

TP n° 4 : Filtrage et Modélisation des signaux aléatoires (Sous Python)

Ce TP a pour objectif :

- Aborder une application directe qui est le filtrage adapté pour la détection d'un signal connu noyé dans du bruit blanc en augmentant le SNR.
- Modéliser un signal aléatoire par les modèles AR et MA (par approximation AR) et d'en extraire les informations utiles. Puis, une application d'identification sera envisagée.

Exercice 1

On veut simuler l'émission et la réception d'un signal radar. On suppose que le signal émis $s(n)$ est un signal sinusoïdal et que le signal reçu $x(n)$ est une version bruitée (bruit blanc) atténuée et retardée du signal émis.

```
import numpy as np; import matplotlib.pyplot as plt; import time; import winsound;
L=300; f0=0.02; N=1000; Te=1; Tar=500;
t = np.linspace(0, L-1, L)*Te; s=np.cos(2*np.pi*f0*t)
var=4; bb = (var**0.5)*np.random.randn(N);
x=bb; x[Tar:Tar+L]=x[Tar:Tar+L]+s;
h=s[::-1] ;y=np.convolve(x,h);
plt.figure(1);plt.subplot(311); plt.plot(s); plt.grid(True); plt.title('signal Emis');
plt.subplot(312); plt.plot(x); plt.grid(True);plt.title('signal Reçu Bruité');
plt.subplot(313); plt.plot(y); plt.grid(True);plt.title('signal Reçu Filtré');
TAR_estim=np.argmax(y)-L+1;
TARR=np.where(y==y.max())
```

1. Montrer que pour $y(t) = a.x(t - T_{AR}) + b(t)$ où $b(t)$ est un bb , on obtient lors d'un filtrage adapté ($h(t) = x^*(T_0 - t)$) en sortie l'expression suivante $R_{yx}(t - T_0) = a.R_{xx}(t - T_0 - T_{AR}) + R_{bx}(t - T_0)$.
2. Que représentent A , s et x ?
3. Arrive-t-on à identifier la partie comportant la sinusoïde dans le signal reçu? pourquoi?
4. Expliciter l'instruction $h=s[::-1]$. Quel est, alors, le lien entre x et h ?
5. Expliquer l'instruction $TAR_estim=np.argmax(y)-L+1$;
6. Commenter le programme plus particulièrement les figures. Quel est le but de ce programme ?
7. Tester d'autres signaux utiles; par ex : $s = np.ones(L)$ et commenter. Pour détecter le temps d'aller-retour, lequel des 2 signaux vous paraît préférable?
8. Faire varier la puissance du bruit ($var=2$ puis 4 en gardant une $moy=0$) et commenter.
9. Simuler le processus de l'aide au stationnement en mettant tout le programme dans une boucle pour un Retard (Tar) de 700 à 0 avec un pas de -50 et rajouter à la fin les instructions :
`time.sleep(TAR_estim/500); winsound.Beep(500, 100)` dans la boucle.
 Mettre les figures en commentaires.

Exercice 2: Télécharger le fichier 'vous avez du courrier en attente.wav' et le placer dans le même répertoire que votre programme commençant comme suit :

Reprendre le programme de l'exo 3 dut TP n°3 (Chargement de la phrase Vous avez du courrier en attente et sélection d'une partie de la phrase)

I. Nous allons commencer par déterminer les paramètres du modèle soit l'étape de modélisation

```
import scipy.linalg as la; import scipy.signal as sp;
""" Modélisation AR """
P=20;          "P Nbre de coefficients du modèle AR>2"
y=y-np.mean(y)
R = np.correlate(y,y,mode='full')[len(y)-1:];
#R=R/R.max(); "Autocor"
C = la.toeplitz(R[0:P]);
B = -R[1:P+1];
a = la.inv(C).dot(B)
a = np.append(1,a)
sigma_carre = sum(a*R[0:P+1]);

""" Visualisation de la réponse fréquentielle du filtre trouvé et du signal y de départ """
b = np.array([1,0]);
f,H= sp.freqz(b,a,L//2); f=f*0.5*fe/np.pi;
TFy = np.fft.fft(y); TFy = TFy[1:L//2+1];
plt.figure(2); plt.subplot(121);plt.plot(f, 20*np.log10(abs(TFy/TFy.max())));
plt.plot(f, 20*np.log10(abs(H/H.max())));
plt.title('Module du Filtre trouvé (orange)+ Signal original modélisé(bleu)');
plt.xlabel('Fréq (Hz)');plt.ylabel('Amp');
```

II. Entamons la partie synthèse en ayant comme données de départ les paramètres du modèle (a_i et sigma). Donc à partir de σ² et des coefficients AR 'a' trouvés, synthétisons et écoutons le son 'zz' .

```
""" Synthèse à partir du filtre trouvé """
bb = (sigma_carre*0.5)*np.random.randn(L); bb=bb-np.mean(bb);
y_synt = sp.lfilter(b,a,bb);
```

1. L'autocorrélation R_{yy} intervenant dans les équations de Yule-Walker est-elle statistique ou temporelle ? Qu'en est-il de celle calculée dans ce programme? Commenter.
2. Visualiser les spectres du son synthétisé t du filtre en db sur le même graphe.
3. Modifier P et commenter les graphes précédents.

III. Rajouter les lignes suivantes et donner une application possible.

```
""" Détermination des fréquences principales """
from math import atan2
freq_form=[];
r=np.roots(a);
for i in r:
    freq_form=np.append(freq_form,atan2(np.imag(i),np.real(i))*fe/(2*np.pi))
freq_form=sorted(freq_form)
```

4. Pourquoi utilise-t-on le modèle AR pour trouver les fréquences principales au lieu du signal 'y'?

IV. Sachant qu'un modèle MA peut être obtenu en identifiant le modèle MA d'ordre M avec un modèle AR d'ordre P>>M :

$$\sum_{i=0}^M b_i z^{-i} = 1 / \sum_{i=0}^{\infty} a_i z^{-i}$$

5. Modifier le programme précédent en rajoutant l'instruction suivante au bon endroit : Dirac=np.zeros(10*P); Dirac[0]=1; bMA = sp.lfilter(b,a,Dirac); aMA = np.array([1,0]); y_synt = sp.lfilter(bMA,aMA,bb);
6. Expliquer comment on s'est servi de ces instructions pour obtenir les b_i.
7. Faire la modification suivante son_synt= filter(b,1,noise); et expliquer pourquoi
8. Observer les spectres de la partie modélisée et synthétisé et commenter.