

TP TRAITEMENT DE SIGNAUX PRINTEMPS 2025

TP 3 : Filtre de Wiener et débruitage

12 mars 2025, 14h15-17h

Débruitage d'un signal 1D

Nous souhaitons débruiter un signal 1D à l'aide du filtre de Wiener et montrer que contrairement à un filtre gaussien classique, le filtre de Wiener est adaptatif.

1. Les signaux à utiliser peuvent être générés grâce à la fonction `load_signal` disponible sur Moodle. Créer un signal régulier par morceaux `f=load_signal('piece-regular')`. Créer un bruit blanc de déviation-standard:

```
sigma = 0.2 * (max(f)-min(f));
```

(vous pouvez utiliser la fonction `randn`). Superposez la densité Spectrale de Puissance du signal f ainsi que celle du bruit. Ajoutez le bruit au signal pour obtenir le signal bruité fb .

2. *Débruitage par un filtre gaussien.* À l'aide des commandes matlab `build_gaussian_filter` et `perform_convolution` que vous pouvez télécharger sur Moodle, créez un filtre gaussien (de taille $n=101$, $\sigma=3.0$, $N=0.5$) et débruitez le signal fb . Étudiez l'influence de l'écart-type du filtre en calculant l'erreur quadratique entre le signal estimé fbh et le signal original f :

```
e = sum( (f(:)-fbh(:)).^2 );
```

Trouvez l'écart-type optimal pour filtrer le signal bruité. Visualisez la réponse fréquentielle du filtre.

3. *Débruitage par un filtre de Wiener.* Trouvez la réponse en fréquence du filtre de Wiener pour débruiter le signal fb et visualisez-la. Appliquez les filtres dans le domaine fréquentiel, déterminez le signal débruité dans le domaine temporel et calculez l'erreur quadratique entre le signal estimé et le signal original. Comparez les résultats avec le filtre gaussien et commentez les avantages et les limitations du filtrage de Wiener.
4. Répétez les étapes précédentes pour un signal `f=load_signal('gaussiannoise')`.

Restauration d'une image 2D

1. Téléchargez l'image `car.jpg` disponible sur Moodle. La lire avec la commande matlab `imread` et la convertir au format `double` avec la commande `im2double`.
2. *Convolution de l'image par un motion kernel.* Avec `fspecial`, créer un motion kernel horizontal de 35 pixels. La commande `[I, OST] = convolve_image(image,kernel)` disponible sur Moodle peut ensuite être utilisée pour filtrer l'image à l'aide de ce kernel. Visualiser l'image floutée avec la commande `imshow`.
3. *Filtre inverse dans le domaine fréquentiel.* Exprimer la déconvolution d'une image dans le domaine fréquentiel en fonction d'un kernel. La commande `convolve_image` que vous avez utilisé ci-dessus retourne la FFT du kernel. Utilisez la pour reconstituer l'image originale dans le domaine fréquentiel puis visualiser le résultat dans le domaine spatial. La commande `fftn` permet de calculer la FFT d'une image RGB et la commande `ifftn` la FFT inverse.

TP TRAITEMENT DE SIGNAUX
PRINTEMPS 2020

4. *Ajout d'un bruit blanc gaussien.* Ajoutez à l'image floutée un bruit blanc gaussien centré en zéro et de variance 1e-4 en utilisant la commande `imnoise`. Visualiser l'image bruitée. Déconvoluer l'image bruitée par filtrage inverse et visualiser le résultat. Quel est le problème ?
5. *Déconvolution de l'image par un filtre de Wiener.* Le filtre de Wiener peut être utilisé pour restaurer une image dans le cas où le motion kernel et le bruit sont connus. Utilisez la commande `deconvwnr` pour restaurer l'image à l'aide du ratio $nsr = \text{variance(bruit)}/\text{variance(signal)}$. Visualiser l'image restaurée. Que se passe-t-il si l'on utilise par erreur un nsr 10 fois plus grand ou 10 fois plus petit ? Quel facteur limite donc l'utilisation d'un filtre de Wiener ?