Sujet d'examen de TP de TNS

Epreuve sur ordinateur sous Matlab. Durée 1h. Les seuls documents autorisés sont l'aide de Matlab avec `help` et les documents suivants disponible en ligne :

`http://www-l2ti.univ-paris13.fr/~dauphin/polyMatlabCpl.pdf`
`http://www-l2ti.univ-paris13.fr/~dauphin/poly.pdf`
`http://www-l2ti.univ-paris13.fr/~dauphin/tp_tns_m1ir.pdf`
`http://www-l2ti.univ-paris13.fr/~dauphin/aide_listing.txt`

Les réponses doivent être conformes aux définitions du traitement du signal vues en cours et ce même si Matlab utilise une convention différente. **Pour chaque exercice, il vous faut donner la réponse aux questions ainsi que le programme ayant permis de trouver ces réponses ou la façon dont vous les avez obtenues**.

Prénom Nom

Exercice 1 *On considère un bruit blanc gaussien noté b_n d'écart-type 2 et de durée 10s. La fréquence d'échantillonnage de ce signal est de 2.5kHz.*

On applique à ce signal b_n un filtre de fonction de transfert

$$H(z) = \frac{1}{1+\frac{1}{2}z^{-1}}$$

La sortie est notée y_n.

Calculez avec 4 chiffres significatifs la moyenne des termes $y_n y_{n-2}$, ainsi définie :

$$m = \frac{1}{N-3}\sum_{n=2}^{N-1} y_n y_{n-2}$$

où N est le nombre d'échantillons du signal.

$m =$	*Commandes essentielles*

Solution :

```
res=[];
for k=1:1e3
 t=0:1/fe:(10-1/fe);
bn=2*randn(1,length(t));
A=[1 +1/2]; B=1;
yn=filter(B,A,bn);
y1n=yn(1:end-2);
y2n=yn(3:end);
m=mean(y1n.*y2n);
res=[res m];
end
min(res), max(res),
 m%    0.8241    2.0242
```

Exercice 2 *On considère le signal $x(t) = \sin(7t)\mathbf{1}_{[0,4[}(t)$. On note $x_q(t)$ le signal quantifié sur 8 bits. Calculez l'erreur quadratique moyenne entre $x(t)$ et $x_q(t)$, définie par $EQ = \sqrt{\frac{1}{t_{\max}-t_{\min}}\int_{t=t_{\min}}^{t_{\max}}(x(t)-x_q(t))^2\,dt}$. La fréquence d'échantillonnage est :* 5kHz.

	Commandes essentielles
EQ=	

solution :

```
fe=5e3;
t=0:1/fe:(4-1/fe);
x=sin(7*t);
a=min(x); b=max(x);
xq=floor((x-a)/(b-a)*2^8)/2^8*(b-a)+a;
figure(1); plot(t,x,'b',t,xq,'r');
figure(2); plot(t,x-xq);
sqrt(mean((x-xq).^2)), %       0.0045
```

Exercice 3 *On considère un signal $x(t)$ défini par*

$$x(t) = \sin(t^2+1)\mathbf{1}_{[0,5[}(t)$$

On note x_n le signal échantillonné à partir de $x(t)$ à la fréquence de $f_e = 500Hz$. On ne considère que les échantillons entre 0 et $5s$. Ce signal est bruité par un bruit blanc additif d'écart-type 1 et noté y_n. On souhaite retrouver une approximation de x_n en appliquant un filtre linéaire temps invariant sur y_n. On souhaite que l'approximation soit la meilleure possible, que proposez-vous comme type de filtre (passe-bas, passe-haut, passe-bande, coupe-bande) et quelles fréquence de coupure ? On suppose ici qu'il n'y a qu'une ou deux fréquencess de coupure. (Indication : représentez le spectre du signal et du bruit, le filtre souhaité doit attenuer le fortement les fréquences où le bruit est plus important que le signal et doit au contraire laisser passer les fréquences où le signal est plus important que le bruit).

Type de filtre :	*Commandes essentielles*
$f_c =$	

Solution :

```
fe=500;
t=0:1/fe:(5-1/fe);
x=sin(t.^2+1);
y=x+randn(size(x));
N=length(x);
ech_f=-fe/2:fe/N:fe/2-fe/N;
Sx=(abs(fftshift(fft(x)))/N).^2;
dB=@(x)10*log10(x);
Sb=(abs(fftshift(fft(y-x)))/N).^2;
figure(1); plot(ech_f,dB(Sx),ech_f,dB(Sb),'g-',...
  ech_f,mean(dB(Sb))-2*std(dB(Sb)),'g-',ech_f,mean(dB(Sb))+2*std(dB(Sb)),'g-');
%fc 3.5  1.6 Hz.
%passe-bas
```

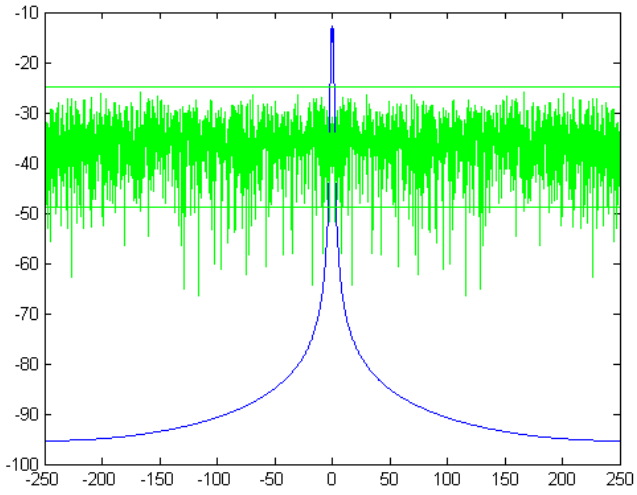

Figure 1: Exercice 3

A Instructions matlab pouvant être utilisées

help, format, :,.*,./,-,+,^,==, sum, mean, zeros, end, length, filter, fir1, triang, window, freqs, freqz, randn, butter, cos, sin, std, abs, fft, min, exp, hamming, window, floor, repmat, fftshift.

Pour la fonction `freqz`, la fonction s'utilise ainsi `[H,F]=freqz(B,A,1000,fe);` les deux premiers arguments à utiliser sont les coefficients du polynômes de variable p mais ordonnés dans le sens des puissances décroissantes. Le troisième argument est le nombre de points et le quatrième argument est la fréquence d'échantillonnage. Matlab ne normalise pas la fonction `fft`, ni la fonction `xcorr` alors que pour un signal temps discret périodique, le calcul est normalisé en traitement numérique du signal. Le sur-échantillonnage d'un facteur M consiste à introduire $M-1$ zéros après chaque échantillon puis à filtrer le résultat obtenu. Le sous-échantillonnage d'un facteur M consiste à filtrer le signal puis à conserver le premier échantillon correspondant à un indice nul puis le $M+1^{\text{ème}}$ échantillon d'indice $\frac{M}{f_e}$ etc... Dans les deux cas, le filtrage introduit un certain retard.