

Traitement du signal avec Scilab : corrélation

L'opération de corrélation est très utilisée en traitement du signal ; elle permet de mesurer le degré de dépendance entre deux signaux. Lorsque cette opération est appliquée à un seul signal, on parle alors d'autocorrélation. L'autocorrélation autorise l'extraction d'un signal noyé dans du bruit, la détermination de périodicité cachée, ainsi que la détermination de la densité spectrale de puissance comme nous le verrons.

Un exemple de solutions aux exercices proposés lors de cette séance sont donnée à la fin.

1. Approche intuitive

Un moyen simple de mesurer le degré de dépendance, donc la similitude entre deux signaux consiste à faire le produit moyenné des signaux, ce qui revient à calculer la puissance moyenne d'interaction (voir TP « puissance et densité spectrale de puissance »). Dans le cas de signaux s_x et s_y complexes (pour plus de généralités) numérisés sur N échantillons, la puissance moyenne peut s'écrire :

$$P_{xy} = \frac{1}{N} \sum_{n=0}^{N-1} s_x(n) s_y^*(n) \quad \text{ou encore} \quad P_{yx} = \frac{1}{N} \sum_{n=0}^{N-1} s_y(n) s_x^*(n)$$

Si on considère par exemple les signaux s_x et s_y comme des tensions et courants sinusoïdaux, les termes P_{xy} et P_{yx} ne seront non nuls que si les fréquences sont identiques (un courant harmonique ne transporte de puissance que s'il dispose d'une tension harmonique de même ordre). Cependant, même à fréquences égales, pour un déphasage de 90° , bien que les signaux aient une forte ressemblance, les termes puissances resteront nuls. Il faut donc introduire un décalage temporel pour un des signaux, et faire le calcul pour chaque valeur du décalage. On arrive alors à la fonction de corrélation de deux signaux définie par :

$$R_{xy}(k) = \frac{1}{N} \sum_{n=0}^{N-1} s_x^*(n) \cdot s_y(n+k)$$

Le calcul est identique au précédent, à la différence qu'un décalage de k échantillons est introduit sur s_y , donnant une valeur particulière $R(k)$ de la fonction de corrélation. En calculant le nombre de valeurs souhaitées (fonction du problème à résoudre), on obtient la fonction complète.

Le facteur $1/N$ évite que la valeur de la corrélation dépende du nombre d'échantillons. Il est parfois omis.

Que le décalage soit défini positivement ou négativement ne change pas fondamentalement le

résultat, aussi trouvera également l'expression $R_{xy}(k) = \frac{1}{N} \sum_{n=0}^{N-1} s_x^*(n) \cdot s_y(n-k)$

Dans le cas d'une fonction d'autocorrélation d'un signal s , on aura :

$$R(k) = \frac{1}{N} \sum_{n=0}^{N-1} s^*(n) \cdot s(n+k)$$

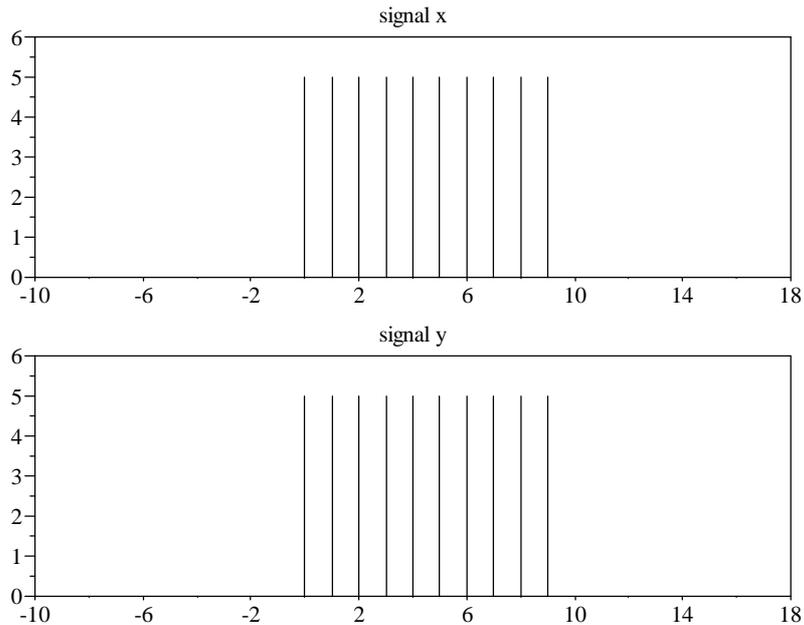
1.1. Réalisation pratique en numérique

L'acquisition d'un signal nécessite un échantillonnage, une numérisation et un fenêtrage (le calcul ne peut se faire sur un nombre infini d'échantillons). Le fait de décaler un signal dans l'opération de corrélation pose alors le problème de la connaissance des échantillons suivants ou précédents la fenêtre d'acquisition (suivant le sens du décalage).

Illustrons ce phénomène pour deux signaux continus, x et y , de 5 V d'amplitude chacun et acquis sur 10 échantillons. Les signaux étant identiques quel que soit le décalage, la fonction de corrélation devrait être constante et égale à la puissance d'interaction des deux signaux, soit $5^2=25$.

Le programme ci-après permet de visualiser l'opération faite pour $R(0)$.

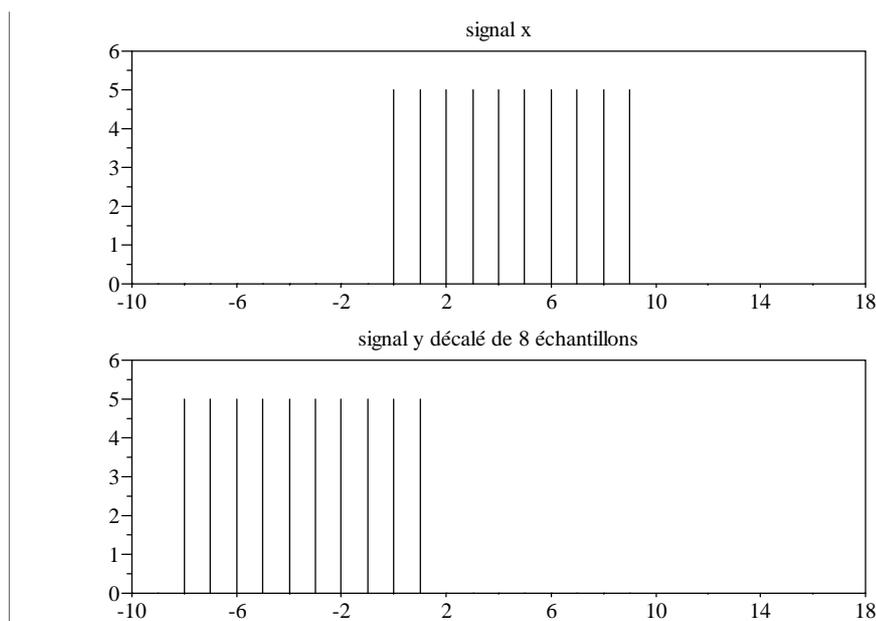
```
clear ;
//
// définition du temps et des signaux
t=(0 :9) ;x=5*ones(1,10) ;y=x ;
//
// affichage
xbasc() ; xset(«font size»,4) ;
xsetech([0,0,1,1/2]) ; plot2d3(t,x,rect=[-10,0,15,6]) ;xtitle(« signal x ») ;
xsetech([0,1/2,1,1/2]) ; plot2d3(t,y,rect=[-10,0,15,6]) ;xtitle(« signal y »)
```



On trouve bien $R_{xy}(0) = \frac{1}{N} \sum_{n=0}^{N-1} x^*(n) \cdot y(n+0) = 1/10 \cdot (10 \cdot 5^2) = 25$

Le programme ci-après permet de visualiser le calcul fait pour un décalage de 8 échantillons de y.

```
clear ;
//
// définition du temps et des signaux
t=(-10 :9) ;x=[zeros(1,10),5*ones(1,10)] ;y=[zeros(1,2),5*ones(1,10),zeros(1,8)] ;
//
// affichage
xbasc() ; xset(«font size»,4) ;
xsetech([0,0,1,1/2]) ; plot2d3(t,x,rect=[-10,0,15,6]) ;xtitle(« signal x ») ;
xsetech([0,1/2,1,1/2]) ; plot2d3(t,y,rect=[-10,0,15,6]) ;xtitle(« signal y décalé de 8 échantillons »)
```



On peut remarquer que l'opération nécessite de connaître les échantillons de y de 2 à 9 pour effectuer la sommation des échantillons de 0 à 9.

Plusieurs stratégies sont alors possibles :

- considérer le signal périodique de période égale à la longueur de la fenêtre d'acquisition, on retrouve alors la valeur 25 précédente.

- considérer les échantillons non acquis comme nuls ; c'est l'option choisie pour l'écriture de la fonction « cor » dans Scilab, l'expression de la corrélation devient alors :

$$R_{xy}(k) = \frac{1}{N} \sum_{n=0}^{N-1} s_x^*(n) \cdot s_y(n+k) = \frac{1}{N} \sum_{n=0}^{N-1-k} s_x^*(n) \cdot s_y(n+k)$$

La conséquence de ce choix est que les calculs correspondants à une valeur élevée de k se font sur peu d'échantillons, mais sont toujours divisé par N le nombre total d'échantillons ; il s'ensuit une atténuation de la fonction lorsque k augmente (le premier indice d'une matrice étant 1 dans Scilab, la sommation se fera en fait de 1 à N-k). Dans notre exemple, le résultat sera alors $R(8)=1/10 \cdot (2.5^2)=5$.

En compensant le défaut précédent par l'expression suivante :

$$R_{xy}(k) = \frac{1}{N-k} \sum_{n=0}^{N-1-k} s_x^*(n) \cdot s_y(n+k)$$

on trouvera alors $R(8)=1/2 \cdot (2.5^2)=25$.

1.2. Cas des signaux analogiques

Comme nous l'avons vu dans les TP précédents, les signaux numériques sont vus comme des signaux à puissance finie. Pour les signaux analogiques, il faut considérer les deux cas suivants :

1.2.1. Cas des signaux à puissance finie

- Corrélation $R_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} s_x^*(t) \cdot s_y(t + \tau) \cdot dt$
- Autocorrélation $R(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} s^*(t) \cdot s(t + \tau) \cdot dt$

1.2.2. Cas des signaux à énergie finie

- Corrélation $R_{xy}(\tau) = \int_{-\infty}^{+\infty} s_x^*(t) \cdot s_y(t + \tau) \cdot dt$
- Autocorrélation $R(\tau) = \int_{-\infty}^{+\infty} s^*(t) \cdot s(t + \tau) \cdot dt$

1.3. Propriétés

1.3.1. Parité pour des signaux réels

Si les signaux sont réels, les fonctions de corrélation et autocorrélation sont paires :

$$R_{xy}(\tau) = R_{xy}(-\tau) \quad \text{et} \quad R(\tau) = R(-\tau)$$

1.3.2. Valeur maximale

Un signal ressemblera d'autant plus à un autre que cet autre est le même signal non décalés :
La fonction d'autocorrélation est donc maximale pour $\tau=0$.

$$\|R(\tau)\| \leq R(0)$$

Cette propriété permettra de détecter les périodicités cachées, la valeur maximale se retrouvant au bout d'une période.

1.3.3. Puissance

Toujours pour la valeur $\tau=0$, la fonction de corrélation correspond à la puissance d'interaction des signaux et la fonction d'autocorrélation à la puissance normalisée :

$$R_{xy}(0) = P_{yx}$$

$$R(0) = P$$

1.3.4. Relation avec la convolution

Un simple changement de variable dans les expressions de la corrélation et autocorrélation permet de mettre en évidence une relation avec le produit de convolution :

$$R_{xy}(\tau) = x^*(-\tau) * y(\tau)$$

$$R(\tau) = x^*(-\tau) * x(\tau)$$

Nous verrons dans le prochain sujet que ces relations donnent de manière simple la fonction de transfert d'un filtre adapté dans une chaîne de transmission numérique.

1.3.5. Théorème de Wiener-Kintchine

Pour comprendre ce théorème, prenons un signal analogique aléatoire. Une suite d'échantillons de ce signal n'a aucune relation temporelle avec une suite d'échantillons prise plus loin dans le temps, tout comme un échantillon donné n'a aucune relation avec le suivant ou le précédent. On peut donc en déduire intuitivement que la fonction d'autocorrélation d'un tel signal se réduit à une simple impulsion de Dirac centrée en $\tau=0$.

D'autre part, un tel signal contient de l'énergie dans toute la bande de fréquence sans favoriser une fréquence plus qu'une autre. Sa densité spectrale de puissance est donc constante sur tout le spectre. Une telle DSP est la transformée de Fourier d'un Dirac.

Le théorème de Wiener-Kintchine s'énonce comme suit pour les signaux à puissance finie :

La transformée de Fourier de la fonction d'autocorrélation d'un signal est la densité spectrale de puissance de ce signal.

Pour les signaux à énergie finie, ce théorème reste valable en utilisant la fonction d'autocorrélation appropriée et la densité spectrale d'énergie.

2. Applications

2.1. Théorème de Wiener-kintchine

Proposer un programme avec Scilab générant un signal aléatoire sur 4096 échantillons, calculer la densité spectrale de ce signal, sa fonction d'autocorrélation ainsi que la transformée de Fourier de cette dernière puis conclure.

Pour le calcul de la densité spectrale de puissance, plutôt que d'élever au carré la module de la transformée de Fourier du signal comme nous l'avons fait jusqu'à présent, ce qui donne, comme nous l'avons vu, une fluctuation importante autour de la courbe, nous utiliserons la fonction « pspect » de Scilab. Celle-ci donne le spectre de puissance en utilisant la méthode du « périodogramme moyenné » qui consiste à fragmenter le signal en sous-ensembles temporels, les fenêtrer et calculer la DSP de chaque sous-ensemble. On calcule ensuite la moyenne de tous les spectres obtenus.

L'appel de cette fonction se fait de la manière suivante :

```
[spectre]=pspect[long_fenêtre, nb_points, 'fenêtre',signal)
```

Le spectre de « signal » est alors calculé sur « nb_points » points, après avoir été fragmenté en fenêtre de « long_fenêtre » points, fenêtrés par la forme « fenêtre ». On obtient des résultats corrects en choisissant une fenêtre triangulaire 'tr' et un nombre de points calculés deux fois plus grand que celui d'une fenêtre de fragmentation, par exemple [spectre]=pspect[256, 512, 'tr',signal).

2.2. Détection de la périodicité d'un signal NRZ pseudo-aléatoire

Ecrire un programme sous Scilab permettant de générer un signal NRZ aléatoire de 64 bits. Celui-ci, que l'on nommera s_nrz sera obtenu comme lors de la séance sur la DSP, avec les fonctions « sign » et « rand », puis par un suréchantillonnage d'un rapport 64 à l'aide des fonctions « matrix » et « ones » :

```
s=sign(rand(1,64,'n'));
s_nrz=5*(matrix(ones(64,1)*s,1,4096));
```

A l'aide de la fonction « corr » afficher l'autocorrélation du signal. Dilater les échelles pour visualiser les premiers échantillons de l'autocorrélation. Faire plusieurs essais et vérifier que l'autocorrélation est toujours de forme identique (triangulaire) pour les premières valeurs. Justifier cette forme par un raisonnement qualitatif.

Modifier maintenant s_nrz pour obtenir une suite de 64 bits.

Produire un autre signal résultant de la mise bout à bout 8 fois de s_nrz. Pour cela on transposera s_nrz en vecteur colonne à l'aide de l'opérateur « ' » , et on le multipliera par une matrice unitaire d'une ligne et 64 colonnes, générée par la fonction « ones(1,64) » :

```
s_per=matrix(s_nrz'*ones(1,8),4096);
```

Quelle fonction d'autocorrélation théorique peut-on attendre d'un tel signal ? Vérifier par l'expérimentation et justifier les différences éventuelles.

2.3. Détection d'un signal sinusoïdal noyé dans du bruit

Considérons un signal sinusoïdal $s(t)$ auquel s'ajoute un bruit blanc $b(t)$. Montrer par un raisonnement qualitatif que l'autocorrélation d'un tel signal est égale, sauf en $\tau=0$, à l'autocorrélation du signal $s(t)$. Calculer l'expression de cette dernière. Conclure sur la possibilité de retrouver un signal noyé dans du bruit par cette méthode.

Proposer un programme sous Scilab permettant d'afficher dans un premier temps l'autocorrélation d'un signal sinusoïdal, puis d'un signal sinusoïdal noyé dans du bruit dans un second temps.

A partir de quelle valeur de rapport signal sur bruit peut-on estimer qu'on ne peut retrouver le signal par cette méthode ?

On peut améliorer la restitution du signal en faisant non pas une autocorrélation, mais une corrélation avec un signal sinusoïdal pur, de fréquence identique à celui recherché. Cette méthode est utilisée par exemple en radioastronomie pour déterminer les pulsations radioélectriques d'étoiles lointaines, dont le signal nous arrive fortement entaché de bruit.

Comme précédemment, simplifier l'expression de la fonction de corrélation pour mettre en évidence l'efficacité de la méthode.

Proposer ensuite une illustration avec Scilab. Jusqu'à quelles valeurs de rapport signal sur bruit peut-on retrouver le signal.

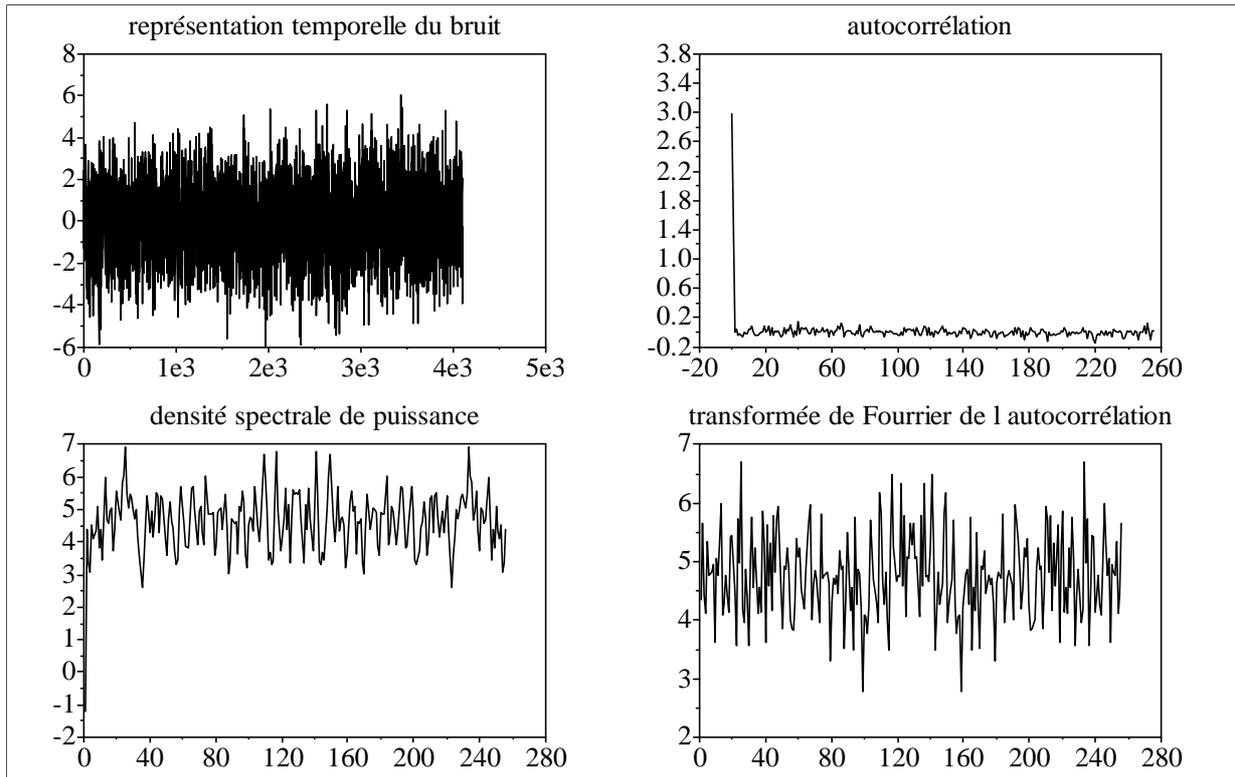
3. Exemples de solutions

3.1. Théorème de Wiener-Kintchine pour un signal aléatoire

Exemple de programme

```
clear
//
//génération d'un signal aléatoire de puissance 3 V2
bruit=sqrt(3)*rand(1,4096,'n');
//
// calcul de la DSP avec pspect
[dsp1]=10*log10(pspect(128,256,'tr',bruit));
//
// calcul de l'autocorrélation et de sa transformation de Fourier
[cov]=corr(bruit,256);
dsp2=10*log10(abs(fft(cov,-1)));
//
//initialisation de l'affichage, réglage pour une taille de fonte de 4, création d'une variable  $\tau$ 
xbasc(); xset("font size",4); tau=(0:255);
//
// affichage
xsetech([0,0,.5,.5]);plot2d(bruit);
xtitle("représentation temporelle du bruit");
//
xsetech([0,.5,.5,.5]);plot2d(dsp1);
xtitle("densité spectrale de puissance");
//
xsetech([.5,0,.5,.5]);plot2d(tau,cov,rect=[-20,-0.2,256,3.5]);
xtitle("autocorrélation");
//
xsetech([.5,.5,.5,.5]);plot2d(dsp2);
xtitle("transformée de Fourier de l'autocorrélation")
```

Voici les résultats obtenus :



On remarque que la fonction d'autocorrélation est (presque) nulle partout, sauf en 0 où elle correspond à la puissance du signal, soit $3V^2$. La transformée de Fourier de la fonction d'autocorrélation ainsi que le résultat donné par la fonction `pspect` correspondent bien à une DSP (presque) blanche de valeur $10 \lg(3)=4,7$.

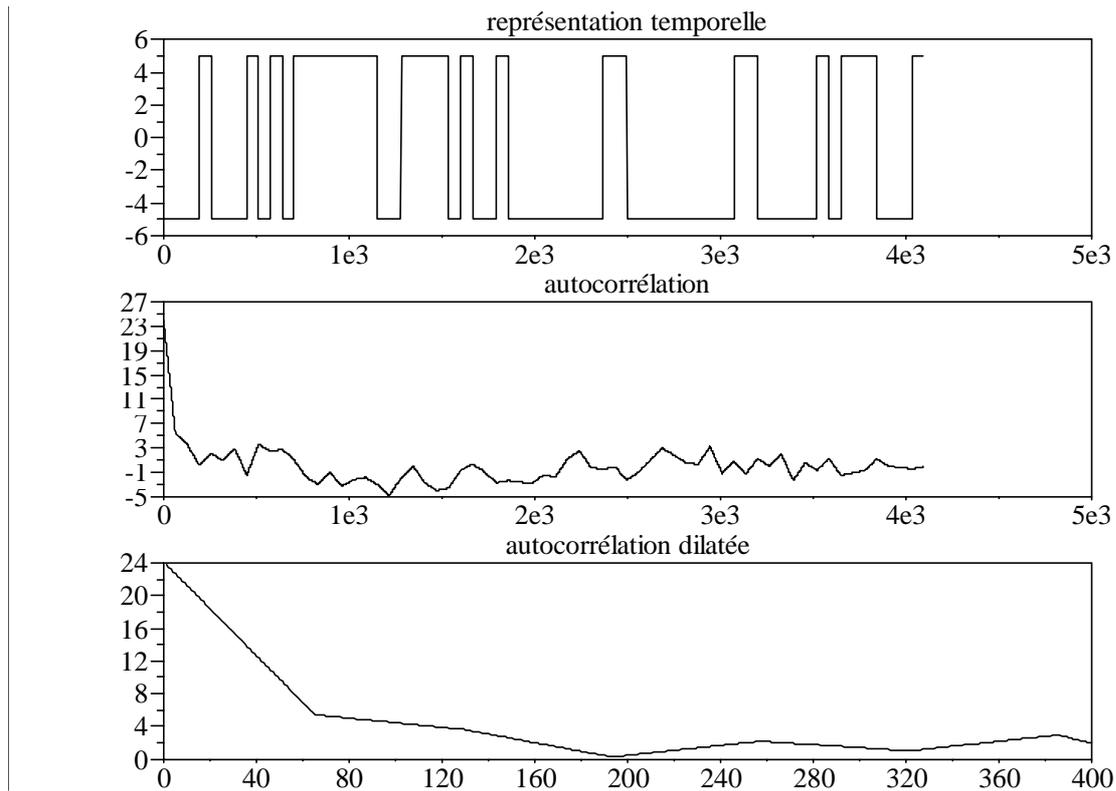
Remarque :

- la transformation de Fourier de la fonction d'autocorrélation donne normalement un réel dans notre cas ; cependant les diverses troncatures et arrondis finissent par faire apparaître une partie imaginaire, d'où la nécessité d'utiliser le module de la transformation pour calculer `dps2`.
- le rôle de la variable « tau » est simplement de permettre l'utilisation de l'instruction « `rect` », afin de mettre en évidence la raie d'amplitude 3 pour l'abscisse 0.

3.2. Périodicité d'un signal NRZ pseudo-aléatoire

Le programme ci-après génère une suite NRZ sur 8 bits avec 64 points par bit.

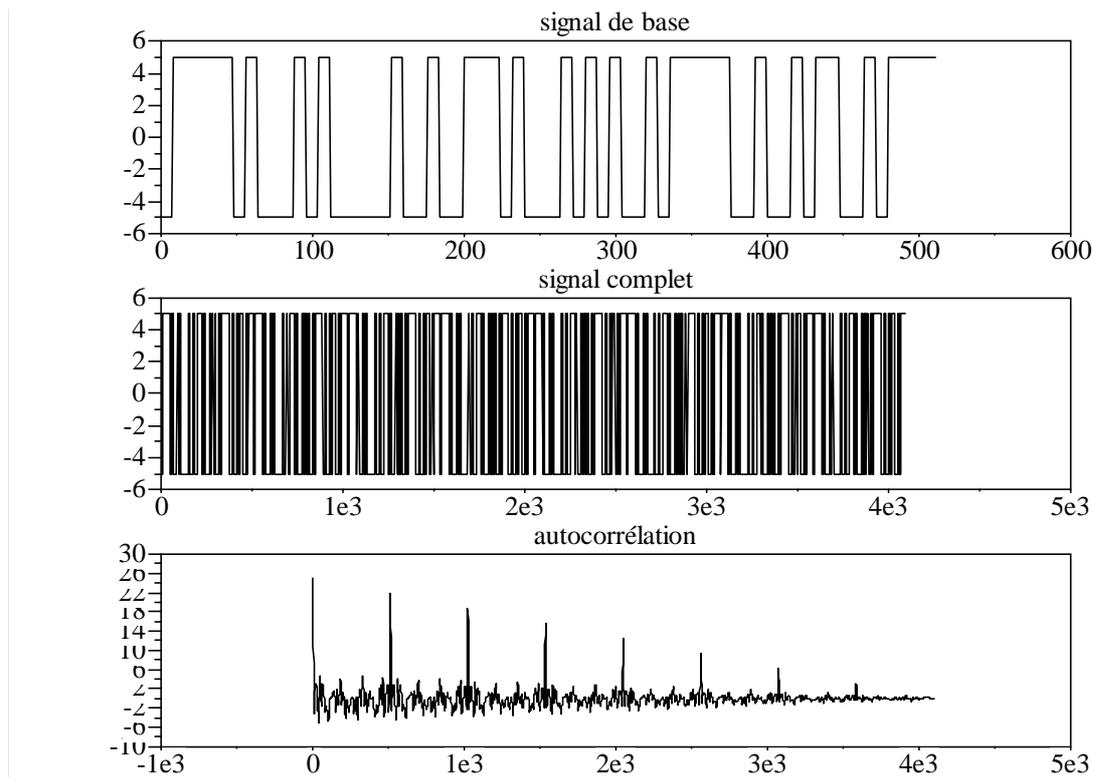
```
clear
//
//génération des informations binaires aléatoire
s=sign(rand(1,64,'n'));
//
// génération du signal NRZ à partir des informations binaires par suréchantillonnage.
s_nrz=5*(matrix(ones(64,1)*s,1,4096));
//
// autocorrélation
[cov]=corr(s_nrz,4096);
//
//affichage des résultats
t=(0:4095);tau=t;
xbas(); xset("font size",4);
xsetech([0,0,1,1/3]);plot2d(t,s_nrz,rect=[0,-5.5,4095,5.5]);xtitle("représentation temporelle");
xsetech([0,1/3,1,1/3]);plot2d(tau,cov);xtitle("autocorrélation");
xsetech([0,2/3,1,1/3]);plot2d(tau,cov,rect=[0,-10,500,30]);xtitle("autocorrélation dilatée")
```



Générons maintenant un signal pseudo-aléatoire;

```
clear
//
//signal de base NRZ de 64 bits suréchantillonné d'un rapport 8
s=sign(rand(1,64,'n')); s_nrz=5*(matrix(ones(8,1)*s,1,512));
//
// 8 périodes de signal 64 bits
s_per=matrix(s_nrz*ones(1,8),4096);
//
// autocorrélation
[cov_per]=corr(s_per,4096);
//
//affichage des résultats
t=(0:511);t_per=(0:4095); tau=t_per;
xbasc(); xset("font size",4);
//
xsetech([0,0,1,1/3]);plot2d(t,s_nrz,rect=[0,-5.5,511,5.5]);
xtitle("signal de base");
//
xsetech([0,1/3,1,1/3]);plot2d(t_per,s_per,rect=[0,-5.5,4095,5.5]);
xtitle("signal complet");
//
xsetech([0,2/3,1,1/3]);plot2d(tau,cov_per,rect=[-10,-10,4096,30]);
xtitle("autocorrélation");
```

Voici les résultats obtenus



Comme on peut le remarquer, l'autocorrélation présente une raie chaque fois que le retard correspond à une période du signal. Théoriquement, les amplitudes de ces raies devraient être identiques, l'atténuation étant due à l'algorithme utilisé par Scilab pour effectuer le calcul.

On peut donc à l'aide de l'autocorrélation détecter les périodes cachées (ou non) d'un signal. On peut également resynchroniser deux signaux décalés (application pour resynchroniser le code d'un récepteur d'une transmission de type AMRC –accès multiples à répartition par le code- dans les systèmes à étalement de spectre par exemple).

3.3. Détection d'un signal noyé dans du bruit

L'autocorrélation du signal R_{SE} peut se décomposer de la manière suivante :

$$R_{SE} = R_S + R_B + R_{BS} + R_{SB}$$

Le signal utile et le bruit n'étant pas corrélés, les deux derniers termes sont nuls. Comme nous l'avons vu, l'autocorrélation d'un bruit blanc produit une impulsion pour un décalage nul et zéro ailleurs. La fonction d'autocorrélation d'un signal sinusoïdal donne un autre signal sinusoïdal. L'information phase est cependant perdue, le maximum de la fonction étant obtenu à l'origine ; on a donc affaire systématiquement à un cosinusoïde.

Le programme suivant génère un bruit blanc aléatoire, un signal sinusoïdal pur de 5 V, et fait la somme des deux. La puissance du bruit est déterminée par le terme SB représentant le rapport signal sur bruit de puissance. On effectue ensuite l'autocorrélation du signal bruité.

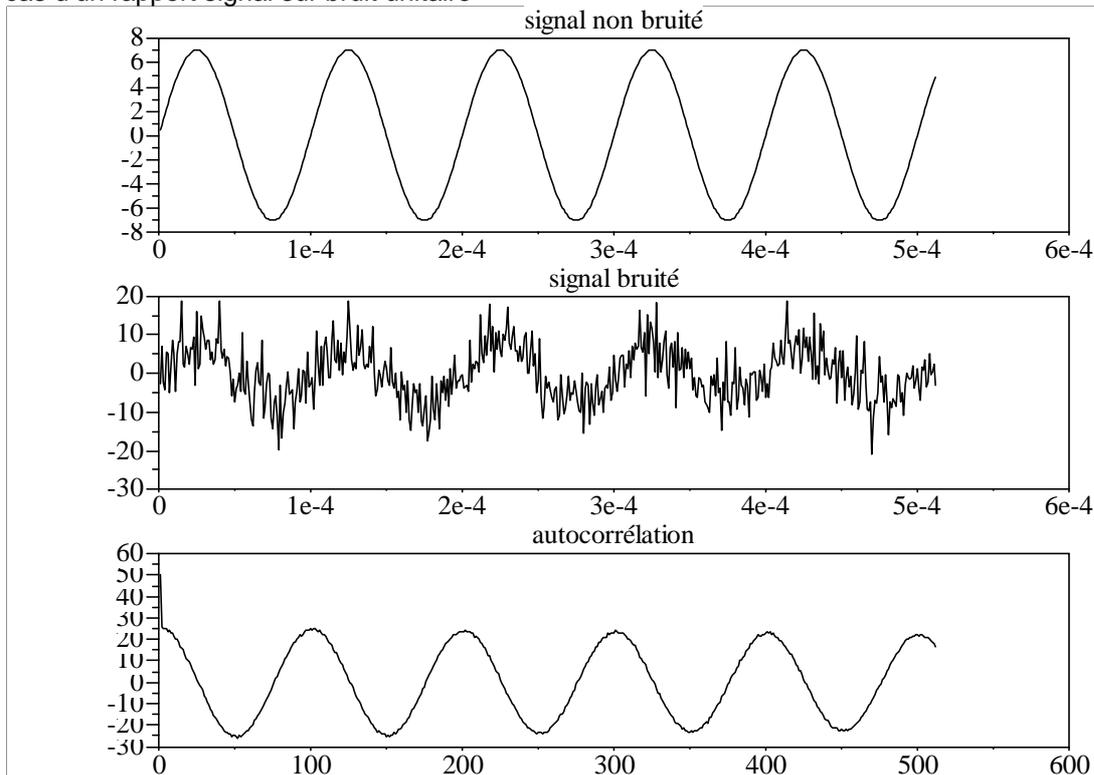
```
clear;
// définition du temps
t=1e-6*(1:4096);
//
// définition du rapport signal sur bruit et des signaux
SB=.1; b=sqrt((5^2)/SB)*rand(1,4096,'n'); s=5*sqrt(2)*sin(2*pi*10e3*t); se=s+b;
//
// autocorrélation
[cov]=corr(se,512);
//
```

```
// affichage
xbasc(); xset("font size",4);
xsetech([0,0,1,1/3]); plot2d(t(1 :512),s(1 :512),rect=[0,-8,512e-6,8]);
xtitle("signal non bruité");
//
xsetech([0,1/3,1,1/3]); plot2d(t(1 :512),se(1 :512));;
xtitle("signal bruité");
//
xsetech([0,2/3,1,1/3]); plot2d(cov);
xtitle("autocorrélation")
```

Remarque : un « bug » de Scilab sous windows (pas de problèmes sous Unix), fait que lorsque l'on cherche à afficher une courbe en limitant le domaine d'affichage avec la fonction « rect », la courbe continue d'être affichée en dehors du rectangle des abscisses et ordonnées extrêmes. Pour pallier cet inconvénient, lorsque la limitation d'affichage ne concerne que l'axe des abscisses, on peut limiter les points affichés en ne prenant par exemple pour le signal « s » que les 512 premiers par la syntaxe « plot2d (t(1 :512),s(1 :512)) ».

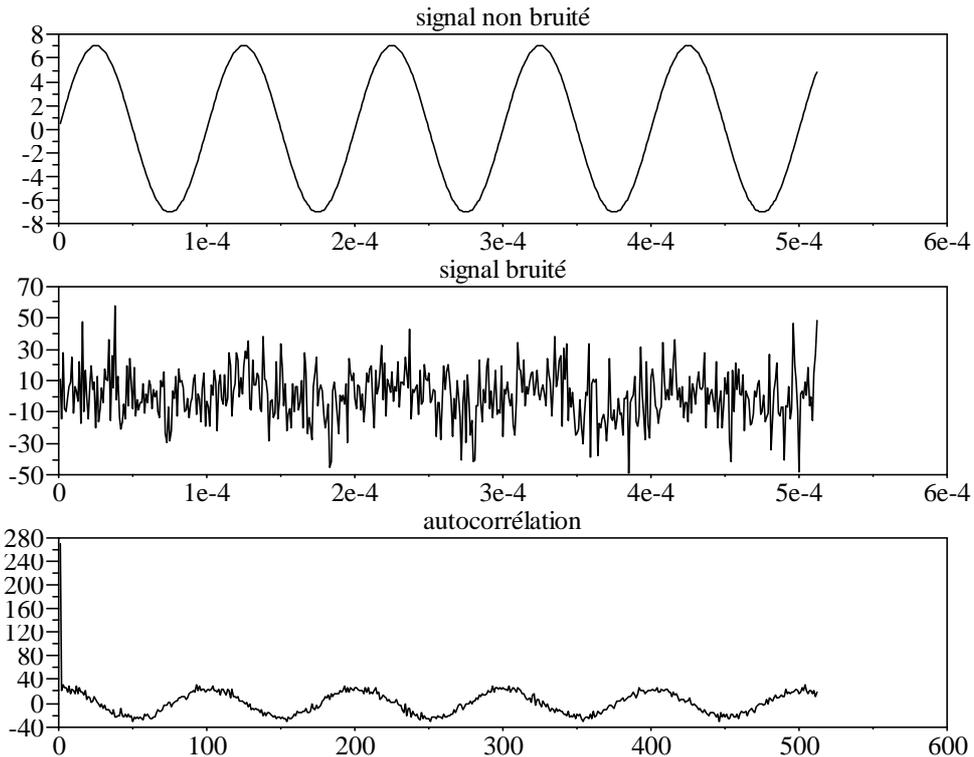
Résultats obtenus

cas d'un rapport signal sur bruit unitaire



cas d'un rapport SB de 0,1

Traitement du signal avec Scilab : corrélation



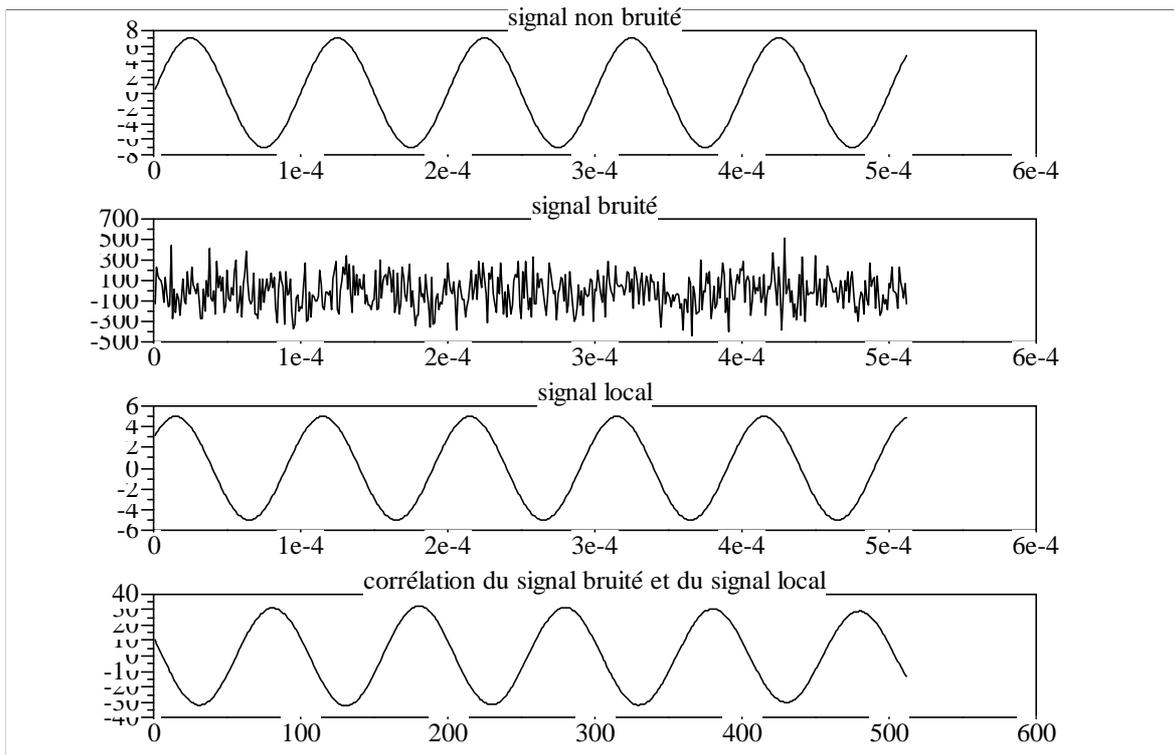
Les résultats obtenus vérifient bien la théorie, la fonction d'autocorrélation présentant une raie importante à l'origine, puis se développant en cosinusoïde. La raie d'origine est d'autant plus importante que le rapport signal sur bruit est faible. De même la pureté de la cosinusoïde décroît avec ce rapport.

La fonction d'autocorrélation n'a été représentée que sur un faible nombre d'échantillons, afin d'éviter l'effet d'atténuation du au moyennage.

Pour retrouver le signal émis, corrélons maintenant le signal reçu avec une sinusoïde de référence interne au récepteur. Le déphasage entre le signal recherché et le signal du récepteur sont à priori quelconques (ici $\pi/5$).

```
clear;
//
// définition du temps
t=1e-6*(1:4096);
//
// définition du rapport signal sur bruit et des signaux
SB=.001;
b=sqrt((5^2)/SB)*rand(1,4096,'n');
s=5*sqrt(2)*sin(2*%pi*10e3*t);
se=s+b;
s_recep=5*sin(2*%pi*10e3*t+%pi/5);
//
// corrélation
[cov]=corr(se,s_recep,512);
//
// affichage
xbasc(); xset("font size",4);
xsetech([0,0,1,.22]); plot2d(t(1 :512),s(1 :512),rect=[0,-8,512e-6,8]);xtitle("signal non bruité");
xsetech([0,1/4,1,.22]); plot2d(t(1 :512),se(1 :512));xtitle("signal bruité");
xsetech([0,2/4,1,.22]); plot2d(t(1 :512),s_recep(1 :512), rect=[0,-6,512e-6,6]);xtitle("signal local");
xsetech([0,3/4,1,.22]); plot2d(cov);xtitle("corrélation du signal bruité et du signal local")
```

Chronogrammes pour un rapport signal sur bruit de 0,001



On observe cette fois une nette amélioration de la restitution du signal en sortie (alors que les chronogrammes ci-dessus sont tracés pour un rapport signal sur bruit 10 fois plus faible que dans l'exemple précédent), avec cette fois la possibilité de retrouver la phase du signal reçu : si ϕ est le déphasage entre les sinusôides reçue et locale, la fonction d'autocorrélation est en effet maximale pour la valeur de τ vérifiant :

$$\omega(t+\tau)+\phi=\omega t+2k\pi, \text{ soit } \omega\tau+\phi=2k\pi$$

connaissant τ on peut alors en déduire ϕ , ce qui est un avantage par rapport à la méthode de l'autocorrélation. La fréquence du signal reçu doit par contre être à peu près connue (une wobulation du signal local est possible, suivi d'une détection de valeur maximale de la corrélation).

Bibliographie

Traitement du signal et acquisition de données par F. Cottet chez Dunod

Transmission de signaux par C. More chez Tec & Doc

Transmission de l'information par Protière, Fraisse, Marty-Dessus chez Ellipse