École de technologie Supérieure

Université du Québec

Rapport première étape de projet

Présenté à Mr. Marcel GABREA

Dans le cadre du cours

*SYS835 – Processeur numérique du signal et ses applications*

PAR

Eric LACERTE LACE23038502

Philippe LAVOIE LAVPXXXXXXXX

MONTRÉAL, LE 8 NOVEMBRE 2017

# Introduction

Ce document traite l’implémentation et la simulation de l’algorithme de suppression du bruit en temps réel élaboré par McAulay et Malpass[[1]](#footnote-1). Tout d’abord Nous décrirons l’implémentation de ce dernier à l’aide du langage Matlab. Nous ferons également un bref survol théorique pour chacune des parties implémentées. Nous terminerons avec les résultats de la simulation et une analyse des résultats obtenus.

# Implémentation

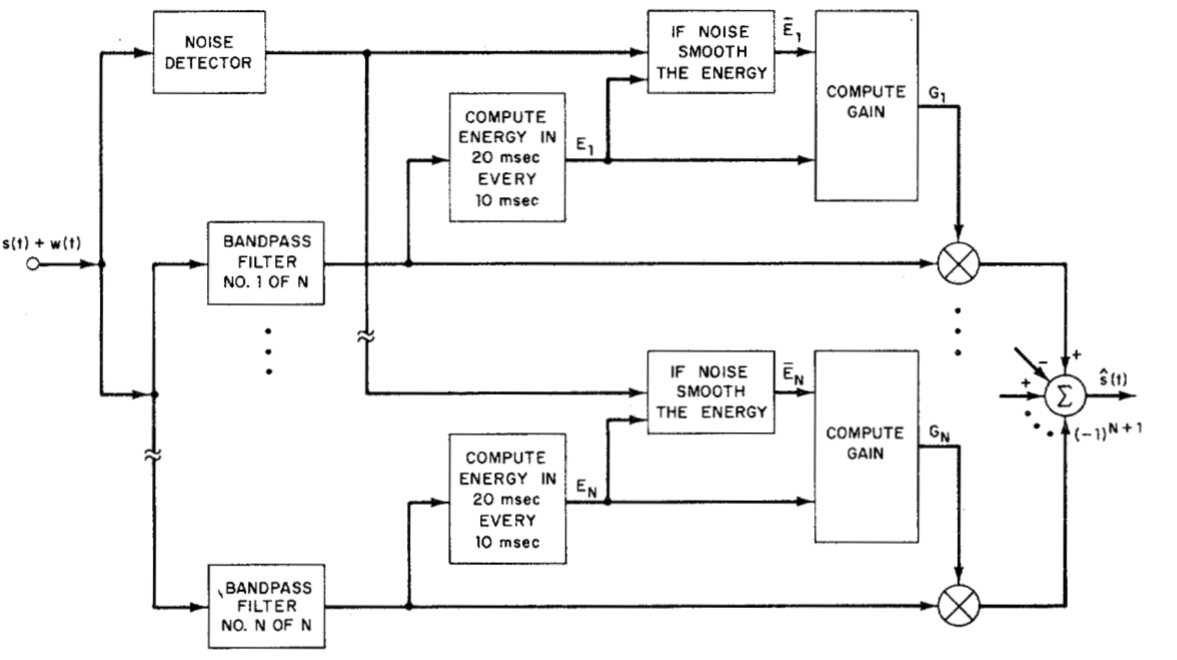


Figure Schéma Bloc Filtre de Suppression de bruit[[2]](#footnote-2)

## Détecteur de bruit

Le signal est divisé en segments de 20 ms. Nous devons déterminer pour chaque segment s’il contient du bruit ou de la voix. Pour ce faire nous utilisons une implémentation de l’algorithme de détection du bruit décrit dans l’article [2].

L’algorithme utilise un histogramme de 4s de l’énergie de chaque segment pour déterminer de façon adaptative le niveau d’énergie correspondant à la frontière entre le bruit et le bruit et la voix.

L’algorithme de l’article détermine qu’un signal n’est pas du bruit si son énergie dépasse 16 bits. Ceci est probablement dû à la quantification utilisée dans système décrit. Pour pouvoir utiliser l’algorithme tel quel, nous réduisons l’échelle de nos échantillons à 8 bits lors du calcul de l’énergie.

## Filtres Passe-bande

Le système utilise 19 filtres Butterworth de 2e ordre pour séparer le signal en bandes de fréquences et déterminer le niveau de bruit moyen et calculer le gain à appliquer dans chaque bande. Dans l’implémentation Matlab, nous générons les coefficients en utilisant les fréquences centrales et bandes passantes de ’échelle de Bark [3] plutôt que les celles utilisées dans l’article, puisque la fréquence d'échantillonnage des signaux que nous traitons est différente de celle du système décrit dans [1] (8 kHz vs 7.575 kHz).

## Calcul de l’énergie

L’énergie est la somme des carrés de chaque échantillon du segment analysé. L’énergie est calculée pour chaque bande selon l’équation:

(Équation énergie)

## Détermination du niveau de bruit

Lorsque le détecteur de bruit détermine qu’un segment ne contient que du bruit, le niveau de bruit moyen dans chaque bande est déterminé par la fonction de lissage récursive sur 1s (50 segments de 20 ms) suivante:

(Équation bruit moyen)

## Détermination du gain

Ce calcul est illustré dans le code Matlab à la section *Détermination du gain*.

Le gain de suppression est déterminé à l’aide d’une fonction de Bessel modifiée :[[3]](#footnote-3)

Insérer équation 4

Où

Insérer équation 5

Nous aurons besoin de déterminer le paramètres gn(m), celui-ci étant le ratio de la différence entre l’énergie du cadre actuelle du canal et l’énergie du bruit du cadre précédent sur l’énergie du cadre actuelle du canal. Plus ce ratio est élevé, plus le cadre actuel peut être considéré comme de la voix. Donc plus ce paramètre est élevé, plus l’atténuation sera faible. Dans le code Matlab, gn(m) est désigné par *ch\_meas\_parms* dans *main.m.*

Le paramètre ε est arbitraire et définit au début du code MatLab par l’utilisateur. Il doit être compris entre 1 et 20.

La calcul du gain est effectué par la fonction *func\_suppress\_curve(eps, v\_parm)* du fichier *func\_suppress\_curve.m*. Cette dernière étant l’implémentation de la fonctions de Bessel modifiée.

## Application du gain et reconstitution du signal

Après avoir déterminer le gain à appliquer au signal du canal, un lissage est effectué afin d’éviter les changements trop brusques du signal. Ces brusques sauts pourraient altérer la qualité du signal sonore perçu par l’usager. Ce lissage est décrit par l’équation suivante :

Insérer équation 6

Où

Insérer équation 7

Cette équation est implémenté par la fonction *calc\_smooth\_gain*  du fichier *calc\_smooth\_gain.m.*

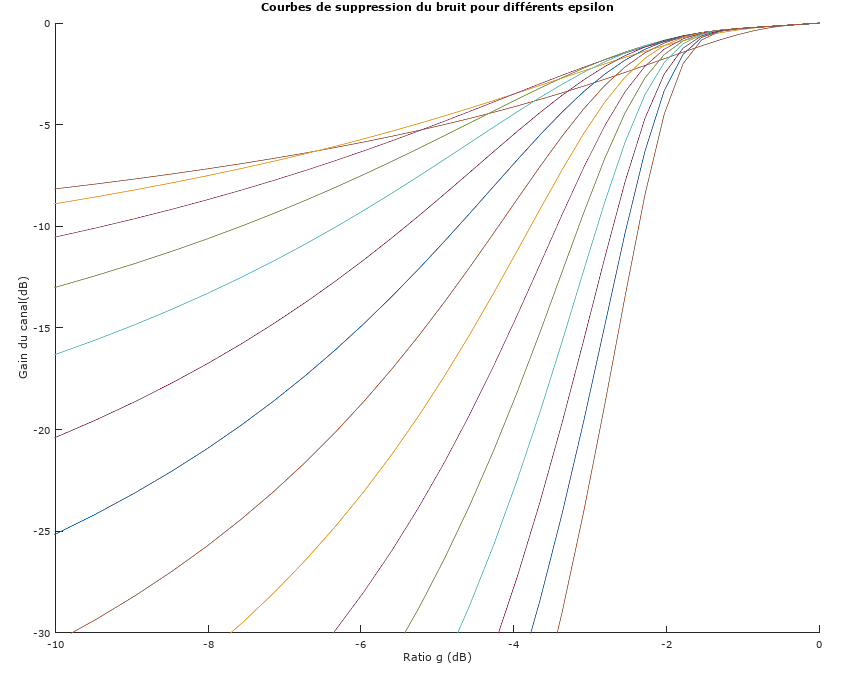
Une fois le gain lissé, celui-ci est appliqué sur le signal du canal. Tel qu’expliquer précédemment, ce gain sera beaucoup plus faible pour les cadres contenant du bruit.

Par la suite, nous recombinons les signaux des différents canaux avec un déphasage de 180° entre eux. Ceci est exprimé dans le code de main.m par une alternance d’addition et de soustraction des signaux de canaux.

# Simulation et Résultats

## Filtres Passe Bande

## Courbe de suppression du bruit



ε=1

ε=20

Figure Courbe de suppression du bruit pour différentes valeurs d'epsilon

# Conclusion et Analyses

# Références

Robert J. McAulay and Marilyn L. Malpass, (1980) A REAL-TIME NOISE SUPPRESSION FILTER FOR SPEECH ENHANCEMENT AND ROBUST CHANNEL VOCODING

Robert J. McAulay and Marilyn L. Malpass, (1980) SPEECH ENHANCEMENT USING A SOFT-DECISION NOISE SUPPRESSION FILTER

1. Robert J. McAulay and Marilyn L. Malpass, (1980) A REAL-TIME NOISE SUPPRESSION FILTER FOR SPEECH ENHANCEMENT AND ROBUST CHANNEL VOCODING [↑](#footnote-ref-1)
2. Robert J. McAulay and Marilyn L. Malpass, (1980) SPEECH ENHANCEMENT USING A SOFT-DECISION NOISE SUPPRESSION FILTER [↑](#footnote-ref-2)
3. Equation 3 de Mckaulay [↑](#footnote-ref-3)