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

On considère une fréquence d'échantillonnage $f_e = 1$kHz, un signal s_n et un bruit b_n synthétisé par les commandes suivantes :

```
n=1:1000;
rand_=sin(sqrt(2)*n.^2);
sn=filter(1/(1-0.2)*[1 -0.5],[1 0.2],cumsum(2*rand_+sin(6*pi*(0:999)/1000)));
sn=sn-0.99*mean(sn); sn=sn/std(sn);
bn=rand_; bn=bn-0.99*mean(bn); bn=bn/std(bn);
```

On pourra vérifier le bon fonctionnement du programme en vérifiant les résultats suivants

```
>>sn(1:3)
ans =    -1.7096  -1.8587  -1.7739
>>bn(1:3)
ans=    1.4100  -0.8601   0.2173
```

Pour chacun des exercices suivants, vous indiquerez sur la feuille *le programme Matlab*, le résultat trouvé et lorsque c'est demandé les graphes correspondant.

Exercice 1 *Réalisez une quantification linéaire du signal* b_n *sur* 13 *niveaux et donnez la valeur du signal quantifié en* $t = 13$ms. *On supposera ici que les valeurs de* b_n *sont contenues dans l'intervalle* $[-1.5, 1.5]$.

```
bnQ=floor(13*(bn+1.5)/(3))/13*(3)-1.5;
bnQ(14)=  0.8077
bnQ2=ceil(13*(bn+1.5)/(3))/13*(3)-fi1.5;
bnQ2(14)=  1.0385
```

Suivant la méthode de discrétisation utilisée, on devrait avoir une valeur entre la première et la deuxième.

Valeurs trouvées	Commandes essentielles

Exercice 2 *On considère un filtre dont la relation entrée-sortie est définie par*

$$y_n = 0.464481y_{n-1} - 0.127961y_{n-2} - 0.024208y_{n-3} + 0.014468y_{n-4}$$
$$-0.114182x_n - 0.112714x_{n-1} - 0.357857x_{n-2} + 0.168156x_{n-3} - 0.057753x_{n-4}$$

où x_n et y_n sont respectivement l'entrée et la sortie. Trouvez une entrée notée x_n nulle pour $n > 3$ (et pour $n < 0$) telle que la sortie notée y_n vérifie

$$\begin{cases} 0.9 \leq y_n \leq 1.1 & \textit{quand} \quad n \in \{3, 4\} \\ |y_n| \leq 0.1 & \textit{quand} \quad n > 4 \end{cases}$$

L'idée consiste à tirer aléatoirement des entrées jusqu'à trouver une entrée dont la sortie vérifie les conditions souhaitées. Vous pouvez par exemple compléter le programme suivant :

```
A=...
B=...
while(1)
  xn=[randn(1,3) zeros(1,40)];
  yn=filter...
  ...
  if ... break; end
end
```

Donnez les valeurs de x_0, x_1, x_2 trouvées.

Fabrication de l'énoncé

```
Jmax=Inf;
B_=randn(1,5); A_=[1, randn(1,4)];
xn_=[randn(1,3) zeros(1,40)];
while(1)
  if rand(1)<0.5 B=B_; else B=B_+0.1*randn(1,5); end
if rand(1)<0.5 A=A_; else A=A_+0.1*[1, randn(1,4)]; end
if rand(1)<0.5 xn=xn_; else xn=xn_+0.1*[randn(1,3) zeros(1,40)]; end
yn=filter(B,A,xn);
J1=max(abs(yn(4:5)-1)); J2=max(abs(yn(6:end)));
J=100*J1*(J1>=0.01)+J2;
if J<Jmax A_=A; B_=B; xn_=xn; Jmax=J; disp([num2str(J1),' ',num2str(J2)]), end
if J1<0.01 && J2<0.01 break; end
end
A=[7.7000  -3.5765   0.9853   0.1864  -0.1114]
B =[  -0.8792  -0.8679  -2.7555   1.2948  -0.4447];
```

Solution

```
A= [1.000000  -0.464481   0.127961   0.024208  -0.014468];
B =[ -0.114182  -0.112714  -0.357857   0.168156  -0.057753];
while(1)
  xn=[randn(1,3) zeros(1,40)];
yn=filter(B,A,xn);
if !(max(abs(yn(4:5)-1))<0.1) continue; end
if !(max(abs(yn(6:end)))<0.1) continue; end
  break;
end
```

Valeurs trouvées	Commandes essentielles

Exercice 3 *On cherche à synthétiser un filtre coupe-bande de réponse impulsionnelle infinie en utilisant les filtres de Butterworth. On se débrouillera pour qu'il y ait* 11 *termes au dénominateurs. Les fréquences de coupures sont* $f_{c1} = 95\text{Hz}$ *et* $f_{c2} = 165\text{Hz}$*. Donnez la fonction de transfert du filtre. Représentez la réponse fréquentielle sur l'intervalle* $[0, 500\text{Hz}]$ **sans** *utiliser l'échelle logarithmique. Recopiez cette densité spectrale sur la copie en faisant attention à l'échelle en fréquence. Combien vaut le module de la réponse fréquentielle en* $f = 170\text{Hz}$ *?*

```
fc1=95; fc2=165; fe=1000;
[B,A]=butter(5,[2*fc1/fe 2*fc2/fe],'stop');
B=[0.487311      -3.4182      12.0272     -27.1273      43.0826 ...
 -50.0662      43.0826     -27.1273      12.0272      -3.4182     0.487311];
A=[  1     -6.01826      18.1417     -35.1463      48.0929     -48.3222 ...
      36.0725     -19.7681      7.64946     -1.90228     0.237472];
 [H,F]=freqz(B,A,300,1000);
 figure(1); plot(F,abs(H));
[val,ind]=min(abs(F-170));
F(ind), %  170
abs(H(ind)), %  0.87
```

Valeurs trouvées	Commandes essentielles

A Instructions matlab pouvant être utilisées

help, format, :,.*,./,-,+,ˆ,==, 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.