
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 tension électrique (signal analogique) par l'intermédiaire d'un microphone. Grâce à un convertisseur Analogique-Numérique (CAN) et un processeur de signaux (Digital Signal Processor, DSP), ce signal 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'un vecteur de données binaires. Ces données sont ensuite 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

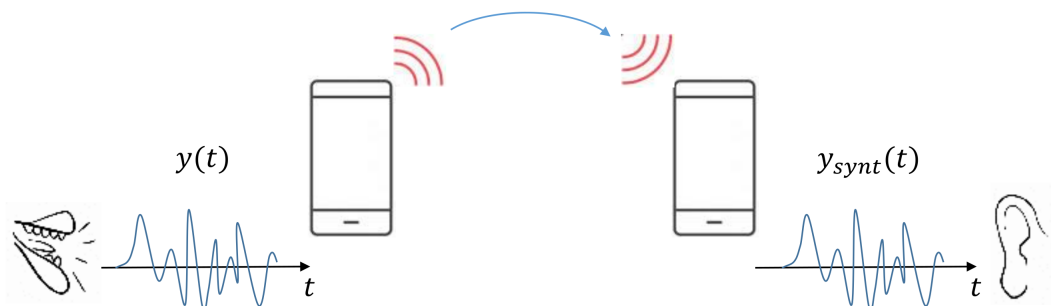


Figure 1: Transmission de la parole

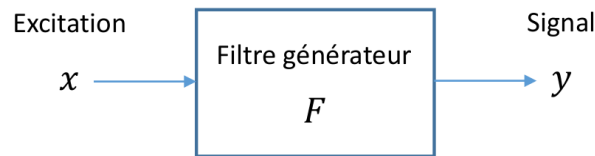


Figure 2: Notion de filtre générateur pour la modélisation d'un signal

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 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 (Partie 1). Il permet aussi de donner quelques premiers éléments sur les signaux de parole (Partie 2) en abordant tour à tour la modélisation des sons voisés, de nature pseudo-périodique (Partie 3), et des sons non-voisés, de nature aléatoire (Partie 4).

Le sujet à traiter comprend un ensemble de jalons qui sont donnés ici avec un temps de passage indicatif :

- Jalon 1 : section 1 (~ 30mn)
- Jalon 2 : section 3.2 (~ 2h00)
- Jalon 3 : section 3.3 (~ 3h00)
- Jalon 4 : section 3.4 (~ 4h00)

La partie 4 peut être traitée en parallèle à partir de la section 3.2. À la fin de chaque jalon, un point sera réalisé avec l'enseignant.

1 Principe de compression/décompression par filtre générateur

On considère un signal discret $y^* = \sum_{k=0}^{N-1} y_k \delta_{kT_s}$ de durée finie $T_a = NT_s = 30$ msec avec $N = 240$ le nombre d'échantillons et $\nu_s = 1/T_s = 8$ kHz la fréquence d'échantillonnage.

1. En admettant que la valeur de chaque échantillon du signal est codée sur 8 bits, combien de bits seront nécessaires pour représenter le signal y^* ?

On considère maintenant le système de convolution de réponse impulsionnelle $h^* = y^*$ pour lequel on admet que la fonction de transfert $F(z)$ est donnée par

$$F(z) = \frac{b_0}{1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}} \quad (1)$$

avec $n_a = 20$. Le système F est un filtre générateur du signal y^* : il est possible d'obtenir les échantillons y_k à partir des paramètres b_0 et $\{a_i\}_{i=1:n_a}$ du filtre. On transmet à travers le réseau mobile les $n_a + 1$ paramètres de ce modèle.

2. En admettant que chaque coefficient b_0 et a_i est codé sur 8 bits, combien de bits seront nécessaires pour représenter l'ensemble des paramètres du filtre générateur ?
3. En déduire le taux de compression obtenu lorsque l'on remplace la transmission des échantillons y_k du signal par celle des paramètres b_0 et a_i du filtre générateur.

Au niveau du récepteur, le signal y^* est reconstruit comme la sortie du filtre générateur pour un signal d'excitation x^* (cf Figure 2). C'est l'étape de synthèse.

4. Donner l'équation de récurrence qui permet de calculer les échantillons $\{y_k\}_{k=0:N-1}$ à partir des paramètres b_0 et $\{a_i\}_{i=1:n_a}$.

Le taux de compression obtenu à la question 3 permet de comprendre l'intérêt de remplacer la transmission des N échantillons $\{y_k\}_{k \in \{0:N-1\}}$ du signal par la transmission des $n_a + 1$ paramètres de la fonction de transfert F du filtre générateur. Il reste donc la question de l'étape d'analyse, à savoir la détermination, pour un signal y^* donné, des paramètres du modèle, à savoir la nature du signal d'excitation x^* et les paramètres de F . Pour cela il est nécessaire d'étudier les caractéristiques des signaux de parole sur lesquels quelques éléments sont donnés dans la partie 2.

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.

2 Le signal de parole

Les sons de la parole sont d'une grande diversité et varient d'une langue à l'autre, d'un individu à l'autre et le choix de leur enchainement permet de créer des mots. 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.

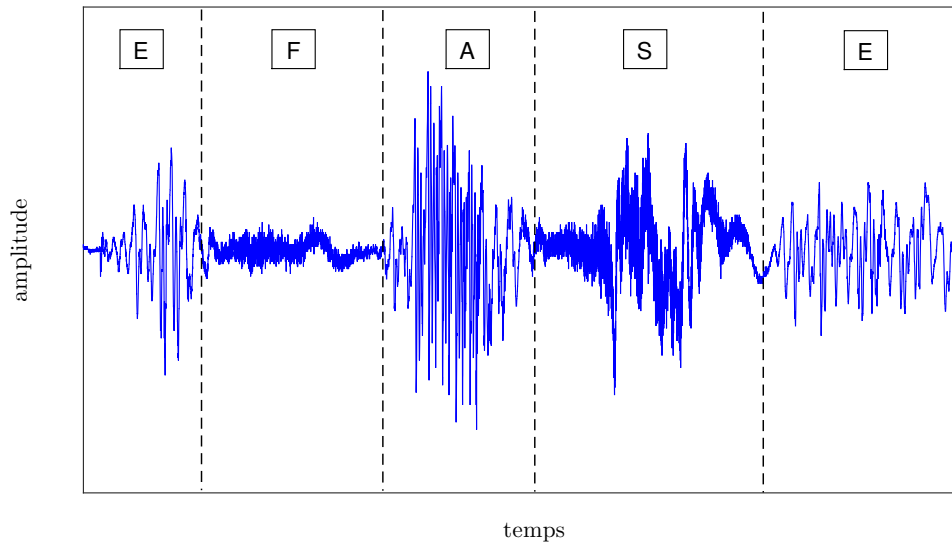


Figure 3: Variation de pression en fonction du temps pour le mot "efface"

La modélisation des signaux de parole nécessite donc de pouvoir prendre en compte cette diversité et dans ce but, on s'est attaché à 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 ensuite 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 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 "boîte 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

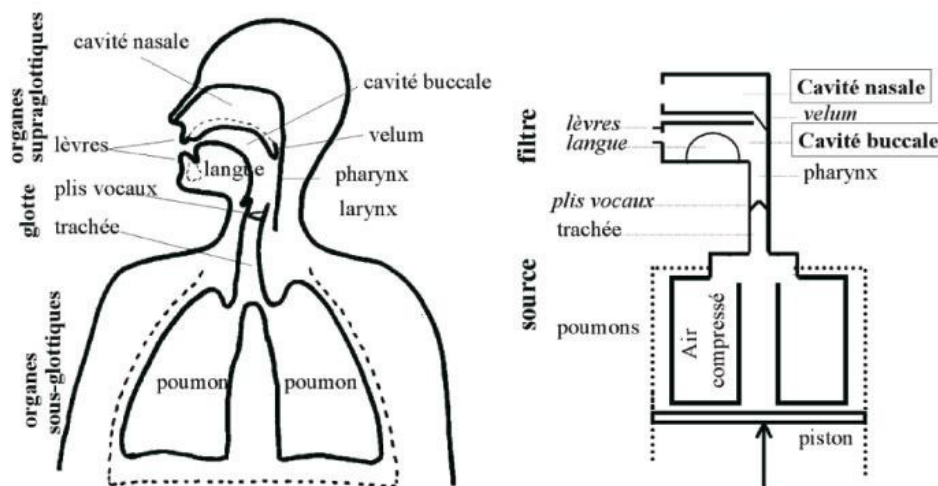


Figure 4: Schéma des organes de production de la parole et modèle source-filtre (d'après <http://www.cairn.info/la-phonetique>)

	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

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. 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 complexe du flux d'air dans les différentes cavités.

Comme illustré sur la figure 5, le même principe de modélisation par filtre générateur sera appliqué pour ces deux types de sons, la différence se faisant sur le signal d'excitation x appliqué en entrée.

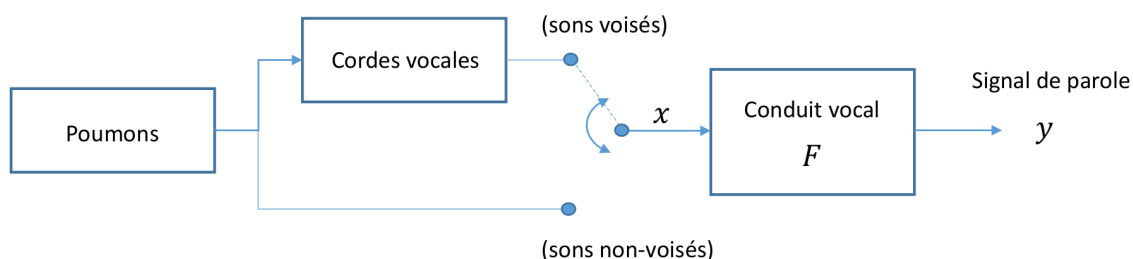


Figure 5: Filtre générateur pour la production des sons voisés et non-voisés

3 Traitement des sons voisés

On s'intéresse dans cette partie à la compression/décompression des sons de parole voisés. Le script matlab `sonsvoises.m` contient le squelette du travail à réaliser pour cette partie. Vous le complétez au fur et à mesure des questions.

3.1 Acquisition d'un son

Utiliser le fichier Simulink `acquisition.slx` pour faire l'acquisition de sons voisés. Celui-ci nécessite de préciser, dans le bloc "From Audio Device", la fréquence d'échantillonnage ν_s des signaux.

5. L'acquisition se fait avec une fréquence d'échantillonnage $\nu_s = 22050$ Hz. Justifier ce choix.
6. Prononcer un son voisé de votre choix dans votre microphone puis lancer l'enregistrement. Prononcer le son pendant 3 ou 4 secondes. Stopper l'enregistrement avant d'arrêter de prononcer le son.
Ne pas hésiter à enregistrer des sons différents : des "A", des "I", des "O"... De même, ne pas hésiter à enregistrer aussi des sons modulés en fréquence en ajoutant une intonation : ces sons illustreront bien l'aspect temps-variant observé plus tard sur leur spectrogramme.

7. Le résultat de l'acquisition est placé dans la variable `simout` dans le `workspace` de Matlab. Pour chaque acquisition, enregistrer cette variable `simout` dans un fichier `NomDuSon_Locuteur.mat` (exemple : `aaa-bob.mat` pour un "a" prononcé par bob) à l'aide de la commande :

```
>> save NomDuSon_Locuteur.mat simout
```

Le contenu de ce fichier sera dès lors accessible ultérieurement à l'aide de la commande :

```
>> load NomDuSon_Locuteur.mat
```

3.2 Analyse du signal

8. Lancer le script matlab `sonsvoises.m` en ayant pris soin de préciser le nom du fichier à étudier. Observer le signal. Est-il périodique ? Si oui, pourquoi ? Si oui, déterminer sa période T_0 .
9. A l'aide de la fonction matlab `fft`, calculer puis tracer le spectre du signal pour les fréquences comprises entre $-\nu_s/2$ et $\nu_s/2$. Vérifier que le spectre obtenu est celui d'un signal périodique et retrouver sa fréquence fondamentale $\nu_0 = 1/T_0$ (le pitch). Le signal périodique est-il éloigné d'un sinus ? Le signal est-il strictement périodique ?
10. Isoler un motif y_m sur une période $T_0 = N_0 T_s$ du signal périodique y . Dans le choix du signal motif, veiller à ce que celui-ci commence et finisse à 0. Calculer puis tracer son spectre en le superposant au spectre du signal complet y . Qu'observe-t-on ? Était-ce prévisible ? Donner la relation théorique entre le spectre d'un signal périodique et le spectre de sa 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.

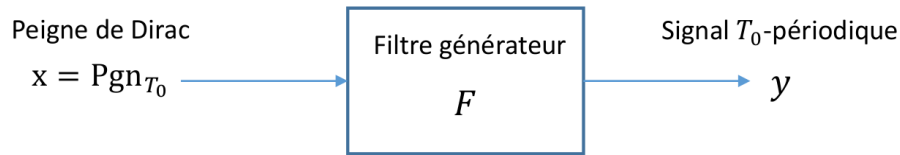


Figure 6: Notion de filtre générateur dans le cas d'un signal périodique

3.3 Compression : modélisation par filtre générateur

On s'attache dans cette section à modéliser les signaux de parole voisés à l'aide d'un filtre générateur. Un signal y périodique de période T_0 se modélise comme la réponse d'une fonction de transfert stable F à un peigne de Dirac de paramètre T_0 (voir figure 6), soit

$$x = \text{Pgn}_{T_0}. \quad (2)$$

11. Rappeler pourquoi la modélisation par filtre générateur utilisant pour signal d'entrée une impulsion de Dirac, soit avec $x = \delta$, n'est pas adaptée aux signaux périodiques.
12. Quel est le lien entre la fonction motif y_m du signal périodique y et la réponse impulsionnelle h du filtre générateur F ? En déduire la relation entre le spectre de la fonction motif y_m et la réponse fréquentielle H du filtre générateur F .

Dans la suite, nous considérons le signal discret y^* , échantillonné à la période T_s sous l'hypothèse que $T_0 = N_0 T_s$ avec $N_0 \in \mathbb{N}$. Le filtre générateur discret est représenté par sa fonction de transfert F définie selon

$$F(z) = \frac{1}{1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}}. \quad (3)$$

La modélisation du signal discret y^* se ramène ainsi à l'identification des coefficients $\{a_i\}_{i=1:n_a}$ du filtre. La méthode de Yule-Walker présentée en annexe A (appliquée ici sur des signaux déterministes) permet d'obtenir ces coefficients à partir des N_0 échantillons $\{y_{mk}\}_{k=0:N_0-1}$ du signal motif. Cette méthode d'identification est implémentée dans la fonction Matlab `aryule` qui sera utilisée dans la suite via la commande

```
>> a = aryule(y_m, n_a);
```

dans laquelle l'ordre du filtre n_a doit être défini par l'utilisateur.

13. En observant le spectre du signal motif y_m , justifier le choix $n_a = 20$ pour l'ordre du filtre.
14. Effacer la commande matlab `return` pour accéder à la suite du script `sonsvoises.m`. A l'aide de la fonction `aryule`, appliquer la méthode de Yule-Walker pour identifier les coefficients du filtre générateur F . Le filtre obtenu est-il stable ?
15. A l'aide de la fonction `filter` de matlab, calculer et tracer la réponse impulsionnelle h^* du filtre générateur F et la comparer avec les résultats attendus. Calculer et représenter la réponse fréquentielle $H^*(\nu)$ du filtre générateur F . Tester votre programme pour plusieurs signaux.

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.

3.4 Décompression : synthèse artificielle

On s'intéresse dans cette partie à la décompression du signal de parole qui consiste en la synthèse d'un signal discret y_{synt}^* périodique de période T_0 à partir du filtre générateur F identifié précédemment et du signal d'entrée x donné en (2).

3.4.1 Méthode nominale

16. Effacer la commande matlab `return` pour accéder à la suite du script `sonsvoises.m`. Créer un vecteur \mathbf{x} contenant N échantillons d'un peigne de Dirac de pas T_0 .
17. Avec la fonction matlab `filter`, calculer le signal y_{synt}^* en sortie du filtre générateur. Observer les représentations temporelle et fréquentielle du signal obtenu. Comparer avec le signal initial y .
18. Écouter le signal synthétisé y_{synt} en utilisant la commande matlab `>> sound(ysynt,nus)`. Le résultat est-il satisfaisant ? Effectuer ces mêmes opérations en changeant le signal initial y , le signal motif y_m et l'ordre n_a du filtre. Commenter les résultats obtenus.
19. À partir de leur représentation temporelle, observer les variations de l'enveloppe, du pitch et de la forme des motifs des signaux de parole enregistrés. Comparer avec les signaux synthétisés.

3.4.2 Adaptation au temps-réel

20. Effacer la commande matlab `return` dans la suite du script `sonsvoises.m` pour observer les spectrogrammes du signal de parole initial y et du signal synthétisé y_{synt} et compléter votre analyse apportée à la question précédente. Des éléments d'explication sur le spectrogramme et sa construction sont donnés en annexe B.
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 (enveloppe, pitch et motif) des signaux de parole dans un cadre temps-réel.
22. En supposant que chaque paramètre du modèle est codé sur 8 bits, combien de bits seront nécessaires pour transmettre 1 sec de parole échantillonnée à 16000 Hz ? Calculer le taux de compression ainsi obtenu.

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.

4 Traitement des sons non-voisés

Dans cette seconde partie du TP, on s'intéresse à la compression/décompression des sons non-voisés. Une démarche identique à celle de la partie précédente est adoptée : analyse des caractéristiques des signaux, modélisation par filtre générateur puis synthèse. Pour cette partie concernant les signaux de parole non-voisés, vous utiliserez le script matlab `sonsnonvoises.m` que vous complèterez au fur et à mesure des questions.

4.1 Acquisition et analyse du signal

23. Répéter les opérations effectuées avec les sons voisés afin d'enregistrer quelques sons non-voisés de votre choix.
24. A partir du script matlab `sonsnonvoises.m`, charger un fichier contenant un de vos signaux de parole non-voisé et observer sa représentation temporelle.
25. Calculer puis représenter sur l'intervalle de fréquences $[-\nu_s/2; \nu_s/2]$ le spectre $Y(\nu)$ du signal y .
26. Observer de la même manière différents enregistrements de sons non-voisés. Quelle semble-t-être la nature de ces signaux ?
27. A l'aide de la fonction matlab `xcorr` avec l'option `biased`¹, calculer puis représenter la fonction d'autocorrélation du signal y . A l'aide de la fonction `fft`, calculer puis représenter une estimation de sa DSP $S_y(\nu)$.

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.

4.2 Compression : modélisation par filtre générateur

Comme observé sur la figure 3 pour le mot "efface" et pour les signaux précédents, les sons non-voisés tels que "F", "S" ou "CH" présentent un caractère aléatoire. Ceci incite naturellement à les modéliser sur la base de signaux aléatoires. La modélisation consiste alors à représenter le signal de parole comme la sortie d'un filtre générateur F prenant pour signal d'entrée x une réalisation temporelle de bruit blanc de puissance σ^2 (voir figure 7).

¹Pour un signal discret x^* à puissance finie, l'autocorrélation est un signal discret de pas T_s dont les échantillons $R_x(n)$ sont définies par :

$$R_x(n) = \lim_{N_{pt} \rightarrow \infty} \left(\frac{1}{2N_{pt} + 1} \sum_{k=-N_{pt}}^{+N_{pt}} x_{k+n} \cdot x_k \right).$$

Dans le cas où l'on ne dispose que de N échantillons du signal, une estimation dite "biaisée" de la fonction d'autocorrélation est obtenue par la formule suivante :

$$\begin{cases} n \geq 0 & \hat{R}_x(n) = \frac{1}{N} \sum_{k=0}^{N-n-1} x_{k+n} \cdot x_k \\ n < 0 & \hat{R}_x(n) = \hat{R}_x(-n) \end{cases} \quad (4)$$

Dans cette formulation, les échantillons $\hat{R}_x(n)$ seront d'autant plus proche des échantillons exactes $R_x(n)$ que n est faible. Cette estimateur s'obtient via la commande

```
>> Rx = xcorr(x, 'biased')
```

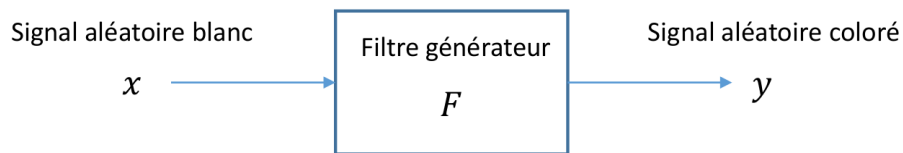


Figure 7: Notion de filtre générateur dans le cas d'un signal aléatoire

28. Donner la relation entre la densité spectrale de puissance $S_y(\nu)$ du signal aléatoire y , et la réponse fréquentielle $H(\nu)$ du filtre générateur.
29. A l'aide de la commande matlab `[a,sigma2]=aryule(y,na)`, identifier les coefficients $\{a_i\}_{i=1:n_a}$ de la fonction de transfert du filtre générateur F ainsi que la puissance σ^2 du bruit blanc d'excitation. Tracer le module de la réponse fréquentielle $H(\nu)$ du filtre générateur obtenu.
30. Vérifiez que la relation donnée à la question 28 est respectée.

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.

4.3 Décompression : synthèse artificielle

31. A l'aide de la fonction matlab `randn`, créer un vecteur \mathbf{x} de N échantillons contenant une réalisation de bruit blanc gaussien de variance σ^2 . A partir de ce signal d'excitation et du filtre générateur précédemment identifié, synthétiser un signal y_{synt} correspondant à un son non-voisé. Écouter le résultat. Est-ce satisfaisant ? Comparer dans le domaine du temps et des fréquences le signal synthétisé y_{synt} avec le signal initial y .
32. Répéter l'opération pour différentes réalisations du signal d'excitation x , puis pour différents sons non-voisés y .

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.

Annexes

A Méthode de Yule-Walker pour l'identification d'un filtre générateur

A.1 Signal déterministe

Soit un signal discret y^* déterministe dont on a mesuré N échantillons $y_{kk \in \{0, \dots, N\}}$. On suppose que ce signal discret correspond à la réponse impulsionnelle du système d'une fonction de transfert

$$F(z) = \frac{1}{1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}}$$

dont on connaît l'ordre n_a mais pour laquelle on ne connaît pas les coefficients $\{a_i\}_{i=1:n_a}$. L'objectif de la méthode de Yule-Walker est de calculer les coefficients $\{a_i\}_{i=1:n_a}$ à partir des échantillons $y_{kk \in \{0, \dots, N\}}$.

Le calcul consiste à résoudre le système de n_a équations linéaires à n_a inconnues :

$$\Gamma \underline{a} = \underline{\gamma} \tag{5}$$

avec

$$\Gamma = \begin{bmatrix} \hat{R}_y(0) & \hat{R}_y(1) & \hat{R}_y(2) & \dots & \dots & \dots & \dots & \hat{R}_y(n_a - 1) \\ \hat{R}_y(1) & \hat{R}_y(0) & \hat{R}_y(1) & \hat{R}_y(2) & \dots & \dots & \dots & \hat{R}_y(n_a - 2) \\ \hat{R}_y(2) & \hat{R}_y(1) & \hat{R}_y(0) & \hat{R}_y(1) & \hat{R}_y(2) & \dots & \dots & \hat{R}_y(n_a - 3) \\ \hat{R}_y(3) & \hat{R}_y(2) & \hat{R}_y(1) & \hat{R}_y(0) & \hat{R}_y(1) & \dots & \dots & \hat{R}_y(n_a - 4) \\ \hat{R}_y(4) & \hat{R}_y(3) & \hat{R}_y(2) & \hat{R}_y(1) & \hat{R}_y(0) & \hat{R}_y(1) & \dots & \hat{R}_y(n_a - 5) \\ \dots & \dots & \dots & \hat{R}_y(2) & \hat{R}_y(1) & \hat{R}_y(0) & \dots & \dots \\ \hat{R}_y(n_a - 2) & \hat{R}_y(n_a - 3) & \hat{R}_y(n_a - 4) & \dots & \dots & \dots & \dots & \hat{R}_y(1) \\ \hat{R}_y(n_a - 1) & \hat{R}_y(n_a - 2) & \hat{R}_y(n_a - 3) & \dots & \hat{R}_y(2) & \dots & \dots & \hat{R}_y(0) \end{bmatrix}$$

et

$$\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ a_{n_a} \end{bmatrix} \quad \text{et} \quad \underline{\gamma} = - \begin{bmatrix} \hat{R}_y(1) \\ \hat{R}_y(2) \\ \vdots \\ \vdots \\ \hat{R}_y(n_a - 3) \\ \hat{R}_y(n_a - 2) \\ \hat{R}_y(n_a - 1) \\ \hat{R}_y(n_a) \end{bmatrix}$$

où $\hat{R}_y(n)$ est une estimation de l'échantillon n de l'autocorrélation du signal discret y^* obtenue par l'équation (4), note de bas de page de la page 9.

L'ensemble des opérations : calcul des $\hat{R}_y(n)$, construction et résolution de (5) est réalisé par la fonction Matlab `aryule` qui s'utilise via la commande

```
>> a = aryule(yk, na);
```

où `yk` est le vecteur qui contient les N échantillons et où n_a est l'ordre de la fonction de transfert F .

A.2 Signal aléatoire

Soit un signal discret y^* la réalisation temporelle d'un signal aléatoire stationnaire et ergodique, dont on a mesuré N échantillons $y_{k \in \{0, \dots, N\}}$. On suppose que ce signal discret correspond à la sortie du système de convolution défini par la fonction de transfert

$$F(z) = \frac{1}{1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}}$$

(dont on connaît l'ordre n_a) avec pour entrée la réalisation temporelle (inconnue !) d'un signal aléatoire bruit blanc. L'objectif de la méthode de Yule-Walker est de calculer les coefficients $\{a_i\}_{i=1:n_a}$ à partir des échantillons $y_{k \in \{0, \dots, N\}}$.

Comme dans le cas d'un signal déterministe, la résolution du système de n_a équations linéaires à n_a inconnues(5) permet de calculer les coefficients $\{a_i\}_{i=1:n_a}$. Elle peut donc être réalisée par la fonction Matlab `aryule`.

B Le spectrogramme : une transformée de Fourier à court terme avec fenêtre glissante

La transformée de Fourier considère un signal sur l'ensemble de son support temporel pour faire apparaître 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. Dans ce but, 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 (rectangle, Hanning, Hamming, ...) 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. A titre d'exemple, la figure 9 présente le spectrogramme du mot "efface". On y reconnaît distinctement la succession des différents phonèmes "É", "F", "A", "S" et "E" dont le contenu fréquentiel diffère de l'un à l'autre.

Dans la construction d'un spectrogramme, trois paramètres sont à choisir par l'utilisateur (voir figurefig-SchemaSpectro) :

- la forme de la fenêtre temporelle w : rectangulaire, Hanning, gaussienne ou autre, le fenêtrage temporel implique une convolution dans le domaine des fréquences. Le choix de la fenêtre permet de gérer le compromis entre finesse du lobe principal et hauteur des lobes secondaires.

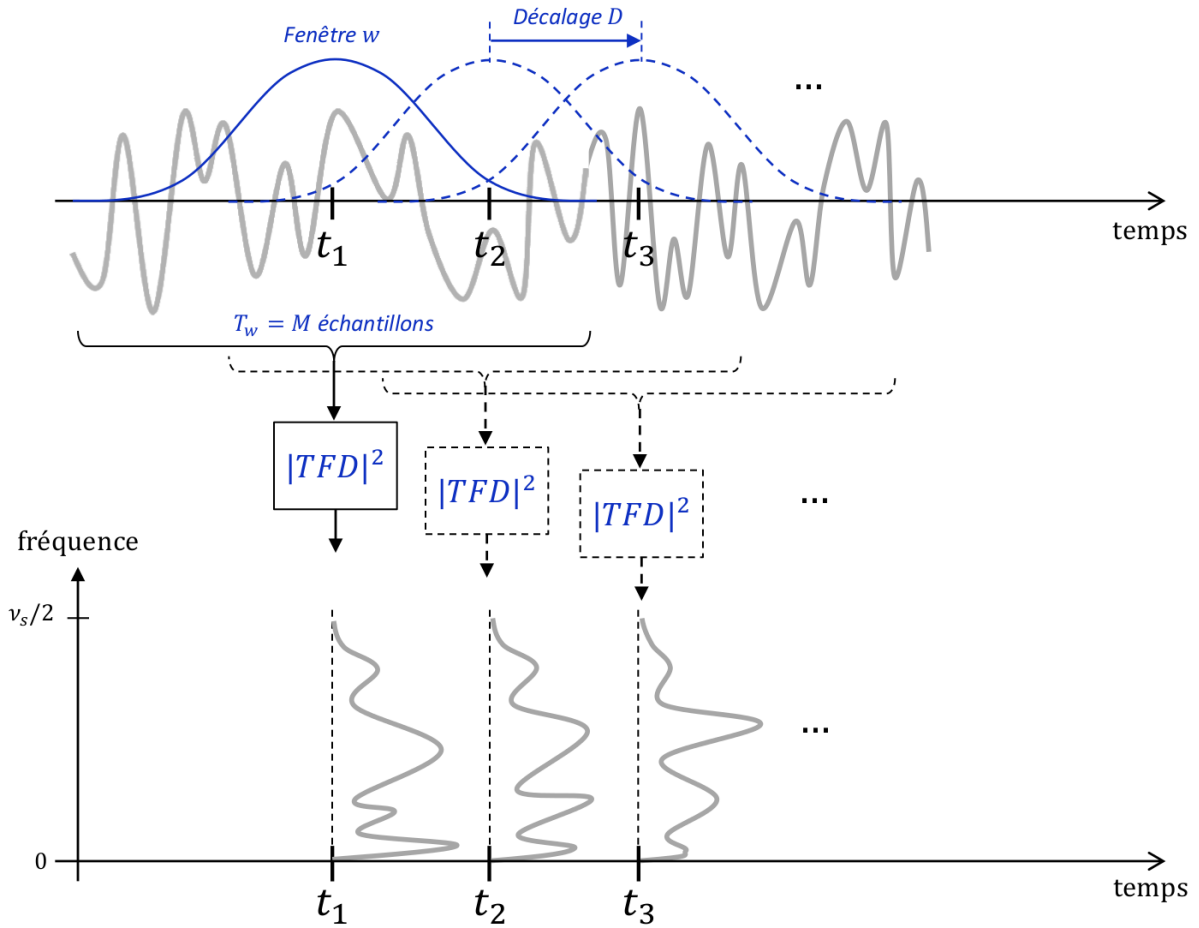


Figure 8: Principe de construction d'un spectrogramme : transformée de Fourier à court terme avec fenêtre glissante

- la taille T_w de la fenêtre temporelle : elle détermine la résolution fréquentielle $d\nu$ de la représentation temps-fréquence, en effet on a $d\nu = 1/T_w$. Une grande fenêtre permet donc une meilleure résolution en fréquence. En revanche, une fenêtre de grande taille diminue la localisation temporelle de l'analyse.
- le décalage D entre 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 tient compte d'un compromis entre la continuité temporelle de la représentation et son temps de calcul.

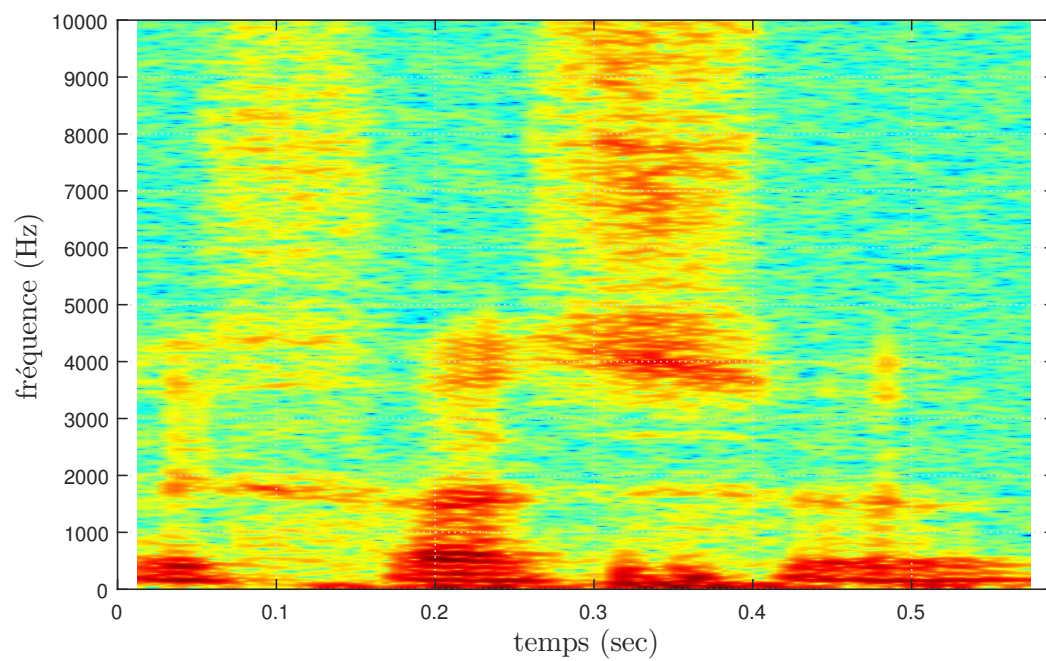


Figure 9: Spectrogramme du mot "efface". Paramètres de construction : fenêtre de Hanning de durée $T_w = 30$ msec, décalage $D = 5$ msec.