

Prédiction linéaire

Olivier Gillet et Gaël Mahé

Université Paris Descartes

Novembre 2008

Plan

- 1 Présentation**
 - Modèle source-filtre
 - Prédiction linéaire
- 2 Résolution du problème de prédiction linéaire**
 - Mise en équations
 - Algorithme de Levinson et Durbin
- 3 Propriétés et applications**
 - Cas où p grand
 - Cas où p petit

Plan

1

Présentation

- Modèle source-filtre
- Prédiction linéaire

2

Résolution du problème de prédiction linéaire

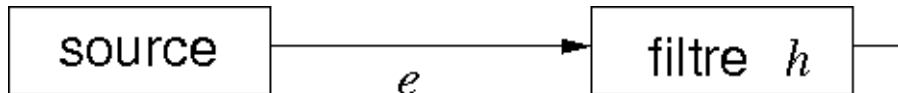
- Mise en équations
- Algorithme de Levinson et Durbin

3

Propriétés et applications

- Cas où p grand
- Cas où p petit

Modèle source-filtre



Modèle général de la production des sons

- Source : élément qui apporte de l'énergie au système et produit une forme d'onde élémentaire.
- Filtre : medium dans lequel l'énergie se propage et se réverbère.

Modèle source-filtre (2)

Exemple 1 : Instrument à percussion

- Source : Coup de baguette (ou de mailloche).
- Filtre : Membrane.

Modèle source-filtre (2)

Exemple 1 : Instrument à percussion

- Source : Coup de baguette (ou de mailloche).
- Filtre : Membrane.

Exemple 2 : Guitare

- Source : Médiator.
- Filtre : Corde, caisse de l'instrument.

Modèle source-filtre (2)

Exemple 1 : Instrument à percussion

- Source : Coup de baguette (ou de mailloche).
- Filtre : Membrane.

Exemple 2 : Guitare

- Source : Médiator.
- Filtre : Corde, caisse de l'instrument.

Exemple 3 : Voix

- Source : Flux d'air venant des poumons - continu ou sous forme d'impulsions.
- Filtre : Cavité buccale (la forme varie en fonction de la position de la mâchoire, des lèvres, de la langue).

Prédiction linéaire

Modèle source-filtre

On se place dans le cas où le filtre a une réponse :

$$H(z) = \frac{1}{1 + \sum_{k=1}^P a_k z^{-k}}$$

P est l'ordre du filtre. Dans ce cas :

$$x(n) = e(n) - \sum_{k=1}^P a_k x(n - k)$$

Prédiction linéaire (2)

Modèle source-filtre (2)

$$x(n) = \hat{x}(n) + e(n)$$

Avec :

- $\hat{x}(n)$ = prédiction linéaire de $x = -\sum_{k=1}^P a_k x(n-k)$
- $e(n)$ = erreur de prédiction

Prédiction linéaire (2)

Modèle source-filtre (2)

$$x(n) = \hat{x}(n) + e(n)$$

Avec :

- $\hat{x}(n)$ = prédiction linéaire de $x = -\sum_{k=1}^P a_k x(n-k)$
- $e(n)$ = erreur de prédiction

Le problème de prédiction linéaire

Trouver les $(a_k)_{1 \leq k \leq P}$ qui minimisent l'erreur de prédiction $e(n)$

Prédiction linéaire (2)

Modèle source-filtre (2)

$$x(n) = \hat{x}(n) + e(n)$$

Avec :

- $\hat{x}(n)$ = prédiction linéaire de $x = -\sum_{k=1}^P a_k x(n-k)$
- $e(n)$ = erreur de prédiction

Le problème de prédiction linéaire

Trouver les $(a_k)_{1 \leq k \leq P}$ qui minimisent l'erreur de prédiction $e(n)$

Minimisation au sens de l'énergie quadratique moyenne

On minimise $\epsilon = E[e(n)^2]$.

Plan

1

Présentation

- Modèle source-filtre
- Prédiction linéaire

2

Résolution du problème de prédiction linéaire

- Mise en équations
- Algorithme de Levinson et Durbin

3

Propriétés et applications

- Cas où p grand
- Cas où p petit

Minimisation au sens E.Q.M. (1)

Expression de l'erreur

$$\begin{aligned}
 \epsilon &= E \left[\left(x(n) + \sum_{k=1}^P a_k x(n-k) \right)^2 \right] \\
 &= E[x(n)^2] + 2E \left[x(n) \sum_{k=1}^P a_k x(n-k) \right] \\
 &+ E \left[\sum_{k=1}^P a_k x(n-k) \sum_{l=1}^P a_l x(n-l) \right] \\
 &= r_x(0) + 2 \sum_{k=1}^P a_k r_x(k) + \sum_{k=1}^P \sum_{l=1}^P a_k a_l r_x(k-l)
 \end{aligned}$$

Minimisation au sens E.Q.M. (2)

Expression matricielle

$$\mathbf{r} = [r_x(1) \quad \dots \quad r_x(P)], \mathbf{R} = \begin{bmatrix} r_x(0) & \dots & r_x(P-1) \\ \vdots & \ddots & \vdots \\ r_x(P-1) & \dots & r_x(0) \end{bmatrix}$$

L'erreur s'écrit donc :

$$\epsilon = r_x(0) + 2\mathbf{r}\mathbf{a} + \mathbf{a}^T \mathbf{R}\mathbf{a}$$

Minimisation au sens E.Q.M. (2)

Expression matricielle

$$\mathbf{r} = [r_x(1) \quad \dots \quad r_x(P)], \mathbf{R} = \begin{bmatrix} r_x(0) & \dots & r_x(P-1) \\ \vdots & \ddots & \vdots \\ r_x(P-1) & \dots & r_x(0) \end{bmatrix}$$

L'erreur s'écrit donc :

$$\epsilon = r_x(0) + 2\mathbf{r}\mathbf{a} + \mathbf{a}^T\mathbf{R}\mathbf{a}$$

Minimisation de l'erreur

$$2\mathbf{r} + 2\mathbf{R}\mathbf{a} = 0$$

$$\mathbf{a} = -\mathbf{R}^{-1}\mathbf{r}$$

Minimisation au sens E.Q.M. (3)

La matrice d'autocorrélation R étant définie positive, il y a toujours une solution au problème.

Minimisation au sens E.Q.M. (3)

La matrice d'autocorrélation R étant définie positive, il y a toujours une solution au problème.

Mais on peut faire mieux...

Car il reste tout de même à effectuer une résolution de système coûteuse pour les ordres élevés !

Algorithme de Levinson et Durbin

L'approche...

- On va calculer les coefficients pour un filtre d'ordre croissant.
- Les coefficients pour l'ordre p vont se calculer très simplement à partir des coefficients calculés pour l'ordre $p - 1$.

Algorithme de Levinson et Durbin

L'approche...

- On va calculer les coefficients pour un filtre d'ordre croissant.
- Les coefficients pour l'ordre p vont se calculer très simplement à partir des coefficients calculés pour l'ordre $p - 1$.

Comment y arriver ?

- 1 Ecrire le système pour un filtre d'ordre p .
- 2 Y faire rentrer une nouvelle équation.
- 3 Trouver une relation intéressante.
- 4 Y voir une solution du système pour un filtre d'ordre $p + 1$.

Algorithme de Levinson et Durbin (2)

Minimisation de l'erreur pour l'ordre p

La condition d'optimalité s'écrit $\mathbf{r} + \mathbf{R}\mathbf{a} = 0$, soit :

$$r_x(k) + \sum_{l=1}^p a_l^p r_x(k-l) = 0; k = 1 \dots P$$

Algorithme de Levinson et Durbin (2)

Minimisation de l'erreur pour l'ordre p

La condition d'optimalité s'écrit $\mathbf{r} + \mathbf{R}\mathbf{a} = 0$, soit :

$$r_x(k) + \sum_{l=1}^p a_l^p r_x(k-l) = 0; k = 1 \dots P$$

Expression de l'erreur pour l'ordre p

$$\begin{aligned} \epsilon_p &= r_x(0) + 2 \sum_{k=1}^p a_k^p r_x(k) + \sum_{k=1}^p \sum_{l=1}^p a_k^p a_l^p r_x(k-l) \\ &= r_x(0) + \sum_{k=1}^p a_k^p r_x(k) \end{aligned}$$

Algorithme de Levinson et Durbin (3)

D'où le système, pour l'ordre p

$$\begin{bmatrix} r_x(0) & \dots & r_x(p) \\ r_x(1) & \dots & r_x(p-1) \\ \vdots & \ddots & \vdots \\ r_x(p) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_1^p \\ \vdots \\ \alpha_p^p \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Algorithme de Levinson et Durbin (3)

D'où le système, pour l'ordre p

$$\begin{bmatrix} r_x(0) & \dots & r_x(p) \\ r_x(1) & \dots & r_x(p-1) \\ \vdots & \ddots & \vdots \\ r_x(p) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_1^p \\ \vdots \\ \alpha_p^p \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

On augmente d'un rang...

$$\begin{bmatrix} r_x(0) & \dots & r_x(p) & r_x(p+1) \\ r_x(1) & \dots & r_x(p-1) & r_x(p) \\ \vdots & \ddots & \vdots & \vdots \\ r_x(p) & \dots & r_x(0) & r_x(1) \\ r_x(p+1) & \dots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_1^p \\ \vdots \\ \alpha_p^p \\ 0 \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \gamma_p \end{bmatrix}$$

Algorithme de Levinson et Durbin (4)

Le coefficient introduit...

$$\gamma_p = r_x(p+1) + \sum_{i=1}^p a_i^p r_x(p+1-i)$$

Algorithme de Levinson et Durbin (4)

Le coefficient introduit...

$$\gamma_p = r_x(p+1) + \sum_{i=1}^p a_i^p r_x(p+1-i)$$

Matrice à structure de Toeplitz = ...

$$\begin{bmatrix} r_x(0) & \dots & r_x(p) & r_x(p+1) \\ r_x(1) & \dots & r_x(p-1) & r_x(p) \\ \vdots & \ddots & \vdots & \vdots \\ r_x(p) & \dots & r_x(0) & r_x(1) \\ r_x(p+1) & \dots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1^p \\ \vdots \\ a_p^p \\ 0 \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \gamma_p \end{bmatrix} \quad (S)$$

Algorithme de Levinson et Durbin (4)

Le coefficient introduit...

$$\gamma_p = r_x(p+1) + \sum_{i=1}^p a_i^p r_x(p+1-i)$$

Ruse diabolique : si $R_{p+1}A = B, R_{p+1}A^\uparrow = B^\uparrow$

$$\begin{bmatrix} r_x(0) & \dots & r_x(p) & r_x(p+1) \\ r_x(1) & \dots & r_x(p-1) & r_x(p) \\ \vdots & \ddots & \vdots & \vdots \\ r_x(p) & \dots & r_x(0) & r_x(1) \\ r_x(p+1) & \dots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 0 \\ a_p^p \\ \vdots \\ a_1^p \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma_p \\ 0 \\ \vdots \\ 0 \\ \epsilon_p \end{bmatrix} \quad (S^\uparrow)$$

Algorithme de Levinson et Durbin (5)

Combinaison linéaire $S + \Gamma_{p+1}S^\dagger$ des deux systèmes

$$R_{p+1} \left(\begin{bmatrix} 1 \\ a_1^p \\ \vdots \\ a_p^p \\ 0 \end{bmatrix} + \Gamma_{p+1} \begin{bmatrix} 0 \\ a_p^p \\ \vdots \\ a_1^p \\ 1 \end{bmatrix} \right) = \left(\begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \gamma_p \end{bmatrix} + \Gamma_{p+1} \begin{bmatrix} \gamma_p \\ 0 \\ \vdots \\ 0 \\ \epsilon_p \end{bmatrix} \right)$$

Algorithme de Levinson et Durbin (5)

Combinaison linéaire $S + \Gamma_{p+1}S^\dagger$ des deux systèmes

$$R_{p+1} \left(\begin{bmatrix} 1 \\ a_1^p \\ \vdots \\ a_p^p \\ 0 \end{bmatrix} + \Gamma_{p+1} \begin{bmatrix} 0 \\ a_p^p \\ \vdots \\ a_1^p \\ 1 \end{bmatrix} \right) = \left(\begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \gamma_p \end{bmatrix} + \Gamma_{p+1} \begin{bmatrix} \gamma_p \\ 0 \\ \vdots \\ 0 \\ \epsilon_p \end{bmatrix} \right)$$

Mise à jour pour l'ordre supérieur...

Si l'on pose :

$$\Gamma_{p+1} = -\frac{\gamma_p}{\epsilon_p}$$

On peut retrouver le système pour l'ordre supérieur.

Algorithme de Levinson et Durbin (6)

La combinaison des deux systèmes devient...

$$R_{p+1} \begin{bmatrix} 1 \\ a_1^p + \Gamma_{p+1} a_p^p \\ \vdots \\ a_p^p + \Gamma_{p+1} a_1^p \\ \Gamma_{p+1} \end{bmatrix} = \begin{bmatrix} \epsilon_p + \Gamma_{p+1} \gamma_p \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Algorithme de Levinson et Durbin (6)

La combinaison des deux systèmes devient...

$$R_{p+1} \begin{bmatrix} 1 \\ a_1^p + \Gamma_{p+1} a_p^p \\ \vdots \\ a_p^p + \Gamma_{p+1} a_1^p \\ \Gamma_{p+1} \end{bmatrix} = \begin{bmatrix} \epsilon_p + \Gamma_{p+1} \gamma_p \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

C'est gagné !

On a réussi à trouver une solution du système pour l'ordre $p + 1$!

Algorithme de Levinson et Durbin (7)

On identifie...

$$\begin{bmatrix} 1 \\ a_1^p + \Gamma_{p+1} a_p^p \\ \vdots \\ a_p^p + \Gamma_{p+1} a_1^p \\ \Gamma_{p+1} \end{bmatrix} \text{ à } \begin{bmatrix} 1 \\ a_1^{p+1} \\ \vdots \\ a_p^{p+1} \\ a_{p+1}^{p+1} \end{bmatrix} \text{ et } \begin{bmatrix} \epsilon_p + \Gamma_{p+1} \gamma_p \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \text{ à } \begin{bmatrix} \epsilon_{p+1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Algorithme de Levinson et Durbin (7)

On identifie...

$$\begin{bmatrix} 1 \\ a_1^p + \Gamma_{p+1} a_p^p \\ \vdots \\ a_p^p + \Gamma_{p+1} a_1^p \\ \Gamma_{p+1} \end{bmatrix} \text{ à } \begin{bmatrix} 1 \\ a_1^{p+1} \\ \vdots \\ a_p^{p+1} \\ a_{p+1}^{p+1} \end{bmatrix} \text{ et } \begin{bmatrix} \epsilon_p + \Gamma_{p+1} \gamma_p \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \text{ à } \begin{bmatrix} \epsilon_{p+1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

D'où la solution

$$\epsilon_{p+1} = \epsilon_p + \Gamma_{p+1} \gamma_p = \epsilon_p - \Gamma_{p+1}^2 \epsilon_p$$

$$a_i^{p+1} = a_i^p + \Gamma_{p+1} a_{p+1-i}^p$$

$$a_{p+1}^{p+1} = \Gamma_{p+1}$$

Algorithme de Levinson et Durbin : Résumé

Initialisation

- 1 $\epsilon_0 = r_x(0)$

Algorithme de Levinson et Durbin : Résumé

Initialisation

- $\epsilon_0 = r_x(0)$

Récursion pour $p = 0 \dots P - 1$

- $a_{p+1}^{p+1} = \Gamma_{p+1} = -\frac{r_x(p+1) + \sum_{i=1}^p a_i^p r_x(p-i+1)}{\epsilon_p}$

- Pour $i = 1 \dots p$, $a_i^{p+1} = a_i^p + \Gamma_{p+1} a_{p+1-i}^p$

- $\epsilon_{p+1} = \epsilon_p (1 - \Gamma_{p+1}^2)$

Plan

1

Présentation

- Modèle source-filtre
- Prédiction linéaire

2

Résolution du problème de prédiction linéaire

- Mise en équations
- Algorithme de Levinson et Durbin

3

Propriétés et applications

- Cas où p grand
- Cas où p petit

Filtrage inverse

Le signal observé s'écrit :

$$x(n) = e(n) - \sum_{k=1}^P a_k x(n-k)$$

Pour retrouver le signal source, on applique le filtre inverse :

$$e(n) = x(n) + \sum_{k=1}^P a_k x(n-k) = x(n) * h_w(n)$$

Avec $h_w(0) = 1$, $h_w(i) = a_i$ pour $1 \leq i \leq P$, $h_w(i) = 0$ pour $i > P$.

Que représente e ?

Cas où p grand

Blanchiment (1)

Propriété de l'erreur de prédiction

Comme $e(n) = x(n) + \sum_{k=1}^P a_k x(n-k)$, on a :

$$r_e(l) = E[e(n)e(n+l)] = E[x(n)e(n+l)] + \sum_{k=1}^P a_k E[x(n)e(n+l+k)]$$

Calculons $E[x(n)e(n+i)]$:

$$E[x(n)e(n+i)] = r_x(i) + \sum_{k=1}^P a_k r_x(i-k)$$

Or le critère de minimisation de l'erreur est tel que cette quantité est nulle pour $i = 1 \dots P$.

Blanchiment (2)

Conclusion

Lorsque $P \rightarrow \infty$, $r_e(l) = 0$ pour $l \neq 0$.

- e est un bruit blanc, de puissance ϵ .
- Le filtre h_w s'appelle filtre de blanchiment.

Blanchiment (2)

Conclusion

Lorsque $P \rightarrow \infty$, $r_e(l) = 0$ pour $l \neq 0$.

- e est un bruit blanc, de puissance ϵ .
- Le filtre h_w s'appelle filtre de blanchiment.

Cas d'un processus $AR - p_0$

$$x(n) = e(n) - \sum_{k=1}^{p_0} a_k x(n-k)$$

, où e est un bruit blanc gaussien. Pour $P \geq p_0$,

$$a_k^P = \begin{cases} a_k & \text{si } k \leq p_0 \\ 0 & \text{sinon} \end{cases}$$

Erreur de prédiction linéaire = e ...c'est un bruit blanc !

Cas où p grand

Spectre LPC à l'ordre p

Rappel...

$$X(z) = H(z)E(z), \text{ avec } H(z) = \frac{1}{1 + \sum_{k=1}^p a_k z^{-k}}$$

Par conséquent, $\gamma_x(f) = |H(f)|^2 \gamma_e(f)$.

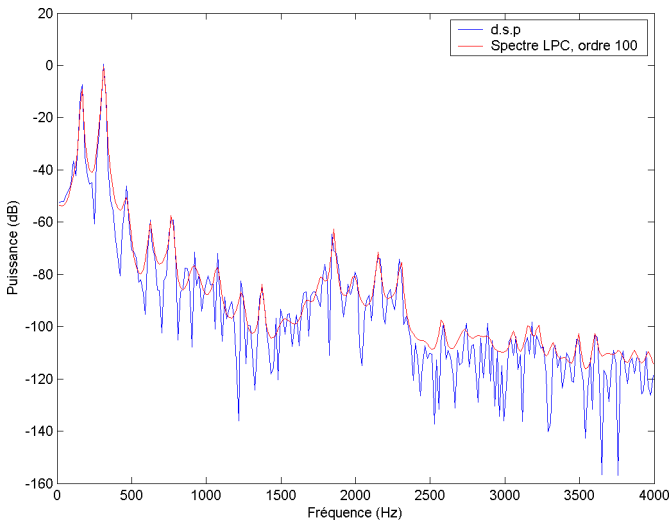
Lorsque p est suffisamment grand

$e(n)$ est un bruit blanc de puissance ϵ . Dès lors,

$$\gamma_x(f) = \epsilon |H(f)|^2 = \text{Spectre LPC.}$$

Cas où p grand

Exemple : voyelle u



Principes de l'analyse

Si p correspond à l'ordre du filtre dans le modèle source-filtre :

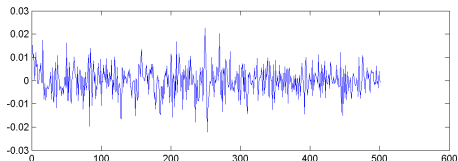
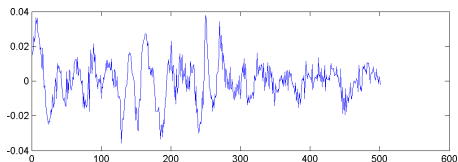
- L'erreur de prédiction e de puissance minimale est le signal source
- Les coefficients a_k sont ceux du filtre

Cas des signaux de parole

- Signal non stationnaire → analyse par trames de 10 à 40ms, où le signal est localement stationnaire
- analyse LPC à l'ordre 10 environ

Cas où p petit

Son non-voisé : consonne "s"

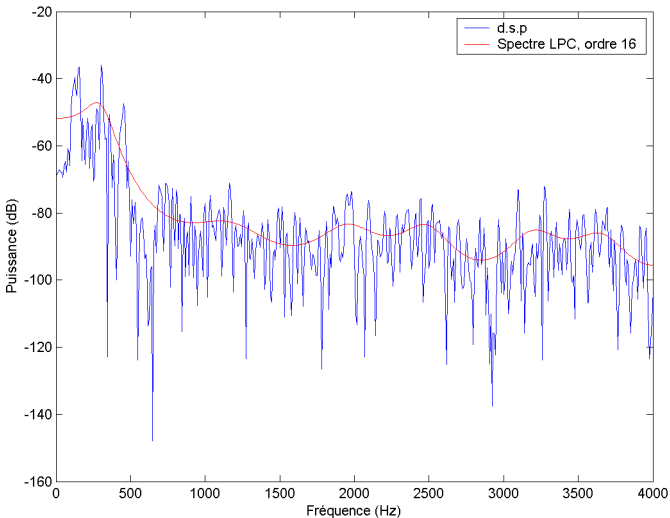


Modèle

Dans le cas du "s", la source est un flux d'air continu. L'erreur de prédiction peut ici être représentée par un bruit blanc.

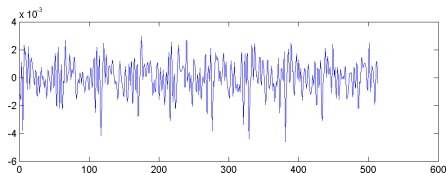
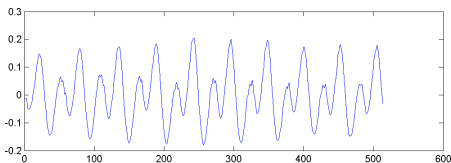
Cas où p petit

Son non-voisé : consonne "s" (2)



Cas où p petit

Son voisé : voyelle "u"

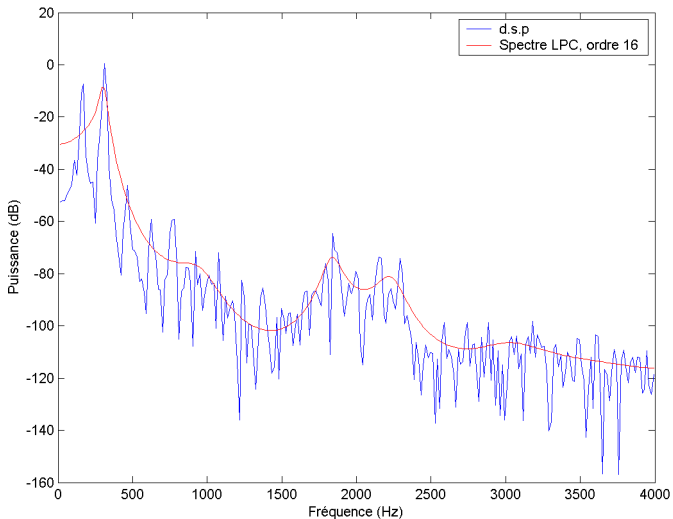


Modèle

Dans le cas des voyelles, la source est un train de "bouffées" d'air s'échappant de la glotte. L'erreur de prédiction peut ici être représentée par un train d'impulsions périodique.

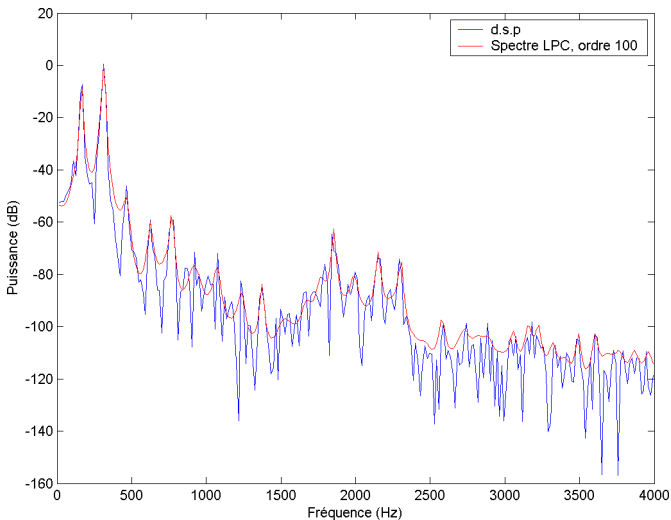
Cas où p petit

Son voisé : voyelle "u" (2)



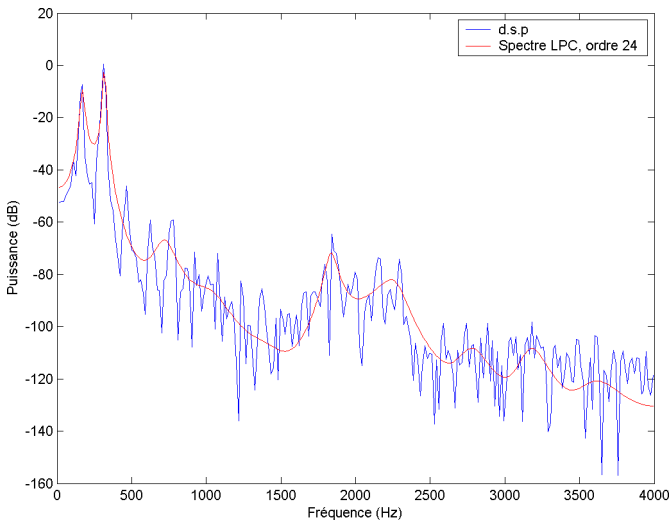
Cas où p petit

Son voisé : voyelle "u" (2)



Cas où p petit

Son voisé : voyelle "u" (2)



Cas où p petit

Son voisé : voyelle "u" (2)

