Compte rendu TP4 de Traitement du Signal : Echantillonnage

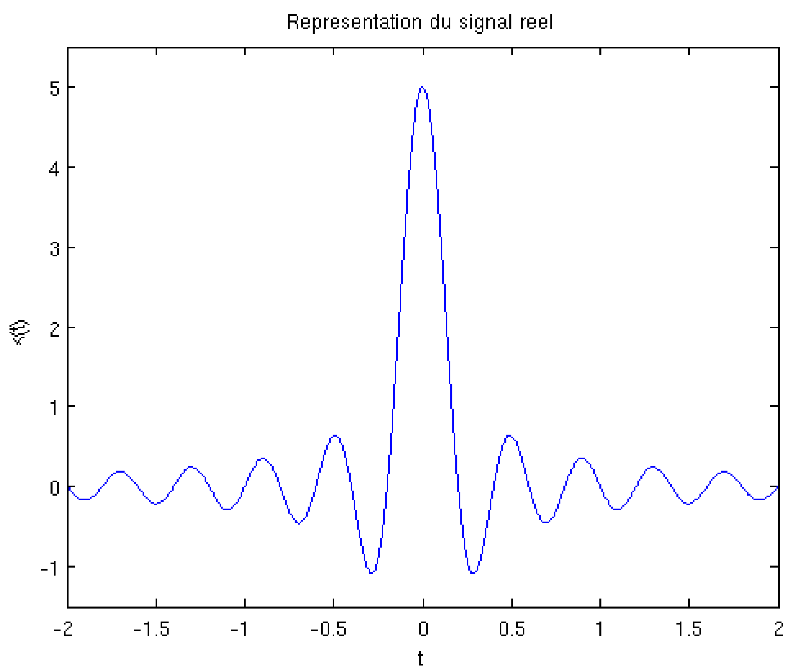
Thomas Bassetto
Antony Gardez

1 Représentation du signal réel

Nous traçons dans un premier temps notre signal $x(t) = \frac{sin(pi*F_0*t)}{pi*t}$ avec un pas très petit (T=0.001) pour simuler un signal continu. Pour tracer le sinus cardinal, on utilise la fonction *sinc_asi* de la manière suivante :

```
f0=5;

% mise en place de l'abscisse pour les courbes
T0=0.001;
t = -2:T0:2;

xt=f0*sinc_asi(f0*t);
```

La courbe obtenue est la suivante :

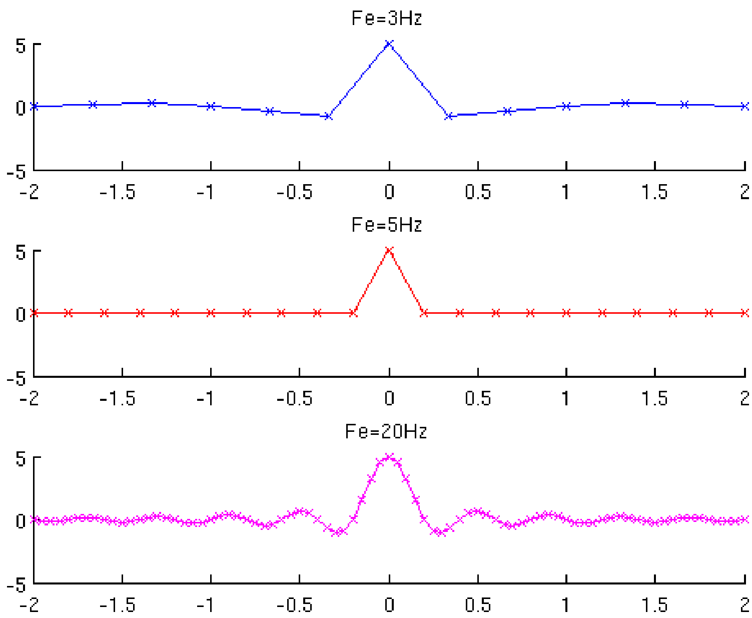


2 Echantillonnage idéal

On discrétise le signal *x(t)* défini précédemment par un échantillonneur idéal, à savoir un peigne de Dirac. Pour échantillonner *x(t)*, nous gardons la même fonction mais nous l'appelons avec un nombre de point plus faible. on récupère les points pour les fréquences *3*, *5* et *20*. Nous utilisons par exemple la commande suivante pour la valeur 3 Hz :

```
t1=-2:1/Fe1:2;
xt1=f0*sinc_asi(f0*t1);
```

Voilà les résultats :



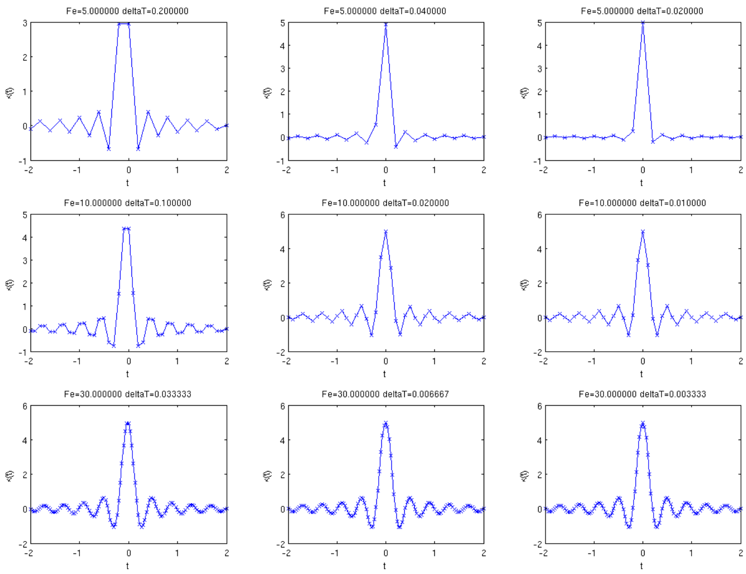
On peut tout de suite conclure que plus la fréquence d'échantillonnage est grande, plus la forme du signal $x_e(t)$ est proche de celle de *x(t)* et donc plus la perte d'information pendant l'échantillonnage est faible.

Cependant pour effectuer un bon échantillonnage, il est quand même important de respecter le théorème de Shannon, c'est à dire $F_e > 2 * F_{max}$. Dans le premier et deuxième cas le théorème n'est pas respecté donc la courbe obtenue n'est pas fidèle à l'originale. De plus, pour la deuxième courbe, F_e est égale à f_0 donc mis à part le pic en t_0 la courbe semble plate.

3 Echantillonnage réel

Pour réaliser un échantillonnage réel on utilise un échantillonneur moyenneur. Au lieu d'avoir une discrétisation avec une impulsion infiniment brève on utilise une impulsion de largeur finie (Te) et on considère la valeur moyenne de x(t) pendant la durée de l'impulsion ! On appelle ça "réel" car dans la "vrai vie"© on ne peut pas avoir de peigne de dirac pour échantillonner, l'impulsion a toujours une durée ΔT.

Voici 9 tests pour différentes valeurs de Fe et ΔT :



On constate que le parametre le plus important est la fréquence d'échantillonage. Plus elle est élevée, c'est à dire plus la période d'échantillonnage est petite, plus la courbe ressemble au sinus cardinal, ce qui est logique : on prend plus de point et on suit donc de manière plus fidèle les courbures. En ce qui concerne Δ, plus sa valeur est petite, plus on est proche d'un échantillonage idéal, avec un peigne de Dirac donc plus la coubr est fidèle à l'originale.

L'échantillonage est donc de meilleure qualité lorsqu'on a une très petite période d'échantillonnage Te et un très petit ΔT.

4 Recontruction

La reconstruction consiste à établir un signal analogique continu à partir d'un signal numérique discret. Le problème est de trouver les valeurs manquantes entre deux valeurs connues. On pratique l'extrapolation. Nous allons donc comparer deux méthodes d'extrapolation : d'ordre 0 et d'ordre 1.

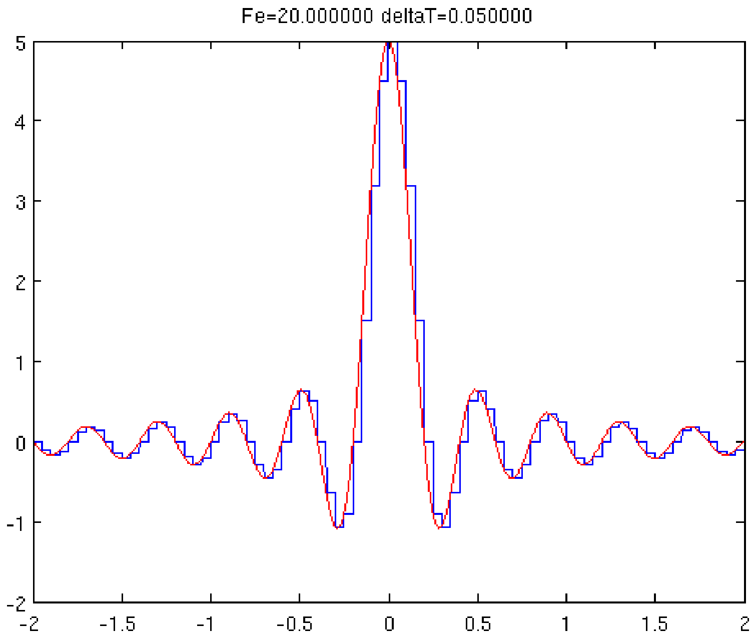
4.1 Ordre 0

On applique l'échantillonneur d'ordre 0 au signal obtenu via l'échantillonnage idéal d'abord. Voici le code utilisé :

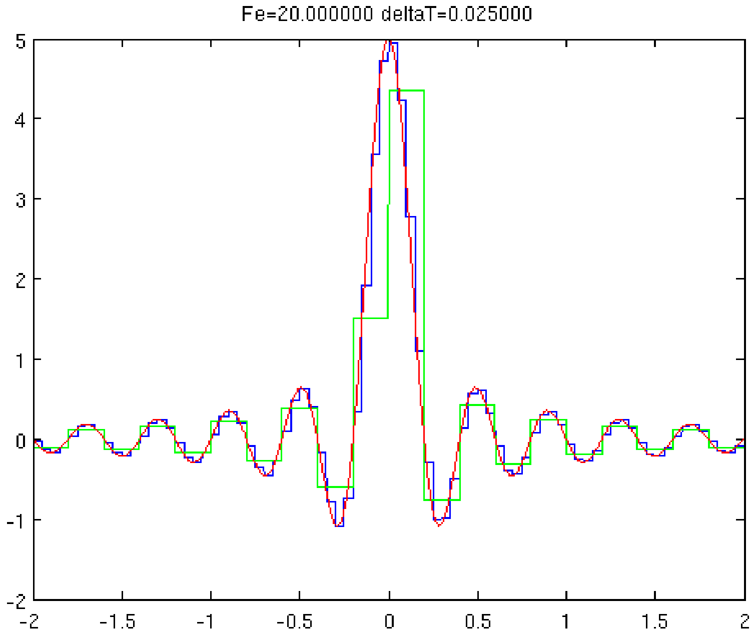
```
Fe3= 20;  % fréquence d'échantillonnage
t3=-2:1/Fe3:2;  % mise en place des abscisse avec le nb de points qui va bien
xt3=f0*sinc_asi(f0*t3);  % on récupère les valeurs pour ces points

for n=1:length(t3)-1
  indice=find( (t>=t3(n)) & (t<t3(n+1)) );
  Mat_Moyideal(indice)=xt3(n);  % on fixe la valeur sur tous les indices correspondants
end

```

```
plot(t(1:length(t)-1), Mat_Moyideal);
hold on
plot(t, xt, 'r');  % affichage de la courbe réelle de base
```

Et voici la courbe obtenue :



Puis on applique également l'échantillonneur d'ordre 0 au signal obtenu via l'échantillonage réel, avec $F_e = 20Hz$ et $\triangle T = \frac{T_e}{2}$.



Le courbe rouge est le signal du départ (question 1). La bleu est celle reconstruite à partie du signal échantilloné avec $F_e = 20Hz$ et en vert celle avec $F_e = 5Hz$.

On voit bien que plus F_e est petit, moins la reconstruction est précise.

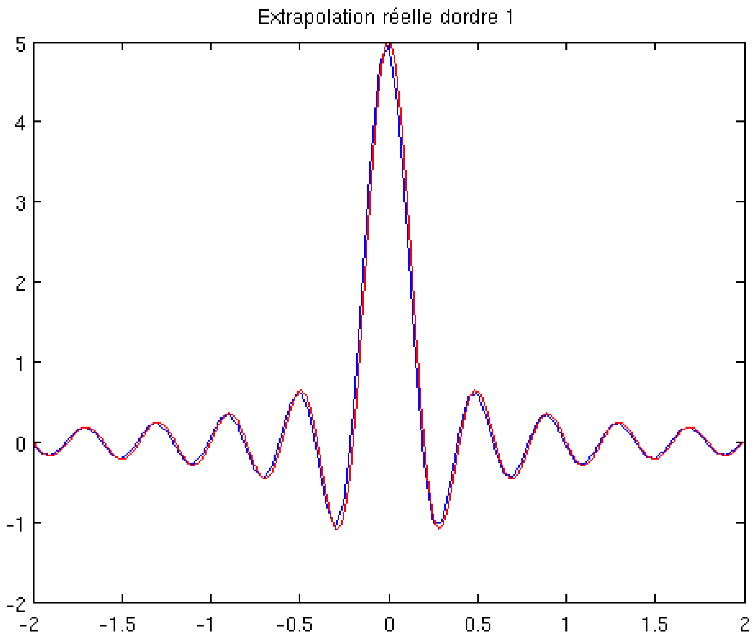
On remarque de plus que lorsque le signal échantillonné de manière idéale est reconstruit, la courbe passe par ses sommets, alors que non pour le signal réel. Ceci s'explique par le fait que dans le cas idéal nous avons pris la valeur exacte à chaque abscisse, alors que nous avons fait la moyenne sur un petit temps pour le signal réel.

4.2 Ordre 1

Pour améliorer la reconstruction de notre signal on utilise maintenant un extrapolateur d'ordre 1, c'est à dire que l'on relie tous les points échantillonnés par une droite dont on a calculé son coefficient directeur. Voici la traduction en code matlab de cette méthode :

```
% préparation du signal
Te = 1/20;
deltaT=Te/2;
ttemp=-2:Te:2;

for n=1:length(ttemp)
    indice=find( (t>=ttemp(n)) & (t<ttemp(n)+deltaT) );
    Mat_Moyreel(n)=mean(xt(indice));
end
% nous venons de recréer un signal échantillonné avec Te = 1/20

for n=1:length(ttemp)-1
    i=find( (t>=ttemp(n)) & (t<ttemp(n+1)) ); % on récupère tous les indices des points de notre droite
% création des coefficients pour la droite reliant les deux points n et n+1
    a=(Mat_Moyreel(n+1)-Mat_Moyreel(n))/(ttemp(n+1)-ttemp(n));
    b=Mat_Moyreel(n);
    ordre1(i)=a*(t(i)-ttemp(n))+b;  % Mise à jour des points dans notre courbe reconstruite
end
```

```
% il ne nous reste plus qu'à afficher tout cela
plot(t(1:length(t)-1),ordre1);
hold on
plot(t, xt, 'r');
```



En effet, la courbe semble évidemment mieux reconstruite puisqu'on ne se contente plus uniquement de fixer les valeurs à certains points, mais on extrapole, on prévoit l'allure de la courbe entre les points que nous connaissons. Ayant un signal relativement simple et correctement échantillonné (le théorème de Shannon est, entre autres, respecté), la courbe est bien reconstruite en sortie.