Méthodes à haute résolution

Augustin HOFF et Gustavo CIOTTO PINTON

Supélec

2014

Table des matières

Outils mathématiques

Définitions

- L'autocorrelation d'un processus aléatoire pour deux indices de temps différents n_1 et n_2 est définie comme

$$r_{xx} = \mathcal{E}\{x[n_1]x^*[n_2]\} \tag{1}$$

- La matrice formée par les valeurs d'autocorrelation :

$$R_{N-1} = \begin{bmatrix} r_{xx}[0] & r_{xx}[-1] & r_{xx}[-2] & \cdots & r_{xx}[-(N-1)] \\ r_{xx}[1] & r_{xx}[0] & r_{xx}[-1] & \cdots & r_{xx}[-(N-2)] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}[N-1] & r_{xx}[N-2] & r_{xx}[N-3] & \cdots & r_{xx}[0] \end{bmatrix} \tag{2}$$

Outils mathématiques

Définitions

- Le rapport entre les puissances du signal et du bruit (SNR) :

$$SNR = \frac{P_{signal}}{P_{bruit}} \tag{3}$$

Il est souvent écrit en *décibels* [dB] :

$$SNR_{dB} = 10 * \log \left[\frac{P_{signal}}{P_{bruit}} \right] \tag{4}$$

- La puissance d'un processus aléatoire est calculée par l'équation suivante

$$P_x = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(\omega) d\omega = \int_{-\frac{1}{2}}^{\frac{1}{2}} S_x(\nu) d\nu \tag{5}$$

où $S_x(f) = \mathcal{F}\{r_{xx}[k]\}$

Outils mathématiques

Application à un signal sinuisoïdal complexe

- On pose le signal d'entrée

$$s[n] = \sum_{k=1}^{P} A_k \exp(j(2\pi f_k nT + \theta_k)) \tag{6}$$

- En supposant que les variables aléatoires θ_k sont idépendantes et uniformement distribuées sur l'intervalle $[0, 2\pi[$, on obtient alors

$$\overline{s}[n] = 0 \quad \text{et} \quad r_{ss}[m] = \sum_{k=1}^{P} A_k^2 \exp(j2\pi f_k mT) \tag{7}$$

De plus :

$$SNR_{dB} = 10 * \log\left[\frac{P_{signal}}{P_{bruit}}\right] = 10 * \log\left[\frac{\sum_{k=1}^{P} A_k^2}{P_{bruit}}\right] \tag{8}$$

Outils mathématiques

Application à un signal sinuisoïdal complexe

- La matrice d'autocorrelation s'écrit en présence de bruit blanc de variance p_w

$$R_{xx} = R_{ss} + R_w = \sum_{k=1}^{P} A_k^2 \boldsymbol{e}_{N-1}(f_k)\boldsymbol{e}_{N-1}^H(f_k) + p_w \boldsymbol{I} \qquad (9)$$

où

$$\boldsymbol{e}_{N-1}(f_k) = \begin{bmatrix} 1 \\ \exp(2\pi j f_k T) \\ \vdots \\ \exp(2\pi j f_k (N-1) T) \end{bmatrix} \qquad (10)$$

Outils mathématiques

Application à un signal sinuisoïdal complexe

- La matrice se réécrit à partir des P valeurs propres de R_{ss}

$$R_{xx} = \sum_{i=1}^{P}(\lambda_i + p_w)\boldsymbol{v}_i\boldsymbol{v}_i^H + p_w \sum_{i=P+1}^{N} \boldsymbol{v}_i\boldsymbol{v}_i^H \tag{11}$$

- Les P premiers vecteurs propres de R_{xx} forment l'espace signal orthogonal à l'espace bruit donné par les N-P autres vecteurs propres.

Algorithme MUSIC

Aspect théorique

- On pose alors :

$$\sum_{k=P+1}^{N} \alpha_k \left| \boldsymbol{e}_{N-1}^H(f) \boldsymbol{v}_k \right|^2 = \boldsymbol{e}_{N-1}^H(f) \left(\sum_{k=P+1}^{N} \alpha_k \boldsymbol{v}_k \boldsymbol{v}_k^H \right) \boldsymbol{e}_{N-1}(f) \qquad (12)$$

- Cette quantité est nulle si $\boldsymbol{e}_{N-1}(f)$ est un des vecteurs de l'espace signal i.e si f est une fréquence du signal car $\boldsymbol{v}_k$ est un vecteur propre de l'espace bruit pour k $\in [\![P+1, N]\!]$
- L'estimateur est donc donné par l'expression

$$P_{MUSIC}(f) = \frac{1}{\boldsymbol{e}_{N-1}^H(f) \left(\sum_{k=P+1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^H \right) \boldsymbol{e}_{N-1}(f)} \qquad (13)$$

Algorithme MUSIC

Mise en oeuvre - *P_music.m*

- Les principales variables

```
function [ resultat ] = P_music(signal_x,  frequences ,
    P)
N = size(signal_x ,2);
T = 1/N;
```

- Calcul de la matrice d'autocorrelation

```
z   = corrmtx( signal_x, 0 );
Rxx = toeplitz(z);
```

Algorithme MUSIC

Mise en oeuvre - *test.m*

- Paramètres initiaux du test

```
clear all; close all; clc;
N = 100;
n = 0: 1 : N - 1;
P = 3;
T = 1/N;
fDebut = 1e0;
fFin   = 7e1;
fPas   = 1e-1;

A = [1 2.5 3];
F = [4.9e1 5e1 6e1];
Theta = 2*pi*[rand(1) rand(1) rand(1)];

SNR = [0.001 40 100 500]; %(dB)
```

Algorithme MUSIC

Mise en oeuvre - *test.m*

- Calcul des pics pour chaqu'un des SNRs

```
bAux = randn(1,N);

%signal
s = zeros (1, N);
for k = 1:P
    s = s + A(k) * exp(2*pi*1i*n*T*F(k) + 1i*Theta(k));
end

%bruit
for l = 1 : size(SNR, 2)

    varianceNoise = sum (A.*A) / ( 10 ^ (SNR(l) / 10));
    b = sqrt(varianceNoise) * bAux;
    x = s + b;

    frequences = fDebut : fPas : fFin;
    result = P_music (x, frequences, P);
    plot(...)
end
```

Algorithme MUSIC

Resultats

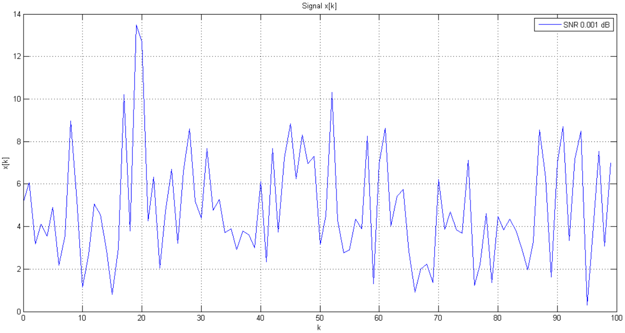

FIGURE 1: Signal avec du bruit pour $SNR_{dB} = 0.001dB$

Algorithme MUSIC

Resultats

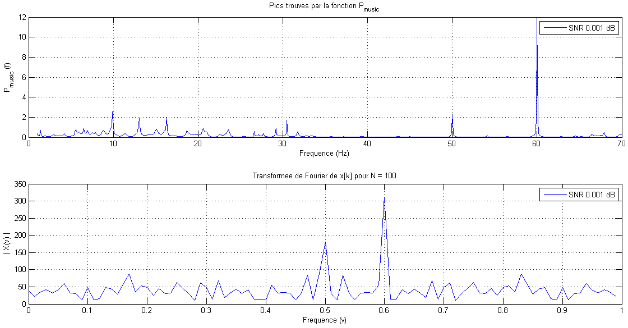

FIGURE 2: Resultats obtenus pour $SNR_{dB} = 0.001dB$

Algorithme MUSIC

Resultats

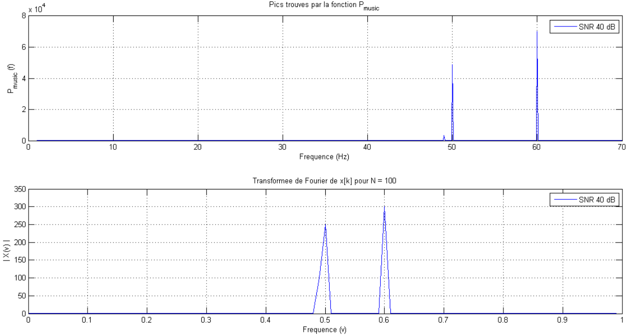

Figure 3: Resultats obtenus pour $SNR_{dB} = 40dB$

Algorithme MUSIC

Resultats

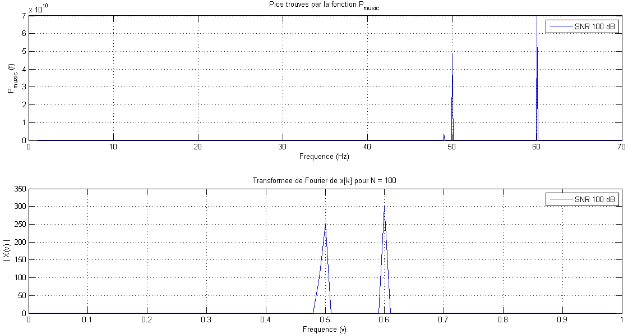

FIGURE 4: Resultats obtenus pour $SNR_{dB} = 100dB$

Algorithme MUSIC

Resultats

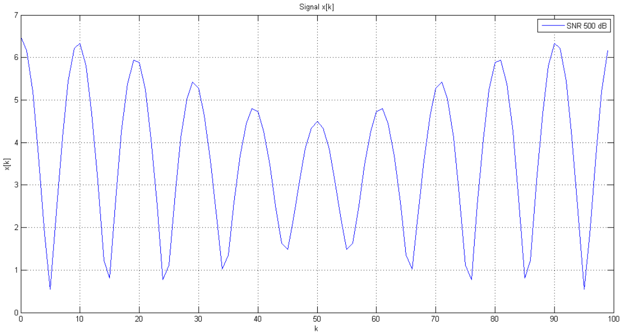

FIGURE 5: Signal avec du bruit pour $SNR_{dB} = 500dB$

Algorithme MUSIC

Resultats

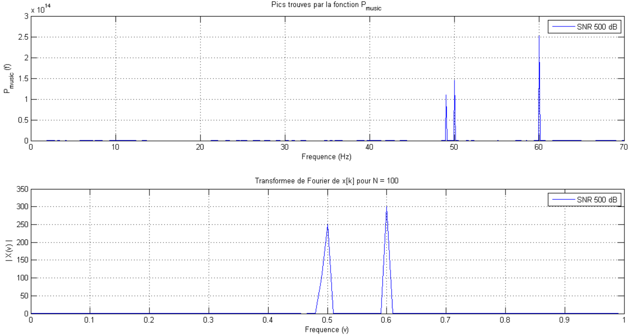

FIGURE 6: Resultats obtenus pour $SNR_{dB} = 500dB$

Critère de Rayleight

Analogie avec la lumière

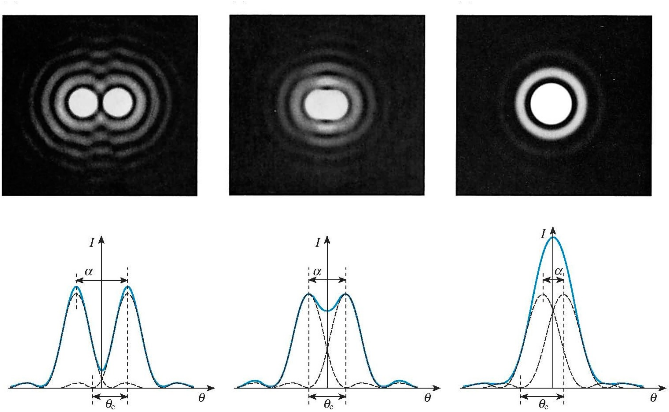

FIGURE 7: Distinction des différentes taches de lumière selon les cas

Algorithme ESPRIT

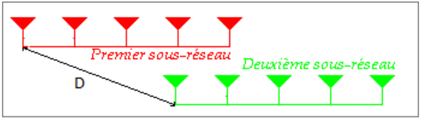

FIGURE 8: Disposition des antennes.

Déphasage temporel de $\frac{Dsin(\phi)}{c}$ où ϕ est l'angle d'incidence du signal

Algorithme ESPRIT

$$S_2 = S_1 \Omega \tag{14}$$

si

$$S_1 = \begin{bmatrix} \boldsymbol{e}_{N-1}(f_1) & \cdots & \boldsymbol{e}_{N-1}(f_P) \end{bmatrix}$$

et

$$\Omega = \begin{bmatrix} \exp(2\pi j f_1 \frac{Dsin(\phi_1)}{c}) & & \text{\huge 0} \\ & \ddots & \\ \text{\huge 0} & & \exp(2\pi j f_p \frac{Dsin(\phi_P)}{c}) \end{bmatrix}$$

Algorithme ESPRIT

- La matrice d'autocorrélation s'écrit alors

$$R_{xx} = \begin{bmatrix} S_1 \\ S_1\Omega \end{bmatrix} \begin{bmatrix} A_1^2 & & 0 \\ & \ddots & \\ 0 & & A_P^2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_1\Omega \end{bmatrix}^H + p_w \boldsymbol{I} \tag{15}$$

- Ce qui donne :

$$U = [u_1 \cdots u_P] = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_1\Omega \end{bmatrix} T = \begin{bmatrix} S_1 T \\ S_1\Omega T \end{bmatrix} \tag{16}$$

Algorithme ESPRIT

- Finalement

$$U_2 = U_1 T^{-1} \Omega T = U_1 \Psi \tag{17}$$

- La méthode des moindres carrés donne alors une estimation de Ψ la matrice semblable à Ω contenant les informations du signal

$$\Psi = (U_1 U_1^H)^{-1} U_1^H U_2 \tag{18}$$

Références

- Institute for Dynamics Systems and Control. *White noise and power spectral density*. http://www.idsc.ethz.ch/Courses/signals_and_systems/ArchiveFall10/lectureNotes8.pdf
- S. Lawrence Marple Jr. *Digital Spectral Analysis with Applications*. Prentice-Hall Inc, 1988
- Bernard Picinbono. *Signaux aléatoires - Tome 2 Fonctions aléatoires et modèles avec problèmes résolus*. Dunod Université, 1993
- *Les méthodes à haute résolution*. HERMES, 1998
- Yves Grenier. *Méthodes haute-résolution en analyse spectrale et localisation*. http://perso.telecom-paristech.fr/~rioul/liesse/2012liesse4/YG-slides.pdf