ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΟΣ

Αποθορυβοποίηση με φίλτρα WIENER

Κωνσταντίνος Ρούμογλου

April 30, 2024

1. ΕΡΩΤΗΜΑ Α

I. ADD NOISE

Αρχικά φορτώνουμε το σήμα μας (y) με τη βοήθεια της εντολής audioread. Στη συνέχεια δημιουργούμε λευκό γκαουσιανό θόρυβο ($noise$) με τυπική απόκλιση 0.01 με τη βοήθεια της συνάρτησης randn. 'Ετσι προσθέτουμε τον θόρυβο στο αρχικό μας σήμα και δημιουργείται το καινούριο σήμα με θόρυβο (x).

$$\boxed{\boldsymbol{noise = 0.01 * randn(N)}}\,, N = length(y)$$

$$\boxed{\boldsymbol{x = y + noise}}$$

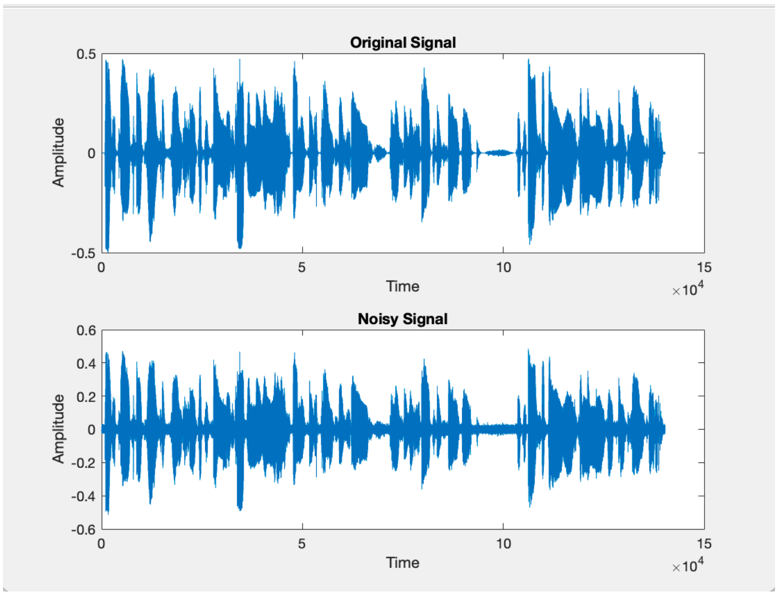

Figure 1: Signals

II. WIENER FILTER

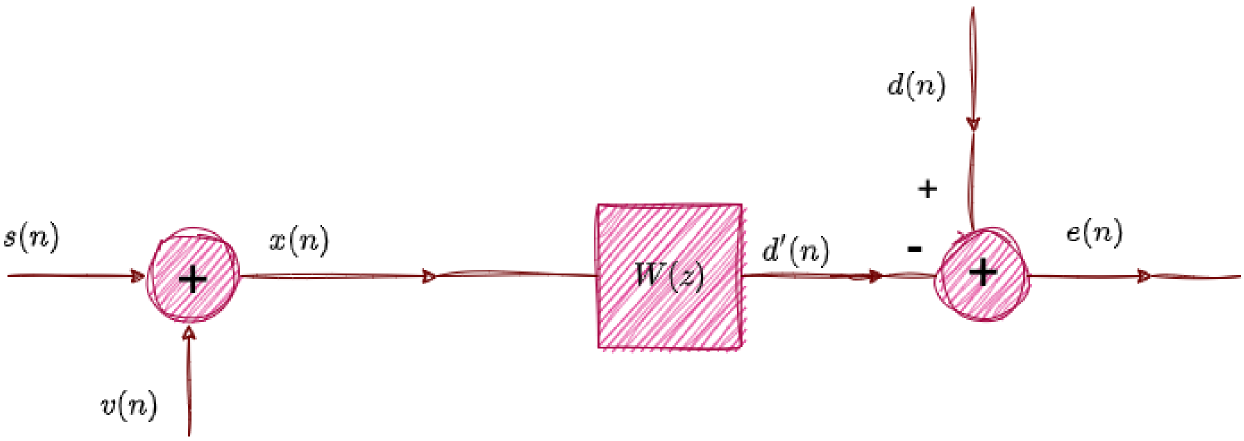

Figure 2: Wiener Filter

$$\hat{d}(n) = \sum_{k=0}^{p-1} w_k(n)x(k-n)$$

$$x(n) = d(n) - \hat{d}(n)$$

Υποθέτουμε ότι $x(n)$ και $d(n)$ είναι WSS και θέλουμε να υπολογίσουμε το $w(k),\ k = 0, 1, 2, ..., p-1.$ ελαχιστοποιώντας το (MSE)

$$\xi \ = min_{w(k)}\ E[|e(n)|^2] = min_{w(k)}\ E[|d(n) - \hat{d}(n)|^2] = \boxed{\frac{1}{n}\sum_{m=0}^{n} e^2[m]\ (1)}$$

$$min_{w(k)}\ \xi \ = min_{w(k)}\ E[|e(n)|^2] \Rightarrow \frac{\partial \xi}{\partial w^*(k)} = 0\ , for\ k = 0, 1, ..., p-1.\ (2)$$

Όμως το σφάλμα $e(n)$ ισούται με :

$$e(n) = d(n) - \hat{d}(n) = d(n) - x(n) * w(n) = \boxed{d(n) - \sum_{l=0}^{p-1} w(l)x(n-l)\ (3)}$$

$$(2) \Rightarrow \frac{\partial E\left\{e(n)e^{\star}(n)\right\}}{\partial w^{\star}(k)} = 0 \Rightarrow \boxed{E\left\{e(n)x^{\star}(n-k)\right\} = 0}, for\ k = 0, 1, ..., p-1.\ (4)$$

$$(4), (3) \Rightarrow E\left\{d(n)x^{\star}(n-k)\right\}\ - \sum_{l=0}^{p-1} w(l)E\left\{x(n-l)x^{\star}(n-k)\right\} = 0 \Rightarrow$$

$$\Rightarrow \boxed{\boldsymbol{r_{dx}(k) - \sum_{l=0}^{p-1} w(l) r_x(k-l) = 0}}\ , for\ k = 0, 1, ...p-1.$$

’Ετσι παίρνουμε το ακόλουθο σύστημα εξισώσεων:

$$\begin{bmatrix} r_x(0) & r_x(1) & \dots & r_x(p-1) \\ r_x(1) & r_x(0) & \dots & r_x(p-2) \\ \dots & \dots & \dots & \dots \\ r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} * \begin{bmatrix} w(0) \\ w(1) \\ \dots \\ w(p-1) \end{bmatrix} = \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \\ \dots \\ r_{dx}(p-1) \end{bmatrix}$$

$$\boxed{\boldsymbol{R_{xx} w = r_{dx}}}$$

Παρακάτω παρουσιάζεται η συνάρτηση υλοποίησης του φίλτρου, με τάξεις 10,20 και 30.

```
function [fin_signal]=my_wiener(x,d,order)

  len=length(x);
  rxx=xcorr(x); % autocorrection for x(y with noise)
  rxd=xcorr(x,d);
  rxx=rxx(len:len+order-1);
  rxd=rxd(len:len+order-1);
  Rxx = toeplitz(rxx);
  w=inv(Rxx)*rxd';

  fin_signal=filter(w,1,x);

end
```

III. SNR

Σε αυτό το ερώτημα μας ζητείται να υπολογίσουμε το SNR για το σήμα με θόρυβο και για τα άλλα 3 αποθορυβοποιημένα σήματα.

Πιο αναλυτικά το Signal-to-Noise-Ratio είναι ένα μέτρο που χρησιμοποιείται στην επεξεργασία σήματος για την ποσοτικοποίηση του επιπέδου ενός επιθυμητού σήματος σε σχέση με το επίπεδο του θορύβου υποβάθρου. Εκφράζεται συνήθως σε ντεσιμπέλ (dB) και υπολογίζεται με τον ακόλουθο τύπο:

$$\boldsymbol{SNR(a,b) = 10 * \log_{10}(\frac{Power(a)}{Power(b)})}$$

$$\boldsymbol{Power = \frac{1}{N}\sum_{i=1}^{N} x(i)^2}$$

Είναι σημαντικό να τονίσουμε πως το αρχικό μας σήμα χωρίς θόρυβο θα έχει υψηλό SNR.Έτσι όσο αυξάνεται ο θόρυβος τόσο μειώνεται το SNR και αντίστοιχα όσο μειώνεται αυξάνεται το SNR.

Τα αποτελέσματα παρουσιάζονται παρακάτω :

- noisy $SNR(y, noise) = 19.3939\ dB$
- order=10: $SNR(z_1, z_1 - y) = 19.7246\ dB$
- order=20: $SNR(z_2, z_2 - y) = 19.7328\ dB$
- order=30: $SNR(z_3, z_3 - y) = 19.7387\ dB$

Για το αποθορυβοποιημένο σήμα αν αφαιρέσουμε το αρχικό θα μας δώσει τον θόρυβο.Τα αποτελέσματα είναι λογικά αλλά όχι και τόσα καλά, δηλαδή παραρηρούμε μικρή βελτίωση. Σε συνδυασμό με τα ακουστικά αποτελέσματα που παίρνουμε συμπεραίνουμε πως το σήμα δεν αποθορυβοποιήθηκε σωστά.

2. ΕΡΩΤΗΜΑ Β

Αφού παρατηρήσαμε πως η αποθορυβοποίηση ολόκληρου του σήματος απέτυχε, σε αυτό το ερώτημα θα εφαρμόσουμε μία διαφορετική τεχνική.

Έτσι θα χωρίσουμε το ενθόρυβο σήμα μας σε παράθυρα μήκους 256 δειγμάτων με overlap=0.5, με τη βοήθεια της συνάρτησης *"frame wind.m "* .

Στη συνέχεια θα εφαρμόσουμε το **Wiener** φίλτρο σε κάθε ένα παράθυρο και τέλος θα επανασυνθέσουμε το φιλτραρισμένο σήμα με τη βοήθεια της *"frame recon.m"*.

Παρακάτω παρουσιάζονται οι δύο συναρτήσεις.

```
function X=frame_wind(x,frame, ovrlp)

  M=length(x);
  K=floor(M/(frame*ovrlp));
  X=zeros(frame,K);
  W=hann(frame);
  for i=1:K-floor(1./ovrlp-1)
    indx=(i-1)*floor(frame*ovrlp);
    X(:,i)=x(indx+1:indx+frame).*W;
  end
  end
```

```
function y=frame_recon(X, ovrlp)

  [frame K]=size(X);
  Mmax=K+floor(1./ovrlp-1);
  y=zeros(1,frame*Mmax*ovrlp);
  for i=1:K
  indx=(i-1)*floor(frame*ovrlp);
  y(indx+1:indx+frame)=y(indx+1:indx+frame)+X(:,i)
    end
    end
```

Η παραπάνω διαδικάσία υλοποιείται από τη συνάρτηση "$wienerframes.m$" . Αφού επανασυνθέσουμε το σήμα ακούμε τα 3 διαφορετικά αποτελέσματα για κάθε τάξη του φίλτρου και υπολογίζουμε και το SNR. Κατανοούμε πως τα αποτελέσματα είναι πολύ καλύτερα.

- order=10: $SNR(z_1, z_1 - y) = 21.56\ dB$
- order=20: $SNR(z_2, z_2 - y) = 21.82\ dB$
- order=30: $SNR(z_3, z_3 - y) = 22.02\ dB$

3. ΕΡΩΤΗΜΑ C

Γνωρίζουμε πως ο λευκός θόρυβος δεν έχει συσχέτιση μεταξύ των δειγμάτων του ,εκτός από το σημείο 0.

Με άλλα λόγια αρκεί να υπολογίσουμε την αυτοσυσχέτιση του θορύβου, η οποία θα πρέπει να είναι μηδέν εκτός από ένα μόνον σημείο.

Παρακάτω παρουσιάζεται η γραφική παράσταση της αυτοσυσχέτισης του λευκού γκαουσιανού θορύβου.

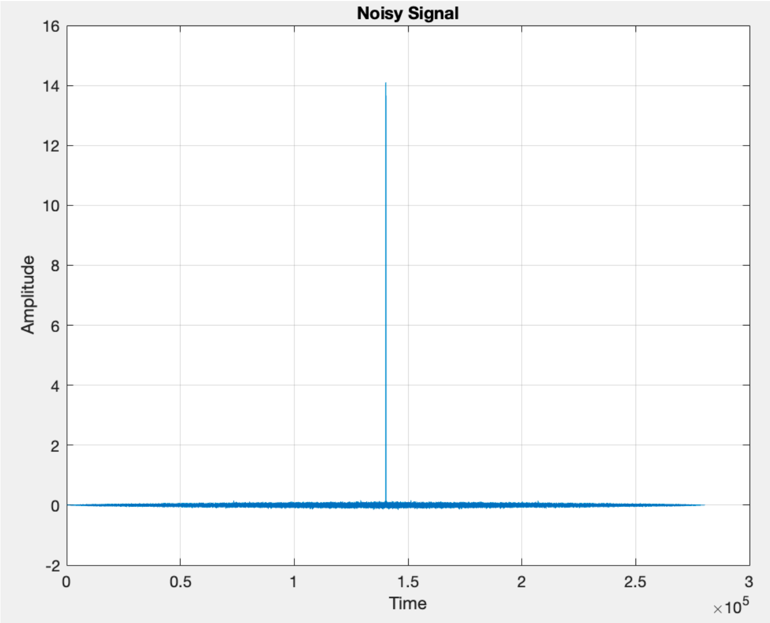

Figure 3: Noise Correlation

4. ΕΡΩΤΗΜΑ D

Θα επαναλάβουμε το ερώτημα a) χωρίς να γνωρίζουμε το αρχικό σήμα και κατ' επέκταση την αυτοσυσχέτιση του. Έτσι θα εντοπίσουμε ένα τμήμα του ενθόρυβου σήματος που έχει ησυχία,δηλαδή μόνο θόρυβο, και από εκεί θα υπολογίσουμε την αυτοσυσχέτιση του θορύβου.Σαφώς το τμήμα αυτό θα είναι μικρότερου μήκους από το αρχικό σήμα, όμως αυτό που μας ενδιαφέρει είναι τα 10,20 ή 30 σημεία από το κέντρο, αναλόγως την τάξη του φίλτρου που θα υλοποιήσουμε.

Ο θόρυβος μας είναι λευκός γκαουσιανός και άρα ασυσχέτιστος.

$$\boldsymbol{noise(n) = v(n)}$$

$$\boldsymbol{r_{yx}(k) = E\left\{y(n)x^{\star}(n+k)\right\} = E\left\{y(n)y^{\star}(n+k)\right\} + E\left\{y(n)v^{\star}(n+k)\right\}\ (*)}$$

Όμως επειδή y, v ασυσχέτιστες : $\boldsymbol{E\left\{y(n)v^{\star}(n+k)\right\} = 0\ (*) \Rightarrow}$

$$\boxed{\boldsymbol{r_{yx}(k) = r_y(k)}}$$

$$\boldsymbol{r_x(k) = E\left\{x(n)x^{\star}(n+k)\right\} = E\left\{y(n)y^{\star}(n+k)\right\} + E\left\{v(n)v^{\star}(n+k)\right\} + E\left\{y(n)v^{\star}(n+k)\right\}}$$
$$\boldsymbol{+E\left\{v(n)y^{\star}(n+k)\right\} \Rightarrow}$$

$$\boxed{\boldsymbol{r_x(k) = r_y(k) + r_v(k)}}$$

Έτσι υλοποιούμε τη συνάρτηση "$find\ silence.m$" η οποία βρίσκει ένα σημείο με θόρυβο και καλούμε την "$neWiener.m$ η οποία υλοποιεί το φίλτρο **Wiener** με κάποιες διαφοροποιήσεις,χωρίς να γνωρίζουμε το αρχικό σήμα. Οι δύο συναρτήσεις παρουσιάζονται παρακάτω.

```
function [start_idx,end_idx]=find_silence(x,space)
N=length(x);
std_noise=0.01;
threshold=2*std_noise;
start_idx=0;
end_idx=0;

for i =1:(N-space)
    segment=x(i:i+space);
    if all(abs(segment)<threshold)

        start_idx=i;
        end_idx=i+space;
        break;
    end

end
end
```

```
function [final]=neWiener(x,noise,order)

  len=length(x);
  rxx=xcorr(x);
  rss=xcorr(noise);
  rss=rss(length(noise):length(noise)+order-1);

  rxx=rxx(len:len+order-1);
%   ryy=ryy(len:len+order-1);
  ryy=rxx-rss;

  Rxx = toeplitz(rxx);
  w=pinv(Rxx,0.0001)*ryy';
  final=filter(w,1,x);

end
```

Έτσι παίρνουμε τα ακόλουθα αποτελέσματα:

- order=10: $SNR(z_1, z_1 - y) = 21.56\ dB$
- order=20: $SNR(z_2, z_2 - y) = 21.82\ dB$
- order=30: $SNR(z_3, z_3 - y) = 22.02\ dB$

5. ΕΡΩΤΗΜΑ Ε

Σε αυτό το ερώτημα μας ζητείται να επαναλάβουμε το ερώτημα b χωρίς να γνωρίζουμε το αρχικό σήμα y και κατά συνέπεια την αυτοσυσχέτισή του.

’Ετσι θα σπάσουμε το σήμα σε 1095 παράθυρα μήκους 256 με τη βοήθεια της ”$frame\ wind$” .Στη συνέχεια θα ανατρέξουμε κάθε παράθυρο ή στήλη και θα βρούμε αν κάθε ένα από αυτά περιέχει φωνή ή θόρυβο με το μηχανισμό που υλοποιήσαμε στο προηγούμενο ερώτημα.

Αν το παράθυρο έχει θόρυβο τότε το χρησιμοποιούμε για να βρούμε την αυτοσυσχέτιση του θορύβου και κατ’ επέκταση του αρχικού σήματος.Χρησιμοποιούμε αυτήν την εκτίμηση ως σταθερό πρότυπο μέχρι να βρεθεί άλλο παράθυρο με θόρυβο.

Σε κάθε μια στήλη εφαρμόζουμε το φίλτρο μας με τη βοήθεια της συνάρτησης ”$neWiener$”.

Τέλος επανασυνθέτουμε το αποθορυβοποιημένο σήμα με τη συνάρτηση ”$frame\ recon$” και παίρνουμε τα ακόλουθα αποτελέσματα.

Όλη η παραπάνω διαδικασία υλοπιείται από τη συνάρτηση ”$neWiener\ frames$”.

Έτσι παίρνουμε τα ακόλουθα αποτελέσματα:

- order=10: $SNR(z_1, z_1 - y) = 21.1040\ dB$
- order=20: $SNR(z_2, z_2 - y) = 20.8816\ dB$
- order=30: $SNR(z_3, z_3 - y) = 20.5546\ dB$

6. ΕΡΩΤΗΜΑ F

Σε αυτό το ερώτημα μας ζητείται να εφαρμόσουμε το φίλτρο **Wiener** χρησιμοποιώντας τη δομή του ερωτήματος **b** ώστε να προβλέψουμε τις επόμενες 2,10 ,15 τιμές του σήματος εισόδου.

Έτσι δημιουργούμε τη συνάρτηση "$wiener\ predict$" η οποία προβλέπει τις επόμενες τιμές του σήματος μας ανάλογα με το όρισμα που θα δώσουμε.

```
function [fin_signal]=wiener_predict(d,order,pos)

len=length(d);
rdd=xcorr(d); % autocorrection for x(y with noise)
rxx=rdd(len:len+order-1);
rdd=rdd(len+pos:len+pos+order-1);

Rxx = toeplitz(rxx);
w=pinv(Rxx,0.0001)*rdd';
fin_signal=filter(w,1,d);
end
```

Η συνάρτηση "$wiener\ predict\ frames$" υλοποιεί το παραπάνω ερώτημα , δηλαδή τη δομή **b** και επιστρέφει τις τιμές που προβλέπουμε.