Déroulé de phase et séparation de sources

Sommaire

Dans ce chapitre, nous intégrons la méthode de déroulé de phase dans le cadre de la séparation de sources. Nous considérons le problème de séparation de sources comme un problème d'optimisation, qui consiste à minimiser une fonction de coût traduisant l'écart entre les mélanges observé et estimé, sous contrainte que le module des sources estimées est fixé à une valeur objectif. Notre idée consiste alors à incorporer dans ce contexte une information à priori sur la phase, calculée par la méthode de déroulé linéaire.

Les aspects techniques de cette procédure itérative ont fait l'objet d'un rapport technique disponible en ligne Magron et al. (2016a). Un article reprenant certaines expériences du chapitre précédent, ainsi que celles présentées dans ce chapitre a été soumis à publication dans la revue *IEEE Transactions on Audio, Speech and Language Processing* Magron et al. (2017b).

Dans la section 5.1, nous proposons une formulation de ce problème de séparation de sources. Dans la section 5.2, nous obtenons donc un algorithme de séparation de sources. Nous évaluons expérimentalement cette technique dans la section 5.3. Nous proposons également une procédure contrainte alternative dans la section 5.4. Enfin, la section 5.5 présente quelques remarques conclusives.

5.1 Position du problème

On raisonne dans cette section (et dans la suivante) dans un point TF (f,t), et on s'affranchit des indices pour plus de clarté. Nous cherchons à obtenir un estimateur $\hat{X}_k$ des K composantes complexes X_k à partir de leur somme X, en supposant que l'on a une estimation de leurs modules V_1,...,V_K (qui peut être obtenue via une NMF préalable, par exemple). Ce problème peut être résolu en minimisant la fonction de coût :

$$|E| = |X - \sum_k \hat{X}_k|. \tag{5.1}$$

Cette fonction possède en général plusieurs zéros. Par exemple, si $K = 2$, considérons deux complexes $X_1 = V_1 e^{i\theta_1}$ et $X_2 = V_2 e^{i\theta_2}$ dont la somme vaut exactement $X = V e^{i\theta}$. Leurs symétriques par rapport à X sont $\bar{X}_1 e^{2\theta}$ et $\bar{X}_2 e^{2\theta}$, où $\bar{z}$ désigne le complexe conjugué de z. Leur somme est :

$$(\bar{X}_1 + \bar{X}_2)e^{2\theta} = \bar{X}e^{2\theta} = V e^{-\theta} e^{2\theta} = X. \tag{5.2}$$

Ainsi, à partir d'un couple de solutions, on peut en identifier aisément un autre, ce qui motive la recherche d'un "bon" minimum. En outre, lorsque $K \geq 3$, il y a une infinité de solutions à ce problème. La stratégie la plus répandue dans la littérature pour obtenir (de façon non-itérative) un zéro de $|E|$ consiste à appliquer un filtrage de type Wiener Liutkus et Badeau (2015). Les estimées sont :

$$\hat{X}_k = \frac{V_k^2}{\sum_l V_l^2} \odot X, \tag{5.3}$$

et l'erreur vaut alors :

$$|E| = |X - \sum_k \hat{X}_k| = |X| \left| 1 - \sum_k \frac{V_k^2}{\sum_l V_l^2} \right| = |X||1 - 1| = 0. \tag{5.4}$$

Cependant, les estimées par filtrage de Wiener ne sont pas des solutions du problème car $|\hat{X}_k| \neq V_k$, et ne conduisent pas toujours à des résultats satisfaisants du point de vue de

la qualité audio lorsque les sources se recouvrent dans le domaine TF (*cf.* chapitre 3). Nous proposons donc d'obtenir de nouvelles estimées des sources par une procédure itérative visant à minimiser $|E|$. Une initialisation par déroulé linéaire confère ainsi à l'estimée obtenue une propriété de régularité temporelle.

5.2 Procédure itérative d'estimation des composantes

Dans cette section, nous présentons l'algorithme que nous avons mis au point pour estimer les composantes complexes à partir de leur mélange.

5.2.1 Exemple dans le cas de deux nombres complexes

Afin d'estimer les phases de chaque source dans le mélange, illustrons notre technique avec le cas particulier de deux nombres complexes X_1 et X_2. Leurs modules V_1 et V_2 sont fixés. La somme X des deux est connue, et on cherche donc à estimer leur argument, ou, formulé différemment, on cherche ces deux complexes avec une contrainte sur leurs modules.

La procédure que nous utilisons est inspirée de Gunawan et Sen (2010), dont nous rappelons le principe. À l'itération (it), on a une estimation de chacun des complexes, $\hat{X}_1^{(it)}$ et $\hat{X}_2^{(it)}$. On peut donc calculer l'erreur entre la somme des deux estimées et le mélange X, soit $E^{(it)} = X - \hat{X}_1^{(it)} - \hat{X}_2^{(it)}$, erreur que l'on redistribue sur les composantes. On normalise ensuite le module de chaque estimée à la valeur objectif, puis on réitère jusqu'à convergence. Ainsi, on effectue à l'itération (it) les opérations suivantes :

1. Distribution de l'erreur : $Y_1^{(it)} = \hat{X}_1^{(it-1)} + \dfrac{E^{(it-1)}}{2}$ et $Y_2^{(it)} = \hat{X}_2^{(it-1)} + \dfrac{E^{(it-1)}}{2}$;
2. Normalisation à la valeur objectif : $\hat{X}_1^{(it)} = \dfrac{Y_1^{(it)}}{|Y_1^{(it)}|} V_1$ et $\hat{X}_2^{(it)} = \dfrac{Y_2^{(it)}}{|Y_2^{(it)}|} V_2$;
3. Calcul de l'erreur : $E^{(it)} = X - \hat{X}_1^{(it)} - \hat{X}_2^{(it)}$;
4. Retour à l'étape 1 jusqu'à convergence.

Cette méthode itérative est illustrée par la figure 5.1. Notons qu'une approche similaire a été utilisée dans Mowlaee et al. (2012) pour l'estimation des phases dans un mélange de deux sources uniquement.

5.2.2 Procédure itérative dans le cas général

Nous étendons cette méthode au cas de la somme de K sources composant un mélange. On a :

$$Y_k^{(it+1)} = \hat{X}_k^{(it)} + \lambda_k E^{(it)}, \tag{5.5}$$

ce qui permet d'ajuster la distribution de l'erreur sur la composante k via un poids positif λ_k. Nous souhaitons par ailleurs que les Y_k somment à X, ce qui revient à imposer la condition suivante :

$$\sum_k \lambda_k = 1. \tag{5.6}$$

Par ailleurs, il est souhaitable que le k-ième poids soit d'autant plus grand que la composante correspondante possède d'énergie : une composante de grande énergie est en effet majoritairement responsable de l'erreur de reconstruction. Ces deux conditions nous conduisent à

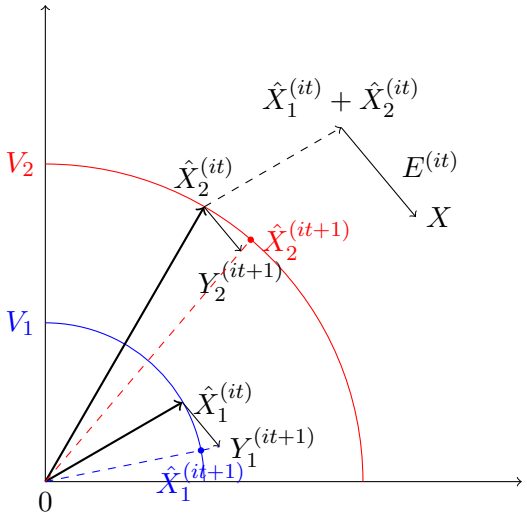

FIGURE 5.1 – Illustration de la procédure itérative consistant à estimer deux nombres complexes dont le module et la somme sont connus.

proposer la définition suivante pour les poids :

$$\lambda_k = \frac{V_k^2}{\sum_l V_l^2}, \tag{5.7}$$

qui correspond au gain de Wiener. D'autres définitions sont possibles, tant qu'elles respectent (5.6). Nous choisissons (5.7) car elle a donné de bons résultats expérimentaux. L'Algorithme 3 détaille cette procédure (dans le cas général où l'on peut utiliser n'importe quelle définition pour les poids λ_k).

5.2.3 Décroissance de la fonction de coût

Nous prouvons ci-après que cette procédure fait décroître la fonction de coût au cours des itérations. On a :

$$\begin{aligned}
\left|E^{(it+1)}\right| &= \left|X - \sum_k \hat{X}_k^{(it+1)}\right| \\
&= \left|\sum_k Y_k^{(it+1)} - \hat{X}_k^{(it+1)}\right| \\
&= \left|\sum_k \hat{X}_k^{(it)} + \lambda_k E^{(it)} - \frac{\hat{X}_k^{(it)} + \lambda_k E^{(it)}}{|\hat{X}_k^{(it)} + \lambda_k E^{(it)}|} V_k\right| \\
&= \left|\sum_k (\hat{X}_k^{(it)} + \lambda_k E^{(it)}) \left(1 - \frac{V_k}{|\hat{X}_k^{(it)} + \lambda_k E^{(it)}|}\right)\right|.
\end{aligned}$$

Algorithme 3 Estimation des sources complexes à partir de leur mélange.

Entrées :

Mélange $X \in \mathbb{C}$,

Modules $V_k \in \mathbb{R}_+$, poids λ_k et valeurs initiales $\hat{X}_k \in \mathbb{C}$, $\forall k \in [\![1, K]\!]$,

Nombre d'itérations N_{it}.

Erreur initiale : $E = X - \sum_k \hat{X}_k$.

pour $it = 1$ à N_{it} **faire**

  pour $k = 1$ à K **faire**

    $Y_k \leftarrow \hat{X}_k + \lambda_k E$,

    $\hat{X}_k \leftarrow \dfrac{Y_k}{|Y_k|} V_k$.

  fin pour

  $E \leftarrow X - \sum_k \hat{X}_k$.

fin pour

Sorties :

$\forall k \in [\![1, K]\!]$, $\hat{X}_k \in \mathbb{C}$.

En appliquant l'inégalité triangulaire, il vient :

$$\begin{aligned}\left|E^{(it+1)}\right| &\leq \sum_k |\hat{X}_k^{(it)} + \lambda_k E^{(it)}| \left|1 - \frac{V_k}{|\hat{X}_k^{(it)} + \lambda_k E^{(it)}|}\right| \\ &\leq \sum_k \left||\hat{X}_k^{(it)} + \lambda_k E^{(it)}| - V_k\right| \\ &\leq \sum_k \left||\hat{X}_k^{(it)} + \lambda_k E^{(it)}| - |\hat{X}_k^{(it)}|\right|.\end{aligned}$$

On utilise alors une propriété des nombres complexes (aisément démontrable en utilisant l'inégalité triangulaire) : $\forall (a, b) \in \mathbb{C}^2$,

$$||a| - |b|| \leq |a - b|. \tag{5.8}$$

En prenant $a = \hat{X}_k^{(it)} + \lambda_k E^{(it)}$ et $b = \hat{X}_k^{(it)}$, on obtient l'inégalité suivante :

$$\left||\hat{X}_k^{(it)} + \lambda_k E^{(it)}| - |\hat{X}_k^{(it)}|\right| \leq \left|\lambda_k E^{(it)}\right|. \tag{5.9}$$

En injectant cette inégalité dans le calcul précédent, on obtient :

$$\left|E^{(it+1)}\right| \leq \sum_k \left|\lambda_k E^{(it)}\right| \leq \left|E^{(it)}\right| \sum_k \lambda_k. \tag{5.10}$$

Or, les λ_k somment à l'unité, comme on l'a vu en (5.6), donc :

$$\left|E^{(it+1)}\right| \leq \left|E^{(it)}\right|, \tag{5.11}$$

ce qui montre la décroissance de la fonction de coût au cours des itérations (et donc sa convergence, vu qu'il s'agit d'une suite à termes positifs).

Remarque : Il est possible d'obtenir l'algorithme 3 en utilisant la méthode de la fonction auxiliaire, similairement à Kameoka et al. (2009). Cette méthode permet d'aboutir naturellement à la procédure itérative, avec une garantie de convergence. Nous avons préféré introduire cette procédure ici de façon intuitive, afin de ne pas surcharger ce chapitre de détails mathématiques. Ceux-ci sont fournis dans l'Annexe B de ce manuscrit.

5.2.4 Initialisation de l'algorithme

D'après les remarques de la section 5.1, le choix de l'initialisation de l'algorithme conditionne fortement la qualité des estimations ainsi que la vitesse de convergence.

Une idée consiste à initialiser les composantes en leur donnant la phase du mélange. Nous montrons dans l'annexe B que cela est un point fixe de la procédure : après une ou deux itérations (selon le choix des poids λ_k), les composantes ne sont plus modifiées, et sont alors égales aux composantes initiales à un déphasage de π près, ce qui n'est pas le but recherché.

Cela nous incite à initialiser l'algorithme avec notre technique de déroulé de phase plutôt qu'en donnant la phase du mélange aux composantes : on s'attend à ce que celle-ci soit assez proche d'un minimum local, et permette non seulement une convergence rapide de l'algorithme, mais également l'obtention d'une solution qui bénéficie de la propriété de régularité temporelle issue du modèle sinusoïdal.

5.2.5 Protocole de séparation de sources

Notre approche repose sur l'utilisation du déroulé linéaire de phase introduit au chapitre précédent. Le déroulé de phase étant récursif, il est initialisé aux trames d'attaque (*cf.* chapitre 4 section 4.1.5).

Nous devons donc faire certaines hypothèses sur les phases des sources dans ces trames. Tout d'abord, nous supposons connues leurs positions. En pratique, on utilise la boîte à outils MATLAB Tempogram Grosche et Müller (2011) sur chaque spectrogramme de source pour les déterminer. On obtient alors la fonction indicatrice des trames d'attaque :

$$\mathbb{1}_k(t) = \begin{cases} 1 & \text{si } t \in \Omega_k \\ 0 & \text{sinon,} \end{cases} \tag{5.12}$$

où Ω_k désigne l'ensemble des trames d'attaque de la source k. La fonction indicatrice du complémentaire est donc $\bar{\mathbb{1}}_k = 1 - \mathbb{1}_k$. Dans les expériences, les phases dans les trames d'attaque seront supposées connues, à l'exception de l'expérience décrite dans la section 5.3.4 où elles seront égales à la phase du mélange, pour une comparaison équitable avec les autres méthodes. Nous effectuerons néanmoins dans la section 5.3.5 une expérience visant à comparer ces deux cas de figure afin de déterminer le potentiel d'amélioration d'estimation des phases d'attaque.

Nous présentons dans l'Algorithme 4 la procédure complète de séparation de sources que nous proposons. Notons que dans cette procédure, l'optimisation est effectuée trame par trame, avant de procéder à l'initialisation de la trame suivante par déroulé linéaire. Nous verrons dans l'expérience 5.3.2 l'intérêt de cette approche par rapport à une application de la procédure d'optimisation directement sur toute la TFCT.

Algorithme 4 Estimation de sources complexes dans un mélange à partir de leurs spectrogrammes en utilisant le déroulé linéaire de phase

Entrées :

Mélange $X \in \mathbb{C}^{F \times T}$,

Spectrogrammes $V_k \in \mathbb{R}_+^{F \times T}$, $\forall k \in [\![1, K]\!]$,

Indicatrices des trames d'attaque $\mathbb{1}_k$, $\forall k \in [\![1, K]\!]$,

Phases d'attaque $\phi_k^o(f,t)$, $\forall t \in \Omega_k$,

Nombre d'itérations N_{it}.

Poids $\lambda_k = \dfrac{V_k^2}{\sum_l V_l^2}$.

pour $t = 1$ à $T-1$ **faire**

  pour $k = 1$ à K **faire**

    si $\mathbb{1}_k(t) = 1$ **alors**

      Phase d'attaque : $\forall f$, $\phi_k(f,t) = \phi_k^o(f,t)$.

    sinon

      Déroulé linéaire (cf. Algorithme 2).

    fin si

    $\forall f$, $\hat{X}_k^{(0)}(f,t) = V_k(f,t)e^{i\phi_k(f,t)}$.

  fin pour

  Calculer $\forall f : E^{(0)}(f,t) = X(f,t) - \sum_k \hat{X}_k^{(0)}(f,t)$.

  pour $it = 1$ à N_{it} **faire**

    Distribution de l'erreur $\forall k, f : Y_k^{(it)}(f,t) = \hat{X}_k^{(it-1)}(f,t) + \lambda_k(t)E^{(it-1)}(f,t)$.

    Normalisation $\forall k, f : \hat{X}_k^{(it)}(f,t) = \dfrac{Y_k^{(it)}(f,t)}{|Y_k^{(it)}(f,t)|}V_k(f,t)$.

    Erreur $\forall f : E^{(it)}(f,t) = X(f,t) - \sum_k \hat{X}_k^{(it)}(f,t)$.

  fin pour

  Composante complexe $\forall k, f : \hat{X}_k(f,t) = \hat{X}_k^{(N_{it})}(f,t)$.

fin pour

Sortie :

$\forall k \in [\![1, K]\!]$, $\hat{X}_k \in \mathbb{C}^{F \times T}$.

5.3 Résultats expérimentaux

5.3.1 Protocole

Les signaux sont échantillonnées à 44100 Hz et la TFCT est calculée avec une fenêtre de Hann de longueur 4096 échantillons (soit 92 ms), 75 % de recouvrement et pas de bourrage de zéros.

Les données utilisées proviennent (à l'exception de certains mélanges de notes de piano issues de la base MAPS Emiya et al. (2010)) de la base DSD100 Ono et al. (2015), pour lesquels nous disposons des sources séparées :

— `bass` : il s'agit de la partie de basse électrique ;

— `drums` : il s'agit des percussions (batterie notamment) ;

— `vocals` : il s'agit de la voix chantée ;

— `other` : ce sont tous les autres instruments d'accompagnement (guitare, piano, sons électroniques etc.).

On peut calculer le spectrogramme de chaque source isolément. Si on utilise ces valeurs dans les différents algorithmes, on parlera de scénario Oracle. Alternativement, dans l'expérience décrite à la section 5.3.4, on considérera des spectrogrammes d'amplitude estimés, afin de tester l'efficacité des différentes méthodes dans un cadre plus réalistes où les V_k ne sont plus parfaitement connus. Ces estimés sont obtenus par une KLNMF effectuée sur chaque spectrogramme de source isolée. Chaque KLNMF utilise 50 itérations de règles de mises à jour multiplicatives et un rang de factorisation égal à 10. Notons que ce n'est pas un scénario "aveugle", puisqu'on estime les spectrogrammes sur chaque source séparée, mais il nous informe néanmoins sur la performance des méthodes lorsque les spectrogrammes ne sont plus égaux à la vérité terrain.

À partir de ces spectrogrammes, on applique les méthodes d'estimation des composantes complexes suivantes :

— Le filtrage de Wiener Févotte et al. (2009), noté **Wiener** ;

— Le filtrage de Wiener consistant Le Roux et Vincent (2013), noté **W-Cons** ;

— Le déroulé horizontal, effectué sur chaque source séparément, sans tenir compte de la phase du mélange, noté **Unwrap** ;

— L'algorithme 3 d'estimation des composantes initialisé par la méthode de déroulé, ce qui correspond donc à l'algorithme 4, noté **Iter**.

La qualité de la séparation de source est mesurée par les SDR, SIR et SAR, calculés par la boîte à outils BSS Eval Vincent et al. (2006).

Le filtrage de Wiener consistant (introduit dans le chapitre 2, section 2.1.4) dépend d'un paramètre γ qui ajuste l'importance relative du filtrage de Wiener et de la contrainte de consistance. On apprend le paramètre γ optimal (au sens du maximum de SDR, SIR et SAR) sur la base de développement (50 morceaux issus de la base DSD100 différents de ceux utilisés pour les tests). L'influence du paramètre γ sur la qualité de la séparation est illustrée sur la figure 5.2. On choisit la valeur $\gamma = 4$ pour l'expérience de séparation de sources qui correspond à une valeur optimale pour l'utilisation du filtrage de Wiener consistant sur ce jeu de données, aussi bien pour le scénario Oracle que pour le cas non-Oracle.

5.3.2 Impact du caractère séquentiel de l'algorithme

Nous proposons tout d'abord une expérience pour justifier l'intérêt de la procédure telle que décrite dans l'Algorithme 4. En effet, dans cette procédure, on agit séquentiellement

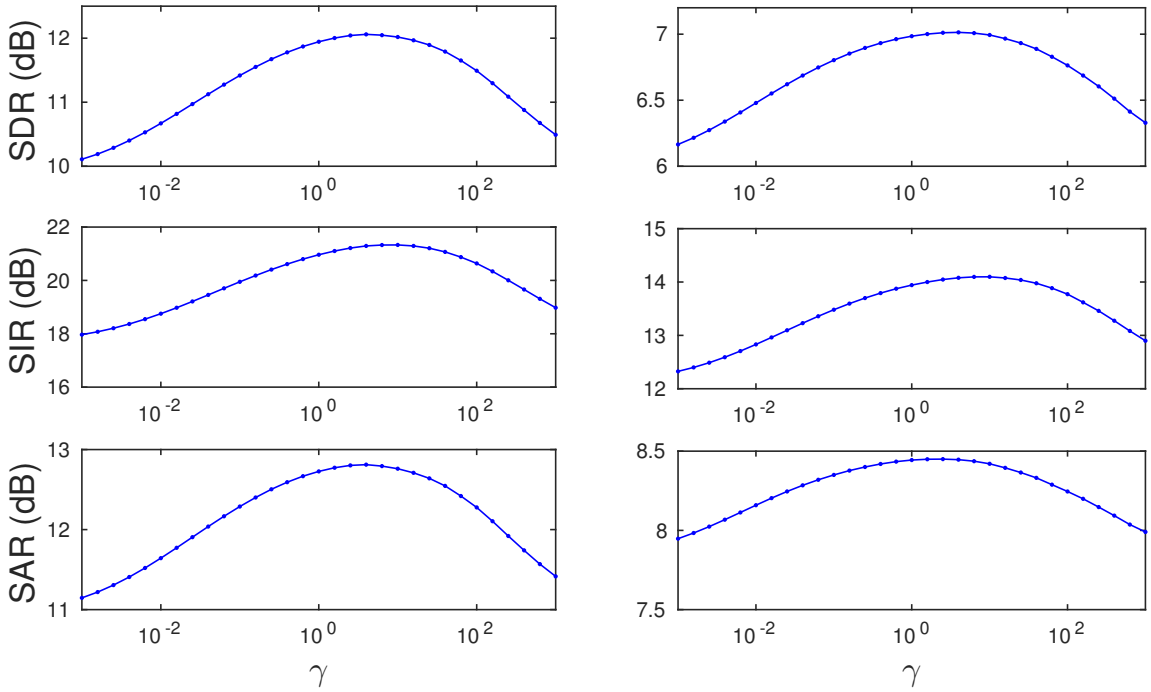

FIGURE 5.2 – Influence du paramètre de concentration γ sur la qualité de la séparation de sources pour le filtrage de Wiener consistant sur la base d'apprentissage de DSD100, dans le cas Oracle (à gauche) et non-Oracle (à droite).

	SDR	SIR	SAR	Temps (s)
Séquentiel	**14.0**	**27.0**	**14.2**	3.8
Direct	12.7	24.7	13.0	**3.6**

TABLEAU 5.1 – Performance de la séparation de sources sur la base DSD100 (SDR, SIR et SAR en dB) pour différentes techniques d'utilisation de la procédure itérative.

sur les trames : ainsi, l'initialisation de l'Algorithme 3 dans une trame donnée est faite par déroulé linéaire à partir de la phase obtenue après application de l'Algorithme 3 dans la trame précédente. On pourrait penser qu'effectuer un déroulé sur toute la TFCT, puis appliquer l'Algorithme 3 matriciellement serait moins gourmand en temps de calcul.

On considère donc 10 morceaux de musique issus de la base DSD100 (extraits de 10 secondes) et on applique deux méthodes de reconstruction des composantes complexes : une méthode *séquentielle*, c'est-à-dire telle que présentée dans l'Algorithme 4, et une méthode *directe*, c'est-à-dire en utilisant un déroulé complet (Algorithme 2) puis une application de l'Algorithme 3 matriciellement. Dans les deux cas, les phases dans les trames d'attaque ainsi que les amplitudes des sources sont supposées connues, et la procédure itérative utilise 10 itérations. Les résultats sont présentés dans le tableau 5.1.

La méthode séquentielle fournit des résultats supérieurs à la méthode directe. En effet, il est préférable d'estimer convenablement les phases dans une trame donnée avant de procéder au déroulé pour initialiser la trame suivante : la procédure itérative bénéficie alors d'une meilleure initialisation, ce qui conduit à de meilleurs résultats. Bien que la méthode directe soit très légèrement plus rapide, la perte de qualité nous incite à conserver la méthode telle que décrite dans l'Algorithme 4 pour la suite des expériences.

Initialisation	SDR	SIR	SAR
Aléatoire	10.4	20.6	10.9
Déroulé	**14.0**	**27.0**	**14.2**

TABLEAU 5.2 – Performance de la séparation de sources sur la base DSD100 (SDR, SIR et SAR en dB) pour différentes initialisations de l'algorithme 3.

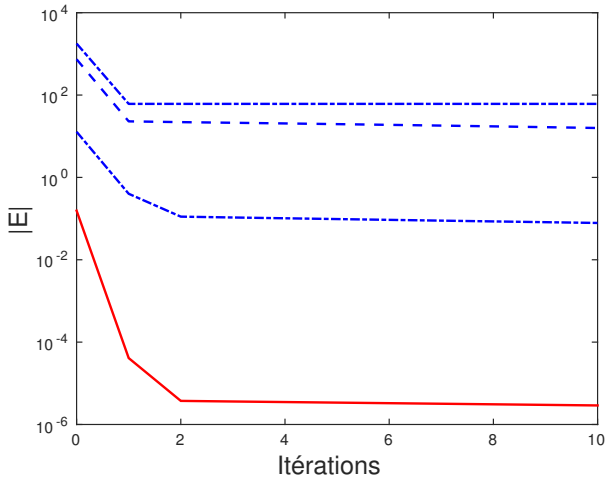

FIGURE 5.3 – Comparaison de l'erreur $|E|$ calculée au cours des itérations au niveau d'un point TF où deux sources (notes de piano C4 et G4) se recouvrent : les courbes en pointillés correspondent à une initialisation aléatoire (pour 30 initialisations différentes, les 3 courbes sont donc le maximum, le minimum et la valeur moyenne de l'erreur), et la courbe en traits pleins correspond à une initialisation par déroulé linéaire.

5.3.3 Influence de l'initialisation

Nous montrons dans cette expérience l'intérêt de l'initialisation par déroulé linéaire dans l'Algorithme 3. On considère 10 morceaux extraits de la base DSD100. Les phases d'attaque sont supposées connues et les phases des partiels sont estimées par 10 itérations de l'algorithme 4 à partir des amplitudes dans le scénario Oracle. Les phases des composantes complexes peuvent être initialisées soit avec des valeurs aléatoires, soit par déroulé linéaire. Les résultats de la séparation de sources sont fournis dans le tableau 5.2.

Initialiser l'algorithme par déroulé linéaire améliore significativement les résultats (gain d'environ 3.5 dB en SDR et SAR, et d'environ 6.5 dB en SIR) par rapport à une initialisation aléatoire.

Pour illustrer ce résultat, on effectue la séparation sur un mélange de notes de piano issues de la base MAPS, C4 et G4. On trace sur la figure 5.3 l'erreur au cours des itérations avec ces deux approches en un point TF où il y a recouvrement. Avec notre approche, l'erreur converge vers une valeur nettement plus basse.

Enfin, afin d'illustrer la différence entre les deux approches après application de l'algorithme, on trace sur la figure 5.4 les parties réelles des signaux originaux et reconstruits par notre algorithme avec différentes initialisations. On constate que l'utilisation du déroulé linéaire conduit à une reconstruction quasi-parfaite du partiel.

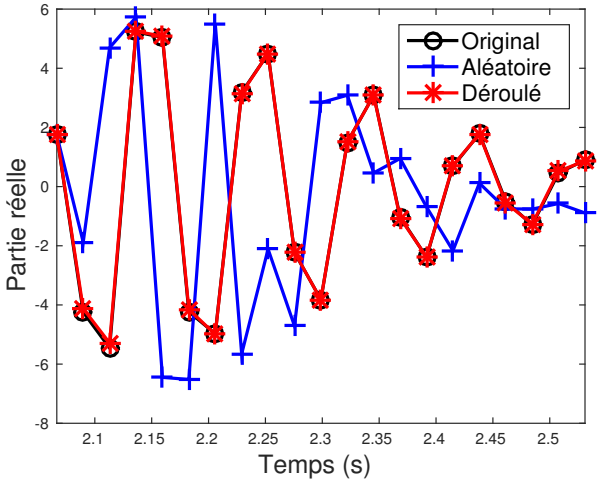

FIGURE 5.4 – Reconstruction d'un partiel de piano (partie réelle) de la note C4 dans la bande de fréquences à 796 Hz, sur une fenêtre de temps où les deux sources (C4 et G4) se recouvrent par application de notre algorithme avec différentes initialisations.

5.3.4 Test sur la base DSD100

On considère dans cette expérience 50 morceaux issus de la base DSD100 (base de test) et on applique les différentes méthodes de séparation de sources présentées dans la section 5.3.1. L'algorithme d'estimation des phases à partir du mélange utilise $N_{it} = 10$ itérations. Les résultats sont illustrés sur la figure 5.5.

Nous constatons que la méthode basée sur l'algorithme 4 conduit à une amélioration des performances par rapport au filtrage de Wiener, notamment en SIR, et à la méthode de déroulé linéaire appliqué isolément sur les sources. Cette dernière mène à des performances moindres du fait des erreurs dues à l'estimation de fréquence qui sont propagées à travers les trames.

L'algorithme que nous proposons mène à des résultats similaires au filtrage de Wiener consistant. Les différences entre ces deux méthodes ne sont pas statistiquement significatives en matière de SDR et SAR, mais notre approche conduit à une légère augmentation de SIR. Dans le cas non-Oracle, notre approche donne des résultats moins bons que le filtrage de Wiener consistant en SDR et SAR, et des résultats similaires pour le rejet d'interférences. Ce résultat est à relativiser car notre méthode présente deux avantages : tout d'abord, son temps de calcul est nettement moins élevé (d'un rapport 7). Par ailleurs, les résultats présentés ici avec le filtrage de Wiener consistant sont optimaux car on a utilisé un paramètre γ appris au préalable sur une base de développement. Notre méthode n'utilise pas de tel paramètre. Les résultats du filtrage de Wiener consistant sont très dépendant du choix de ce paramètre, et celui-ci peut varier significativement selon la base de données utilisée (pour les données de l'article originel LE ROUX et VINCENT (2013), la valeur optimale est $\gamma = 10^6$ contre 4 ici).

On illustre ce résultat sur un mélange simple, constitué de deux notes de piano (C4 et G4) qui se recouvrent dans le plan TF. On sépare les sources par différentes méthodes et on trace la partie réelle de la première source (C4) estimée dans une bande de fréquences et sur un intervalle de temps où le recouvrement est observé. Le résultat présenté sur la figure 5.6 montre une reconstruction du partiel quasi-parfaite avec la méthode basée sur l'algorithme 4. Le filtrage de Wiener conduit quant à lui à utiliser la phase du mélange dans laquelle les battements sont marqués, ce qui mène à une nette dégradation du signal reconstruit. Enfin, la technique de déroulé linéaire seule n'est pas parfaite mais est néanmoins relativement proche

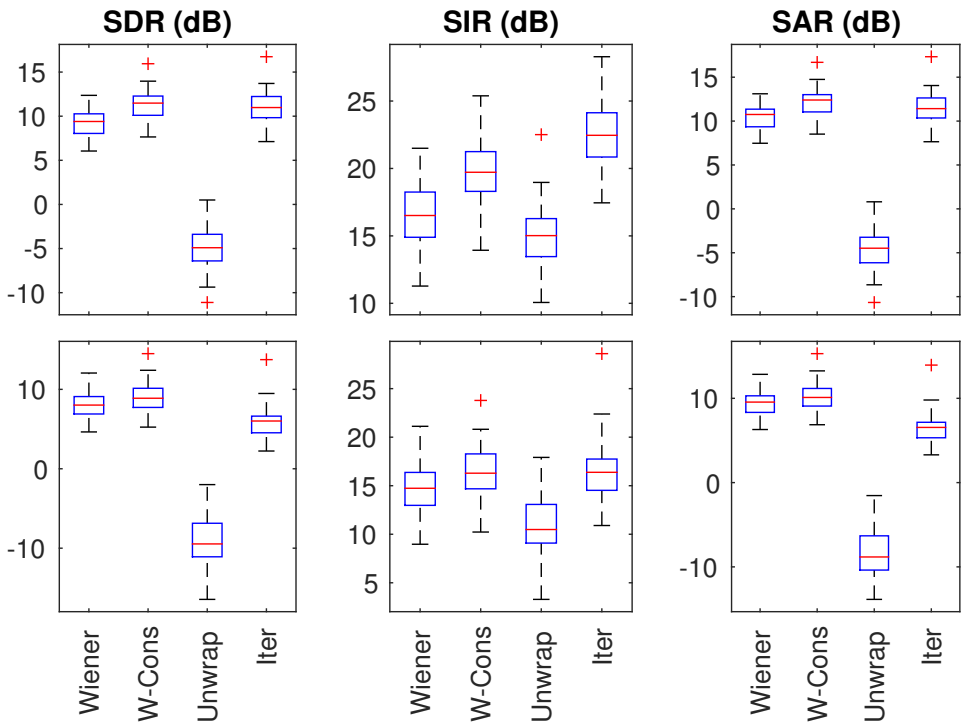

FIGURE 5.5 – Performance de la séparation de sources (SDR, SIR et SAR en dB) sur des morceaux de musiques de la base DSD100. Amplitudes Oracle (en haut) et estimées (en bas).

du résultat original (en comparaison avec le filtrage de Wiener), et permet notamment de s'affranchir des battements. Le filtrage de Wiener consistant mène à un résultat similaire au filtrage de Wiener.

Une évaluation subjective informelle de notre part montre que ces conclusions sont également constatées perceptivement. Le filtrage de Wiener crée des interférences entre sources à cause du recouvrement TF. Le déroulé seul élimine ces interférences, mais au prix de la création d'artéfacts qui dégradent la qualité globale des signaux. L'approche utilisant le déroulé linéaire ainsi que la phase du mélange s'affranchit de ces problèmes et conduit à des signaux proches de la reconstruction parfaite, dans lesquels ni interférences ni artéfacts ne sont audibles dans le cas Oracle.

5.3.5 Potentiel d'amélioration dans les trames d'attaque

Les phases dans les trames d'attaque étaient dans les expériences précédentes obtenues en prenant la phase du mélange ou en les supposant connues. Nous proposons donc d'appliquer l'algorithme 3 dans ces deux cas. La comparaison entre ces deux approches permettra une estimation de l'impact de la phase d'attaque sur la reconstruction du reste des phases, et pourra nous renseigner sur le potentiel d'amélioration d'estimation des phases d'attaque. Les amplitudes sont ici supposées connues.

Les résultats présentés dans le tableau 5.3 montrent qu'en supposant une connaissance parfaite des phases dans les trames d'attaque, on peut améliorer les résultats d'environ 2 dB en SDR et SAR et de 3 dB en SIR. Il existe donc une marge de progression possible pour

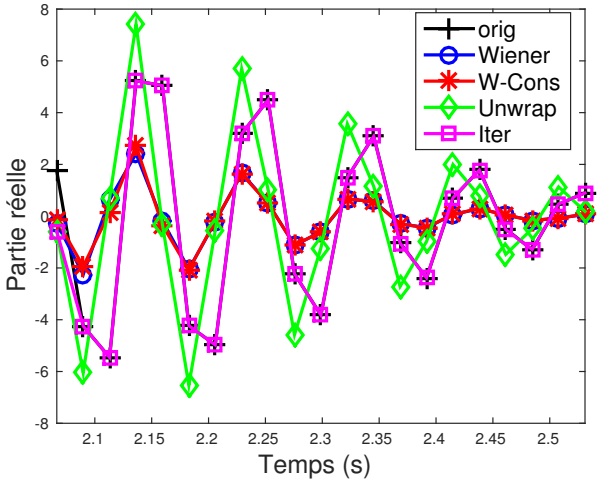

FIGURE 5.6 – Reconstruction d'un partiel de piano (partie réelle) de la note C4 dans le canal fréquentiel à 796 Hz, sur une fenêtre de temps où les deux sources (C4 et G4) se recouvrent. Plusieurs algorithmes de reconstruction des partiels sont comparés dans le cas Oracle.

Phases d'attaque	SDR	SIR	SAR
Mélange	11.2	22.3	11.7
Oracle	**13.2**	**25.4**	**13.6**

TABLEAU 5.3 – Comparaison des performances de séparation de sources selon la technique utilisée pour les phases d'attaque.

dépasser les résultats obtenus en utilisant la phase du mélange au niveau des attaques. Cela traduit l'importance de ces phases d'attaque, et montre qu'il peut être intéressant de travailler à les reconstruire plus proprement.

5.4 Algorithme contraint par le déroulé de phase

5.4.1 Principe

L'approche présentée dans ce chapitre consiste à minimiser une fonction de coût $|E|$ qui mesure l'écart entre le mélange et la somme des sources estimées. L'information à priori sur la phase est introduite par le biais de l'initialisation. Dans cette section, nous proposons d'ajouter à cette fonction de coût une contrainte sur la phase, pour que celle-ci reste relativement "proche" du candidat que nous proposons (la composante obtenue par déroulé linéaire). C'est ce type d'approche qui est proposé dans les algorithmes de NMF complexes à phase contrainte, comme LE ROUX et al. (2009) ou BRONSON et DEPALLE (2014).

Cette approche est posée comme un problème d'optimisation, qui consiste à minimiser la fonction :

$$\tilde{\mathcal{C}}(\theta) = |X - \sum_k \hat{X}_k|^2 + \sigma \sum_k |\hat{X}_k - \tilde{X}_k|^2 \tag{5.13}$$

sous la contrainte $|\hat{X}_k| = V_k$. Cette fonction objectif comporte un terme d'attache aux données $|E|^2$ et un terme d'attache à l'à priori, l'importance relative des deux étant réglée par le paramètre σ. L'à priori est $\tilde{X}_k$, dont la phase est obtenue par déroulé linéaire.

Pour minimiser cette fonction, on applique la méthode de la fonction auxiliaire de la même façon que cela a été fait pour l'algorithme 3 (pour rappel, les détails de ce calcul sont présentés

Algorithme 5 Estimation des sources complexes à partir de leur mélange, avec contrainte sur la phase.

Entrées :

Mélange $X \in \mathbb{C}$,

Modules $V_k \in \mathbb{R}_+$, poids λ_k et valeurs initiales $\hat{X}_k \in \mathbb{C}$, $\forall k \in [\![1, K]\!]$,

Candidat $\tilde{X}_k$, $\forall k \in [\![1, K]\!]$ et paramètre $\sigma \geq 0$,

Nombre d'itérations N_{it}.

Erreur initiale : $E = X - \sum_k \hat{X}_k$.

pour $it = 1$ à N_{it} **faire**

  pour $k = 1$ à K **faire**

    $Y_k \leftarrow \hat{X}_k + \lambda_k E$,

    $\hat{X}_k \leftarrow \dfrac{Y_k + \sigma\lambda_k \tilde{X}_k}{|Y_k + \sigma\lambda_k \tilde{X}_k|} V_k$.

  fin pour

  $E \leftarrow X - \sum_k \hat{X}_k$.

fin pour

Sorties :

$\forall k \in [\![1, K]\!]$, $\hat{X}_k \in \mathbb{C}$.

dans l'annexe B). La fonction auxiliaire est obtenue en utilisant l'inégalité de Jensen pour la fonction convexe $|.|^2$ ce qui requiert d'introduire des poids λ_k. Les mises à jour des variables auxiliaires Y_k sont inchangées, puisque le terme supplémentaire dans (5.13) ne dépend pas de ces variables. Enfin, la mise à jour des $\hat{X}_k$ ne se fait plus dans la direction des Y_k mais des $Y_k + \sigma\lambda_k \tilde{X}_k$.

Les règles de mise à jour pour l'estimation de ce modèle sont résumées dans l'Algorithme 5.

Les algorithmes 3 et 5 sont similaires, mais dans le 5, les composantes sont contraintes à rester proche du candidat $\tilde{X}_k$.

5.4.2 Résultats expérimentaux

Nous comparons ici expérimentalement les performances méthodes de séparation basées sur les algorithmes 3 et 5. On considère 10 morceaux extraits de la base DSD100, et on fait varier le paramètre σ. L'algorithme 5 peut être initialisé :

— Soit avec des composantes à phases aléatoires ;

— Soit avec des composantes dont les phases sont obtenues par déroulé linéaire.

On tire un certain nombre de conclusions à partir des résultats présentés sur la figure 5.7 :

— Si on initialise l'algorithme par déroulé linéaire, augmenter la valeur de σ à partir de 0 n'améliore pas les résultats. En d'autres termes, la performance maximale est obtenue pour une contrainte de phase nulle. Comme l'algorithme est initialisé convenablement, mieux vaut favoriser le terme d'attache aux données plutôt que forcer les solutions à rester trop proches de cette valeur initiale.

— Lorsque le poids σ est grand, on obtient avec cet algorithme (pour les deux initialisations) le même résultat qu'avec la méthode **Unwrap**. Le terme dans la fonction de coût (5.13) qui mesure l'écart entre modèle et mélange devient alors négligeable : les estimées des sources sont alors simplement données par les $\tilde{X}_k$.

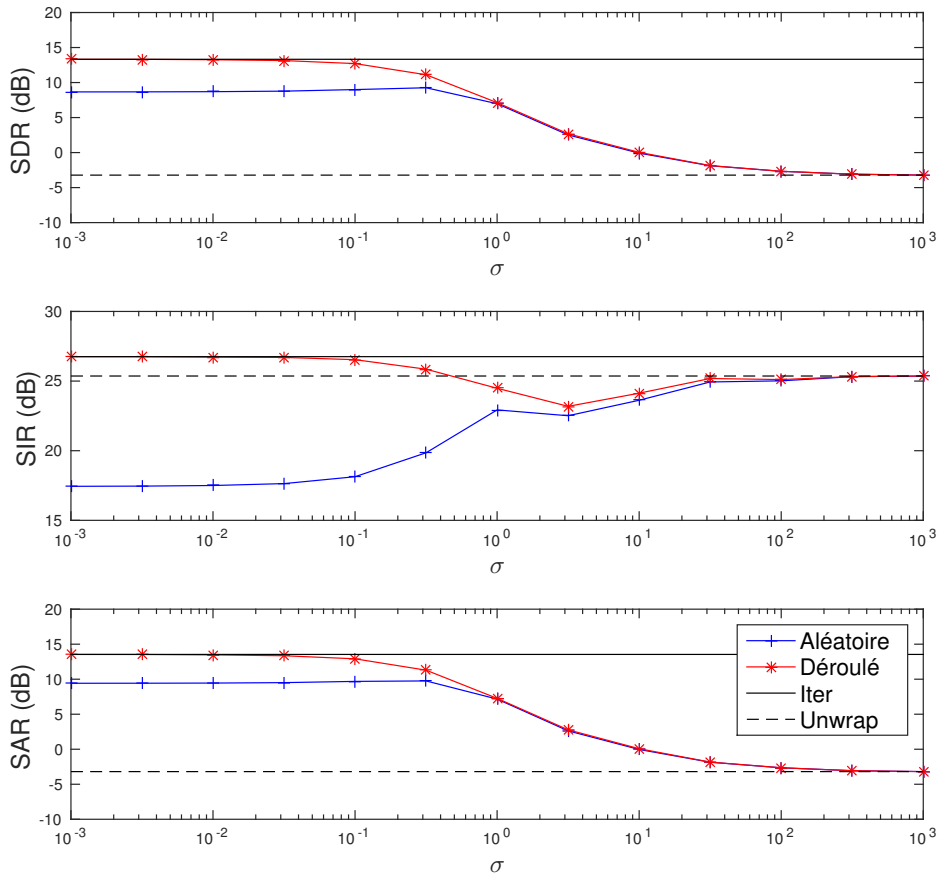

FIGURE 5.7 – Séparation de sources (SDR, SIR et SAR en dB) pour différentes initialisations (Aléatoire ou par Déroulé) de l'algorithme 5 sur le jeu de données E, et comparaison avec les approches **Iter** et **Unwrap**.

— Avec une initialisation aléatoire, on constate qu'en augmentant la valeur de σ, les indicateurs augmentent jusqu'à un certain seuil à partir duquel ils redescendent (pour le SDR et le SAR), ou alors continuent globalement d'augmenter (pour le SIR). Ceci peut s'expliquer par le fait qu'une initialisation aléatoire ne donne pas de très bons résultats pour l'algorithme non contraint, et que forcer les composantes estimées à être proches des $\tilde{X}_k$ augmentent nécessairement la qualité des résultats. Néanmoins, au-delà d'une certaine valeur de σ, la contrainte devient trop importante par rapport à l'erreur de reconstruction, ce qui se traduit par une chute des indicateurs SDR et SAR, même si le SIR continue de croître (ce qui s'interprète aisément, compte-tenu du fait qu'en négligeant le terme d'attache aux données, on ne tient plus compte du mélange, ce qui implique une suppression des interférences).

— L'initialisation avec déroulé linéaire, couplée à une contrainte nulle, correspond à la méthode **Iter** et donne donc les mêmes résultats. À partir de cette valeur, une augmentation de σ conduit à une augmentation des SDR et SAR qui reste négligeable (non visible sur la figure).

En fin de compte, on peut constater que la borne supérieure de résultats pour l'algorithme avec contrainte est donnée par la méthode **Iter**, valeur atteinte lorsqu'on initialise l'algorithme

avec le déroulé linéaire et que l'on fixe une contrainte nulle. En d'autres termes, cela revient à utiliser directement la méthode **Iter** : contraindre les phases n'apporte pas d'amélioration significative par rapport à une initialisation bien choisie.

5.5 Conclusion

Les expériences présentées dans ce chapitre ont montré l'intérêt de la technique de déroulé linéaire de phase pour la séparation de sources. Nous avons introduit une procédure itérative pour résoudre ce problème, qui permet, via son initialisation, d'introduire un à priori sur la phase. Cette méthode a été expérimentalement testée sur une base de musiques réalistes et nous avons obtenu des résultats comparables au filtrage de Wiener consistant avec un net bénéfice en temps de calcul.

La phase au niveau des trames d'attaque était dans ces expériences soit supposée connue, soit égale à la phase du mélange. Néanmoins, l'écart entre les deux identifié dans la section 5.3.5 suggère qu'il est possible d'améliorer les performances en travaillant à la reconstruction de phases d'attaque, ce qui fera l'objet du chapitre 6.

En outre, la procédure introduite ici consiste à fixer à chaque itération les amplitudes à une valeur objectif, qui n'est pas forcément réaliste. Il pourrait donc être préférable de mettre au point un modèle dans lequel on s'accorde une certaine incertitude sur les amplitudes, et dans lequel le mélange estimé est bien égal au mélange observé : cela fera l'objet du chapitre 8. Enfin, il sera intéressant de mettre en place un cadre de séparation plus réaliste dans lequel les amplitudes et les phases seront estimées conjointement à partir du mélange : c'est ce que nous proposons au chapitre 7.