

# Modèle de phase d'attaque basé sur la répétition d'évènements audio

## Sommaire

---

<b>6.1</b>	<b>Modèle de phase d'évènements audio répétés</b>	<b>94</b>
<b>6.2</b>	<b>Validation expérimentale préliminaire</b>	<b>96</b>
6.2.1	Précision du modèle de phase	96
6.2.2	Combinaison avec le déroulé linéaire	97
<b>6.3</b>	<b>Modèle de mélange de sources</b>	<b>98</b>
<b>6.4</b>	<b>Estimation des phases des composantes</b>	<b>99</b>
6.4.1	Contrainte stricte	99
6.4.2	Contrainte relaxée	102
<b>6.5</b>	<b>Résultats expérimentaux</b>	<b>105</b>
6.5.1	Influence du paramètre $\sigma$	105
6.5.2	Performance du modèle de phase	106
6.5.3	Modèle de phase d'attaque et déroulé linéaire	107
6.5.4	Prise en compte de la phase du mélange pour le déroulé	108
<b>6.6</b>	<b>Conclusion</b>	<b>109</b>

---

Comme on l’a indiqué dans l’introduction de ce manuscrit, la majorité des techniques de séparation de sources reposent sur le caractère répétitif des événements qui constituent les données. En audio, on peut voir une source comme étant constituée de la répétition d’un événement élémentaire (comme une note). La NMF repose sur le postulat de l’existence d’un spectre associé à chaque événement audio, qui est activé à différents instants avec un gain variable. Néanmoins, les répétitions de phase ne sont guère exploitées. Or, comme c’est l’évènement audio (un signal temporel) qui est redondant, on peut intuitivement penser qu’il existe une forme de redondance qui se retrouve dans sa TFCT, aussi bien au niveau du spectrogramme d’amplitude que de la phase.

Nous proposons donc d’exploiter ces redondances au niveau des attaques des sources musicales pour reconstruire les phases dans les trames d’attaque dans le domaine TF. On postule l’existence d’une *phase de référence* et on suppose que pour une source donnée, la phase au niveau des trames d’attaque est égale à cette phase de référence, à laquelle est ajouté un décalage qui est une fonction linéaire de la fréquence. Ce modèle est testé sur divers signaux afin d’en attester la validité. Nous proposons également de le combiner à l’algorithme de déroulé linéaire afin de restaurer complètement la phase d’un signal, dans un cadre supervisé (les phases de référence sont préalablement apprises à partir d’une base de données externe).

Nous proposons ensuite d’intégrer cette propriété dans un modèle de mélange de sources complexes au niveau des attaques. Ce modèle de mélange est estimé par deux algorithmes, la contrainte de phase pouvant être stricte ou relaxée. Une fois la restauration des phases complétée par l’application de l’algorithme de déroulé linéaire, nous obtenons alors une procédure complète de reconstruction de phase dans le cadre de la séparation de sources.

Les contributions de ce chapitre ont fait l’objet d’une publication à la conférence WASPAA 2015 [MAGRON et al. \(2015a\)](#).

La section 6.1 présente ce modèle de phase, qui est expérimentalement validé dans la section 6.2. Cette propriété est intégrée à un modèle de mélange dans la section 6.3, dont l’estimation des paramètres est décrite dans la section 6.4. Des expériences de séparation de sources sont conduites dans la section 6.5, et nous concluons dans la section 6.6.

## 6.1 Modèle de phase d’évènements audio répétés

Considérons un signal réel  $x(n)$ . On rappelle l’expression de sa TFCT, pour chaque canal fréquentiel  $f \in \llbracket 0, F - 1 \rrbracket$  et trame  $t \in \llbracket 0, T - 1 \rrbracket$  :

$$X(f, t) = \sum_{n=0}^{N_w-1} x(n + tS)w(n)e^{-2i\pi \frac{f}{F}n}, \quad (6.1)$$

où  $w$  est une fenêtre d’analyse de taille  $N_w$ , et  $S$  est le décalage temporel (en échantillons) entre deux trames successives. On note  $x_w^t(n) = x(n + tS)w(n)$  le signal fenêtré correspondant à la trame d’analyse  $t$ , et on a donc :

$$X(f, t) = \sum_{n=0}^{N_w-1} x_w^t(n)e^{-2i\pi \frac{f}{F}n}. \quad (6.2)$$

Supposons que le signal  $x$  représente un événement audio élémentaire qui est activé plusieurs fois (par exemple une note de piano) : les indices de trames  $t_m$ ,  $m \in \llbracket 0, M - 1 \rrbracket$ , correspondent aux  $M$  trames d’attaque de cette source dans le domaine TF. L’idée qui est au coeur du modèle de ce chapitre consiste à supposer que les signaux fenêtrés  $x_w^{t_m}(n)$  dans les trames

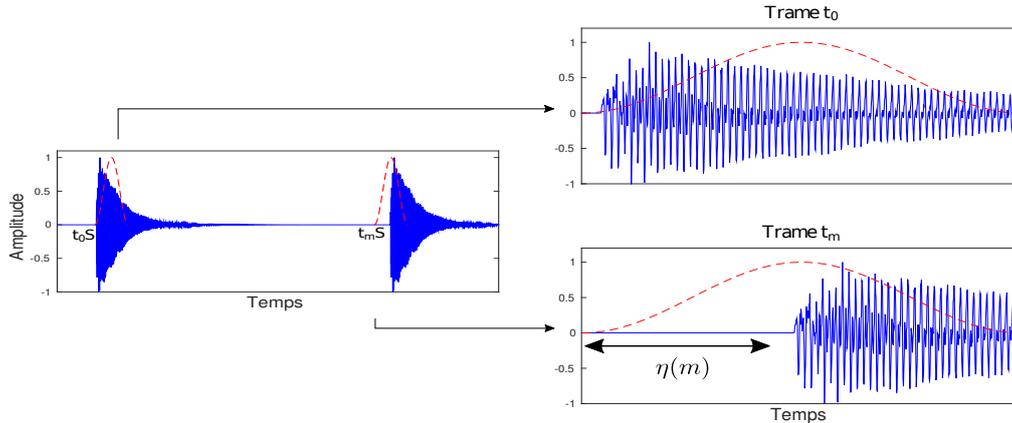


FIGURE 6.1 – Répétition d'un évènement audio (traits pleins) et positionnement de la fenêtre d'analyse (pointillés) : on suppose que les signaux fenêtrés sont identiques à un délai et facteur d'amplitude près.

d'attaque sont égaux, à un délai  $\eta$  et un facteur d'amplitude  $\rho$  près. Ainsi, en prenant comme référence le signal fenêtré dans la première trame d'attaque, on a :

$$x_w^{t_m}(n) \approx \rho(m)x_w^{t_0}(n - \eta(m)), \quad (6.3)$$

comme c'est illustré sur la figure 6.1. Le délai de  $\eta(m)$  échantillons<sup>1</sup> dépend donc du positionnement du signal par rapport à la fenêtre d'analyse. Cette hypothèse n'est vérifiée que pour une fenêtre d'analyse rectangulaire et lorsque la totalité du signal se trouve à l'intérieur de la fenêtre : cela ne correspond pas à des cas d'application pratiques, puisqu'on utilise généralement une fenêtre plus sophistiquée (afin d'éviter le phénomène de fuite spectrale), et que les signaux sont tronqués par l'opération de fenêtrage. En outre, ce modèle repose sur l'hypothèse que la forme d'onde des signaux correspondants aux diverses occurrences d'un même évènement sonore soient identique (au facteur d'amplitude près) : pour plus de réalisme, il faudrait considérer des variations de forme d'onde qui correspondent aux différentes nuances du jeu du musicien. Nous étudierons expérimentalement la validité de ce modèle lorsque l'on considère des occurrences d'une source qui comportent des variations de nuance.

La TFCT de  $x$  dans la trame  $t_m$  est donc :

$$\begin{aligned} X(f, t_m) &= \sum_{n=0}^{N_w-1} x_w^{t_m}(n) e^{-2i\pi \frac{f}{F} n} \\ &\approx \sum_{n=0}^{N_w-1} \rho(m) x_w^{t_0}(n - \eta(m)) e^{-2i\pi \frac{f}{F} n} \\ &\approx \rho(m) e^{2i\pi \frac{f}{F} \eta(m)} X(f, t_0). \end{aligned}$$

On pose  $\lambda(m) = \frac{2\pi\eta(m)}{F}$ , ce qui conduit à :

$$X(f, t_m) \approx \rho(m) e^{i\lambda(m)f} X(f, t_0). \quad (6.4)$$

1. Cette écriture suppose que le délai  $\eta(m)$  soit entier. En effet, lorsque celui-ci devient fractionnaire, l'expression (6.3) n'a plus de sens. Néanmoins, on introduit ultérieurement la notation  $\lambda(m)$  dans le domaine TF, qui permet d'étendre la notation  $\zeta$  des valeurs continues du délai.

En notant  $X$  sous forme polaire (module  $V$  et argument  $\phi$ ), (6.4) devient :

$$V(f, t_m)e^{i\phi(f, t_m)} \approx \rho(m)V(f, t_0)e^{i(\phi(f, t_0) + \lambda(m)f)}. \quad (6.5)$$

La relation (6.5) contient à la fois une information sur les modules et sur les phases. Si on note  $W(f) = V(f, t_0)$  et  $H(t_m) = \rho(m)$ , on obtient  $V(f, t_m) \approx W(f)H(t_m)$ . En généralisant ce résultat à toutes les trames, on retrouve le modèle NMF sur les amplitudes. Ainsi, ce modèle d'évènements audio répétés conduit naturellement au modèle NMF traditionnel sur les spectrogrammes d'amplitude. Ce qui nous intéresse ici est la relation sur les phases :

$$\phi(f, t_m) \approx \psi(f) + \lambda(m)f, \quad (6.6)$$

avec  $\psi(f) = \phi(f, t_0)$ , à laquelle on se réfère sous le nom de *phase de référence*. L'équation (6.6) traduit donc le modèle de phase que nous proposons : dans une trame d'attaque, la phase de la TFCT  $X$  est égale à une phase de référence (ne dépendant pas de l'indice de trame) à laquelle est ajouté un délai linéaire en fréquence. La forme de la phase de référence  $\psi(f)$  caractérise alors le timbre de l'instrument (et notamment les relations de phase entre partiels), alors que le délai  $\lambda$  informe sur le positionnement temporel du signal au sein de la fenêtre d'analyse.

## 6.2 Validation expérimentale préliminaire

On cherche dans cette partie à mettre en évidence la phase de référence introduite précédemment. On suppose connues les trames d'attaque  $t_m$ ,  $m \in \llbracket 0, 1 \rrbracket$ , bien que l'on pourrait par la suite les estimer (à partir par exemple de la MATLAB Tempogram Toolbox [GROSCHÉ et MÜLLER \(2011\)](#)).

### 6.2.1 Précision du modèle de phase

On considère dans cette expérience des signaux constitués d'une source (un évènement audio) qui est activée à deux reprises. Ces signaux sont :

- des notes de piano tirées aléatoirement de la base MAPS [EMIYA et al. \(2010\)](#),
- des notes de guitare électrique tirées aléatoirement de la base IDMT-SMT-GUITAR [KEHLING et al. \(2014\)](#).

Ces sources peuvent être activées soit à l'identique, c'est-à-dire sans variation de forme d'onde, soit avec une certaine nuance (afin de reproduire de façon plus réaliste les variations de jeu de l'instrumentiste). Dans le cas des notes de piano, les nuances possibles sont "mezzo-forte", "forte" et "piano". Pour les notes de guitare, les nuances sont obtenues en considérant des notes qui sont jouées à différents endroits du manche (en effet, avec la guitare, on peut produire une même note de plusieurs façons différentes, selon la case et la corde choisie).

On calcule la différence de phase  $\Delta\phi(f) = \phi(f, t_1) - \phi(f, t_0)$  entre les trames d'attaque. On s'attend à ce que cet écart de phase soit linéaire en  $f$  pour qu'il respecte le modèle (6.6). On calcule donc une approximation de ce décalage entre phases d'attaque par régression linéaire, et on illustre cette expérience sur la figure 6.2 dans le cas où il n'y a pas de variation de forme d'onde entre les deux occurrences de notes.

Afin de mesurer la pertinence de ce modèle, nous calculons (sur 25 signaux dans chaque cas de figure) l'erreur moyennée sur les fréquences (en radians) entre décalages de phases d'attaque observés et estimés par le modèle, et nous présentons les résultats dans le tableau 6.1.

La première ligne de ce tableau confirme ce que l'on constate visuellement sur la figure 6.2 : lorsqu'une source est activée "à l'identique", c'est-à-dire sans variation de forme d'onde, le

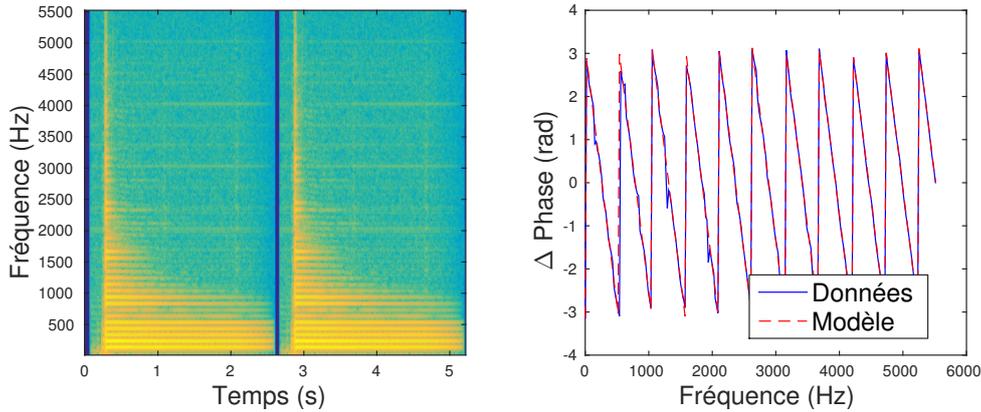


FIGURE 6.2 – Spectrogram comportant deux occurrences d’une note de guitare sans variation de forme d’onde (gauche) et décalages de phase entre attaques (droite).

	Piano	Guitare
Sans nuance	0.21	0.15
Avec nuance	1.48	1.49

TABLEAU 6.1 – Erreur moyenne (en radians) entre décalage de phases d’attaque observé et estimé par régression linéaire pour plusieurs types de données.

modèle est relativement précis. Lorsque les notes sont activées avec une certaine nuance variable, il reste possible d’exploiter une phase de référence pour caractériser la phase dans les trames d’attaque, même si ce modèle perd nettement en précision par rapport au cas de sources identiques.

### 6.2.2 Combinaison avec le déroulé linéaire

Nous combinons à présent le modèle de phase (6.6) et le déroulé linéaire introduit au chapitre 4 pour achever la reconstruction des phases à partir de celle des attaques.

#### Phase apprise à partir d’une première occurrence de la note

Nous considérons des mélanges constitués de deux occurrences d’une note (piano ou guitare) avec ou sans nuance. Nous reconstruisons la phase au niveau de la deuxième activation à partir de la première. Après avoir reconstruit la phase dans la deuxième trame d’attaque, nous achevons la reconstruction de phase de la deuxième occurrence de note par déroulé linéaire. Nous effectuons également une reconstruction par l’algorithme de Griffin et Lim (GL) qui utilise 200 itérations, et on mesure la qualité de la reconstruction par le SDR [VINCENT et al. \(2006\)](#). Les résultats sont présentés dans le tableau 6.2.

Globalement, notre modèle de phase améliore la qualité de reconstruction du signal par rapport à l’algorithme GL. Cela montre l’importance de cette phase d’attaque, qui est non seulement indispensable pour conduire à un résultat perceptif satisfaisant (attaque nette) mais également pour initialiser un déroulé pertinent des phases des partiels.

#### Base de phases de référence

En pratique, comme toutes les phases sont inconnues, nous n’avons pas à disposition la phase de référence. Une possibilité est alors d’utiliser une base de données de phases d’attaque.

	Griffin Lim	Modèle + Déroulé
Piano (sans nuance)	-3.4	<b>-0.5</b>
Guitare (sans nuance)	-2.6	<b>-0.3</b>
Piano (avec nuance)	-6.1	<b>-5.1</b>
Guitare (avec nuance)	-1.0	<b>0.3</b>

TABLEAU 6.2 – Qualité de reconstruction de phase (SDR en dB) d’une deuxième activation d’une source à partir de la première via un modèle de phase d’attaque combiné au déroulé linéaire.

	Griffin Lim	Modèle+Déroulé	Oracle+Déroulé
Morceaux de piano	-4.4	-3.3	-0.1
Morceaux de guitare	-4.3	-3.7	-0.6

TABLEAU 6.3 – Qualité de la reconstruction de signal (SDR en dB) en utilisant une base de données de phases d’attaque combinée au déroulé linéaire.

Nous considérons donc deux ensembles de notes (piano et guitare) à partir desquels nous fabriquons deux dictionnaires de phases de référence. Les dictionnaires sont construits avec une seule nuance par note.

On considère ensuite deux jeux de données : des morceaux de piano tirés de la base MAPS et des morceaux de guitare tirés de la base IDMT-SMT-GUITAR (ces bases comportent en effet aussi bien des notes isolées que des morceaux de musique, qui comportent donc plusieurs nuances des notes). Les trames d’attaque sont détectées grâce à la boîte à outils MATLAB Tempogram toolbox. Le délai  $\lambda$  est estimé grâce à un calcul sur les amplitudes similaire à celui conduit dans le chapitre 4 (section 4.4.1). Il est alors appliqué à la phase de référence contenue dans la base afin de reconstruire la phase d’attaque du signal. Les phases des partiels sont ensuite reconstruites par déroulé linéaire. À titre de comparaison, on reconstruit également les phases par déroulé linéaire à partir de phases d’attaque connues (cas Oracle). Nous testons enfin l’algorithme de Griffin Lim qui utilise 200 itérations.

Les résultats présentés dans le tableau 6.3 montrent l’intérêt d’utiliser une base de phases de références pour la reconstruction des phases d’attaque. Ce modèle améliore en effet légèrement les résultats par rapport à l’algorithme GL, même si sa performance reste éloignée de la performance Oracle.

### 6.3 Modèle de mélange de sources

Considérons à présent la TFCT  $X$  d’un mélange de  $K$  sources  $X_k$ , dont les amplitudes et les phases sont notées  $V_k$  et  $\phi_k$  respectivement. On note  $t_m$ ,  $m \in \llbracket 0, M-1 \rrbracket$  les indices des  $M$  trames d’attaque de  $X$ . On peut extraire de  $X$  la matrice des attaques  $Y \in \mathbb{C}^{F \times M}$  :

$$Y(f, m) = X(f, t_m) = \sum_{k=1}^K V_k(f, t_m) e^{i\phi_k(f, t_m)}. \quad (6.7)$$

On introduit le modèle de phase (6.6) pour chaque source complexe :

$$\phi_k(f, t_m) = \psi_k(f) + \lambda_k(m)f. \quad (6.8)$$

On aboutit alors au modèle de mélange suivant :  $\forall (f, m) \in \llbracket 0, F-1 \rrbracket \times \llbracket 0, M-1 \rrbracket$ ,

$$\hat{Y}(f, m) = \sum_{k=1}^K V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f}. \quad (6.9)$$

**Remarque :** Il serait plus rigoureux de considérer des ensembles de trames d'attaque différents pour chaque source, car les paramètres  $\lambda_k$  et  $\psi_k$  n'ont de sens que dans les trames d'attaque pour la source  $k$ . Néanmoins, dans ce chapitre, on considère des données pour lesquelles une trame d'attaque du mélange est une trame d'attaque pour toutes les sources, ou bien telle que si une trame d'attaque pour une source ne l'est pas pour une autre, alors cette autre source a une énergie nulle dans cette trame. Ainsi, le fait de considérer les amplitudes dans le modèle de mélange permet de rendre compte de cette hypothèse. Pour des données réalistes plus complexes, il est nécessaire de distinguer les trames d'attaque selon les sources : c'est ce que nous proposons dans le chapitre 7.

Ce modèle basé sur la répétition de phases permet de réduire la dimension des données : les phases des sources sont initialement représentées par  $KFM$  paramètres alors que notre modèle utilise  $K(F + M - 1)$  paramètres. Si  $M > 1$  (ce qui est en pratique le cas, puisque on suppose qu'il y a plusieurs activations des sources), on observe bien que  $K(F + M - 1) < KFM$ . Pour un mélange de 3 sources avec 5 trames d'attaque et une transformée de Fourier à 512 canaux, ce modèle utilise presque cinq fois moins de paramètres que les données originales.

Ce modèle peut se réécrire sous la forme matricielle suivante :

$$\hat{Y} = \sum_{k=1}^K V_k \odot (\Psi_k \Lambda_k), \quad (6.10)$$

où  $\Psi_k = \text{diag}(e^{i\psi_k(0)}, \dots, e^{i\psi_k(F-1)})$  est une matrice diagonale de dimensions  $F \times F$  contenant les phases de référence, et  $\Lambda_k$  est une matrice de Vandermonde de dimensions  $F \times M$  contenant les paramètres de délai :

$$\Lambda_k = \begin{pmatrix} 1 & \dots & 1 \\ e^{i\lambda_k(0)} & \dots & e^{i\lambda_k(M-1)} \\ \vdots & & \vdots \\ e^{i\lambda_k(0)(F-1)} & \dots & e^{i\lambda_k(M-1)(F-1)} \end{pmatrix} \quad (6.11)$$

## 6.4 Estimation des phases des composantes

Dans cette partie, nous estimons les paramètres du modèle (6.9). On suppose que les amplitudes des sources  $V_k$  dans les trames d'attaque sont connues. On cherche donc à estimer les paramètres  $\psi_k(f)$  et  $\lambda_k(m)$ ,  $\forall (f, m, k)$ . Une première méthode consiste à minimiser une fonction de coût représentant l'erreur entre les données et le modèle (contrainte *stricte*).

Lorsque les données ne correspondent plus exactement au modèle, une contrainte stricte peut être trop forte pour estimer les paramètres. On s'inspire donc de [RIGAUD et al. \(2013\)](#) pour proposer une méthode alternative d'estimation, qui repose sur une contrainte *relaxée*.

### 6.4.1 Contrainte stricte

On considère la fonction de coût suivante :

$$\mathcal{C}_s = \sum_{f,m} \left| Y(f, m) - \sum_{k=1}^K V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2. \quad (6.12)$$

Nous minimisons cette fonction par rapport à chacune des variables successivement, ce qui conduit à une procédure itérative. Cette méthode est connue sous le nom de méthode de relaxation.

## Estimation de $\psi_k(f)$

Réécrivons  $\mathcal{C}_s$  en isolant les termes dépendant de  $\psi_k(f)$  :

$$\mathcal{C}_s = \sum_m \left| Y(f, m) - \sum_{l \neq k} V_l(f, t_m) e^{i\psi_l(f)} e^{i\lambda_l(m)f} - V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2 + \sum_m \sum_{p \neq f} \left| Y(p, m) - \sum_l V_l(p, t_m) e^{i\psi_l(p)} e^{i\lambda_l(m)p} \right|^2. \quad (6.13)$$

On pose

$$B_k(f, m) = Y(f, m) - \sum_{l \neq k} V_l(f, t_m) e^{i\psi_l(f)} e^{i\lambda_l(m)f}, \quad (6.14)$$

et on note  $\stackrel{c}{=}$  l'égalité à une constante additive près, ne dépendant pas de la variable considérée, ici  $\psi_k(f)$ . On peut alors écrire :

$$\mathcal{C}_s \stackrel{c}{=} \sum_m \left| B_k(f, m) - V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2, \quad (6.15)$$

que l'on développe :

$$\begin{aligned} \mathcal{C}_s &\stackrel{c}{=} \sum_m \left| B_k(f, m) - V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2 \\ &\stackrel{c}{=} \sum_m |B_k(f, m)|^2 + |V_k(f, t_m)|^2 - 2\Re(\overline{B_k(f, m)} V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f}) \\ &\stackrel{c}{=} -2\Re \left( \sum_m \overline{B_k(f, m)} V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right) \\ &\stackrel{c}{=} -2\Re \left( e^{i\psi_k(f)} \sum_m \overline{B_k(f, m)} V_k(f, t_m) e^{i\lambda_k(m)f} \right), \end{aligned}$$

où  $\Re$  désigne la partie réelle. On pose  $z_s = \sum_m B_k(f, m) V_k(f, t_m) e^{-i\lambda_k(m)f}$  (on enlève les indices  $k$  et  $f$  pour plus de clarté), que l'on note en coordonnées polaires  $z_s = |z_s| e^{i\xi_s}$ . On a :

$$\begin{aligned} \mathcal{C}_s &\stackrel{c}{=} -2|z_s| \Re \left( e^{i\psi_k(f)} e^{-i\xi_s} \right) \\ &\stackrel{c}{=} -2|z_s| \cos(\psi_k(f) - \xi_s). \end{aligned}$$

La fonction de coût, ainsi écrite, est aisée à minimiser par rapport à  $\psi_k(f)$ . En effet, minimiser  $\mathcal{C}_s$  revient à maximiser le cosinus de  $\psi_k(f) - \xi_s$ , ce qui conduit à  $\psi_k(f) = \xi_s = \angle(z_s)$ , soit :

$$\psi_k(f) = \angle \left( \sum_m B_k(f, m) V_k(f, t_m) e^{-i\lambda_k(m)f} \right). \quad (6.16)$$

**Remarque :** Il est possible d'obtenir ce résultat en passant par la dérivée de  $\mathcal{C}_s$  par rapport à  $\psi_k(f)$  (ce que nous avons proposé dans [MAGRON et al. \(2015a\)](#)), qui est :

$$\frac{\partial \mathcal{C}_s}{\partial \psi_k(f)} = i \sum_m B_k(f, m) V_k(f, t_m) e^{-i\lambda_k(m)f} e^{-i\psi_k(f)} - \overline{B_k(f, m)} V_k(f, t_m) e^{i\lambda_k(m)f} e^{i\psi_k(f)}. \quad (6.17)$$

Annuler cette dérivée partielle conduit à l'estimation suivante pour  $\psi_k(f)$  :

$$\psi_k(f) = \pm \angle \left( \sum_m B_k(f, m) V_k(f, t_m) e^{-i\lambda_k(m)f} \right). \quad (6.18)$$

Pour lever l'ambiguïté sur le signe de l'argument, on peut examiner le signe de la dérivée seconde. Pour la valeur donnée par (6.16), celle-ci est positive, ce qui confirme que cette solution est bien un minimum de  $\mathcal{C}_s$ .

### Estimation de $\lambda_k(m)$

On écrit à présent  $\mathcal{C}_s$  de façon similaire à (6.15) en isolant les termes dépendant de  $\lambda_k(m)$  :

$$\mathcal{C}_s \stackrel{c}{=} \sum_f \left| B_k(f, m) - V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2. \quad (6.19)$$

Minimiser  $\mathcal{C}_s$  par rapport à  $\lambda_k(m)$  revient donc à minimiser la quantité suivante :

$$\tilde{\mathcal{C}}_s(k, m) = \sum_f \left| B_k(f, m) e^{-i\psi_k(f)} - V_k(f, t_m) e^{i\lambda_k(m)f} \right|^2. \quad (6.20)$$

On note :

$$\begin{aligned} \beta_k(f, m) &= B_k(f, m) e^{-i\psi_k(f)}, \\ \underline{\beta}_k(m) &= [\beta_k(0, m), \dots, \beta_k(F-1, m)]^T, \\ \underline{\Lambda}_k(m) &= [1, e^{i\lambda_k(m)}, \dots, e^{i\lambda_k(m)(F-1)}]^T, \\ \underline{V}_k(m) &= [V_k(0, t_m), \dots, V_k(F-1, t_m)]^T. \end{aligned}$$

La fonction (6.20) se reformule alors comme suit :

$$\tilde{\mathcal{C}}_s(k, m) = \|\underline{V}_k(m) \odot \underline{\Lambda}_k(m) - \underline{\beta}_k(m)\|^2, \quad (6.21)$$

où  $\|\cdot\|$  désigne la norme euclidienne pour les vecteurs. La solution à ce problème peut être obtenue en s'inspirant de l'algorithme ESPRIT [HUA et al. \(2004\)](#). En effet, lorsque la fonction de coût  $\tilde{\mathcal{C}}_s$  est nulle, on remarque que (on s'affranchit des indices  $k$  et  $m$  pour plus de lisibilité) :

$$\underline{\beta}_\downarrow^H \underline{\beta}_\uparrow = (\underline{V} \odot \underline{\Lambda})_\downarrow^H (\underline{V} \odot \underline{\Lambda})_\uparrow = \underline{V}_\downarrow^H \underline{V}_\uparrow e^{i\lambda}, \quad (6.22)$$

où  $\cdot^H$  désigne le transposé Hermitien, et la notation  $\underline{v}_\downarrow$  (respectivement  $\underline{v}_\uparrow$ ) désigne le vecteur obtenu en retirant le dernier élément (respectivement le premier) d'un vecteur  $\underline{v}$ . Ainsi :

$$e^{i\lambda} = \frac{\underline{\beta}_\downarrow^H \underline{\beta}_\uparrow}{\underline{V}_\downarrow^H \underline{V}_\uparrow}. \quad (6.23)$$

L'estimation de  $\lambda_k(m)$  est donc :

$$\lambda_k(m) = \angle \left( \underline{\beta}_k(m)_\downarrow^H \underline{\beta}_k(m)_\uparrow \right). \quad (6.24)$$

---

**Algorithme 6** Estimation des paramètres de phase sous contrainte stricte
 

---

**Entrées :**  $Y \in \mathbb{C}^{F \times M}$ ,  $V_k \in \mathbb{R}_+^{F \times M}$ ,  $\psi_k^{ini} \in \mathbb{R}^{F \times 1}$ ,  $\lambda_k^{ini} \in \mathbb{R}^{1 \times M}$ .

**Initialisation :**

$$\psi_k = \psi_k^{ini}, \lambda_k = \lambda_k^{ini}.$$

$$\hat{Y}_k(f, m) = V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f}.$$

$$\hat{Y}(f, m) = \sum_{k=1}^K \hat{Y}_k(f, m).$$

$$B_k(f, m) = Y(f, m) - \hat{Y}(f, m) + \hat{Y}_k(f, m).$$

**tant que** critère non atteint **faire**

**pour**  $k = 1$  à  $K$ ,  $f = 0$  à  $F - 1$  et  $m = 0$  à  $M - 1$  **faire**

**Calculer**  $\psi$

$$\psi_k(f) = \angle \left( \sum_m B_k(f, m) V_k(f, t_m) e^{-i\lambda_k(m)f} \right).$$

**Calculer**  $\underline{\beta}$

$$\beta_k(f, m) = B_k(f, m) e^{-i\psi_k(f)},$$

$$\underline{\beta}_k(m) = [\beta_k(0, m), \dots, \beta_k(F - 1, m)]^T.$$

**Calculer**  $\lambda$

$$\lambda_k(m) = \angle \left( \underline{\beta}_k(m)_{\downarrow}^H \underline{\beta}_k(m)_{\uparrow} \right).$$

**Calculer**  $\hat{Y}$

$$\hat{Y}_k(f, m) = V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f},$$

$$\hat{Y}(f, m) = \sum_{k=1}^K \hat{Y}_k(f, m).$$

**Calculer**  $B$

$$B_k(f, m) = Y(f, m) - \hat{Y}(f, m) + \hat{Y}_k(f, m).$$

**fin pour**

**fin tant que**

**Sorties :**  $\hat{Y}, \hat{Y}_k, \psi, \lambda$ .

---

**Algorithme d'estimation strict**

Les équations (6.16) et (6.24) fournissent des règles de mises à jour pour estimer itérativement les paramètres de phase. L'Algorithme 6 détaille cette procédure. Cet algorithme prend en entrée la matrice des attaques, obtenue au préalable, grâce par exemple à la boîte à outils MATLAB Tempogram Toolbox [GROSCHÉ et MÜLLER \(2011\)](#). L'initialisation pourrait par ailleurs être affinée en prenant pour  $\psi_k(f)$  la valeur de la phase du mélange en une trame où l'on est sûr que seule la source  $k$  est active (la phase du mélange est alors égale à la phase de la source, prise comme référence dans cette trame).

Notons que l'ordre dans lequel sont effectuées les mises à jour des paramètres dans l'algorithme 6 (et de même plus loin dans l'algorithme 7) est arbitraire. Nous avons obtenu des résultats similaires en inversant cet ordre.

### 6.4.2 Contrainte relaxée

On considère à présent une contrainte relaxée, ce qui mène à la fonction de coût suivante :

$$C_r = \sum_{f,m} \left| Y(f, m) - \sum_{k=1}^K V_k(f, t_m) e^{i\phi_k(f, t_m)} \right|^2 + \sigma \sum_{f,m,k} V_k(f, t_m)^2 \left| e^{i\phi_k(f, t_m)} - e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2. \quad (6.25)$$

Le paramètre  $\sigma$  permet de donner plus ou moins d'importance à la contrainte issue du modèle de phase. Pour  $\sigma = 0$ , les phases d'attaque ne sont plus contraintes, et lorsque  $\sigma$  devient très grand, les sources ne somment plus au mélange : ainsi, il est nécessaire de choisir convenablement ce paramètre pour obtenir un compromis entre attache aux données et contrainte de phase (c'est l'objet de l'expérience conduite dans la section 6.5.1).

### Estimation de $\psi_k(f)$

On applique la même méthode que dans le cas strict, qui consiste à isoler dans la fonction de coût les termes ne dépendant que de  $\psi_k(f)$  :

$$\mathcal{C}_r \stackrel{c}{=} \sum_m V_k(f, t_m)^2 \left| e^{i\phi_k(f, t_m)} - e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2. \quad (6.26)$$

On a alors :

$$\begin{aligned} \mathcal{C}_r &\stackrel{c}{=} \sum_m V_k(f, t_m)^2 \left( 1 + 1 - 2\Re(e^{-i\phi_k(f, t_m)} e^{i\psi_k(f)} e^{i\lambda_k(m)f}) \right) \\ \mathcal{C}_r &\stackrel{c}{=} -2\Re \left( \sum_m V_k(f, t_m)^2 e^{-i\phi_k(f, t_m)} e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right) \\ \mathcal{C}_r &\stackrel{c}{=} -2\Re \left( e^{i\psi_k(f)} \sum_m V_k(f, t_m)^2 e^{-i\phi_k(f, t_m)} e^{i\lambda_k(m)f} \right). \end{aligned}$$

On pose  $z_r = \sum_m V_k(f, t_m)^2 e^{-i\phi_k(f, t_m)} e^{i\lambda_k(m)f}$  (on enlève les indices  $k$  et  $f$  pour plus de clarté), que l'on écrit en coordonnées polaires  $z_r = |z_r| e^{i\xi_r}$ . On a :

$$\mathcal{C}_r \stackrel{c}{=} -2|z_r| \cos(\psi_k(f) - \xi_r).$$

Minimiser  $\mathcal{C}_r$  par rapport à  $\psi_k(f)$  revient à maximiser le cosinus de  $\psi_k(f) - \xi_r$ , ce qui conduit à  $\psi_k(f) = \xi_r = \angle(z_r)$ , soit :

$$\psi_k(f) = \angle \left( \sum_m V_k(f, t_m)^2 e^{i\phi_k(f, t_m)} e^{-i\lambda_k(m)f} \right). \quad (6.27)$$

### Estimation de $\lambda_k(m)$

Nous estimons  $\lambda_k(m)$  en appliquant une méthode similaire au cas de la contrainte stricte. Cela requiert d'introduire un nouveau paramètre  $\gamma$  (similaire au  $\beta$  de la partie précédente) défini comme suit :

$$\gamma_k(f, m) = V_k(f, t_m) e^{i\phi_k(f, t_m)} e^{-i\psi_k(f)}. \quad (6.28)$$

On pose :

$$\underline{\gamma}_k(m) = [\gamma_k(0, m), \dots, \gamma_k(F-1, m)]^T. \quad (6.29)$$

L'adaptation de la méthode ESPRIT comme utilisée précédemment mène à l'estimation :

$$\lambda_k(m) = \angle \left( \underline{\gamma}_k(m) \downarrow^H \underline{\gamma}_k(m) \uparrow \right). \quad (6.30)$$

---

### Estimation de $\phi_k(f, t_m)$

Enfin, il faut estimer, en plus des paramètres du modèle de phase, les termes  $\phi_k(f, t_m)$ . La méthode est la même que pour les termes  $\psi_k(f)$  : on réécrit la fonction de coût en isolant les termes dépendant uniquement de  $\phi_k(f, t_m)$ . On note ici, de façon similaire à la partie précédente :

$$B_k(f, m) = Y(f, m) - \sum_{l \neq k} V_l(f, t_m) e^{i\phi_l(f, t_m)}, \quad (6.31)$$

et on écrit donc la fonction de coût de la façon suivante :

$$\mathcal{C}_r \stackrel{c}{=} \left| B_k(f, t_m) - V_k(f, t_m) e^{i\phi_k(f, t_m)} \right|^2 + \sigma V_k(f, t_m)^2 \left| e^{i\phi_k(f, t_m)} - e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right|^2, \quad (6.32)$$

que l'on réécrit :

$$\begin{aligned} \mathcal{C}_r \stackrel{c}{=} & |B_k(f, t_m)|^2 + |V_k(f, t_m)|^2 - 2\Re(\overline{B}_k(f, t_m) V_k(f, t_m) e^{i\phi_k(f, t_m)}) \\ & + \sigma V_k(f, t_m)^2 \left( 1 + 1 - 2\Re(e^{i\phi_k(f, t_m)} e^{-i\psi_k(f)} e^{-i\lambda_k(m)f}) \right). \end{aligned} \quad (6.33)$$

En retirant les termes ne dépendant pas de  $\phi_k(f, t_m)$ , on peut simplifier cette expression :

$$\begin{aligned} \mathcal{C}_r \stackrel{c}{=} & -2V_k(f, t_m) \Re \left( \overline{B}_k(f, t_m) e^{i\phi_k(f, t_m)} + \sigma V_k(f, t_m) e^{i\phi_k(f, t_m)} e^{-i\psi_k(f)} e^{-i\lambda_k(m)f} \right) \\ \stackrel{c}{=} & -2V_k(f, t_m) \Re \left( e^{i\phi_k(f, t_m)} (\overline{B}_k(f, t_m) + \sigma V_k(f, t_m) e^{-i\psi_k(f)} e^{-i\lambda_k(m)f}) \right). \end{aligned}$$

On pose  $z_\phi = B_k(f, t_m) + \sigma V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f}$  (on enlève les indices  $k, f$  et  $m$  pour plus de clarté), que l'on réécrit en coordonnées polaires  $z_\phi = |z_\phi| e^{i\xi_\phi}$ , et on a :

$$\mathcal{C}_r \stackrel{c}{=} -2V_k(f, t_m) |z_\phi| \cos(\phi_k(f, t_m) - \xi_\phi). \quad (6.34)$$

En appliquant la même technique que précédemment, on a finalement :

$$\phi_k(f, t_m) = \angle \left( B_k(f, t_m) + \sigma V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f} \right). \quad (6.35)$$

### Algorithme d'estimation relaxé

L'Algorithme 7 détaille la procédure d'estimation itérative des paramètres de phase obtenue grâce aux équations (6.27), (6.30) et (6.35) dans le cas relaxé.

---

**Algorithme 7** Estimation des paramètres de phase sous contrainte relaxée

---

**Entrées**  $Y \in \mathbb{C}^{F \times M}$ ,  $V_k \in \mathbb{R}_+^{F \times M}$ ,  $\psi_k^{ini} \in \mathbb{R}^{F \times 1}$ ,  $\lambda_k^{ini} \in \mathbb{R}^{1 \times M}$ ,  $\phi_k^{ini} \in \mathbb{R}^{F \times T}$ ,  $\sigma \in \mathbb{R}_+$ .

**Initialisation :**

$$\phi = \phi_{ini}, \psi = \psi_{ini}, \lambda = \lambda_{ini}.$$

$$\hat{Y}_k(f, m) = V_k(f, t_m) e^{i\phi_k(f, t_m)}.$$

$$\hat{Y}(f, m) = \sum_{k=1}^K \hat{Y}_k(f, m).$$

$$B_k(f, m) = Y(f, m) - \hat{Y}(f, m) + \hat{Y}_k(f, m).$$

**tant que** critère non atteint **faire**

**pour**  $k = 1$  à  $K$ ,  $f = 0$  à  $F - 1$  et  $m = 0$  à  $M - 1$  **faire**

**Calculer**  $\phi$

$$\phi_k(f, t_m) = \angle(B_k(f, m) + \sigma V_k(f, t_m) e^{i\psi_k(f)} e^{i\lambda_k(m)f}).$$

**Calculer**  $\psi$

$$\psi_k(f) = \angle\left(\sum_m V_k(f, t_m)^2 e^{i\phi_k(f, t_m)} e^{-i\lambda_k(m)f}\right).$$

**Calculer**  $\gamma$

$$\gamma_k(f, m) = V_k(f, t_m) e^{i\phi_k(f, t_m)} e^{-i\psi_k(f)},$$

$$\gamma_k(m) = [\gamma_k(0, m), \dots, \gamma_k(F - 1, m)]^T.$$

**Calculer**  $\lambda$

$$\lambda_k(m) = \angle\left(\gamma_k(m)_\downarrow^H \gamma_k(m)_\uparrow\right).$$

**Calculer**  $\hat{Y}$

$$\hat{Y}_k(f, m) = V_k(f, t_m) e^{i\phi_k(f, t_m)},$$

$$\hat{Y}(f, m) = \sum_{k=1}^K \hat{Y}_k(f, m).$$

**Calculer**  $B$

$$B_k(f, m) = Y(f, m) - \hat{Y}(f, m) + \hat{Y}_k(f, m).$$

**fin pour**

**fin tant que**

**Sorties :**  $\hat{Y}, \hat{Y}_k, \psi, \lambda, \phi$ .

---

## 6.5 Résultats expérimentaux

Dans cette partie, nous présentons des expériences menées pour évaluer le potentiel de notre méthode. Les signaux sont échantillonnés à  $F_s = 11025$  Hz. La TFCT est calculée avec une fenêtre de Hann de longueur 512 échantillons et 75 % de recouvrement. Les phases des sources sont calculées par 100 itérations de nos algorithmes (strict et relaxé), la performance n'étant pas améliorée au-delà.

La boîte à outils MATLAB Tempogram Toolbox est utilisée pour estimer les trames d'attaque. Nous extrayons ensuite la matrice d'attaque  $Y$  à partir de la matrice complète  $X$ . Nous utilisons les SDR, SIR et SAR comme mesure de la qualité de la séparation de sources.

### 6.5.1 Influence du paramètre $\sigma$

Dans cette expérience, on cherche à estimer les paramètres de phases d'attaque sur des mélanges de deux sources ( $K = 2$ ) dans lesquels on observe successivement chaque source seule, puis la superposition des deux. Nous testons les algorithmes tout d'abord sur des données synthétiques qui sont fabriquées en suivant le modèle (6.9), puis sur des données plus réalistes, des mélanges de notes de piano.

On teste les algorithmes avec contrainte stricte et avec contrainte relaxée pour différentes valeurs du paramètre de relaxation  $\sigma$ . Les valeurs des amplitudes des sources  $V_k$  sont supposées

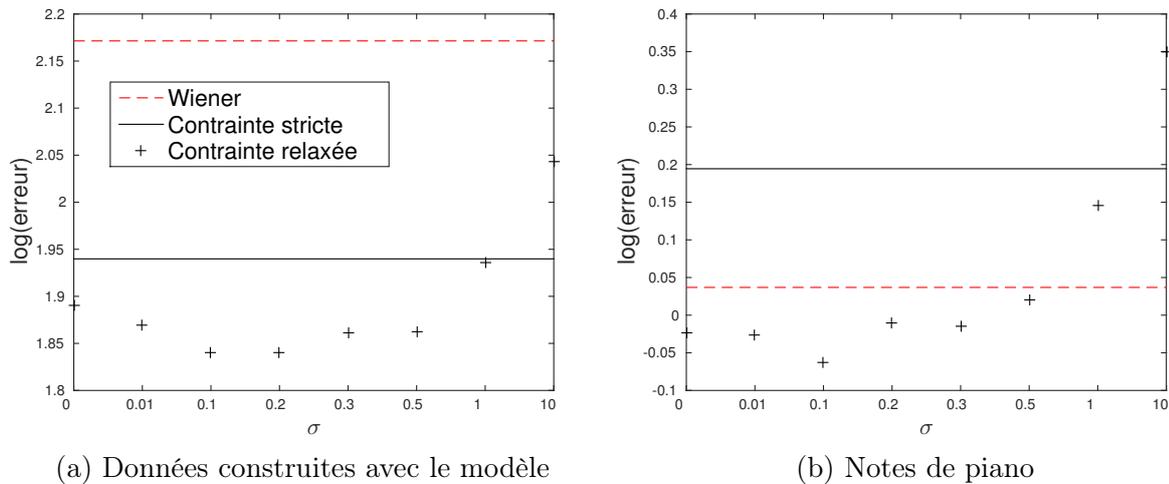


FIGURE 6.3 – Erreur (6.36) entre données et modèle estimé au niveau des attaques.

connues afin de s'intéresser spécifiquement à la reconstruction de phase. Ces algorithmes sont comparés à l'approche traditionnelle du filtrage de Wiener. Pour toutes ces méthodes, nous calculons l'erreur d'estimation moyennée sur les sources :

$$\frac{1}{K} \sum_{k=1}^K \|Y_k - \hat{Y}_k\|_2, \quad (6.36)$$

où  $\hat{Y}_k$  est l'estimée de la  $k$ -ième source  $Y_k$  dans les trames d'attaque. Les résultats sont moyennés sur 30 signaux pour chaque jeu de données, et présentés sur la figure 6.3.

Lorsque les données suivent exactement notre modèle, l'algorithme avec contrainte stricte conduit à un meilleur résultat que le filtrage de Wiener. Pour des données réelles, qui ne suivent donc plus rigoureusement le modèle, l'algorithme d'estimation strict ne donne pas de meilleurs résultats que le filtrage de Wiener, certainement en raison d'une contrainte trop forte qui ne correspond plus à la réalité des données. Néanmoins, lorsque l'on applique l'algorithme relaxé, on voit que pour certaines valeurs de  $\sigma$ , les résultats sont meilleurs qu'avec le filtrage de Wiener. Cela traduit le potentiel d'un tel modèle de phase, à condition qu'il soit utilisé de façon relaxée pour prendre en compte les écarts au modèle.

### 6.5.2 Performance du modèle de phase

Nous proposons d'appliquer cette technique de reconstruction de phase à la séparation de sources. Nous considérons plusieurs jeux de données :

- A : 30 mélanges de deux sources, composées de sinusoides amorties synthétiques. Les sources ne se recouvrent pas dans le domaine TF ;
- B : 30 mélanges de deux sources, composées de sinusoides amorties synthétiques. Les sources se recouvrent dans le domaine TF ;
- C : 30 mélanges de deux notes de piano tirées de la base MAPS ;
- D : Un extrait MIDI de 1.57 secondes. Il est composé de plusieurs occurrences de trois notes de basse, de trois notes de piano et d'un accord de guitare.

Pour les mélanges dans les jeux de données A, B et C, chaque source est successivement observée seule, puis les deux sources sont activées simultanément. On applique l'algorithme

Données	SDR	SIR	SAR
A	0.16	0.3	0.14
B	0.002	0.003	0.001
C	0.05	0.1	0.02
D	-0.09	0.15	-0.3

TABLEAU 6.4 – Variation de score (en dB) entre méthodes de reconstruction de phase par modèle d'attaque et par filtrage de Wiener.

Données	Méthode	SDR	SIR	SAR
A	Wiener	<b>34.8</b>	<b>44.9</b>	<b>35.3</b>
	RePU	6.9	28.9	6.9
B	Wiener	<b>11.4</b>	15.7	<b>13.6</b>
	RePU	8.7	<b>28.4</b>	8.8
C	Wiener	<b>12.6</b>	<b>17.5</b>	<b>14.6</b>
	RePU	2.0	15.5	2.6
D	Wiener	<b>15.9</b>	<b>18.6</b>	<b>19.7</b>
	RePU	5.5	15.2	6.4

TABLEAU 6.5 – Performance de la séparation de source (SDR, SIR et SAR en dB) pour le filtrage de Wiener et la méthode RePU.

relaxé avec  $\sigma = 0.1$  pour restaurer les phases des attaques. Alternativement, on peut utiliser le filtrage de Wiener pour la restauration des phases d'attaque. Par suite, les phases des partiels des composantes sont estimées par filtrage de Wiener, ce qui permet de comparer les deux méthodes utilisées pour restaurer les phases d'attaque en employant la même technique pour reconstruire les phases des autres trames.

Nous présentons dans le tableau 6.4 la valeur  $\Delta = SR(\text{Méthode proposée}) - SR(\text{Wiener})$ , ou  $SR$  représente le score considéré (SDR, SIR ou SAR). Ainsi, une valeur positive de cet écart traduit une amélioration des performances par notre méthode par rapport au filtrage de Wiener pour la restauration des phases d'attaque.

On constate que le modèle de phase améliore légèrement les résultats par rapport au filtrage de Wiener sur ces jeux de données. Cette amélioration est faible en valeur absolue et en erreur relative puisque les niveaux de performance sont déjà très élevés (entre 20 et 30 dB). Les hauts niveaux de performance sont expliqués par la simplicité des données, et la relative modération de l'augmentation due au fait que les signaux reconstruits ne diffèrent qu'au niveau des attaques. Ainsi, nous allons par la suite utiliser deux approches différentes pour la reconstruction des phases de partiels.

### 6.5.3 Modèle de phase d'attaque et déroulé linéaire

L'expérience précédente montre l'intérêt d'utiliser notre modèle de phase d'attaque pour l'estimation des composantes complexes dans le mélange. Néanmoins, les phases des partiels étaient estimées par filtrage de Wiener. À présent, on estime les phases des partiels en appliquant l'algorithme de déroulé linéaire de phases introduit dans le chapitre 4. Cette méthode est désignée par l'acronyme **RePU** (pour *Repeating Phase with Unwrapping*). Nous la comparons au filtrage de Wiener intégral (attaques + partiels) et présentons les résultats dans le tableau 6.5.

Ces résultats montrent que le filtrage de Wiener fournit globalement de meilleurs résultats que la combinaison du modèle de phases d'attaque et du déroulé linéaire.

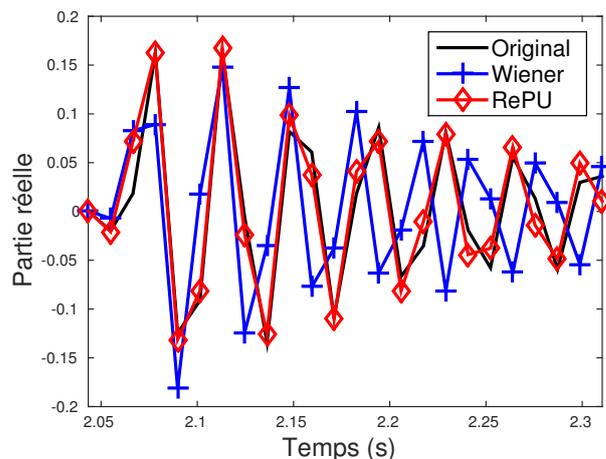


FIGURE 6.4 – Partie réelle de la source correspondant à la note G4 dans la bande de fréquences autour de 796 Hz, sur une fenêtre de temps où deux sources se recouvrent.

Une évaluation perceptive informelle de notre part sur les données audio obtenues suggère cependant que les indicateurs utilisés dans ces tests (les SDR, SIR et SAR) ne sont pas aptes à rendre compte de certaines propriétés des signaux séparés. Par exemple, le phénomène de battements observé dans le jeu de données B (ainsi que sur d’autres tests effectués sur notes de piano) est complètement supprimé lors de la séparation de sources avec notre méthode, alors que le filtrage de Wiener ne permet pas de s’en affranchir. On illustre ce phénomène par la figure 6.4 : considérant un mélange de deux notes de piano (C4 et G4), on représente sur cette figure la partie réelle de la deuxième source (note G4) sur la fenêtre de temps et dans une bande de fréquences où l’on observe un recouvrement. La méthode RePU approche mieux le signal original que le filtrage de Wiener, qui conduit à des interférences entre sources.

Bien qu’une perte de définition des transitoires d’attaque soit également à déplorer avec notre technique, ces critères perceptifs ne sont donc pas nécessairement capturés par ces indicateurs.

On constate donc que similairement aux expériences menées dans le chapitre 5, lorsque le déroulé est appliqué isolément sur chaque source (c’est-à-dire sans tenir compte de la phase du mélange), les performances sont moindres qu’en utilisant le filtrage de Wiener. La baisse de performance que nous avons constatée dans cette expérience provient donc probablement de l’utilisation isolée du déroulé linéaire, qui n’exploite pas la phase du mélange au niveau des partiels.

#### 6.5.4 Prise en compte de la phase du mélange pour le déroulé

Dans cette expérience, nous cherchons à comparer uniquement l’impact de la méthode choisie pour la reconstruction des phases d’attaque. Les phases des partiels sont reconstruites en utilisant l’algorithme 3 introduit au chapitre 5. Trois choix sont possibles pour estimer les composantes dans les trames d’attaque :

- Utiliser la phase du **Mélange** ;
- Le **Modèle** de phase introduit ici ;
- Une approche **Oracle** : les phases d’attaque sont supposées connues.

Les tests sont conduits sur des mélanges de sinusoides synthétiques avec recouvrement TF et des mélanges de notes de piano, et les résultats sont présentés dans le tableau 6.6.

Données	Méthode	SDR	SIR	SAR
Données synthétiques	<b>Mélange</b>	46.0	55.2	46.6
	<b>Modèle</b>	46.7	56.1	47.3
	<b>Oracle</b>	49.5	58.5	50.1
Notes de piano	<b>Mélange</b>	20.4	27.1	21.5
	<b>Modèle</b>	21.0	27.9	22.1
	<b>Oracle</b>	22.6	29.8	23.6

TABLEAU 6.6 – Performance de la séparation de sources (SDR, SIR et SAR en dB) pour diverses méthodes de reconstruction des phases d’attaque.

Les interprétations que l’on peut faire à partir de ce tableau sont les mêmes pour les deux jeux de données. Une comparaison entre **Mélange** et **Oracle** montre qu’en utilisant la technique de déroulé linéaire, il existe une certaine marge de progression possible entre une donner la phase du mélange aux sources dans les trames d’attaque et une approche Oracle (phases d’attaque connues). Cette marge de progression est d’environ 3 dB selon l’indicateur et le jeu de données considérés. Notre méthode conduit à un gain moyen de 0.7 dB par rapport à **Mélange**.

La méthode proposée ici combine modèle de phases d’attaque et déroulé linéaire pour les partiels. L’impact du déroulé linéaire est plus significatif que celui du modèle de phase dans les trames d’attaque, mais ceci est expliqué par le fait que les trames concernées par cette méthode (qui ne sont pas des trames d’attaque) représentent une proportion des données très importante : sur un mélange de notes de piano tel qu’utilisé ici, il y a 3 trames d’attaque pour un total de 262 trames. Le déroulé linéaire concerne donc presque 99 % des trames, ce qui explique que son impact sur le résultat final soit plus important que celui de la technique utilisée pour la reconstruction des phases d’attaque. Comme le déroulé linéaire dépend cependant de son initialisation (effectuée au niveau des trames d’attaque), la méthode employée pour reconstruire les phases d’attaque influe à la fois sur la qualité des attaques reconstruites mais également sur celle des partiels via le déroulé.

## 6.6 Conclusion

Nous avons introduit un modèle de phase basé sur la répétition d’évènements audio qui exploite cette propriété au sein des trames d’attaque. Ce modèle a montré expérimentalement sa pertinence sur plusieurs types de données. Combiné au déroulé linéaire dans le cas aveugle, on constate que cette méthode permet une meilleure reconstruction du signal que l’utilisation d’une phase aléatoire. Nous l’avons également intégré à un modèle de mélange de sources dans les trames d’attaque pour la séparation de sources. De futures expériences pourraient être conduites sur des signaux de musique plus réalistes et complexes. Néanmoins celles-ci sont compliquées à réaliser car il faudrait pour cela disposer des amplitudes de chaque atome spectral (i.e. de chaque note), ce qui peut être compliqué à obtenir pour un scénario Oracle.

Au demeurant assez simple, ce modèle pourrait être affiné, en prenant en compte le fait qu’une partie du signal est tronquée d’une trame d’attaque à l’autre, et que la forme de la fenêtre d’analyse et son positionnement pourraient également être utilisés. Les premiers calculs que nous avons effectués sur ce sujet n’ont cependant pas abouti à la formulation d’un modèle analytiquement simple, aussi nous avons préféré inclure ce premier modèle dans le cadre de la séparation de sources afin d’évaluer son potentiel. Les résultats de séparation étant mitigés du fait de la dépendance vis-à-vis du paramètre  $\sigma$ , il pourrait être utile de calculer automatiquement un paramètre de relaxation optimal. Par ailleurs, outre la modélisation des

---

répétitions de phase, il pourrait être intéressant de modéliser les phases au sein des trames d'attaque afin d'exploiter les corrélations entre canaux fréquentiels [KIRCHHOFF et al. \(2014\)](#); [BADEAU et PLUMBLEY \(2014\)](#) et donc de structurer la phase de référence  $\psi$ .

Les amplitudes des sources étaient supposées connues, ce qui n'est pas le cas en pratique. Le chapitre 7 introduit cette contrainte de répétition dans un modèle de NMF complexe pour l'estimation conjointe des amplitudes et des phases des différentes sources.