



Institut Mines-Télécom

Processus et synthèse vocale sous Matlab

Roland Badeau



Contexte académique } **sans modifications**

Voir page 5

Master Sciences et Technologies - Parcours ATIAM - UE FpA



1 Étude théorique : Analyse du mélange de deux processus

Soit un signal que l'on modélise par un processus X centré, réel et SSL. Un autre signal est lui aussi modélisé par un processus centré et SSL noté Y . Les processus X et Y sont supposés indépendants. Le processus X est filtré par un filtre dont la RI est h , que l'on ne connaît pas. Il s'ajoute à Y pour donner un processus Z que l'on mesure. Cela s'écrit :

$$Z = h * X + Y$$

On admet sans démonstration que pour toute suite sommable g , le processus $g * X$ est indépendant du processus Y .

On note $T = h * X$. Montrer que

$$S_Z(\nu) = S_Y(\nu) + S_T(\nu)$$

(pour cette preuve, il suffit d'utiliser le fait que T et Y sont indépendants, pas besoin de faire intervenir explicitement le filtre h).

On suppose que l'on a mesuré le processus X . Par exemple X est l'un des instruments participant à un orchestre devant lequel se trouve un micro qui ne mesure que cet instrument. Z est le son produit par tout l'orchestre mesuré loin de tous les instruments. Et on veut annuler l'effet de l'instrument X .

Montrer que quelque soit la suite sommable h^1 et on notant Z^1 le processus $Z^1 = Z - h^1 * X$ on a

$$R_{Z^1}(0) \geq R_Y(0)$$

et que si $h^1 = h$ alors

$$R_{Z^1}(0) = R_Y(0)$$

Interprétation :

- On interprète Z^1 comme le processus Z débarrassé de tout écho dû à X (on a éliminé l'effet de l'un des instruments).
- On rappelle que $R_B(0)$ (où B est un processus SSL centré) est la puissance de ce processus. L'inégalité démontrée porte donc sur la puissance de Z^1 par rapport à celle de Y .
- Les inégalités démontrées permettent de dire que le problème de retrouver h se réduit à la minimisation de la puissance de $Z - h^1 * X$ où h^1 est l'inconnue.
- Il peut paraître étrange de supposer connu un processus, ici X , que l'on modélise par de l'aléa. Mais il y a des cas où cela est une hypothèse réaliste. Un autre exemple, à part l'orchestre de musique, est la communication entre un téléphone mobile et son antenne relais. Dans ce cas X est une séquence connue que l'antenne envoie au téléphone portable afin d'identifier le canal (les échos des ondes sur les bâtiments, véhicules...) c'est-à-dire trouver h . Et Y est un bruit électromagnétique inconnu.

Dans les sections suivantes nous allons mettre en pratique cette idée dans différents scénarios.

Un programme : Une brique fondamentale vous est donnée. Il s'agit d'un programme Matlab `moindres_carres.m`. Il prend en entrée un entier p qui est la taille du h à trouver -1 (par souci d'homogénéité des notations avec la partie suivante). Il prend aussi deux signaux x et z qui jouent les rôles des processus X et Z . Il renvoie une séquence h^1 de taille $p + 1$ qui minimise l'énergie du signal $z - h^1 * x$.

Téléchargez le fichier **TP_Alea_Matlab.zip** depuis le Moodle du Master ATIAM. Il contient des données et des programmes. Dans Matlab vous travaillerez dans le répertoire résultant de sa décompression.

1.1 Première application : Identifier un canal

Dans le cas de l'identification de canal, nous ne connaissons pas le canal, c'est-à-dire la RI h . On se donne un signal X que l'on connaît et un signal Y parasite que l'on ne connaît pas. On observe

$$Z = h * X + Y$$

On veut retrouver h à partir de X et de Z . Cela s'écrit en Matlab :

```
h=[1 1/2];
N=1000;%Longueur de la séquence d'apprentissage
X=randn(1,N); % génère un bruit blanc de moyenne nulle et de puissance 1
sigma=0.1; %sigma est le niveau du parasitage
Y=sigma*randn(1,N);
Z=filter(h,1,X)+Y; %on convole h avec X et on ajoute Y
hestime=moindres_carres(1,X,Z); % On retrouve h à partir de X et Z
```

Pour différentes valeurs du bruit estimez l'erreur commise sur l'estimation de h , en Matlab l'erreur relative commise s'écrit

```
norm(hestime-h)/norm(h)
```

(on prendra au moins $\sigma = 0.1, 1, 10$)

Que constatez vous lorsque σ devient trop grand? Pour $\sigma = 10$, à partir de quelle valeur de N l'estimation de h redevient-elle acceptable (moins de 10% d'erreur relative, par exemple)?

Dans le programme que nous venons de faire nous connaissons la longueur de la séquence h . Que proposez-vous si la longueur de h n'est pas connue?

Comparez, pour $\sigma = 1$ l'efficacité du choix d'une séquence $X = \text{randn}(1, N)$ avec l'approche intuitive consistant à prendre $X = \delta$, i.e introduire une impulsion dans le canal et écouter la sortie. À quoi attribuez-vous le manque d'efficacité de l'approche par envoi d'une impulsion?

1.2 Seconde application : Suppression d'un son qui en parasite un autre.

Dans ce cas, Y est un son qui nous intéresse mais n'est pas connu. X est un son connu, dont on veut éliminer la présence dans Z . Étant donné Z et X on veut estimer h et en déduire une estimation de Y par l'équation

$$Y = Z - h * X$$

Après avoir chargé le fichier "parasite.mat" par la commande `load parasite.mat`, la variable Matlab **Xparasite** contient le signal X et la variable **Zparasite** contient le son mesuré Z dont on veut éliminer l'influence de X . En utilisant un programme sur le modèle suivant

```
hestim=moindres_carres(p_est_inconnu,Xparasite,Zparasite);
Y=Zparasite-filter(hestim,1,Xparasite);
```

Comment avez-vous fait pour estimer la taille du filtre? Quel filtre h avez-vous trouvé? Écoutez **Zparasite** et **Y** par la commande `soundsc(VARIABLE,22050);`¹ pour vous assurer que vous avez bien éliminé l'influence du son parasite.

1. les sons de ce TP sont échantillonnés à 22050 Hz

2 Le modèle auto-régressif pour la synthèse de parole

Dans le modèle auto-régressif on suppose que le signal enregistré est X et qu'il est obtenu à partir d'un bruit blanc ϵ suivant le filtre récursif suivant

$$X_n + h_1 X_{n-1} + \dots + h_p X_{n-p} = \epsilon_n$$

Ceci modélise le fait qu'un signal qui possède les statistiques d'un bruit blanc, ici ϵ passe dans un filtre qui se trouve être récursif. Dans ce problème il y a deux inconnus, les h_j et le signal ϵ .

Un autre programme : Le programme `lpc_morceau.m` calcule les coefficients h_j . Il est très semblable au programme `moindres_carres` : il minimise la norme de $h * X$, sauf qu'écrit comme cela il suffit de prendre $h = 0$ pour obtenir une solution imbattable. On force $h_0 = 1$ pour éviter une solution triviale et on minimise $\|h * X\|$.

2.1 Test du modèle et du programme fourni

À l'issue des commandes suivantes, on voudrait avoir **hestim=h**

```
h=[1 1/2];
sigma=1;
epsilon=sigma*randn(1,1000);
X=filter(1,h,epsilon); % Taper "help filter" pour voir en quoi cela correspond au modèle
hestim=lpc_morceau(1,X);
hestim
```

Retrouvez-vous bien le filtre h utilisé ? Que se passe-t-il sous Matlab lorsque la première ligne est remplacée par **h=[1 2]** ? Comment expliquez-vous cela ? (on peut se reporter à la partie TZ du polycopié).

Revenez à un filtre h qui n'a pas de pôles à l'extérieur du cercle unité (causal).

Expliquez la valeur affichée par la suite de commandes Matlab (pour cela il faut bien lire l'aide Matlab de la commande filter, "help filter")

```
tmp=filter(hestim,1,X); %Remarquer la position de hestim
sum(tmp.^2)/1000/sigma^2
```

2.2 Application du modèle à un signal de parole

Le programme `lpc.m` découpe un signal en trames d'une taille prescrite et recherche pour chaque trame les coefficients h qui expliquent le mieux le signal observé. Il renvoie un tableau dont chaque ligne est les coefficients de h pour la trame correspondante et un vecteur colonne qui contient l'écart-type de ϵ pour chacune des trames.

Le programme `joue_lpc.m` fabrique un son en générant un bruit blanc ϵ de même écart-type que celui calculé par le programme lpc puis le fait passer dans le filtre spécifié pour chaque trame.

Chargez des sons de piano et de voix par la commande `load modelear.mat`.

```
[lpcs,res]=lpc(aeiou,15,2000);
joue_lpc(lpcs,res,2000,22050);
```

Qu'entendez-vous ? Que se passe-t-il si on élimine le préfiltrage au début du programme `lpc` ?

Remarquez-vous une grande amélioration/dégradation lorsque le paramètre p varie ?

Modifiez le programme `joue_lpc` afin qu'il régénère le son non pas en introduisant un bruit blanc mais un signal formé d'impulsions éloignées de 0.01 secondes entre elles (on rappelle que les sons traités ici sont échantillonnés à 22050Hz).

En utilisant le programme `spectrogramme` dire à quelle fréquence les impulsions devraient être envoyées dans le filtre.

Que pouvez-vous suggérer pour améliorer la qualité de la restitution sonore ?

Essayez le modèle auto-régressif sur le son "piano". Décrire un modèle de génération qui vous paraît plus réaliste pour le génération d'un son de piano et comment l'utiliser pour une compression d'un son de piano ?

3 Estimation de la densité spectrale de puissance : périodogramme

On cherche à estimer la densité spectrale de puissance (DSP) d'un processus X_n stationnaire au sens large et de moyenne nulle, observé sur un intervalle $n \in [0 \dots N-1]$, à l'aide de la transformée de Fourier discrète (TFD), implémentée dans la fonction Matlab `fft`. On rappelle que la DSP $S_X(e^{2i\pi\nu})$ est définie comme la transformée de Fourier à temps discret (TFTD) de la fonction d'autocovariance $R_X(k)$: $S_X(e^{2i\pi\nu}) = \sum_{k \in \mathbb{Z}} R_X(k) e^{-2i\pi\nu k}$.

1. Soit $\hat{R}_N(k)$ l'estimateur biaisé de la fonction d'autocovariance du processus X_n . On rappelle que le périodogramme de X , que nous noterons $\hat{S}_N(e^{2i\pi\nu})$, est défini comme $\hat{S}_N(e^{2i\pi\nu}) = \sum_{k=-N+1}^{N-1} \hat{R}_N(k) e^{-2i\pi\nu k}$. Exprimer $\hat{S}_N(e^{2i\pi\nu})$ en fonction de la TFTD du signal $X_0 \dots X_{N-1}$.
2. En déduire un algorithme de calcul de $\hat{S}_N(e^{2i\pi k/M})$ pour $k \in [0 \dots M-1]$, où $M \geq N$ est l'ordre de la TFD utilisée, que vous implémenterez sous Matlab. Testez-le sur un bruit blanc.
3. Comment obtenir la séquence $\hat{R}_N(k)$ de l'estimateur biaisé de la fonction d'autocovariance à partir de $\hat{S}_N(e^{2i\pi k/M})$? Quelle précaution faut-il prendre sur M ? Écrire une fonction Matlab qui calcule $\hat{R}_N(k)$ en exploitant cette propriété. Testez-la sur un bruit blanc. Comparer sa complexité algorithmique à celle d'un calcul direct de $\hat{R}_N(k)$.
4. Toujours dans le cas du bruit blanc, estimer la variance du périodogramme pour plusieurs horizons d'observation N , comme fonction de la fréquence réduite ν . Qu'observez-vous ?
5. Implémentez la méthode du périodogramme fenêtré présentée en cours, que vous testerez sur un bruit blanc. Estimez la variance de ce périodogramme fenêtré pour plusieurs horizons d'observation N . Qu'en concluez-vous ?



Contexte académique } sans modifications

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après, et à l'exclusion de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage dans un cadre académique, par un utilisateur donnant des cours dans un établissement d'enseignement secondaire ou supérieur et à l'exclusion expresse des formations commerciales et notamment de formation continue. Ce droit comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document à destination des élèves ou étudiants.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel et non exclusif. Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : sitepedago@telecom-paristech.fr