Département Électronique Électrotechnique Automatique

Équipe Automatique et Traitement du Signal

STI tc2 Traitement du Signal

TP Parole :

Compression et décompression de la parole par filtre générateur

---

Contexte et objectifs du TP

Lors d'une communication, un téléphone mobile transforme la parole (onde acoustique) en un signal électrique par l'intermédiaire d'un microphone. Grâce à un convertisseur Analogique-Numérique (CAN) et un processeur de signaux (Digital Signal Processor, DSP), ce signal électrique analogique est échantillonné, quantifié, puis compressé et codé dans un format binaire (voir l'AF STI tc3 "Conversion analogique-numérique"). Le signal de parole initial prend à ce stade la forme d'une séquence de données binaires. Ces données sont alors transmises via le réseau au téléphone mobile de l'interlocuteur. Celui-ci décode et décompresse les données afin de recréer un signal de parole numérique, le convertit en signal analogique, et finalement génère à l'aide d'un haut-parleur une onde acoustique qui sera perçue par l'interlocuteur (cf Figure 1).

Le développement de la téléphonie mobile s'est accompagné d'une augmentation massive de la quantité de données transmises à travers les réseaux de communication. Afin de ne pas saturer la capacité de ces réseaux, le développement d'algorithmes simples et efficaces permettant la compression et la décompression de la parole a été nécessaire. Ces algorithmes sont issus des techniques d'analyse et de synthèse des signaux de parole et reposent sur la notion de filtre

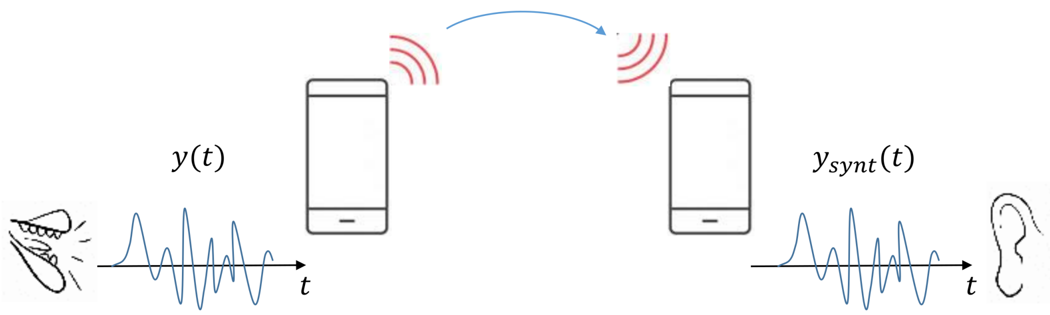

Figure 1 – Transmission de la parole

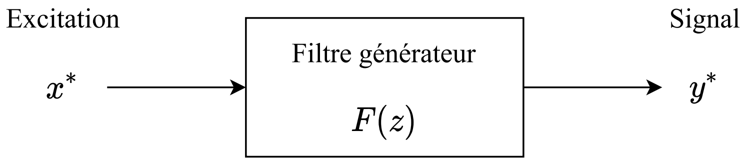

FIGURE 2 – Notion de filtre générateur pour la modélisation d'un signal

générateur (cf Figure 2).

L'objectif de ce TP est de se familiariser avec le principe de la modélisation des signaux par filtre générateur et avec les algorithmes de compression (analyse) et de décompression (synthèse) qui en sont issus. Il permet de donner quelques éléments sur les signaux de parole à travers l'analyse d'enregistrements sonores (Partie 1) puis aborde tour à tour la modélisation et la synthèse des sons voisés, de nature pseudo-périodique (Partie 2), et des sons non-voisés, de nature aléatoire (Partie 3).

Organisation du travail

Vous travaillerez par groupe de 4 élèves. La partie 1 sera traitée par l'ensemble du groupe. Ensuite, les parties 2 et 3 seront traitées en parallèle par la moitié du groupe de travail.

Dans les deux cas, le travail à effectuer comprend quatre étapes marquées par des jalons donnés ici avec un temps de fin indicatif :
- Jalon 1 : Acquisition et analyse de signaux de parole ($\sim$ 1h00)
- Jalon 2 : Analyse préliminaire des sons voisés/non-voisés ($\sim$ 2h00)
- Jalon 3 : Compression - Identification du filtre générateur ($\sim$ 3h00)
- Jalon 4 : Décompression - Synthèse du signal ($\sim$ 4h00)

Pour chacun de ces jalons, un point sera réalisé avec l'enseignant.

Il n'est pas demandé de délivrer un rapport en fin de séance. La restitution de votre travail aura lieu lors d'une séance de 2h, notée TP2 dans votre emploi du temps, et prendra la forme d'un exposé oral par groupe de 6 élèves (ces groupes ne seront pas nécessairement les mêmes que les groupes formés pour ce TP). Dans la perspective de cette restitution, il est fortement conseillé de garder une trace écrite la plus précise possible du travail réalisé en TP via un document en ligne partagé par exemple.

1 Les signaux de parole

Les sons de la parole sont d'une grande diversité et varient d'une langue à l'autre et d'un individu à l'autre. À titre d'exemple, la figure 3 présente le signal de parole correspondant au mot "efface". On y observe une succession de phonèmes ("E", "F", "A", "S" et "E") pour lesquels le signal admet des caractéristiques très différentes. Pour les phonèmes "E" et "A", le signal admet une évolution (quasi-)périodique à l'intérieur d'une enveloppe temporelle dont la forme marque assez nettement le début et la fin du phonème. Pour les phonèmes "F" et "S", le signal est très irrégulier et admet une évolution aléatoire.

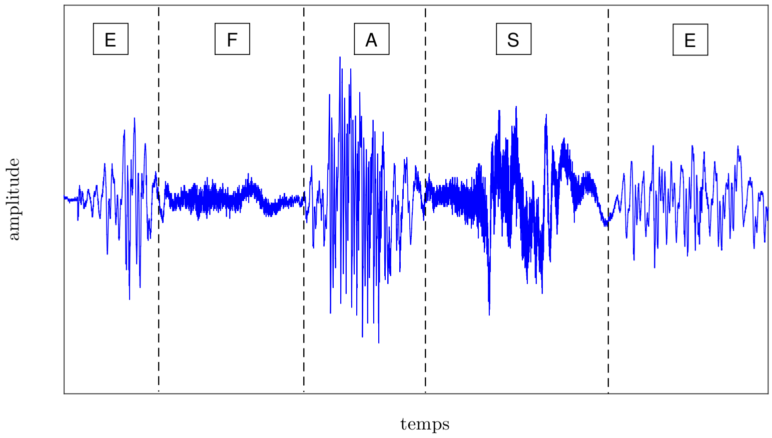

FIGURE 3 – Variation de pression en fonction du temps pour le mot "efface"

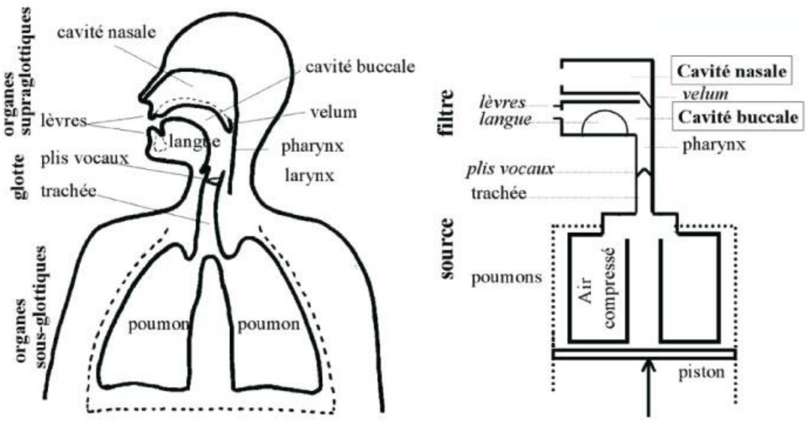

FIGURE 4 – Schéma des organes de production de la parole et modèle source-filtre (d'après `http://www.cairn.info/la-phonetique`)

La modélisation des signaux de parole nécessite donc de pouvoir prendre en compte cette diversité et de comprendre leur mode de production. Un signal de parole est une onde de pression acoustique née d'un flux d'air initié par les poumons et traversant une série de conduits et de cavités : la trachée, le larynx, la cavité buccale et/ou la cavité nasale (voir figure 4, partie gauche). Il en découle qu'une modélisation naturelle et intuitive des signaux de parole se fonde sur des filtres générateurs (voir figure 4, partie droite) dans lesquels le filtre représente le

	Enfant	Femme	Homme
Pitch	de 200 Hz à 600 Hz	de 150 Hz à 400 Hz	de 70 Hz à 250 Hz

TABLE 1 – Intervalles de pitch selon les classes d'individus

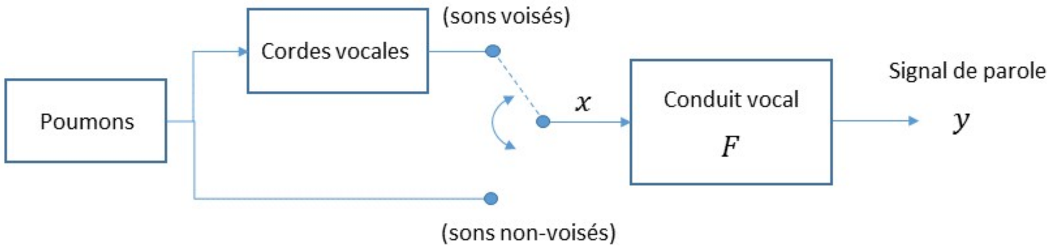

FIGURE 5 – Filtre générateur pour la production des sons voisés et non-voisés

conduit vocal et l'excitation sera fonction de la nature du son émis (voir plus loin). Dans ce TP nous ne chercherons pas à expliciter la relation entre les paramètres de la fonction de transfert du filtre et les caractéristiques physiques du conduit vocal. Cette approche de la modélisation, dite "boite noire", est en effet bien adaptée au compromis entre complexité du signal étudié et contraintes d'efficacité, de performance et de robustesse liées au contexte de transmission temps-réel de la parole.

Deux types de signaux de parole seront étudiés dans le cadre de ce TP : les sons voisés et les sons non-voisés. Pour les sons voisés, le flux d'air émis par les poumons subit la vibration des cordes vocales avant d'entrer dans le conduit vocal. Les phonèmes tels que "A", "E", "I", "O", "U", "OU", "ON", "AN" mais aussi "J", "Z" ou "V", appartiennent à cette catégorie de sons. La vibration des cordes vocales produisant un signal périodique, il en résulte que les sons voisés sont de nature périodique et admettent ainsi une fréquence fondamentale $\nu_0 = 1/T_0$ que l'on nomme "pitch". Cette fréquence fondamentale varie en fonction de la classe d'âge et du sexe des individus (voir la table 1), mais également en fonction des émotions du locuteur. Le contenu harmonique du signal, c'est à dire l'amplitude des composantes fréquentielles multiples de la fréquence fondamentale ν_0 caractérise ensuite et locuteur et le phonème prononcé. Pour les sons non-voisés, par exemple les phonèmes "F", "S" et "CH", les cordes vocales ne vibrent pas. Seul le conduit vocal est responsable des caractéristiques du signal, jugé aléatoire, qui résulte de l'écoulement turbulent du flux d'air dans les différentes cavités. Comme illustré sur la figure 5, le même principe de modélisation par filtre générateur peut être utilisé pour ces deux types de sons, la différence se faisant sur le signal d'excitation x qui sera appliqué en entrée.

1.1 Acquisition de différents signaux de parole

Utiliser le fichier Simulink `acquisition.slx` pour faire l'acquisition de différents signaux de parole. Par défaut, la fréquence d'échantillonnage est fixée à $\nu_s = 16$ kHz.

1. Prononcer le son voisé de votre choix dans votre microphone puis lancer l'enregistrement. faire durer le son pendant 3 secondes. Stopper l'enregistrement avant d'arrêter de prononcer le son.

2. Les échantillons y_k du signal discret mesuré y^* sont automatiquement enregistrés dans le vecteur `yk` visible dans le `workspace` de Matlab. Sauvegarder le signal dans un fichier

`Son_Locuteur.mat` (exemple : iii_bob.mat pour un "i" prononcé par bob) à l'aide de la commande :
`» save Son_Locuteur.mat yk`
Ce fichier contenant les échantillons y_k du signal sera dès lors accessible à l'aide de la commande :
`» load Son_Locuteur.mat`

3. Répéter l'opération pour enregistrer plusieurs sons différents : des "I", des "U", des "O"..., mais aussi des fricatives non-voisés comme des "FF", des "SS" et des "CH".

1.2 Analyse des signaux

4. Créer un script matlab `AnalyseSignal.m` et récupérer un enregistrement à l'aide des commandes suivantes en ayant pris soin de préciser le nom du fichier à étudier :

```
load Son_Locuteur.mat         % mettre le nom de votre fichier

N = 2*floor(length(yk)/2);    % taille du signal (ech) (multiple de 2)
yk = yk(1:N);                 % troncature du signal à N échantillons
yk = yk/max(abs(yk));         % normalisation en amplitude

nus = 16000;     % fréquence d'échantillonnage (Hz)
Ts = 1/nus;      % période d'échantillonnage (sec)
```

5. Créer un vecteur des temps en seconde et tracer la représentation temporelle du signal y^*. Il est possible d'écouter le signal en utilisant la commande matlab `sound(yk,nus)`.

6. À l'aide de la fonction matlab `fft`, calculer puis tracer le spectre du signal y^* pour les fréquences comprises entre $-\nu_s/2$ et $\nu_s/2$. Observer également le spectre en utilisant une échelle logarithmique pour l'axe des ordonnés [1] en remplaçant la commande `plot` par la commande `semilogy`.

7. Observer les différents sons enregistrés et identifier les différences entre les sons voisés et non-voisés.

Fin du jalon 1 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

Pour la suite du TP, diviser votre groupe de travail en 2 pour étudier respectivement les sons voisés (Partie 2) et les sons non-voisés (Partie 3).

1. Utiliser une échelle logarithmique permet de représenter les valeurs du spectre en dB ($Y_{dB}(\nu) = 20 \log Y(\nu)$). Cette échelle est intéressante car elle est plus représentative de la puissance sonore perçue par l'oreille humaine.

2 Traitement des sons voisés

On s'intéresse dans cette partie à la compression/décompression des sons de parole voisés. Les signaux correspondants étant périodiques de période T_0, ils se modélisent comme la sortie d'un système de convolution discret dont la fonction de transfert est donnée par

$$F(z) = \frac{b_0}{1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}}. \tag{1}$$

prenant pour signal d'entrée un peigne de Dirac de paramètre T_0, soit

$$x^* = \mathrm{Pgn}_{T_0}\,. \tag{2}$$

La figure 6 illustre cette modélisation.

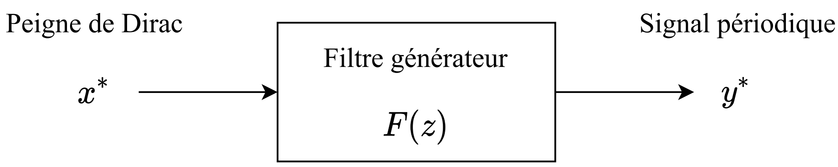

FIGURE 6 – Modélisation par filtre générateur d'un signal périodique.

2.1 Analyse préliminaire des sons voisés

8. Renommez votre script matlab `sonsvoises.m`. Vous le compléterez au fur et à mesure du travail réalisé dans cette partie.

9. Choisir un son voisé et déterminer sa fréquence fondamentale $\nu_0 = 1/T_0$. Le signal est-il strictement périodique ?

10. Extraire une fonction motif y_m^* du signal périodique y^* et tracer ses représentations temporelle et fréquentielle. Pour cela, extraire du vecteur `yk` un segment de N_0 échantillons tel que $T_0 = N_0 T_s$ à l'aide de la commande

```
ym = yk(n1:n1+N0-1);          % fonction motif
```

   Vous choisirez la valeur de `n1` de sorte que le premier échantillon de `ym` soit proche de 0.

11. Quel est le lien entre la réponse impulsionnelle h^* du filtre générateur et la fonction motif y_m^* du signal à modéliser ?

12. Quel est le lien entre la réponse fréquentielle H^* du filtre générateur et la transformée de Fourier Y_m^* de la fonction motif ?

Fin du jalon 2 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

2.2 Compression : identification des paramètres du filtre

La méthode de Yule-Walker permet de déterminer les coefficients $\{a_i\}_{i=1,\ldots,n_a}$ de la fonction de transfert du filtre générateur à partir des échantillons de la fonction motif du signal enregistré. Cette méthode d'identification est implémentée dans la fonction Matlab `myaryule` qui sera utilisée dans la suite via la commande :

```
[b0,a] = myaryule(ym,na,'det');
```

dans laquelle :

- `ym` contient les échantillons de la fonction motif ;
- `na` est l'ordre du filtre générateur et doit être défini par l'utilisateur ;
- `'det'` permet de préciser que le signal à modéliser est déterministe ;
- `a` contient les $n_a + 1$ coefficients $\{a_i\}_{i=0,\ldots,n_a}$ avec $a_0 = 1$;
- `b0` est le gain b_0 du filtre.

13. À partir de l'observation du spectre du signal y_m^* à modéliser, proposer une valeur pour l'ordre n_a du filtre.
14. Appliquer la méthode de Yule-Walker sur le signal y_m^* pour identifier les paramètres de la fonction de transfert du filtre générateur. À l'aide de la commande `roots(a)`, vérifier que le filtre obtenu est bien stable.
15. A l'aide de la fonction `freqz`[2], calculer et représenter la réponse fréquentielle H^* du filtre générateur et la comparer avec le résultat attendu.
16. Donner l'expression du taux de compression obtenu avec cette technique pour un signal voisé de N échantillons et un filtre générateur d'ordre n_a.

Fin du jalon 3 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

2.3 Décompression : synthèse du signal

On s'intéresse dans cette partie à la décompression de l'information qui consiste en la synthèse d'un signal discret y_{synt}^*, périodique de période T_0, à partir du filtre générateur donné en (1) dont les paramètres ont été identifiés précédemment et du signal d'entrée x^* donné en (2).

17. Donner l'équation de récurrence qui permet de calculer les échantillons du signal y_{synt}^* à partir des paramètres b_0 et $\{a_i\}_{i=1,\ldots,n_a}$.

2.3.1 Méthode stationnaire

18. Créer un vecteur `xk` contenant $N = T_a/T_s$ échantillons (avec T_a la durée totale du signal à générer) d'un peigne de Dirac discret de paramètre T_0.

---

2. La fonction matlab `freqz` permet de calculer la réponse fréquentielle H^* d'un filtre numérique possédant une fonction de transfert F rationnelle, soit telle que

$$F(z) = \frac{B(z)}{A(z)} = \frac{\sum_{j=0}^{n_b} b_j z^{-j}}{\sum_{i=0}^{n_a} a_i z^{-i}}$$

Elle s'utilise avec la syntaxe `Hstar=freqz(B,A,nu,nus)` dans laquelle :

— `B` contient le vecteur $[b_0, \ldots, b_{n_b}]$ des coefficients du polynôme au numérateur ;

— `A` contient le vecteur $[a_0, \ldots, a_{n_a}]$ des coefficients du polynôme au dénominateur ;

— `nu` est le vecteur des fréquences sur lesquelles la réponse fréquentielle du filtre sera calculée ;

— `nus` est la fréquence d'échantillllonnage ;

Le vecteur `Hstar` obtenu en sortie est de même taille que le vecteur des fréquences `nu`.

19. En utilisant la fonction matlab `filter`[3], calculer le signal y^*_{synt} obtenu en sortie du filtre générateur et tracer ses représentations temporelle et fréquentielle. Comparer le résultat avec le signal initial y^*.

20. Écouter le signal synthétisé y_{synt} en utilisant la commande matlab `sound(ysynt,nus)`. Le résultat est-il satisfaisant ? Analyser les résultats obtenus en modifiant l'ordre n_a du filtre générateur et pour différents sons voisés.

2.3.2 Adaptation aux variations temporelles

En annexe A est décrit le principe de construction d'une représentation temps-fréquence appelée spectrogramme. Cette représentation permet d'observer l'évolution temporelle du contenu fréquentiel d'un signal. À titre d'exemple, la figure 9 présente le spectrogramme du signal de parole "efface".

21. À partir des lignes de commande suivantes, calculer et tracer les spectrogrammes du signal de parole original y^* et du signal synthétisé y^*_{synt} et observer leurs différences.

```
Tw = 0.03;                  % durée de la fenêtre d'analyse (sec)
Nw = 3*round(Tw/Ts/3);  % taille de la fenetre (ech) (multiple de 3)
figure, spectrogram(y,Nw,Nw/3,2*Nw,nus,'yaxis');
```

22. En s'inspirant du principe de construction du spectrogramme, proposer une adaptation de la méthode de synthèse précédente permettant de prendre en compte l'aspect temps-variant des signaux de parole. On choisira une fenêtre d'analyse de Hanning de durée 30 msec et un décalage de 10 msec.

Fin du jalon 4 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

3. La fonction matlab `filter` permet de calculer la sortie d'un filtre pour un signal d'entrée donné. Elle s'utilise selon la syntaxe `ysynt = filter(B,A,x);` dans laquelle `B` et `A` sont des vecteurs contenant les coefficients des polynômes du numérateur et du dénominateur de la fonction de transfert du filtre, et `x` est un vecteur contenant les échantillons du signal d'entrée. Le vecteur `ysynt` obtenu en sortie est de même taille que le signal d'entrée.

3 Traitement des sons non-voisés

On s'intéresse dans cette partie à la compression/décompression des sons de parole non-voisés. Les signaux correspondants étant aléatoires, ils se modélisent comme la sortie d'un système de convolution discret dont la fonction de transfert est donnée par

$$F(z) = \frac{b_0}{1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}}. \tag{3}$$

prenant pour signal d'entrée une réalisation b^* d'un bruit blanc gaussien de variance 1, soit

$$x^* = b^*. \tag{4}$$

La figure 7 illustre cette modélisation.

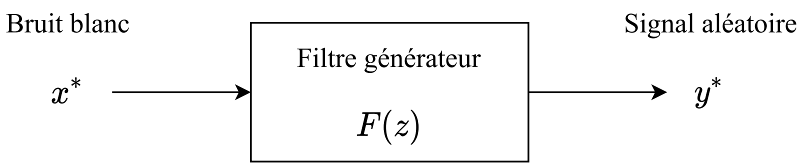

FIGURE 7 – Modélisation par filtre générateur d'un signal aléatoire.

3.1 Analyse préliminaire des sons non-voisés

8. Renommer votre script matlab `sonsnonvoises.m`. Vous le compléterez au fur et à mesure du travail réalisé dans cette partie.
9. À l'aide de la fonction matlab `xcorr`[4] calculer puis représenter la fonction d'autocorrélation $R_{y^*}(\tau)$ du signal discret y^*. Le signal aléatoire y^* est-il une réalisation de bruit blanc ?
10. À l'aide de la fonction matlab `fft`, calculer puis représenter la densité spectrale de puissance $S_{y^*}(\nu)$ du signal discret y^*. Le signal aléatoire y^* est-il une réalisation de bruit blanc ?
11. Quel est le lien entre la réponse fréquentielle H^* du filtre générateur et la densité spectrale de puissance $\gamma_{\mathbf{Y}^*}(\nu)$ du processus aléatoire $\mathbf{Y}^*$ dont y^* est une réalisation particulière ?

Fin du jalon 2 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

---

4. La fonction matlab `xcorr` permet de calculer les échantillons de la fonction d'autocorrélation d'un signal discret. Dans le cas d'un signal y^* discret à puissance finie (mais mesurés sur un intervalle de temps fini $T_a = NT_s$), elle s'utilise avec la syntaxe `Ry = xcorr(yk,'biased');` dans laquelle le vecteur `yk` contient les N échantillons du signal, et le vecteur `Ry` contient $2N-1$ échantillons $R_y(n)$, $n \in \{-(N-1), ..., N-1\}$, calculés selon :

$$\begin{cases} R_y(n) = \dfrac{1}{N} \displaystyle\sum_{k=0}^{N-1-n} y_{k+n} y_k & \text{pour } n = 0, ..., N-1 \\ R_y(n) = R_y(-n) & \text{pour } n = -(N-1), ..., -1 \end{cases}$$

Il s'agit d'une estimation biaisée de la fonction d'autocorrélation de y^* car on ne fait pas tendre N vers l'infini, et la division par N ne correspond pas au nombre de termes de la somme (sauf pour $n = 0$).

3.2 Compression : identification des paramètres du filtre

La méthode de Yule-Walker permet de déterminer les coefficients $\{a_i\}_{i=1,...,n_a}$ de la fonction de transfert du filtre générateur à partir des échantillons du signal enregistré. Cette méthode d'identification est implémentée dans la fonction Matlab `myaryule` qui sera utilisée dans la suite via la commande :

```
[b0,a] = myaryule(yk,na,'rand');
```

dans laquelle :

- `yk` contient les échantillons du signal à modéliser ;
- `na` est l'ordre du filtre générateur et doit être défini par l'utilisateur ;
- `'rand'` permet de préciser que le signal à modéliser est aléatoire ;
- `a` contient les $n_a + 1$ coefficients $\{a_i\}_{i=0,...,n_a}$ avec $a_0 = 1$;
- `b0` est le gain b_0 du filtre.

12. À partir de l'observation de la densité spectrale de puissance du signal y^* à modéliser, proposer une valeur pour l'ordre n_a du filtre.
13. Appliquer la méthode de Yule-Walker sur le signal y^* pour identifier les paramètres de la fonction de transfert du filtre générateur. À l'aide de la commande `roots(a)`, vérifier que le filtre obtenu est bien stable.
14. A l'aide de la fonction `freqz`[5], calculer et représenter la réponse fréquentielle H^* du filtre générateur et la comparer avec le résultat attendu.
15. Donner l'expression du taux de compression obtenu avec cette technique pour un signal non-voisé de N échantillons et un filtre générateur d'ordre n_a.

Fin du jalon 3 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

3.3 Décompression : synthèse du signal

On s'intéresse dans cette partie à la décompression de l'information qui consiste en la synthèse d'un signal aléatoire discret y^*_{synt} à partir du filtre générateur donné en (3) dont les paramètres ont été identifiés précédemment et du signal d'entrée x^* donné en (4).

16. Donner l'équation de récurrence qui permet de calculer les échantillons du signal y^*_{synt} à partir des paramètres b_0 et $\{a_i\}_{i=1,...,n_a}$.

5. La fonction matlab `freqz` permet de calculer la réponse fréquentielle H^* d'un filtre numérique possédant une fonction de transfert F rationnelle, soit telle que

$$F(z) = \frac{B(z)}{A(z)} = \frac{\sum_{j=0}^{n_b} b_j z^{-j}}{\sum_{i=0}^{n_a} a_i z^{-i}}$$

Elle s'utilise avec la syntaxe `Hstar=freqz(B,A,nu,nus)` dans laquelle :

— `B` contient le vecteur $[b_0, ..., b_{n_b}]$ des coefficients du polynôme au numérateur ;

— `A` contient le vecteur $[a_0, ..., a_{n_a}]$ des coefficients du polynôme au dénominateur ;

— `nu` est le vecteur des fréquences sur lesquelles la réponse fréquentielle du filtre sera calculée ;

— `nus` est la fréquence d'échantilllonnage ;

Le vecteur `Hstar` obtenu en sortie est de même taille que le vecteur des fréquences `nu`.

3.3.1 Méthode stationnaire

17. En utilisant la fonction `randn`, créer un vecteur `xk` contenant $N = T_a/T_s$ échantillons (avec T_a la durée totale du signal à générer) d'une réalisation de bruit blanc gaussien de variance 1.

18. En utilisant la fonction matlab `filter`[6], calculer le signal y^*_{synt} obtenu en sortie du filtre générateur et tracer ses représentations temporelle et fréquentielle. Comparer le résultat avec le signal initial y^*.

19. Écouter le signal synthétisé y^*_{synt} en utilisant la commande matlab `sound(ysynt,nus)`. Le résultat est-il satisfaisant ? Analyser les résultats obtenus en modifiant l'ordre n_a du filtre générateur et pour différents sons non-voisés.

3.3.2 Adaptation aux variations temporelles

En annexe A est décrit le principe de construction d'une représentation temps-fréquence appelée spectrogramme. Cette représentation permet d'observer l'évolution temporelle du contenu fréquentiel d'un signal. À titre d'exemple, la figure 9 présente le spectrogramme du signal de parole "efface".

20. À partir des lignes de commande suivantes, calculer et tracer les spectrogrammes du signal de parole original y^* et du signal synthétisé y^*_{synt} et observer leur différence.

```
Tw = 0.03;                 % durée de la fenêtre d'analyse (sec)
Nw = 3*round(Tw/Ts/3);     % taille de la fenetre (ech) (multiple de 3)
figure, spectrogram(y,Nw,Nw/3,2*Nw,nus,'yaxis');
```

21. En s'inspirant du principe de construction du spectrogramme, proposer une adaptation de la méthode de synthèse nominale permettant de prendre en compte l'aspect temps-variant des signaux de parole. On choisira une fenêtre d'analyse de Hanning de durée 30 msec et un décalage de 10 msec.

Fin du jalon 4 - veuillez faire le bilan du travail réalisé dans cette section, mettre à jour le code matlab correspondant, puis appeler l'enseignant.

6. La fonction matlab `filter` permet de calculer la sortie d'un filtre pour un signal d'entrée donné. Elle s'utilise selon la syntaxe `ysynt = filter(B,A,x);` dans laquelle `B` et `A` sont des vecteurs contenant les coefficients des polynômes du numérateur et du dénominateur de la fonction de transfert du filtre, et `x` est un vecteur contenant les échantillons du signal d'entrée. Le vecteur `ysynt` obtenu en sortie est de même taille que le signal d'entrée.

Annexes

A Le spectrogramme : une représentation conjointe en temps et en fréquence

La transformée de Fourier considère un signal sur l'ensemble de son support temporel pour faire apparaitre l'amplitude et la phase des composantes fréquentielles qui le composent. Cette analyse de Fourier est cependant mal adaptée aux signaux non-stationnaires, c'est-à-dire pour lesquels le contenu fréquentiel évolue au cours du temps. En effet, à la lecture de la représentation graphique d'une transformée de Fourier, il est difficile de savoir quelles composantes fréquentielles sont présentes à un instant donné et lesquelles ne le sont pas. Les représentations temps-fréquence, comme le spectrogramme, permettent de rendre explicite l'évolution temporelle du contenu fréquentiel d'un signal.

Un spectrogramme est une collection de Transformées de Fourier Discrètes (TFD) appliquées sur des petits segments successifs d'un signal (voir figure 8). Chaque segment est obtenu par application d'une fenêtre temporelle permettant de localiser temporellement l'analyse de Fourier et donc de préserver l'information de temps. La TFD appliquée sur chaque segment en donnera le contenu fréquentiel. Mises cotes à cotes, cette collection de TFD forme une représentation conjointe en temps et en fréquence qui permet de représenter de façon intuitive les signaux non-stationnaires. On représente généralement le spectrogramme sous la forme d'une image graduée en temps (abscisse) et en fréquence (ordonnée) et dans laquelle le module au carré de chaque TFD est représenté suivant un code couleur. À titre d'exemple, la figure 9 présente le spectrogramme du mot "efface". On y reconnait distinctement la succession des différents phonèmes "É", "F", "A", "S" et "E" dont le contenu fréquentiel diffère de l'un à l'autre.

Pour construire un spectrogramme, trois paramètres doivent être choisis (voir figure 8) :

— la forme de la fenêtre d'analyse w : Hanning, Hamming, gaussienne ou autre. Le fenêtrage temporel implique une convolution dans le domaine des fréquences. Le choix de la fenêtre résulte du compromis entre finesse du lobe principal et hauteur des lobes secondaires de sa transformée de Fourier.

— la taille T_w de la fenêtre d'analyse : elle détermine la résolution fréquentielle de la représentation temps-fréquence qui vaut $1/T_w$. Augmenter la taille de la fenêtre permet d'augmenter la résolution fréquentielle, mais diminue la localisation temporelle de l'analyse.

— le décalage D entre les temps initiaux de deux segments successifs : il détermine la graduation temporelle de la représentation temps-fréquence, c'est-à-dire son axe des temps. Le choix de ce décalage résulte du compromis entre continuité temporelle de la représentation et temps de calcul.

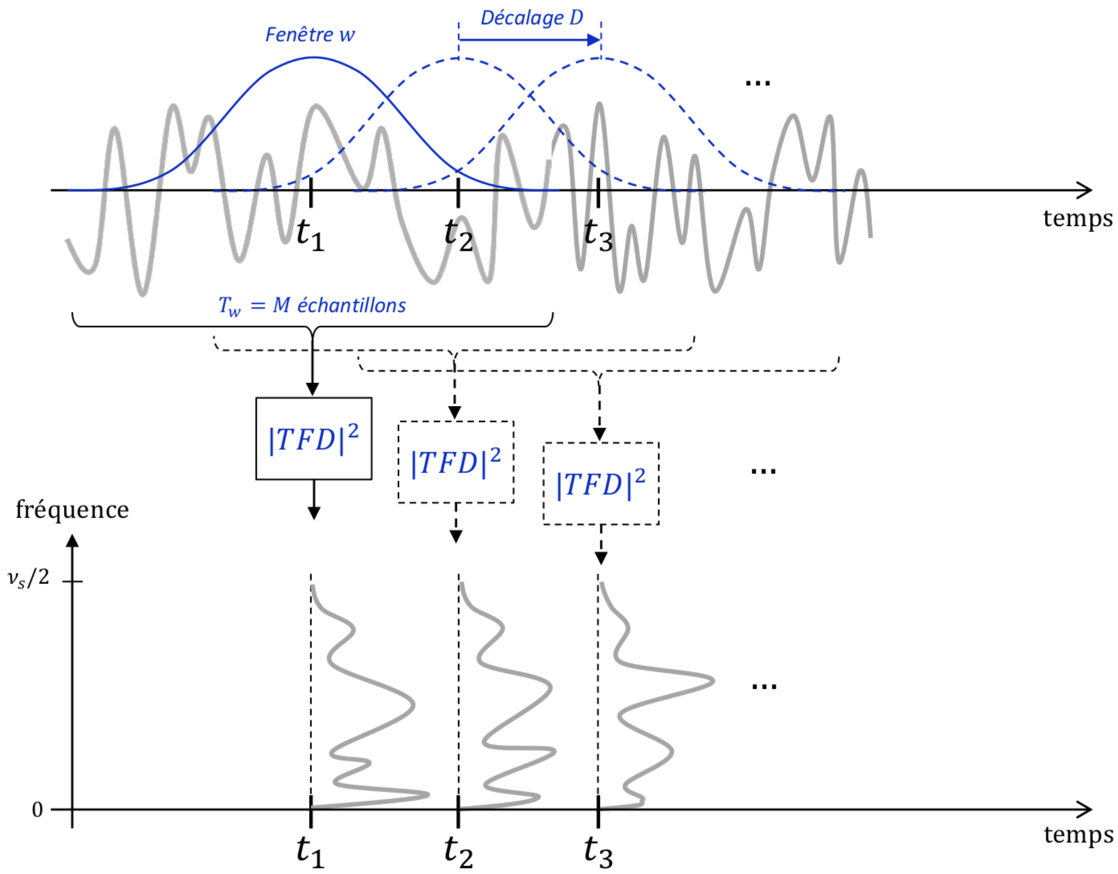

FIGURE 8 – Principe de construction d'un spectrogramme : transformée de Fourier à court terme avec fenêtre glissante

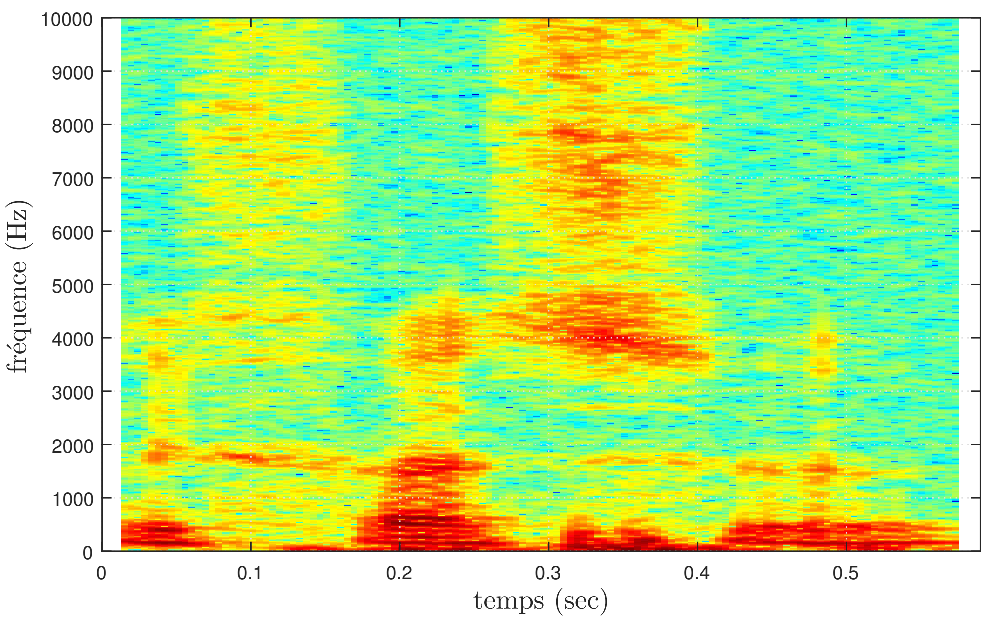

FIGURE 9 – Spectrogramme du mot "efface". Paramètres de construction : fenêtre de Hamming de durée $T_w = 30$ msec, décalage $D = 10$ msec.