instability of Periodic Solutions and Chaos in Physical Models of Musical Instruments", *ICMC 1994*, Aarhus, Denmark, 1994
- [18] Sivanand, S., "Variable-delay implementation using digital FIR filters", *Int. Conf. Signal Processing Theory and Applications (ISPAT '92)*, Boston, 1992
- [19] Smith, J. O., "Bandlimited Interpolation - Introduction and algorithm", non-publié, 1994
- [20] Tassart, S., "Frequential Study of the Lagrange Interpolator", *Rapport Interne de l'IRCAM*, 1995
- [21] Tassart, S., "Les retards fracionnaires", *Rapport Interne de l'IRCAM*, 1996

- [22] Tassart, S., Depalle, Ph., "Fractionnal Delays using Lagrange Interpolator", NAM '96, Helsinki, 1996
- [23] Thiran, J.-P., "Recursive digital filters with maximally flat group delay", IEEE Trans. Circuit Theory, vol. 8, n°6, 1971
- [24] Välimäki, V., "Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters", Dissertation for the degree of Doctor, Helsinki, Finland, 1995
- [25] Verhelst, W., Nilens, P., "A modified superposition speech synthesizer and its applications", Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP '86), Tokyo, 1986
- [26] Zetterberg, L. H., and Zhang, Q., "Elimination of transients in adaptative filters with application to speech coding", Signal Processing, vol. 15, n°4, 1988

* * *