Université de Toulouse

Master 2 SIA2

Compte rendu

TP1 : Mise en œuvre de l'algorithme FastICA pour la séparation aveugle de sources

- Module : *Apprentissage automatique et séparation de sources*
- Réalisé par :
  - *KABOU Abdeldjalil*

2025 / 2026

Sujet 1 : Tests Mise en œuvre de l'algorithme FastICA pour la séparation aveugle de sources

I. Génération d'un mélange artificiel de deux sources :

1) **Question 1 :** *elle est donnée dans le code Matlab*

Commentaire : On fixe la valeur du générateur aléatoire (seed = 1) pour obtenir toujours les mêmes résultats. Ensuite, on crée deux signaux aléatoires s1 et s2 contenant chacun 2000 valeurs.

2) **Question 2 :** *elle est donnée dans le code Matlab*

Commentaire : On centre les signaux en retirant leur moyenne afin d'obtenir une moyenne nulle. Ensuite, on les normalise pour que leur puissance soit égale à 1. Les résultats montrent que les moyennes sont proches de zéro et que les puissances valent 1.

3) **Question 3 :** *elle est donnée dans le code Matlab*

Commentaire : Le coefficient de corrélation entre s1 et s2 est proche de zéro (ρ = −0.036). Cela montre qu'il n'y a pas de relation linéaire entre les deux sources. Comme elles ont été générées indépendamment, on peut les considérer comme indépendantes.

4) **Question 4 :** *elle est donnée dans le code Matlab*

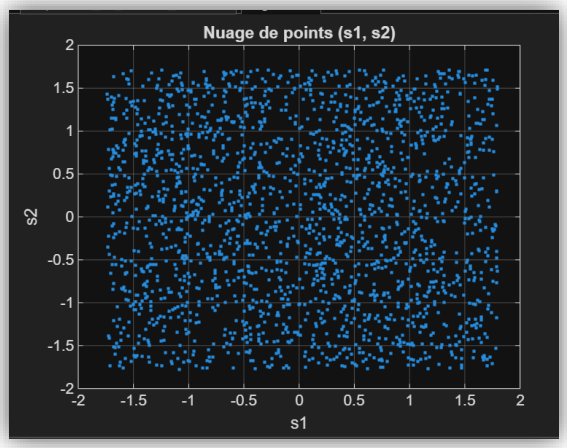


Figure 1 : Représentation du nuage de points bidimensionnel (s1 et s2)

Commentaire : Le nuage de points ne présente aucune structure particulière. Si l'on connaît s1 = 1.5, cela ne permet pas de prédire une valeur précise de s2, qui peut prendre différentes valeurs. Comme les deux signaux ont été générés séparément et que la corrélation est proche de zéro, on peut les considérer comme indépendants.

5) **Question 5 :** *elle est donnée dans le code Matlab*

Commentaire : Les deux sources sont mélangées à l'aide de la matrice A. On obtient ainsi deux observations x1 et x2 qui sont des combinaisons linéaires des sources originales.

6) **Question 6 :** *elle est donnée dans le code Matlab*

Conclusion : Le coefficient de corrélation entre x1 et x2 vaut 0.9452, ce qui indique une forte dépendance linéaire. Le nuage de points présente une structure allongée. On conclut que les observations x1 et x2 ne sont pas indépendantes.

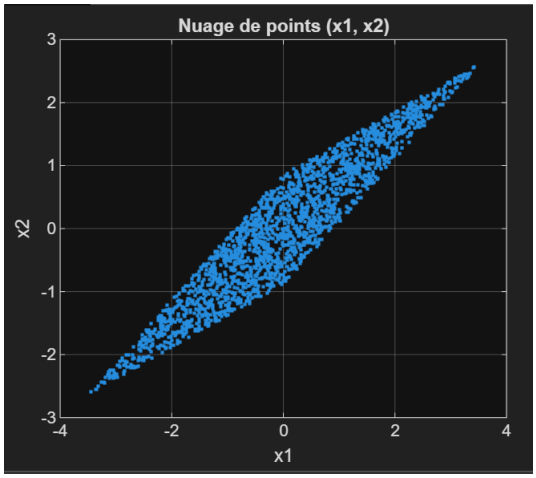


Figure 2 : Représentation du nuage de points bidimensionnel (x1 et x2)

II. Blanchiment :

1) **Question 1 :** elle est donnée dans le code Matlab
2) **Question 2 :** elle est donnée dans le code Matlab

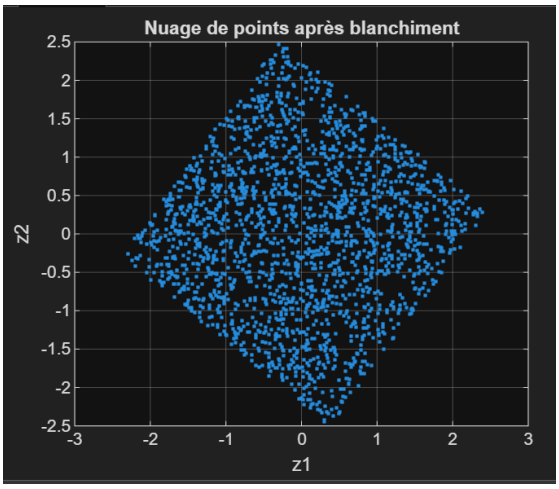


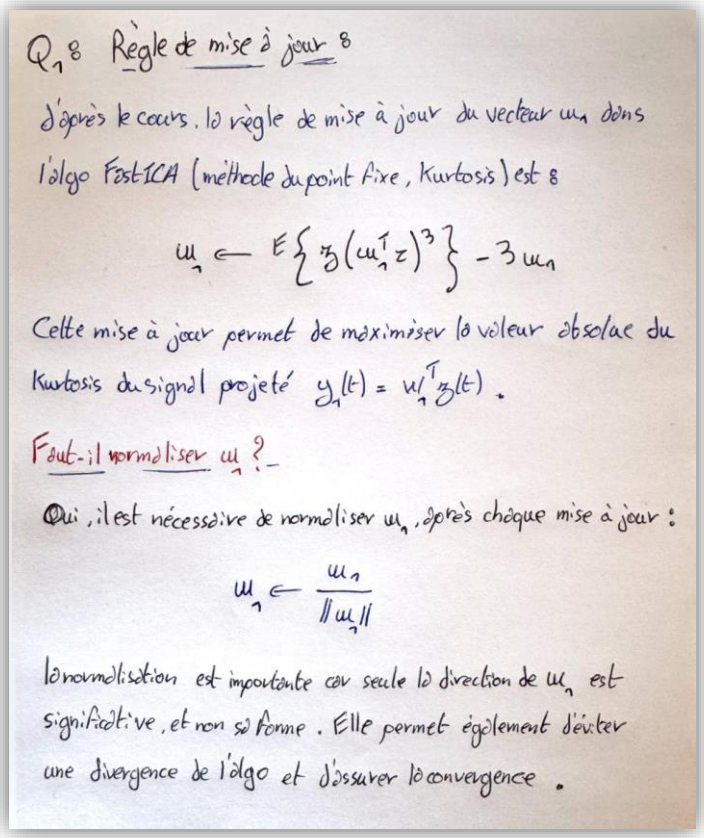
Figure 3 : Représentation du nuage de points après blanchiment

Conclusion :

Après blanchiment, la corrélation entre z_1 et z_2 est proche de zéro : les signaux sont donc non corrélés. Les variances sont proches de 1, ce qui confirme la normalisation. En revanche, ils ne sont pas forcément indépendants, car le blanchiment enlève seulement la corrélation.

III. Estimation des sources :

1) **<u>Question</u> 1 :**



Q1.8 Règle de mise à jour :

d'après le cours, la règle de mise à jour du vecteur w_1 dans l'algo FastICA (méthode du point fixe, kurtosis) est :

$$w_1 \leftarrow E\{z(w_1^T z)^3\} - 3w_1$$

Cette mise à jour permet de maximiser la valeur absolue du kurtosis du signal projeté $y_1(t) = w_1^T z(t)$.

Faut-il normaliser w_1 ?

Oui, il est nécessaire de normaliser w_1, après chaque mise à jour :

$$w_1 \leftarrow \frac{w_1}{\|w_1\|}$$

la normalisation est importante car seule la direction de w_1 est significative, et non sa forme. Elle permet également d'éviter une divergence de l'algo et d'assurer la convergence.

2) **<u>Question</u> 2 :** elle est donnée dans le code MATLAB

Commentaire : On initialise w1 aléatoirement avec randn (seed=1). On applique ensuite une mise à jour itérative de type point fixe. À chaque itération, on calcule $y_1(t)$ = w1^T * z(t), on mesure |kurtosis(y_1)| et on normalise w_1.

3) **<u>Question</u> 3 :** elle est donnée dans le code MATLAB

Conclusion :

La valeur absolue du kurtosis augmente très rapidement au début, puis devient presque constante. On considère que l'algorithme a convergé après environ 3 itérations, car la courbe se stabilise.

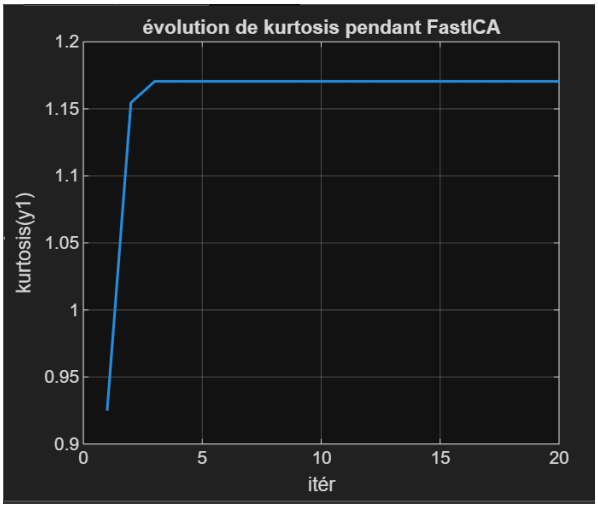


Figure 4 : Représentation de l'évolution de la valeur absolue du kurtosis

4) **<u>Question</u> 4 :** elle est donnée dans le code MATLAB

Conclusion :

Le signal estimé $y_1(t)$ correspond à la source $s_1(t)$. La corrélation entre y_1 et s_1 est très proche de 1 (≈ 0.9999), tandis que la corrélation avec s_2 est proche de zéro.

Visuellement, la différence entre les signaux n'est pas toujours clairement visible sur la figure, car les courbes sont très denses et se superposent. C'est pourquoi une vérification numérique par le calcul de la corrélation a été effectuée afin de confirmer la correspondance entre y_1 et s_1.

On conclut que FastICA a correctement extrait une des deux sources.

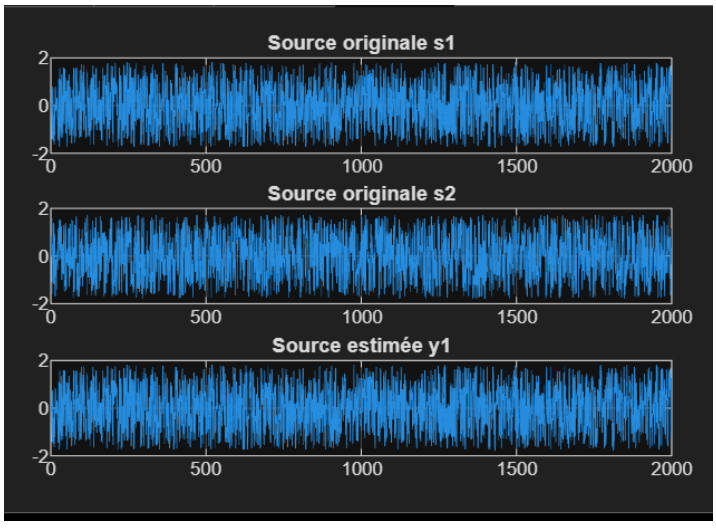
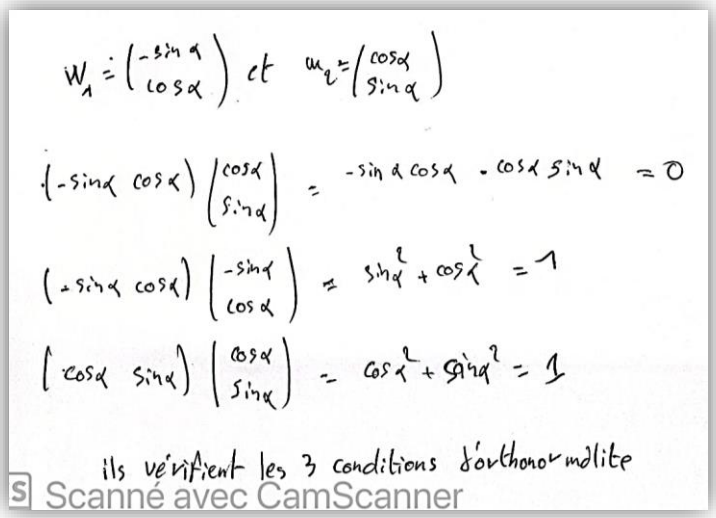


Figure 5 : Représentation qui montre la comparaison entre s1, s2 et y1

5) **<u>Question</u> 5 :**

Dans l'espace des signaux blanchis, la matrice de covariance est l'identité. La séparation des sources revient alors à effectuer une rotation. Or une rotation conserve les normes et impose des directions perpendiculaires. Ainsi, les vecteurs w_1 et w_2 sont orthogonaux et de norme unitaire : ils sont donc orthonormaux.

6) **Question 6 :**



7) **Question 7 :** elle est donnée dans le code MATLAB

Conclusion :

La corrélation entre y_2 et s_2 est très proche de ±1 (≈ −0.9988), tandis que la corrélation avec s_1 est proche de zéro. On en déduit que $y_2(t)$ correspond à la source $s_2(t)$, à un signe près.

L'algorithme FastICA a donc correctement séparé les deux sources.

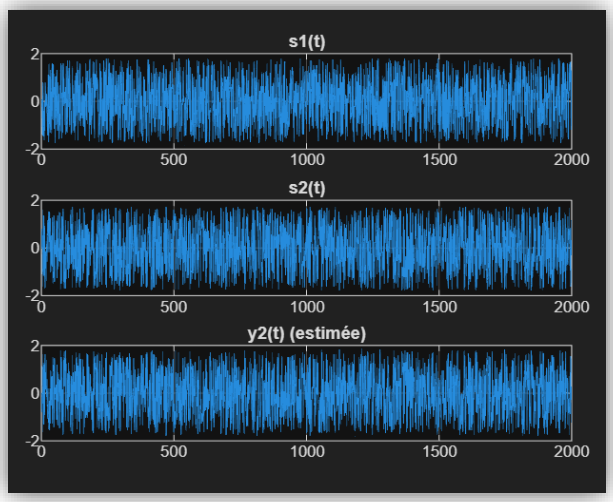


Figure 6 : Représentation qui montre la comparaison entre s1, s2 et y2

8) **Question 8 :** elle est donnée dans le code MATLAB

Conclusion :

Le coefficient de corrélation entre y_1 et y_2 est proche de zéro, ce qui montre qu'ils sont non corrélés. Le nuage de points ne présente pas de structure particulière.

On peut donc considérer que les deux sources estimées sont indépendantes, ce qui confirme le bon fonctionnement de l'algorithme FastICA.

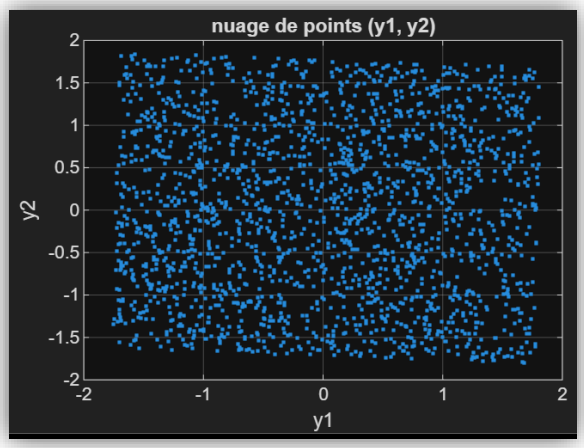
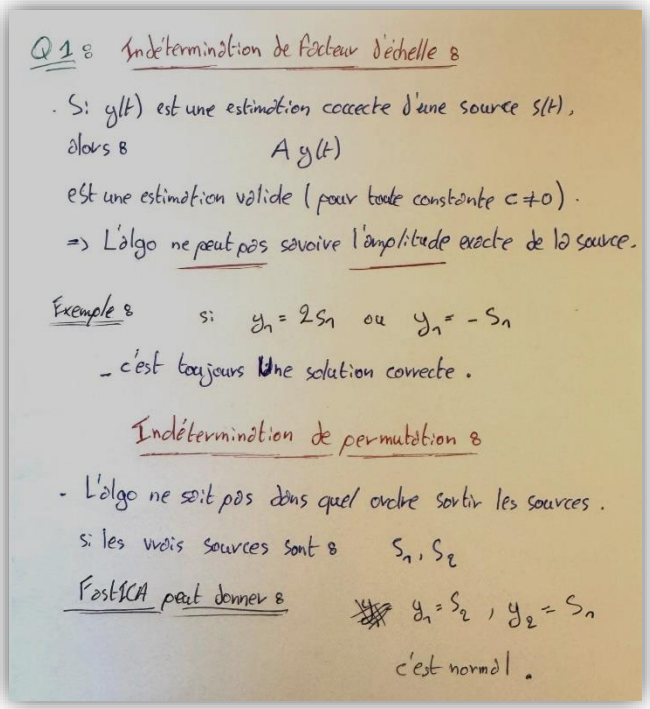


Figure 7 : Représentation du nuage de points bidimensionnel (y1 et y2)

IV. Mesure de performances de l'algorithme :

1) **Question 1**


Q1 : Indétermination de facteur d'échelle :

- Si $y(t)$ est une estimation correcte d'une source $s(t)$, alors : $A\,y(t)$ est une estimation valide (pour toute constante $c \neq 0$).
⇒ L'algo ne peut pas savoire l'amplitude exacte de la source.

Exemple : si $y_1 = 2s_1$ ou $y_1 = -s_1$
- c'est toujours une solution correcte.

Indétermination de permutation :

- L'algo ne sait pas dans quel ordre sortir les sources.
si les vrais sources sont : s_1, s_2
FastICA peut donner : ~~$y_1 = s_2$, $y_2 = s_1$~~
c'est normal.

2) **Question 2 :** elle est donnée dans le code MATLAB

Conclusion :

Le RSI obtenu est d'environ 31.8 dB, ce qui indique une très bonne qualité de séparation. Les deux sources sont correctement estimées, avec une erreur très faible par rapport aux signaux originaux.

L'algorithme FastICA a donc permis une séparation aveugle efficace des sources.

V. Tests avec d'autres signaux :

1) **Question 1 :** elle est donnée dans le code MATLAB

Explication :

Avec des sources gaussiennes, FastICA ne fonctionne pas car l'algorithme utilise la non-gaussianité (kurtosis) pour séparer les sources. Or toute combinaison linéaire de signaux gaussiens reste gaussienne, donc il n'existe pas de direction privilégiée pour retrouver les sources.

2) **Question 2 :** elle est donnée dans le code MATLAB

Conclusion :

Les deux fichiers audios contenaient un mélange d'une voix chantée et d'un accompagnement au piano. À l'écoute des signaux mélangés, on entend simultanément la voix et le piano dans chaque canal.

Après application du blanchiment puis de l'algorithme FastICA, on obtient deux signaux estimés. À l'écoute, l'un des signaux contient principalement la voix chantée avec très peu de piano, tandis que l'autre contient principalement le piano avec très peu de voix.

L'algorithme a donc réussi à séparer efficacement les deux sources audios.

3) **Question 3 :** elle est donnée dans le code MATLAB

Conclusion :

On part de deux images mélangées (chaque mélange contient les deux images superposées). Après blanchiment puis application de FastICA pixel par pixel, on obtient deux images estimées. Visuellement, chaque image séparée ressemble principalement à une des images originales. Cela montre que FastICA permet aussi de séparer des sources sous forme d'images. (les figures ils sont données dans le code Matlab).

VI. Séparation de plusieurs sources :

1) Question 1

Q18

Dans FASTICA symétrique (kurtosis), on met à jour simultanément les vecteurs w_i par une règle de point fixe :

$$w_i \leftarrow E\{z(w_i^T z)^3\} - 3w_i$$

Après la mise à jour, on applique une orthogonalisation/normalisation globale pour garantir l'orthonormalité des vecteurs :

$$W \leftarrow (WW^T)^{-1/2} W$$

Cela évite que plusieurs vecteurs convergent vers la même source et assure la séparation de plusieurs sources.

2) Question 2 : la fonction est donnée dans le code MATLAB

Commentaires : On implémente FastICA symétrique : les mélanges sont centrés puis blanchis. Ensuite, on initialise une matrice W et on met à jour simultanément ses lignes avec la règle du point fixe basée sur le kurtosis. À chaque itération, on orthogonalise W afin de garantir des vecteurs de séparation orthonormaux. Les sources estimées sont obtenues par Y=WZ.

3) Question 3 : elle est donnée dans le code MATLAB

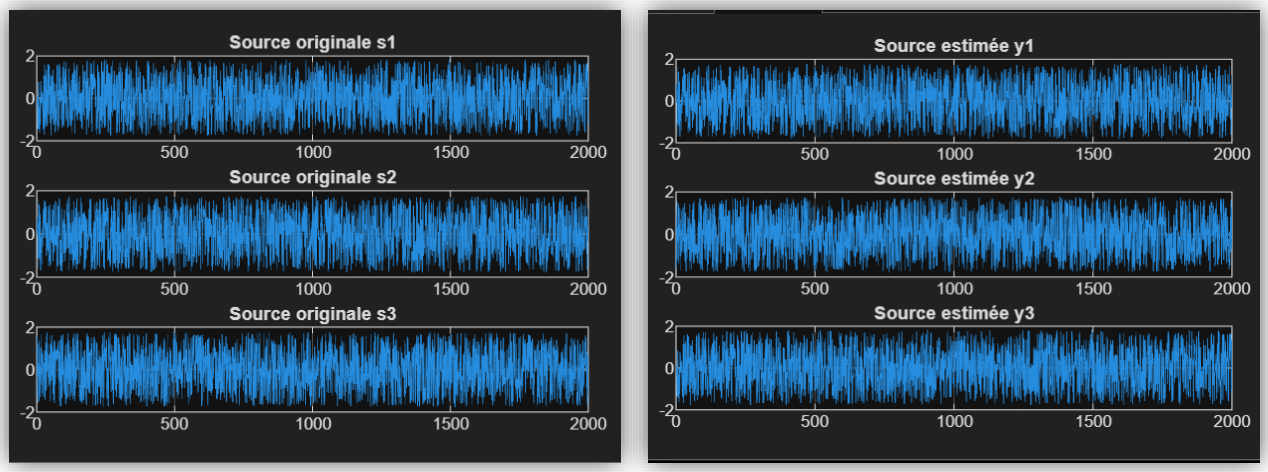


Figure 8 : Représentation des sources séparées et des sources estimées

Conclusion :

Les corrélations montrent que :

- Y_1 correspond à S_2
- Y_2 correspond à S_3
- Y_3 correspond à S_1

Les coefficients de corrélation sont très proches de 1 (≈ 0.999), ce qui confirme une séparation très précise.
L'ordre des sources estimées est permuté par rapport aux sources originales, ce qui est normal en ICA.
Le RSI moyen d'environ 31.5 dB confirme une très bonne qualité de séparation.

4) <u>Question</u> 4 : elle est donnée dans le code MATLAB

Conclusion :

Les neuf fichiers contenaient des mélanges de plusieurs sources musicales (des chansons). À l'écoute des mélanges, plusieurs instruments ou sons étaient superposés dans chaque signal.

Après application de l'algorithme FastICA symétrique, les neuf signaux estimés ont été écoutés séparément. On observe que chaque sortie contient principalement une seule source musicale, avec une séparation globalement nette et intelligible.

Il peut subsister une légère interférence résiduelle ou une variation d'amplitude, ce qui est normal en séparation aveugle.

On conclut que FastICA permet une séparation efficace de plusieurs sources audio simultanément.

Conclusion générale :

Au cours de ce TP, nous avons étudié et implémenté l'algorithme FastICA pour la séparation aveugle de sources, dans différents contextes (signaux synthétiques, audio et images).

Dans un premier temps, nous avons généré des sources aléatoires non gaussiennes, les avons mélangées linéairement, puis appliqué le blanchiment et FastICA basé sur le kurtosis. Les résultats ont montré que les sources originales sont correctement retrouvées, à un signe et une permutation près, ce qui correspond aux indéterminations théoriques de la séparation aveugle. Les valeurs élevées du RSI (≈ 30 dB) ont confirmé la très bonne qualité de séparation.

Nous avons ensuite vérifié que l'algorithme ne fonctionne pas avec des sources gaussiennes, ce qui confirme que FastICA repose sur la non-gaussianité des signaux pour effectuer la séparation.

L'algorithme a également été appliqué avec succès à des signaux audio (voix et instruments), montrant une séparation clairement perceptible à l'écoute. Enfin, la version symétrique de FastICA a permis de séparer simultanément plusieurs sources (3 puis 9 signaux audio), avec des résultats satisfaisants.

Ce TP met ainsi en évidence :

- l'importance du blanchiment,
- le rôle central de la non-gaussianité,
- les indéterminations d'échelle et de permutation,
- et l'efficacité pratique de FastICA pour la séparation de sources.

En conclusion, FastICA constitue une méthode puissante et efficace pour la séparation aveugle de sources dans des contextes variés.