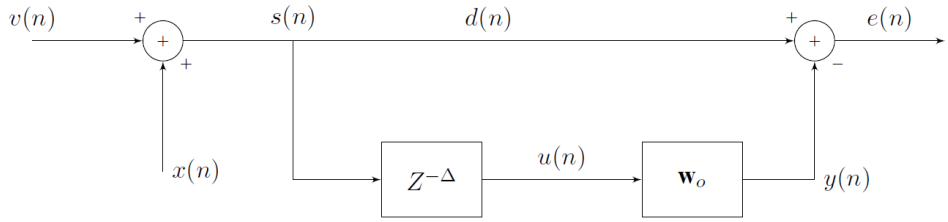
ΨΗΦΙΑΚΑ ΦΙΛΤΡΑ

3η Εργασία: Εξουδετέρωση Περιοδικής Παρεμβολής Χωρίς Σήμα Αναφοράς

Βαγενάς Θεόδωρος- Παναγιώτης

Α.Ε.Μ: 8112

Θεωρητική ανάλυση



$$x(n) = A\left(\sin\left(2\pi f_o n + \phi\right) + \cos\left(4\pi f_o n + \phi\right) + \cos\left(7\pi n + \frac{\phi}{3}\right)\right),$$
$$s(n) = x(n) + v(n),\ u(n) = s\left(n - \Delta\right), d(n) = s(n),\ e(n) = d(n) - y(n),$$
$$f_o = \frac{1}{4}, \phi = \frac{\pi}{2},\ A = 4.2,\ \Delta = 10$$

Το σήμα υ(n) προσομοιώνεται με λευκός θόρυβο μηδενικής μέσης τιμής και διακύμανσης $\sigma_v^2 = 0.54$

α)Το φίλτρο prediction error μπορεί να χρησιμοποιηθεί για την αποθορυβοποίηση μιας στάσιμης στοχαστικής διαδικασίας διακριτού χρόνου όταν χρησιμοποιηθεί φίλτρο με αρκετά μεγάλη τάξη. Αυτό γίνεται γιατί υπάρχει συσχετισμός μεταξύ γειτονικών δειγμάτων. Έτσι όσο αυξάνουμε την τάξη του φίλτρου τόσο μειώνουμε τη συσχέτιση μεταξύ των γειτονικών δειγμάτων της εισόδου μέχρι να πετύχουμε την τάξη που να παράγει μία διαδικασία εξόδου από ασυσχέτιστα δείγματα με αποτέλεσμα να παράγεται το καθαρό σήμα στην έξοδο του φίλτρου. Στη συγκεκριμένη άσκηση έχουμε περιοδική παρεμβολή την οποία πρέπει να αφαιρέσουμε για να καθαρίσουμε το σήμα. Επομένως στην έξοδο e =d-y αφαιρούμε από το επιθυμητό σήμα, το οποίο έχει δύο συνιστώσες την πληροφορία και την περιοδική παρεμβολή, το y το οποίο είναι σχεδόν ίδιο με το x(n.) Το σήμα y(n) προσεγγίζει το x(n) ενώ το σήμα e(n) προσεγγίζει το καθαρό από περιοδική παρεμβολή σήμα πληροφορίας υ(n). Αυτό φαίνεται και από το διάγραμμα e v τα οποία έχουν μικρή απόκλιση κατά την εκτέλεση του φίλτρου σε matlab με τον αλγόριθμο του Joint Process Estimator αλλά και μέσα από το φίλτρο wiener (όμως για Δ=8) τα οποία παρουσιάζονται παρακάτω και υλοποιούνται στα αρχεία matlab που επισυνάπτονται.

β) Στο αρχείο matlab γίνεται ο υπολογισμός των συντελεστών του φίλτρου wiener.

Χρησιμοποιήθηκε η σχέση: $w_0 = R^{-1}r$ όπου R είναι ο πίνακας toeplitz συμπεριλαμβανομένου και του r(0) και r το διάνυσμα αυτοσυσχέτισης χωρίς το r(0).

γ) Στο ίδιο αρχείο matlab με πριν εφαρμόζεται ο αλγόριθμος Joint Process Estimator. χρησιμοποιείται ο αλγόριθμος Levinson-Durbin και υπολογίζονται οι παράμετροι που ζητούνται. Έπειτα παρουσιάζονται τα σφάλματα των παραμέτρων αυτών από τις παραμέτρους που υπολογίζονται από την έτοιμη συνάρτηση της matlab (levinson()). Υπολογίζονται τα backward errors με τον τύπο $b(n) = L * u(n)$ όπου $u(n) =$

$[u(n), u(n-1), \dots, u(n-M)]^T$, $b(n) = [b_0(n), b_1(n), \dots, b_M(n),]^T$

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_{1,1} & 1 & \dots & 0 \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ a_{M,M} & a_{M,M-1} & \dots & 1 \end{bmatrix}$$

Επιπλέον υπολογίζονται οι optimal parameters για τον joint process estimator με τους τύπους $D = diag(P), p(i) = rb(lags == 0),\ g_0 = D \backslash p$

Τέλος το αποτέλεσμα επιβεβαιώνεται μέσω της σχέσης $w_0 = L^T g_0$.

Τα σφάλματα (με norm()) όπως προκύπτουν από την levinson συνάρτηση που χρησιμοποιήθηκε σε σχέση με την συνάρτηση της matlab levinson() :

Filter coefficient norm(a-mat_a): 7.559812e-13
G norm(G-mat_g): 7.326243e-13
Prediction Error power norm(Dp(end) - mat_p): 3.032297e-15
norm(L.'*go-wo):2.859435e-01
MSE L.'*go = wo :8.095417e-04

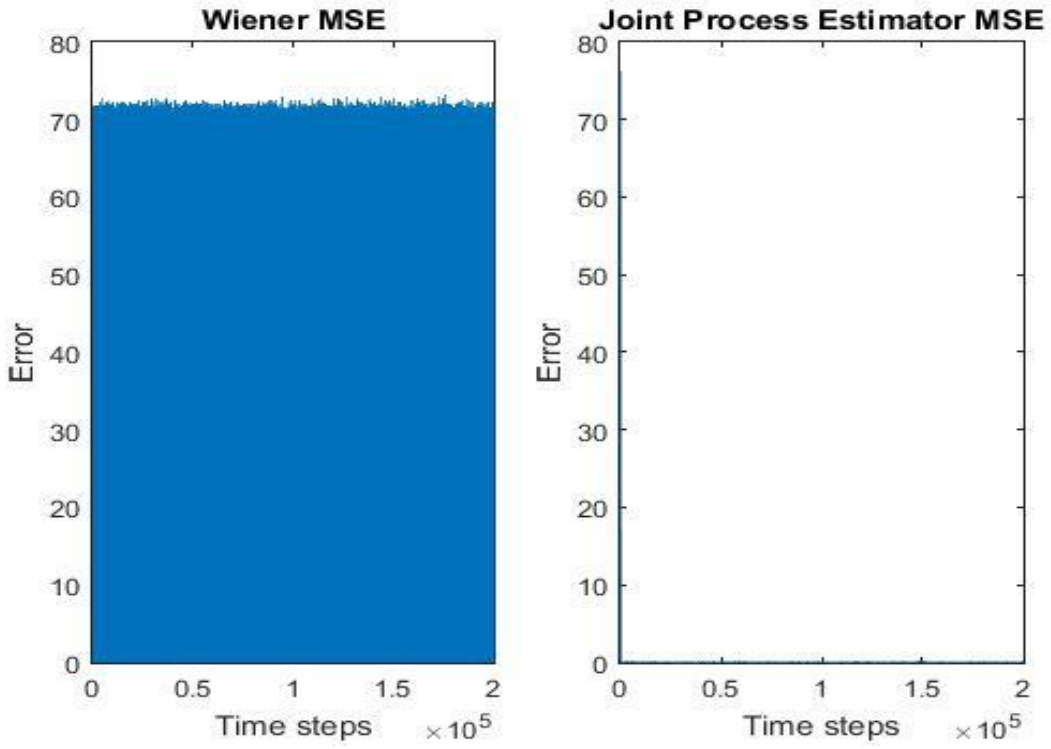
δ) Το κόστος του υπολογισμού των συντελεστών είναι μικρότερο στο φίλτρο του Joint Process Estimator. Αυτό συμβαίνει γιατί οι είσοδοι, δηλαδή τα backward prediction errors, είναι ορθογώνια μεταξύ τους και επομένως ο πίνακας αυτοσυσχέτισης τους που χρησιμοποιείται στη σχέση $g_0 = D \backslash p$ είναι διαγώνιος. Επομένως οι συντελεστές μπορούν να υπολογιστούν πολύ πιο γρήγορα. Επιπλέον μετρώντας τον χρόνο υπολογισμού των συντελεστών στη matlab προέκυψε ότι ο χρόνος υπολογισμού στον Joint Process Estimator ήταν 10 φορές μικρότερος για το μέγεθος του προβλήματος που δοκιμάστηκε.

ε) Για το κομμάτι αυτό σχεδιάστηκε και πάλι ένα φίλτρο Joint Process Estimator αλλά με κάποιες διαφορές προκείμενου να γίνει εφικτή η επεξεργασία του κομματιού ήχου. Το φίλτρο χρησιμοποιεί M=100 συντελεστές. Για τον υπολογισμό της ετεροσυσχέτισης μεταξύ του επιθυμητού σήματος s και των συντελεστών b προτιμήθηκε να υπολογίζεται ξεχωριστά κάθε συντελεστής επειδή ο συνολικός πίνακας των b ήταν πολύ μεγάλος n x (M+1) καταλαμβάνοντας μνήμη περίπου 3,9gb και προκαλώντας μεγάλη καθυστέρηση στην εκτέλεση του φίλτρου. Για να γίνει αυτό χρησιμοποιήθηκε η συνάρτηση της matlab filter(b,a,x) με `filter(L(i,1:i),1,u)` που είναι η Rational Transfer Function δηλαδή τη διαφορική εξίσωση $\alpha(1)y(n) = b(1)x(n) + b(2)x(n-1) + \dots + b(n_b+1)x(n-n_b) - \alpha(2)y(n-1) - \dots - \alpha(n_a+1)y(n-n_a)$.Έπειτα υπολογίζεται η έξοδος y και e και αναπαράγεται το τραγούδι το οποίο βρίσκεται στην έξοδο e του φίλτρου.
Όνομα τραγουδιού : Joshua Fit The Battle Of Jericho , Sidney Bechet

Επίσης υπολογίζονται οι έξοδοι και για τα 2 φίλτρα προκειμένου να βρεθεί το μέσο τετραγωνικό σφάλμα της εξόδου και του επιθυμητού σήματος για το καθένα και να παρουσιαστούν οι γραφικές παραστάσεις τους.

ΔΙΑΓΡΑΜΜΑΤΑ

Για Δ=10:



Για Δ=8: καθώς το φίλτρο wiener έδωσε στην έξοδο το καθαρό σήμα μόνο για αυτήν την τιμή.