PARTIE 2

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 r(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 . Le signal transmis (et donc reçu) est alors :

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

Le signal c(t) peut ensuite être « démodulé » au point de réception pour extraire chacun des signaux $s_i(t)$. On construit pour cela le 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= -50 secondes et b=50 secondes), réalisez les opérations de modulation et démodulation avec Matlab pour le signal suivant :

$$s(t)=\sum_{n=1}^{4} n.cos(2\pi.n.t)$$

Visualisez les spectres des différentes fonctions et comparez avec les résultats théoriques.

IV. Restauration d'images par filtre de Wiener

Description

De nombreuses dégradations d'images (flou, bougé, défauts d'optique …) peuvent se modéliser par le passage de l'image idéale $I_{Idéale}(u,v)$ dans un filtre linéaire de dégradation $H(u,v)$. Si on arrive à modéliser de façon suffisamment précise ce filtre de dégradation $H(u,v)$, on peut alors atténuer les effets de la dégradation sur l'image $D(u,v)$ par filtrage inverse. En fréquence, cela s'écrira de façon simpliste :

$$D(u,v)=I_{Idéale}(u,v).H(u,v) \quad \text{et donc} \quad I_{Idéale}(u,v)=D(u,v)/H(u,v)$$

De façon, plus réaliste, il faut tenir compte du bruit, toujours plus ou moins présent :

$$D(u,v)=I_{Idéale}(u,v).H(u,v)+B(u,v) \quad \text{et donc}$$
$$I_{Idéale}(u,v)=[D(u,v)-B(u,v)] \ / \ H(u,v)$$

Cependant, pour pouvoir utiliser cette formule, il faudrait connaître précisément ce bruit $B(u,v)$, ce qui n'est pas possible, car il est aléatoire par nature. De plus, les zéros de $H(u,v)$ posent un problème évident. Pour pouvoir néanmoins atténuer la dégradation de l'image $D(u,v)$, on peut construire des filtres de restauration qui utilisent des caractéristiques statistiques de ce bruit. Le filtre de Wiener $W(u,v)$ est un exemple de ce type de filtres. Il est donné par :

$$W(u,v)=\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+\dfrac{P_B(u,v)}{P_I(u,v)}}$$

et $I_{idéale}(u,v)$ sera alors assez bien estimée par $D(u,v).W(u,v)$

Ici, $P_B(u,v)$ et $P_I(u,v)$ sont des estimations des spectres de puissance du bruit et de l'image idéale respectivement. Ce ne sont que des estimations puisqu 'elles ne sont pas accessibles.

Application

L'image '*imageFloue.png*' de 512 x 512 pixels a été obtenue à la suite d'une mauvaise opération de numérisation de la page. Nous voulons cependant savoir ce que raconte ce texte que l'on devine. Nous choisissons pour cela d'appliquer la méthode de restauration par filtre de Wiener. Pour pouvoir calculer $W(u,v)$, nous devrons déterminer les trois fonctions $H(u,v)$, $P_B(u,v)$ et $P_I(u,v)$.

Le filtre de dégradation $h(x,y)$, de fonction de transfert $H(u,v)$, sera synthétisé pour traduire la dégradation observée. C'est la convolution d'une image $i(x,y)$ par l'image $h(x,y)$ (ou la multiplication des deux transformées de Fourier $I(u,v)$ et $H(u,v)$) qui donne la dégradation observée.

Les deux spectres de puissance $P_B(u,v)$ et $P_I(u,v)$ seront estimés à partir d'une image de référence $i_R(x,y)$ contenue dans '*imageRef.png*', qui présente des caractéristiques statistiques vraisemblablement proches de celles de notre image idéale. $P_I(u,v)$ est approximé par le spectre de puissance de $i_R(x,y)$. $P_B(u,v)$ est approximé par le spectre de puissance du bruit $b(x,y)$ de quantification qui apparaît lors du passage de $d_R(x,y)$ à $d_Q(x,y)$ qui n'est codée que sur des nombres entiers. Ainsi :

$d_Q(x,y)=\mathrm{int}(\ d_R(x,y)\ )$

$d_R(x,y)=i_R(x,y)$ © $h(x,y)$ et $D_R(u,v)=I_R(u,v).H(u,v)$

$d_Q(x,y)=i_R(x,y)$ © $h(x,y)+b(x,y)$ et $D_Q(u,v)=I_R(u,v).H(u,v)+B(u,v)$

avec © correspondant au produit de convolution.

Questions

- Construisez et affichez les images correspondant aux représentations fréquentielles du filtre et des différentes images. On réalisera un recadrage sur une échelle logarithmique pour l'affichage des amplitudes en fréquence.
- Programmez la chaine de traitements à réaliser pour restaurer l'image par filtrage de Wiener. Il faudra pour cela trouvez le modèle du filtre de dégradation sachant qu'il correspond à une translation verticale. La réponse impulsionnelle correspond ici à un rectangle (largeur 3, hauteur à déterminer) centré à l'origine, d'intégrale 1, le reste de l'image étant nul.
- Comparez le résultat obtenu par la méthode de Wiener avec ce que l'on obtiendrait avec un filtrage « simpliste ».
- Question subsidiaire : réalisez la convolution spatiale entre une image et un filtre défini par une réponse impulsionnelle égale à un carré de 3x3 pixels de valeur 1/9. Quel est l'effet de ce filtre sur l'image ?

Rappels

Ouverture de l'image 'imageFloue.png' :	[im,map]=imread('imageFloue.png');
Construction d'une palette 'Niveau de Gris'	gris=([0:255]/255)'*[1 1 1] ;
Affichage de l'image :	figure(1) image(im) colormap(gris)
Conversion en réels pour les calculs	im=double(im) ;
Fast Fourier Transform 2D :	IM=fft2(im) ;
Permutation des cadrans :	IM=fftshift(IM) ;
Calcul du spectre d'amplitude recadré	logIM=log(abs(IM)+1) ; maxi=max(max(logIM)) ; mini=min(min(logIM)) ;
Affichage	figure(2) image((logIM-mini)/(maxi-mini)*255) colormap(gris)

Attention : En Matlab, le produit simple terme à terme entre tableau est : .*
Le produit * est un produit matriciel