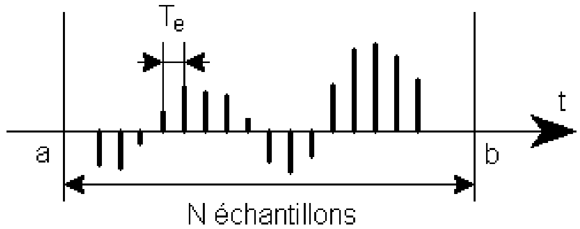
TP MATLAB
Transformée de FOURIER

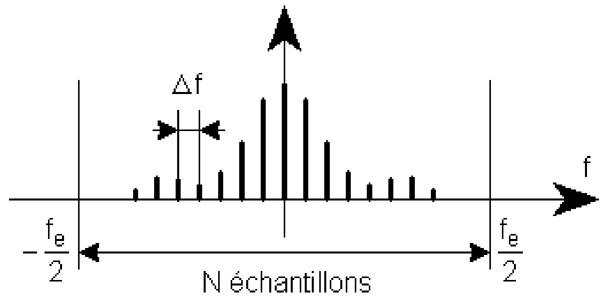
I. La fonction Transformée de FOURIER discrète

Description

On considère un signal analogique $x_a(t)$, fonction continue de la variable temps. Pour constituer le signal discret x(n) de N valeurs, on "échantillonne" ce signal $x_a(t)$ avec une période T_e, sur l'intervalle [a, b[. En pratique, cela revient à constituer x(n) selon :

$$x(n) = x_a(\ (n-1).T_e + a\ ) \quad \text{avec } T_e = (b-a) / N \quad \text{et} \quad n = 1, 2, \dots, N.$$





T_e est la **période d'échantillonnage**
$f_e = 1/T_e$ est la **fréquence d'échantillonnage**

La fonction Matlab `X=tfour(x)` donne la transformée de FOURIER discrète de ce signal discret x(n) dans un vecteur X de N valeurs. Ce vecteur X correspond à l'échantillonnage de la fonction transformée de FOURIER de $x_a(t)$. Les fréquences représentées sont les fréquences de $-f_e/2$ à $f_e/2$ (ou à peu près). L'écart, en fréquence, entre deux échantillons successifs dans X est donc de :

$$\Delta f = f_e/N = 1/(T_e\,.\,N) = 1/(b-a) \text{ Hertz} \quad (\text{ ou } s^{-1}).$$

La fonction Matlab `x=tfourinv(X)` permet de calculer la transformée de FOURIER inverse de la fonction discrétisée X et donc de retrouver x.

Application

On choisit N = 16384 échantillons, a = -25 secondes et b = 25 secondes.
Avec ces paramètres, on échantillonne les 8 fonctions suivantes :

$x_0(t) = C$

$x_1(t) = \cos(2.\pi.f_0.t)$

$x_2(t) = \sin(2.\pi.f_0.t)$

$x_3(t) = \exp(-\beta.t).U(t)$

$x_4(t) = \delta(t-\Delta t)$

$x_5(t) = \exp(i.2.\pi.f_0.t)$

$x_6(t) = rect_\tau(t)$

$x_7(t) = \exp(-\pi.t^2)$

pour différentes valeurs de C, f_0, β, τ et Δt.

Questions :

- Quelle est la période d'échantillonnage ?
- Quelle est la fréquence d'échantillonnage ?
- Affichez ces fonctions ainsi que leurs spectres (utiliser les spectres amplitude / phase ou partie réelle / imaginaire selon les plus représentatifs).
- Pour les fonctions périodiques, on choisira une fréquence f_0 qui donne un nombre entier de périodes entre a et b.
- Que se passe-t-il lorsque la fréquence f_0 choisie donne un nombre non-entier de périodes entre a et b ? Donner un exemple et expliquer.
- Vérifiez que la fonction `tfourinv(X)` permet bien de récupérer le signal d'origine.
- Essayez de construire une version périodique de $x_6(t)$. Comment se transforme son spectre ? Expliquez.
- La fonction $x_7(t)$ est une gaussienne. Calculez sa transformée de Fourier théorique et vérifiez sur les graphiques, sachant que :

$$\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}.$$

ATTENTION :

*Les graphiques "temporels" des fonctions devront avoir **l'échelle des abscisses en secondes**, les graphiques "fréquentiels" devront avoir **l'échelle des abscisses en Hertz**. On repérera donc les pas de discrétisation temporelle et fréquentielle, la fréquence nulle, les fréquences extrêmes représentées … Utilisez pour cela, les fonctions Matlab plot, figure, hold on, hold off, axis … pour les affichages, et real, imag, abs et angle pour avoir la partie réelle, la partie imaginaire, le module et l'argument des fonctions complexes.*

II. Echantillonnage et aliasing

On considère la famille de fonctions g_f , de paramètre f, définies par :

$$g_f(t) = -\cos(2\pi.f.t) + \cos(2\pi.(f + \Delta f).t) + \cos(2\pi.(f + 2.\Delta f).t) \quad \text{avec } \Delta f = 10$$

Questions :

- Quels sont les spectres théoriques des fonctions g_{50} et g_{250} ?
- Reprenez les paramètres précédents (N = 16384 échantillons, a= -25 secondes et b=25 secondes) et échantillonnez les deux fonctions précédentes. Expliquez les différences observées par rapport aux versions théoriques.
- Pour quelles valeurs positives de la fréquence f obtient-on (ou plutôt a-t-on l'impression d'obtenir) un simple cosinus ? Quelles sont les valeurs possibles de la fréquence de ce simple cosinus ? Expliquez.

III. Transmission par modulation d'amplitude

Description

La transmission d'informations à travers un canal unique (câbles, fibres optiques, air, espace...) nécessite bien souvent le codage et l'adaptation de ces informations au canal de transmission (utilisation des fréquences qui se propagent ...).
Le problème est le suivant : on veut transmettre simultanément plusieurs signaux $s_i(t)$ vers un destinataire distant à travers un seul canal (de l'air par exemple). Le signal reçu c(t) contient tous les $s_i(t)$ et on veut pouvoir extraire indifféremment chacun de ces signaux de c(t).
La modulation d'amplitude est une des façons les plus simples pour résoudre ce problème : on se sert des $s_i(t)$ pour moduler l'amplitude de signaux sinusoïdaux de fréquences f_i . Chaque émetteur construit son propre signal modulé. Le signal résultat transmis et donc reçu est alors :

$$c(t) = \sum_i s_i(t).\cos(2\pi f_i t)$$

Le signal c(t) doit ensuite être « démodulé » au point de réception pour extraire chacun des signaux $s_i(t)$. On construit pour cela de signal $d_i(t)$ en remultipliant c(t) par $\cos(2\pi.f_i.t)$ pour reconstruire $s_i(t)$.

- Que donne le calcul théorique pour le spectre du signal modulé c(t) et le spectre du signal démodulé $d_i(t)$?
- Quel est l'effet de la modulation d'un cosinus par un signal $s_i(t)$?
- Quels sont les critères de choix des f_i ?
- Quels traitements faut-il ensuite appliquer à $d_i(t)$ pour retrouver $s_i(t)$?

Application

En reprenant les paramètres précédents (N = 16384 échantillons, a= -25 secondes et b=25 secondes), réalisez les opérations de modulation et démodulation avec Matlab avec différentes valeurs de porteuses f_i et pour les signaux suivants :

$$s_1(t)=\sum_{n=1}^{4} n.\cos(2\pi.n.t) \qquad s_2(t)=\sum_{n=1}^{4} (5-n).\cos(2\pi.n.t)$$

IV. Localisation de formes par Corrélation

Description

La localisation d'un signal court F de forme particulière dans un signal plus long S a de nombreuses applications en reconnaissance de formes, tracking ou identification. Lorsque l'on dispose d'un modèle précis des formes que l'on cherche, la façon la plus directe de procéder à cette localisation est de comparer le signal S à tous les signaux F_i recherchés, en réalisant toutes les superpositions possibles par translation. Pour chaque superposition, un indicateur de ressemblance est calculé. Si il est suffisamment élevé, on considère que la forme est trouvée.
Cet indicateur de ressemblance peut être, par exemple, construit à partir d'une somme de différences ou d'une somme de produit. Dans le deuxième cas, nous pouvons construire une carte de ressemblance C à partir de la formule (pour un signal 1D, continu):

$$Corr(t) = \int_{-\infty}^{+\infty} S(\tau).F(t+\tau)d\tau$$

Cette carte *Corr(t)* correspond au signal de *corrélation* entre S et F. Pour une translation particulière t, la valeur de *Corr(t)* traduit la ressemblance de S et de F. La valeur *Corr(t)* est maximale si S et F sont identiques pour cette translation.

Pour un signal discret à deux dimensions, comme une image par exemple, la formule précédente devient :

$$Corr(l,c) = \sum_i \sum_j S(i,j).F(l+i,c+j)$$

En terme d'efficacité de la méthode, on se rend vite compte du grand nombre de calcul à réaliser (une somme de produits pour chaque translation …), d'autant plus grand que la forme recherchée est grande. Il est pourtant possible d'améliorer les choses en remarquant la similitude entre cette formule de corrélation et celle du produit de convolution, dont l'expression discrète pour un signal à deux dimensions est :

$$Conv(l,c) = \sum_i \sum_j S(i,j).F(l-i,c-j)$$

Ce produit de convolution présente l'avantage de pouvoir se calculer rapidement dans l'espace fréquentielle puisqu'il correspond à un produit simple dans cet espace. Nous utiliserons donc la convolution pour calculer la corrélation. En effet, la convolution de S(l,c) avec F(-l,-c) correspond à la corrélation de S(l,c) avec F(l,c).

Quelles sont à votre avis, les avantages et limitations de cette méthode de localisation de formes ?

Application

On cherche à compter le nombre d'occurrences de trois formes bien définies F_1, F_2 et F_3, dans l'image **`'ImageFormes.png'`**. Utilisez la corrélation pour réaliser cela. Vous définirez les formes par inspection directe de l'image.

V. Filtrage

A partir de la fonction gaussienne de la première partie, on peut construire les fonctions de transfert de filtres très utilisés en pratique, applicables sur des signaux 1D, 2D ou plus. La définition de ces filtres dans un espace fréquentiel 2D serait :

$$H(u,v) = \exp(-K.(u^2 + v^2))$$

Testez les effets de ces filtres sur les images en niveaux de gris de votre choix pour différentes valeurs de K. Conclusions ?

Application

On veut diviser par 4 chaque dimension de l'image « GirafesIV.png » qui occupe beaucoup de place, en la sous-échantillonnant. Pour cela, on construit une première image en prenant un point sur 4 dans l'image (en ligne et en colonne).
Quelles sont les raisons de la forte dégradation constatée ?
Comment améliorer ce résultat ?

Quelques fonctions spéciales images :

```
function test

% Lecture de l'image
[im, map]=imred('imageNB.bmp') ;

% Affichage de l'image sur la figure 1
figure(1)
image(im)
colormap(map)

% Calcul de la FFT 2D
IM=fftshift(fft2(im));

% Recadrage du spectre d'amplitude
% pour affichage sous forme d'image niveaux de gris
affIM=abs(IM)+1;
maxi=max(max(affIM));
mini=min(min(affIM));
affIM=(log(affIM)-log(mini))/(log(maxi)-log(mini))*255;

figure(2)
image(affIM)
colormap(map)
```

Enfin, le produit simple entre deux matrices A et B se fait par : `A.*B`