CHAPITRE 4

MISE EN ŒUVRE DE LA SOLUTION CHOISIE

La mise en œuvre de la solution choisie a été faite premièrement en simulation sous Matlab puis s'est réalisée en programme exécutable sous langage C. La simulation sous Matlab part de la figure 6 du schéma synoptique de notre application au chapitre 3. Les différentes opérations sont décrites ci-après.

4.1 Préparation du signal

À partir d'un signal échantillonné, nous lui avons ajouté dans un premier temps un bruit de telle sorte qu'on obtienne un rapport signal sur bruit de 5 dB ou 3 dB selon les signaux essayés. Par la suite nous l'avons fait passer à travers deux filtres RIF qui modélisent l'effet convolutif réverbérant, la réponse fréquentielle de ces filtres est donnée sur la figure 15. On obtient ainsi deux signaux semblant venir de deux microphones illustrés sur la figure 14, où l'abscisse est exprimée en terme du nombre d'échantillon. Les programmes ayant servit à cette préparation sont mis en annexe et s'intitulent; sigbruit.m et mixconvolFIR4.m .

4.2 Détermination de la fréquence fondamentale f0

Chaque canal est traité par la TOD sur trois échelles pour relever les maxima et calculer les périodes associées de la fréquence fondamentale, cela se fait avec le programme freqf0.m, le résultat global est donnée sur les figure 16 et 17 pour chaque canal.

La simulation sur Matlab de la détermination de la fréquence fondamentale f0 selon l'approche vue dans la solution retenue [10] est mise en œuvre dans le programme freqf0.m placé en annexe I, où la fonction CWT (continuous wavelet function) est écartée mais remplacée par son calcul équivalent, ceci afin de mieux exporter sa fonctionnalité vers le langage C qui sera compilé sur DSP éventuellement.

À partir d'un échantillon de voix, on opère la TOD puis pour chaque segment analysé on calcul la fréquence fondamentale f0 qui est la donnée cruciale pour le reste des algorithmes de l'application.

La TOD sur les trois échelles, se fait sur segment de 256 échantillons, le résultat global est illustré sur les figures 16 et 17. La période et la fréquence fondamentale sont calculées sur les crêtes maximales supérieures à un seuil fixé, comme vu au chapitre 3.

4.3 Élaboration d'un banc de filtres RIF sous Matlab

Pour illustrer les principes avancés dans cette partie du document, le programme rédigé sous Matlab dont le nom de fichier est bancFIR.m mis en annexe I simule la construction d'un banc de filtre RIF à partir d'une fréquence fondamentale mesurée sur un échantillon vocal. Le listing est disponible en annexe I. La figure 18 montre le banc de filtre RIF élaboré avec une fenêtre de Kaiser et on montre sa réponse fréquentielle pour une f0 de 128 Hz sur le canal 1, l'autre banc sur le canal 2 étant sensiblement le même. Les figure 19 à 23 illustrent les résultats obtenus pour chaque passe-bande des canaux respectifs donnant les signaux filtrés.

4.4 Extraction de l'enveloppe par transformée de Hilbert

Nous avons également sur les figure 19 à 23 l'allure des enveloppes des différents signaux filtrés obtenues par la transformée de Hilbert.

4.5 La partie couvrant l'ACI

Cette partie débute l'approche de l'ACI vu selon les auteurs [10] qu'on doit considérer dans un sens très large, elle a été également vue brièvement au chapitre 3 où nous avons discuté des similitudes et des différences en rapport avec la définition de l'ACI adoptée par l'ensemble de la communauté des chercheurs scientifiques.

Chaque tranche fréquentielle filtrée sur les deux canaux est prise en entrée avec leur enveloppe respective et traitée séparément en fonction de la f0 ou de ses harmoniques. On élabore le signal synthétisé de chaque tranche fréquentielle avec un signal analytique (une exponentielle) modulée en amplitude par son enveloppe associée mais qui est pour l'instant dépourvue du déphasage original. Elle est donc mise à l'entrée d'un filtre de Wiener de type LMS dont la séquence d'apprentissage est le signal venant de la tranche fréquentielle filtrée. La sortie de ce traitement donne les signaux synthétisés $x_{i,k}(t)$ sur la fondamentale et ses harmoniques avec leurs déphasages corrigés.

C'est le programme ica.m mis à l'annexe I qui assume cette partie. Les figures 24 à 27 donnent les résultats de cette partie de l'ACI pour chaque première tranche fréquentielle synthétisée sur chaque canal et sur la tranche fréquentielle intermédiaire, car dans notre exemple le banc de filtre sur chaque canal est composé de cinq passe-bandes comme l'illustre la figure 18.

Puis vient la seconde partie du traitement de l'ACI où est mise en œuvre la matrice de séparation conjuguée avec le prédicteur linéaire de rehaussement vus à la fin du chapitre 3 et schématisés sur la figure 13. Le programme exécutant ces fonctions se nomme ica1.m et peut être consulté dans l'annexe I. Les figures 28 à 32 donnent les résultats du processus sur chaque composante fréquentielle des signaux synthétisés et combinés des deux canaux. La figure 33 montre le signal original et son estimé avec le délai de la réponse du traitement.

Les algorithmes construits pour la simulation sous Matlab nous ont permis de réaliser les blocs fonctionnels sous programmation en langage C. Chaque partie a été vérifiée par rapport à la simulation faite sur Matlab.

La différence majeure réside dans le traitement segmenté mais inclus cependant l'ensemble des algorithmes dans le programme en langage C alors que la simulation

sous Matlab, a été réalisée sous différents blocs fonctionnels. Le listing du programme intégral en langage C est mis dans l'annexe II.

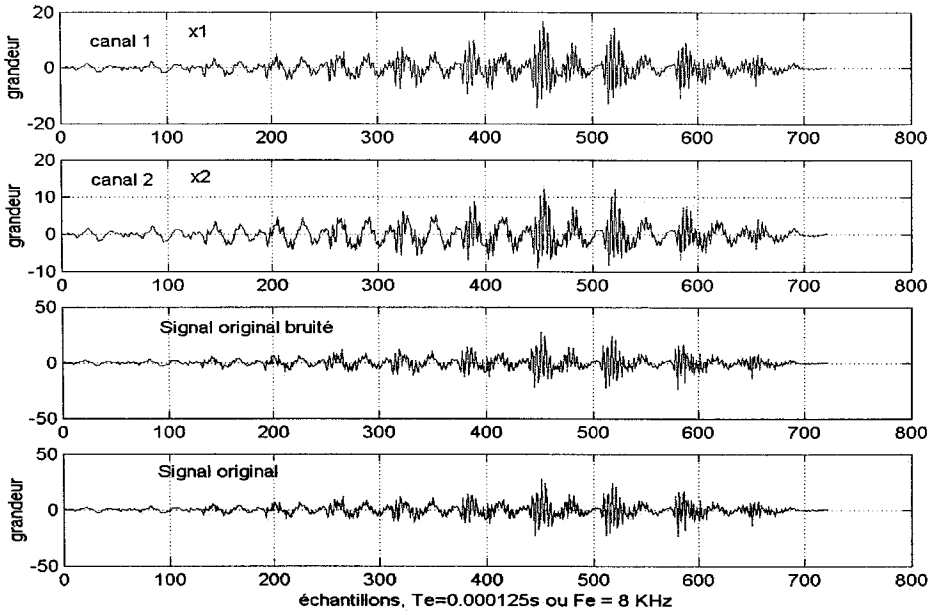

Figure 14 Signaux captés par microphones 1 et 2

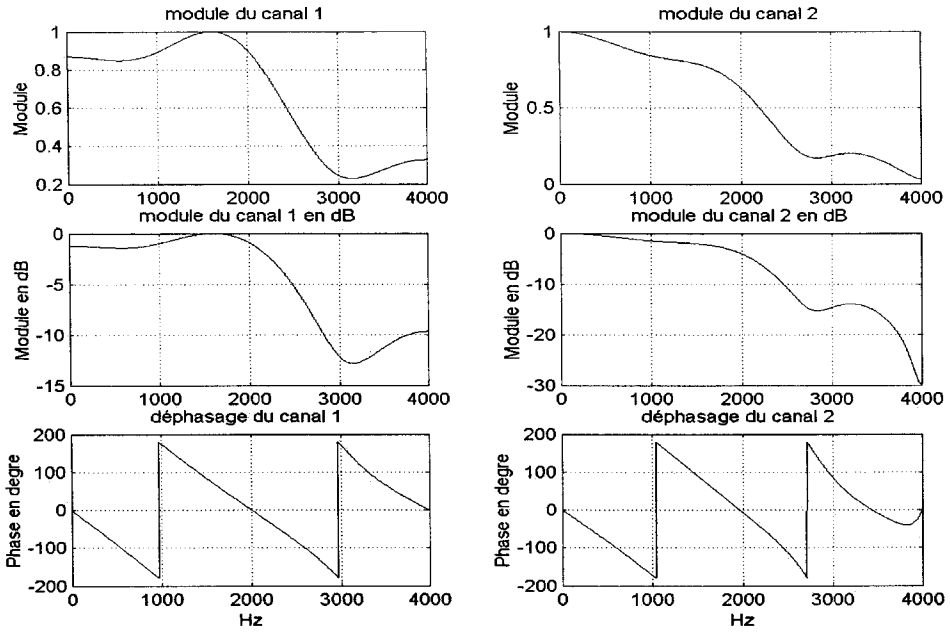

Figure 15 Réponse fréquentielle des filtres RIF des deux canaux

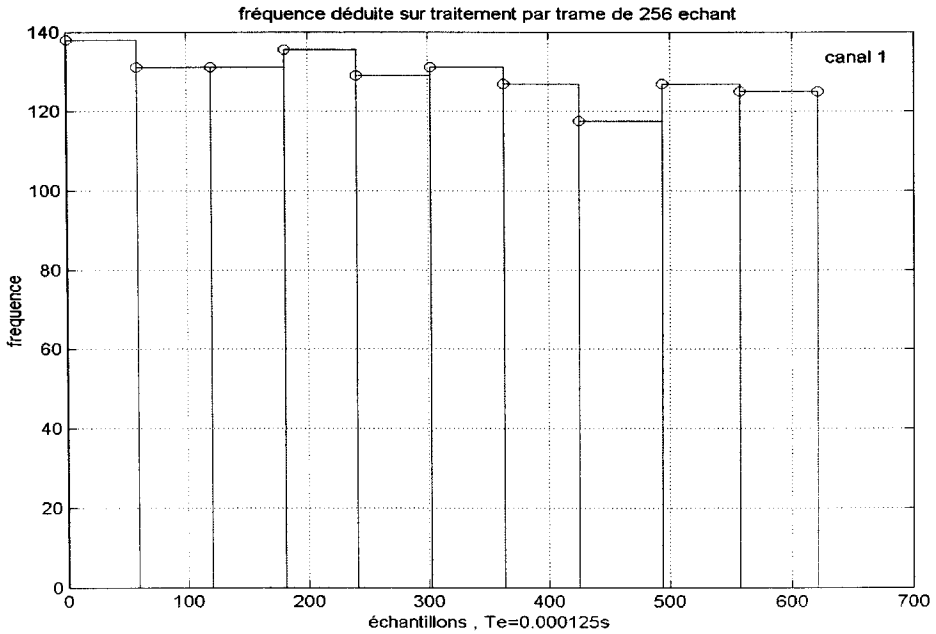

Figure 16 Détection de la fréquence fondamentale f0 sur canal 1

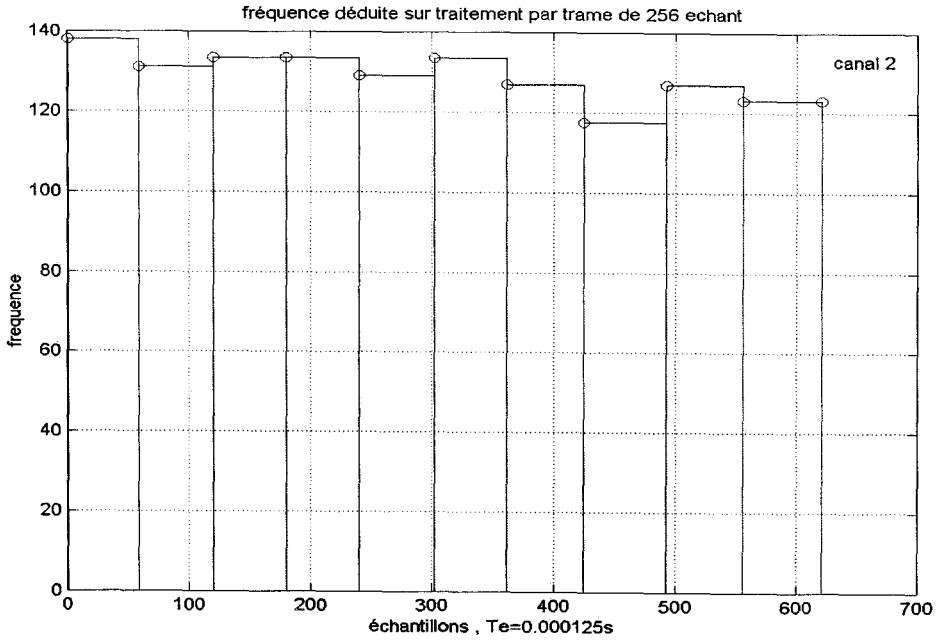

Figure 17 Détection de la fréquence fondamentale f0 sur canal 2

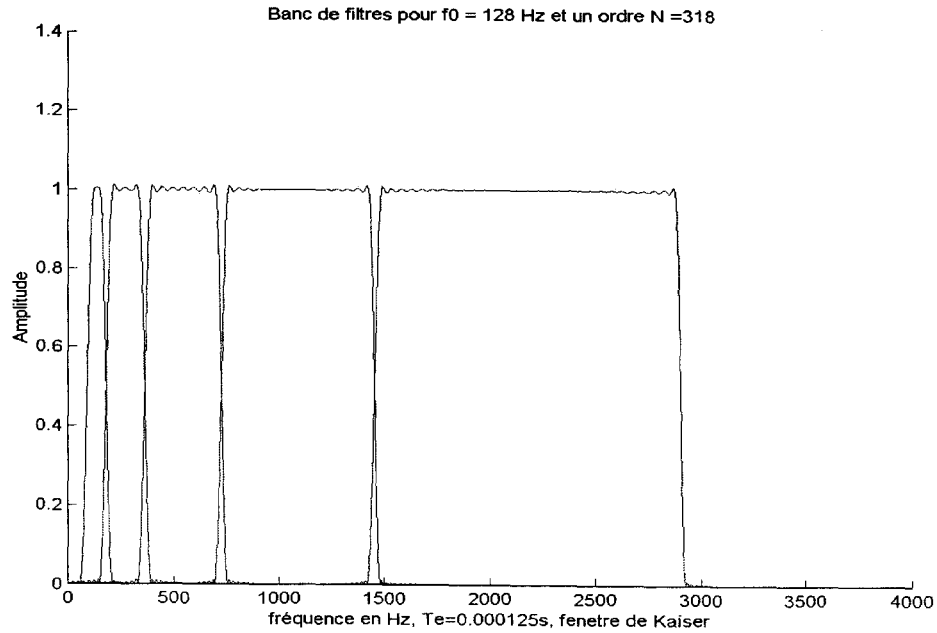

Figure 18 Réponse fréquentielle du banc de filtre canal 1

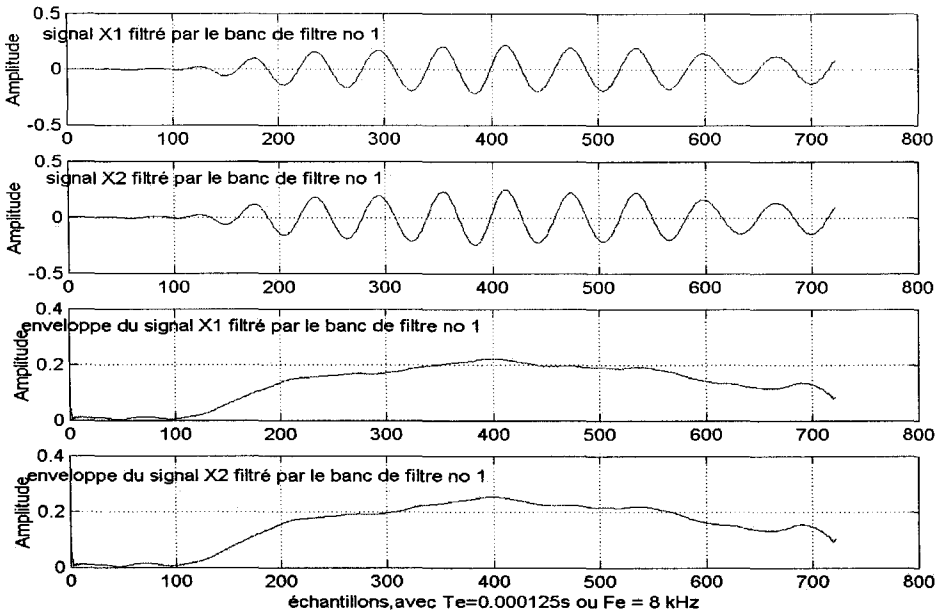

Figure 19 Premier passe-bande filtré autour de f0

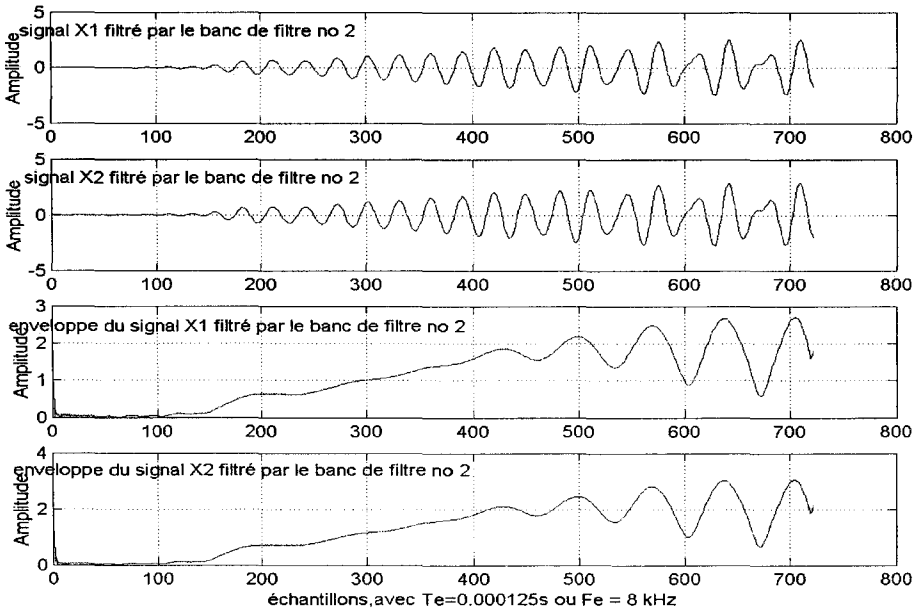

Figure 20 Second passe-bande filtré autour de 2 f0

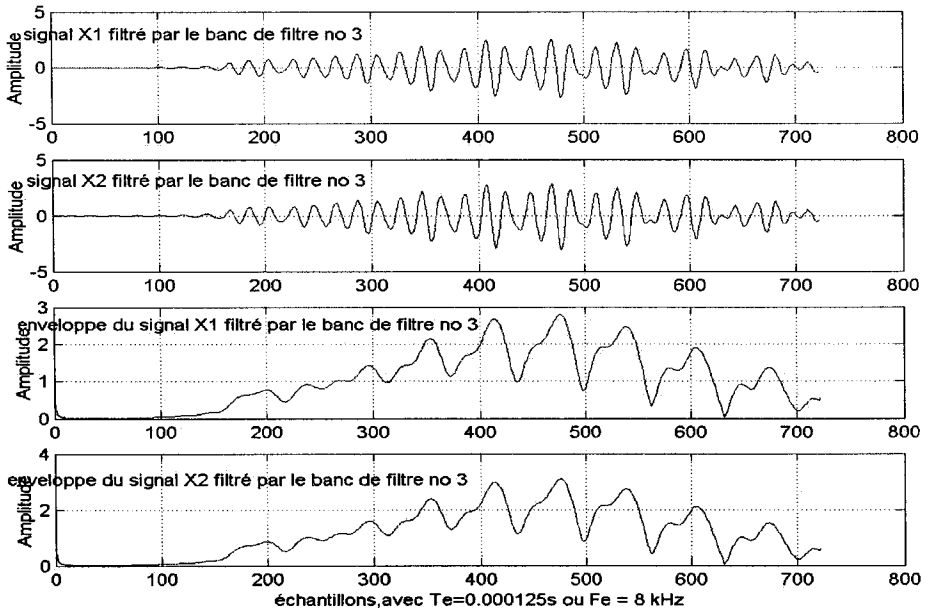

Figure 21 Troisième passe-bande filtré autour de 4 f0

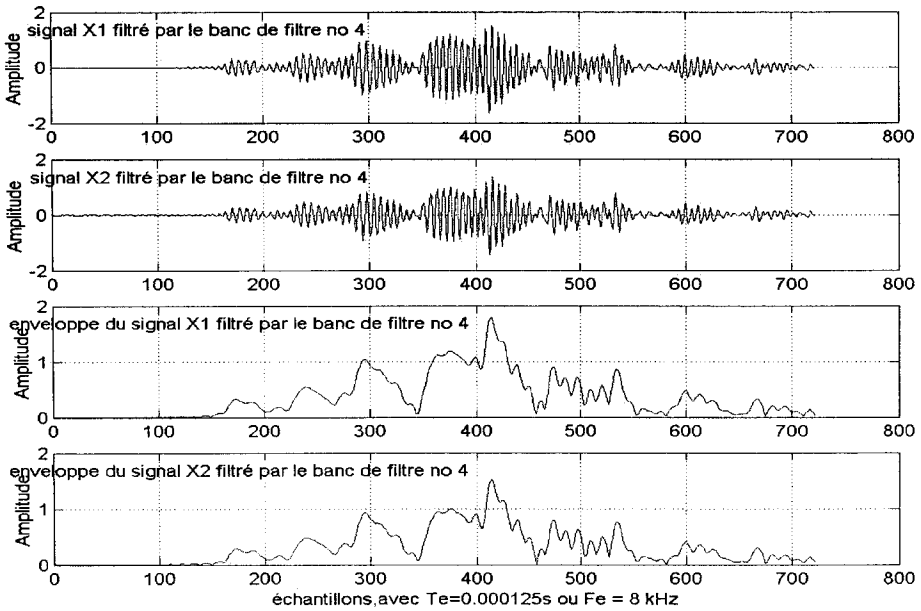

Figure 22 Quatrième passe-bande filtré autour de 8 f0

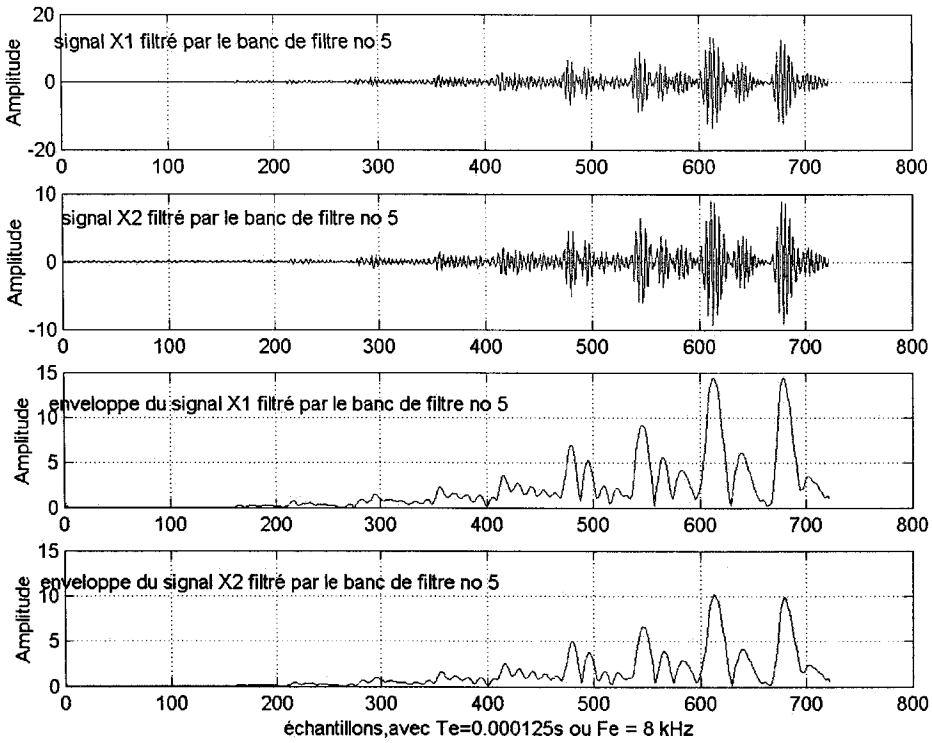

Figure 23 Cinquième passe-bande filtré autour de 16 f0

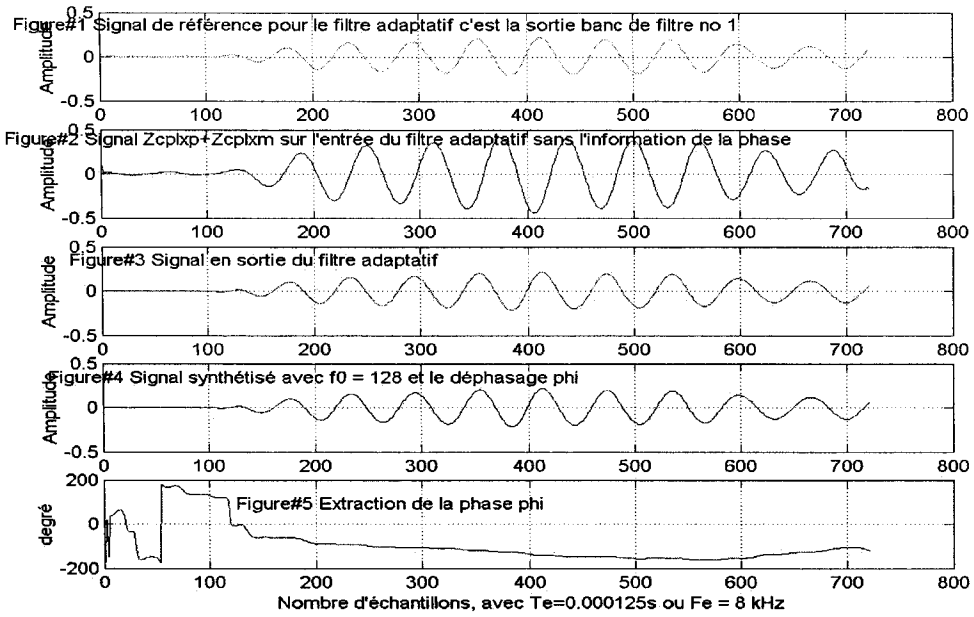

Figure 24 Signal synthétisé avec la f0 sur canal 1

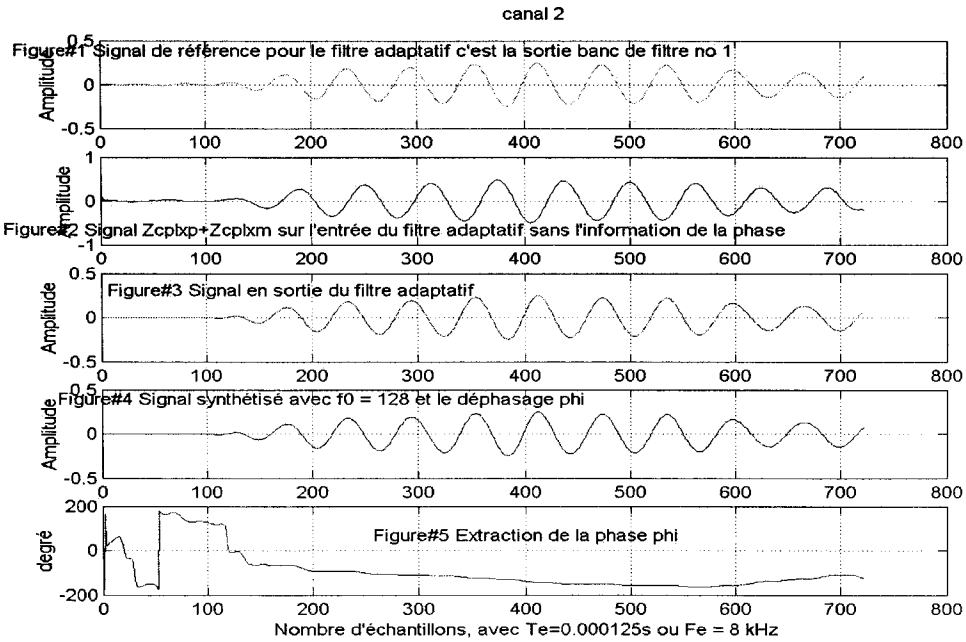

Figure 25 Signal synthétisé avec la f0 sur canal 2

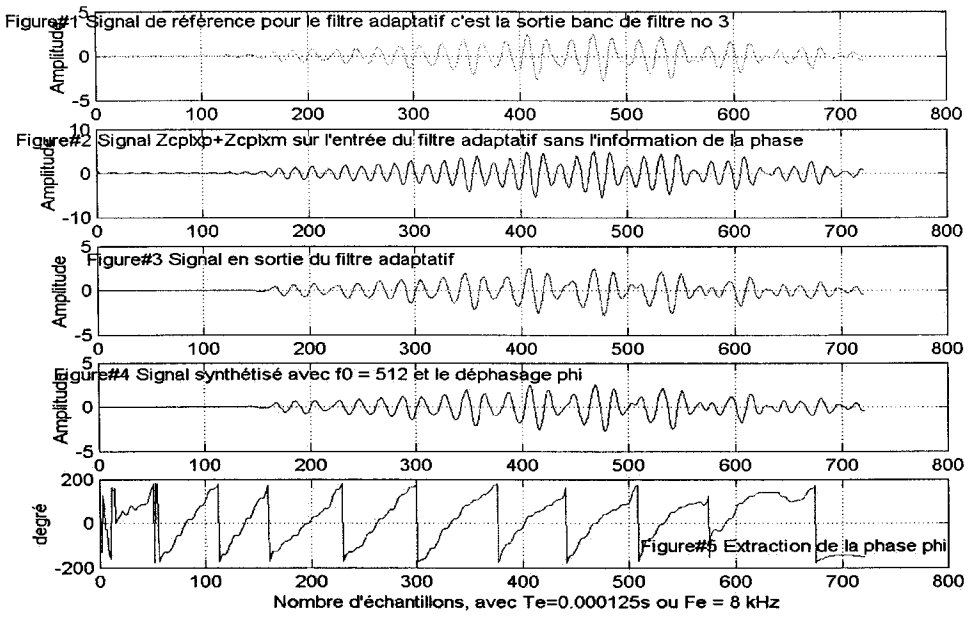

Figure 26 Signal synthétisé avec la 4 f0 sur canal 1

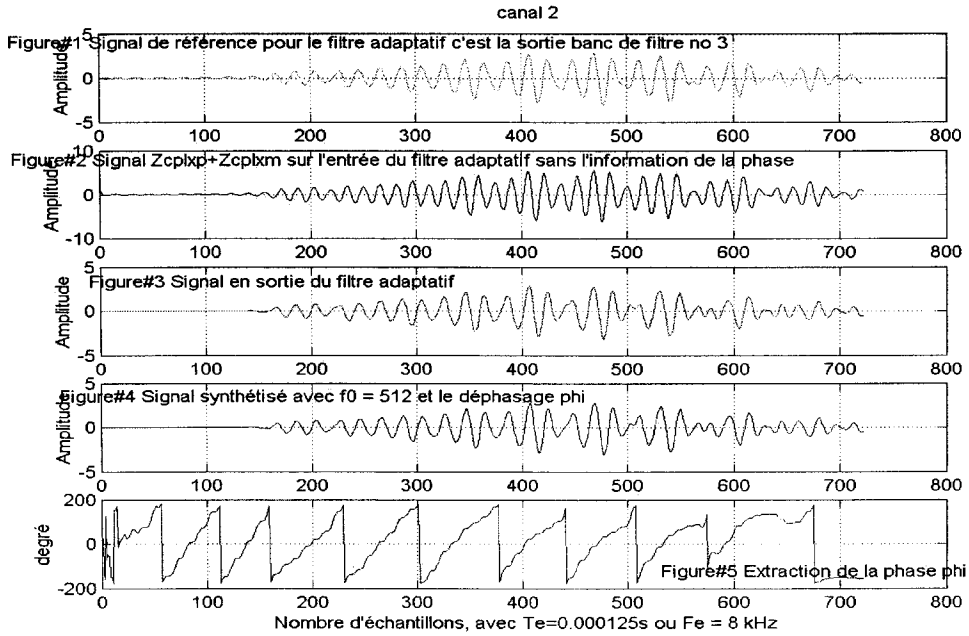

Figure 27 Signal synthétisé avec la 4 f0 sur canal 2

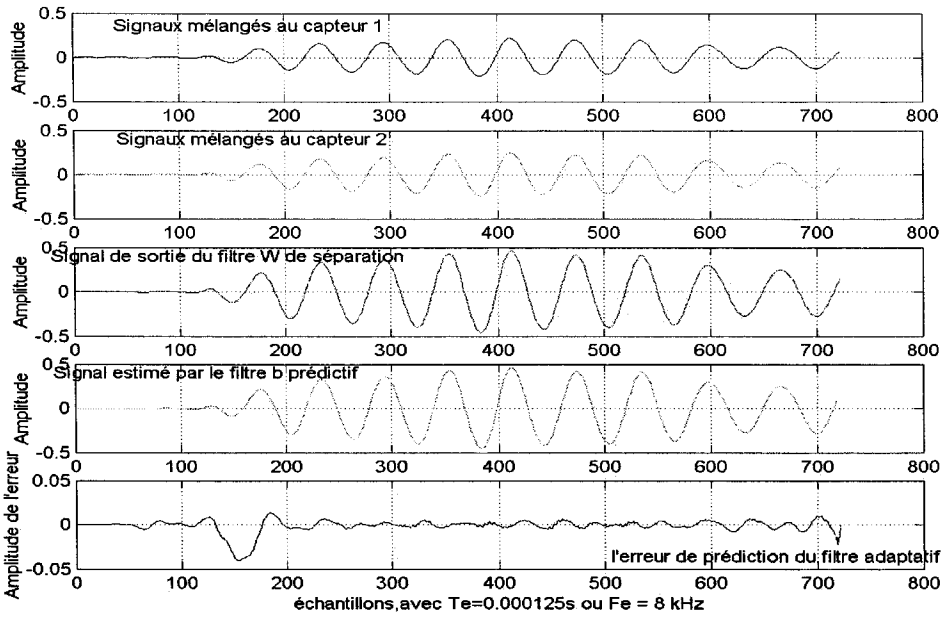

Figure 28 Extraction des composantes de la source sur f0

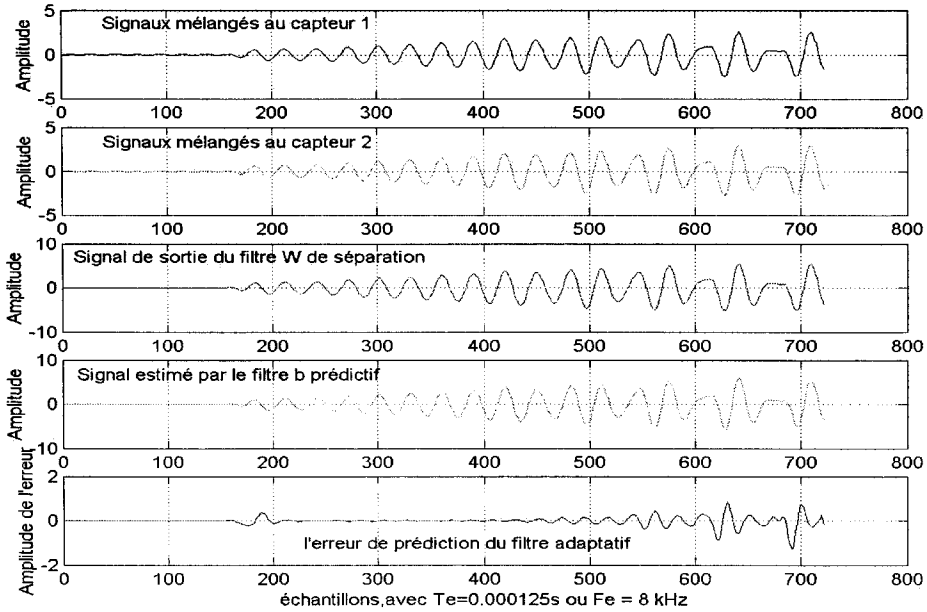

Figure 29 Extraction des composantes de la source sur 2 f0

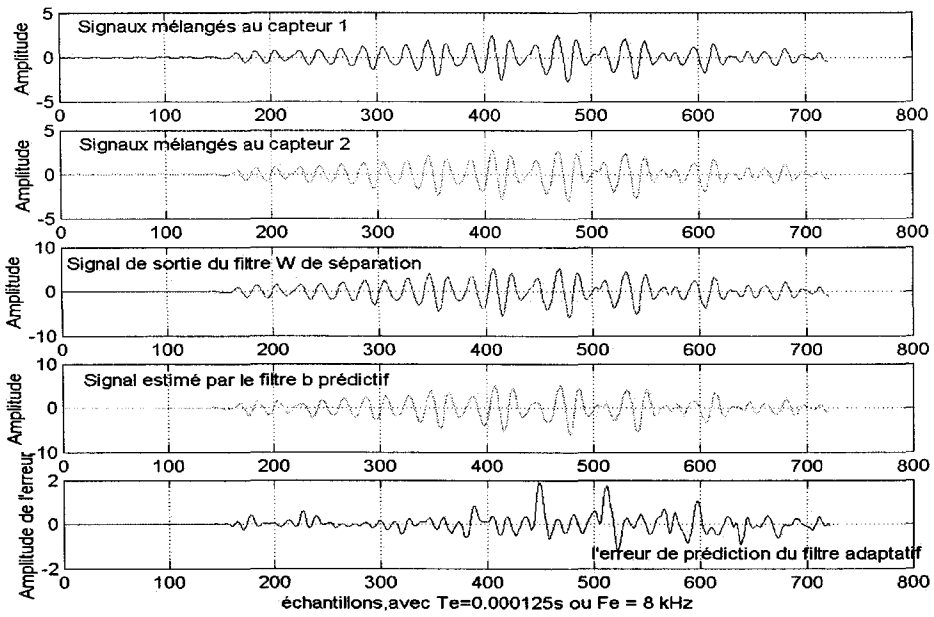

Figure 30 Extraction des composantes de la source sur 4 f0

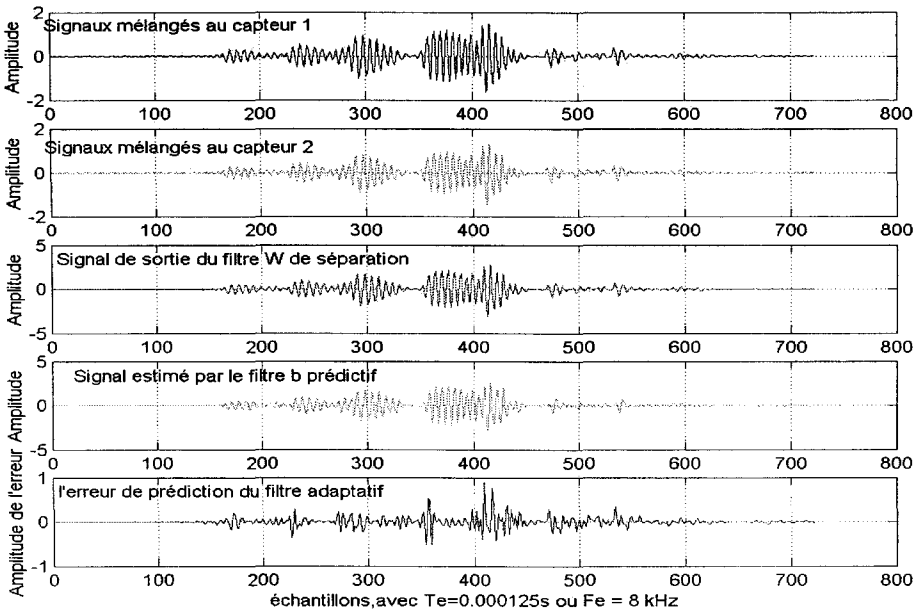

Figure 31 Extraction des composantes de la source sur 8 f0

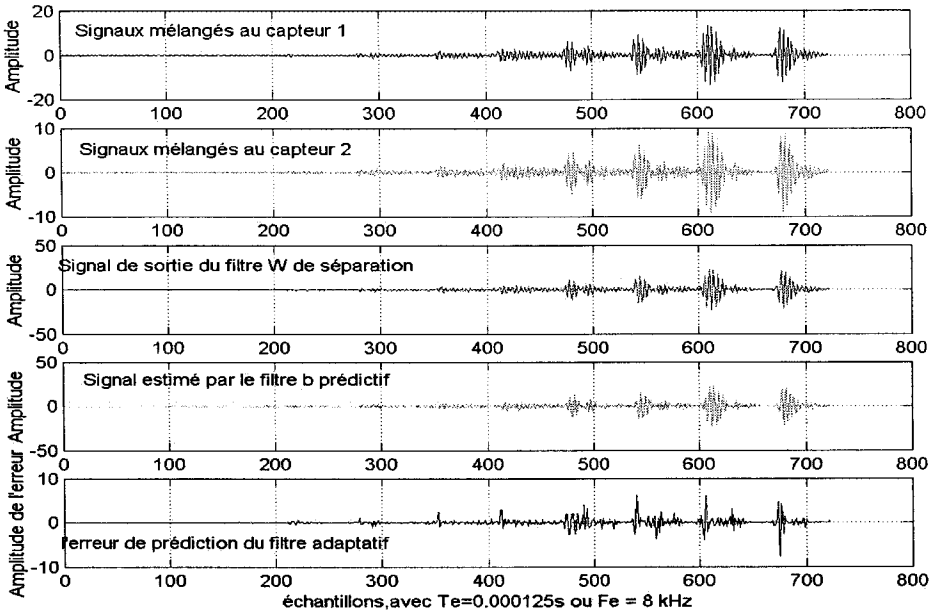

Figure 32 Extraction des composantes de la source sur 16 f0

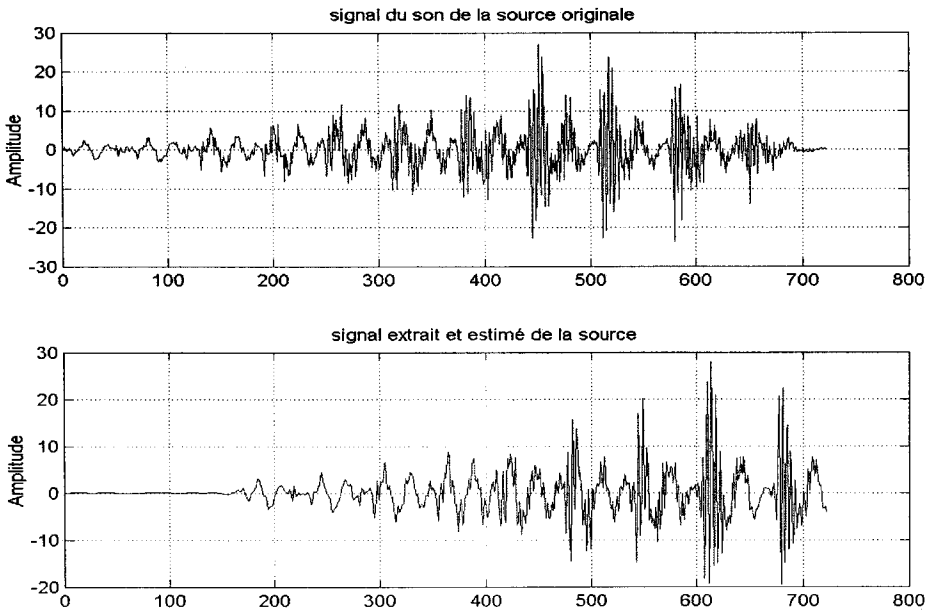

Figure 33 Sommation des composantes donnant la source estimée