

Ψηφιακά Φίλτρα

Εργασία 3 Εξουδετέρωση Περιοδικών Παρεμβολών Χωρίς Σήμα Αναφοράς & Εξουδετέρωση Ηχούς

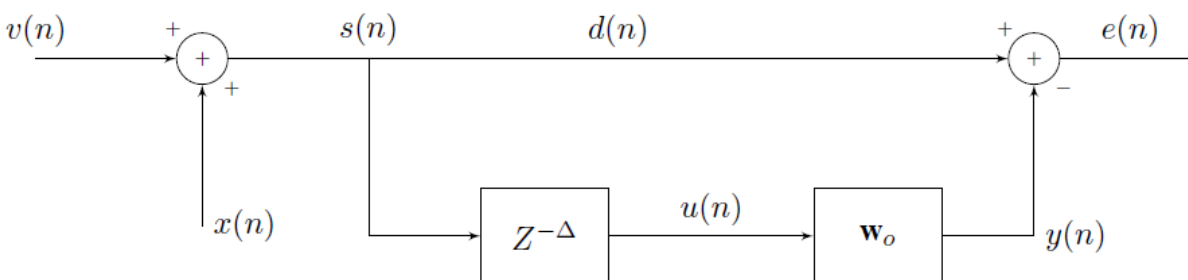
ΧΑΤΖΗΘΩΜΑ ΑΝΤΡΕΑΣ

AEM: 8026

antreasc@ece.auth.gr

Μέρος Α

Για το Μέρος Α, έχουμε την παρακάτω διάταξη:



Ισχύουν οι παρακάτω σχέσεις:

$$x(n) = A \left(\sin(2\pi f_o n + \phi) + \cos(4\pi f_o n + \phi) + \cos\left(7\pi n + \frac{\phi}{3}\right) \right) \quad \text{όπου } f_o = \frac{1}{4}, \quad \phi = \frac{\pi}{2}, \quad A = 2.3$$

$$s(n) = x(n) + v(n)$$

$$u(n) = s(n - \Delta)$$

$$d(n) = s(n)$$

$$e(n) = d(n) - y(n)$$

Όπου $v(n) = \text{λευκός θόρυβος με } \sigma_v^2 = 0.34 \text{ και } \Delta = 10$

Α) Η λειτουργία της Διάταξης

Αρχικά, έχουμε το σήμα εισόδου $x(n)$ στο οποίο προστίθεται ο θόρυβος $v(n)$. Ακολούθως, το νέο σήμα $s(n)$ περνάει από ένα φίλτρο Wiener, αφότου έχει προηγηθεί μια καθυστέρηση κατά Δ , και έτσι προκύπτει μια προσέγγιση του χρήσιμου σήματός, $y(n)$. Τέλος, η προσέγγιση $y(n)$ αφαιρείται από το desired $d(n)$ το οποίο ισούται με το σήμα $s(n)$ το οποίο με τη σειρά του προέκυψε από την άθροιση του χρήσιμου σήματος και του θορύβου. Έτσι, προκύπτει το σήμα $e(n)$ το οποίο είναι προσέγγιση του σήματος εισόδου $x(n)$.

Συνεπώς η περιοδική παρεμβολή εμφανίζεται στο $e(n)$ και το καθαρό σήμα στο $y(n)$.

Β) Υπολογισμός των R, p, w_o του φίλτρου Wiener για 100 συντελεστές

Όπως ξέρουμε από τη θεωρία αλλά και από της προηγούμενες εργασίες, για φίλτρο Wiener ισχύουν οι εξισώσεις Wiener-Hopf: $Rw_o = p$

Ο υπολογισμός των ζητούμενων πινάκων, έγινε στο MATLAB στο αρχείο *merosA_abc.m* κάτω από τα σχόλια `%% Question B - Wiener filter` της γραμμής 20.

Γ) Αντικατάσταση του φίλτρου *Wiener* με ένα *Joint Process Estimator*

Σε αυτό το ερώτημα ζητήθηκε η αντικατάσταση του αντικατάσταση Wiener με ένα joint Process Estimator και ο υπολογισμός των συντελεστών/παραμέτρων α_m, Γ_m , και γ χρησιμοποιώντας τον αλγόριθμο Levinson-Durbin.

Η υλοποίηση του συγκεκριμένου ερωτήματος βρίσκεται στο αρχείο *merosA_abc.m* κάτω από τα σχόλια `% Question G - levinson-durbin` της γραμμής 31.

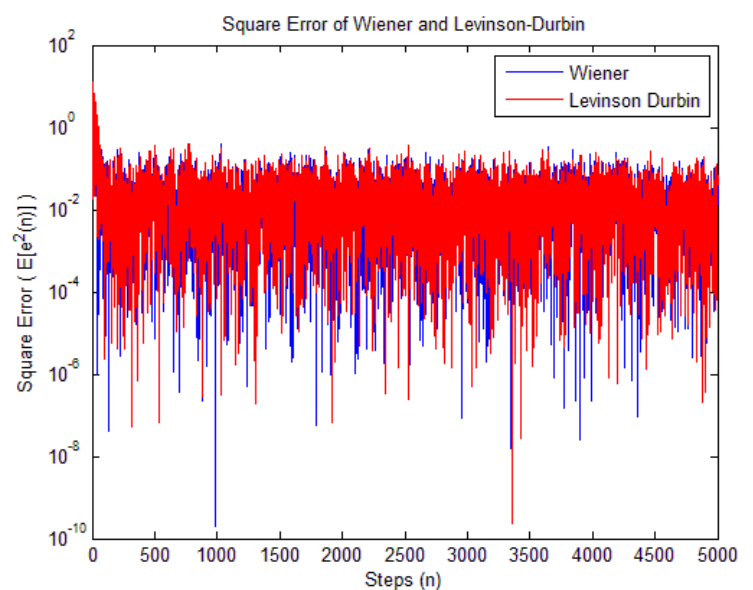
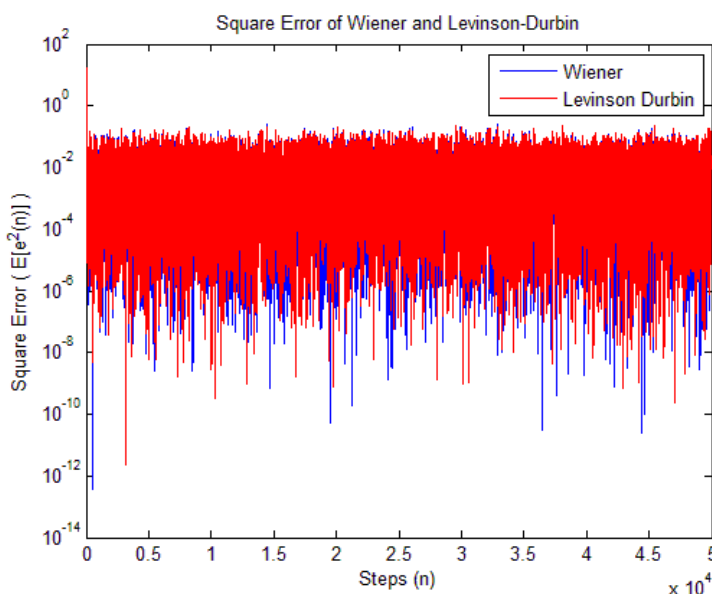
Για την υλοποίηση αυτού του ερωτήματος, χρησιμοποιήθηκαν οι έτοιμες συναρτήσεις Levinson-Durbin του κ. Σισμάνη, οι οποίες βρίσκονται στην πλατφόρμα eThmmy. Συγκεκριμένα, χρησιμοποιήθηκαν οι συναρτήσεις που βρίσκονται στα αρχεία *LevinsonDurbin.m* και *LevinsonDurbin_iterative.m*. Επίσης, βοηθητικό υπήρξε και το αρχείο *LDprediction.m*.

Παρακάτω παρουσιάζονται τα σφάλματα των διανυσμάτων των συντελεστών και η διαφορά των forward και backward prediction power errors, συγκριτικά με αυτά που υπολογίζονται χρησιμοποιώντας τις έτοιμες συναρτήσεις του MATLAB:

```
>> meros_A
Norm of difference with MATLAB levinson (iterative) 1.917354e-14
Norm of difference with MATLAB levinson (recursive) 1.917354e-14
Norm of difference with gama coefficients and Wiener 3.136307e-01
Norm of difference with Matlab backward errors 3.764949e-15
```

Από τα παραπάνω αποτελέσματα παρατηρούμε ότι το σφάλμα στον επαναληπτικό καθώς και στον αναδρομικό Levinson-Durbin σε σχέση με την υλοποίηση του MATLAB, είναι της τάξης 10^{-14} . Η διαφορά του backward error είναι της τάξης 10^{-15} . Και τέλος, επιβεβαιώνεται η σχέση που συνδέει τους συντελεστές γ και w_o με τους συντελεστές Wiener, καθώς έχουμε σφάλμα της τάξης 10^{-1} .

Παρακάτω παρουσιάζονται 2 διαγράμματα με διαφορετικό step, για το τετραγωνικό σφάλμα της εκτίμησης από το φίλτρο Wiener και από τον Estimator.



Από τα διαγράμματα, παρατηρούμε ότι το σφάλμα του Levinson-Durbin (Estimator) ταυτίζεται με αυτό του Wiener και έτσι ο αλγόριθμος Levinson-Durbin λειτουργεί όπως το φίλτρο Wiener, καθαρίζοντας το σήμα, μετά από μικρό αριθμό βημάτων. Επίσης, στο 2^ο διάγραμμα φαίνεται ότι στα πρώτα δείγματα το σφάλμα του estimator είναι σχετικά μεγάλο, και σταδιακά μειώνεται μετά από μικρό αριθμό βημάτων. Όπως φαίνεται και στα διαγράμματα, το φίλτρο Wiener έχει γενικά μικρότερο σφάλμα σε σχέση με τον estimator. Επίσης, το φίλτρο Wiener λειτουργεί καλύτερα σε σχέση με τον estimator, καθώς διαθέτει εξ αρχής όλη την πληροφορία του σήματος με αποτέλεσμα να είναι σε θέση να καθαρίσει καλύτερα το σήμα. Παρόλα αυτά, η προσέγγιση του Levinson-Durbin γίνεται με μεγάλη ακρίβεια.

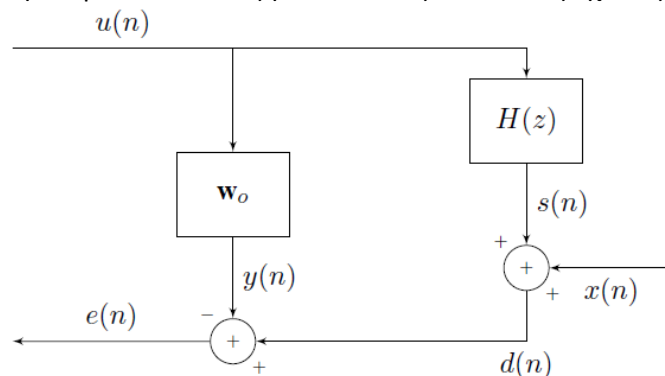
Δ) Απομάκρυνση Παρεμβολών από το μουσικό κομμάτι

Σε αυτό το υποερώτημα ζητήθηκε η εκ νέου σχεδίαση ενός joint process estimator για την απομάκρυνση των παρεμβολών του μουσικού κομματιού *music.bat* που δίνεται. Η υλοποίηση για αυτό το υποερώτημα βρίσκεται στο αρχείο *merosA_d.m*. Η μεθοδολογία που ακολουθήθηκε είναι αντίστοιχη του προηγούμενου υποερωτήματος και έγινε πάλι η χρήση της συνάρτησης *LevinsonDurbin.m* του κ. Σισμάνη.

Οπότε, τρέχοντας το αρχείο *merosA_partD.m* μπορεί κανείς να ακούσει το μουσικό κομμάτι χωρίς τις παρεμβολές.

Μέρος Β

Για το Μέρος Β, έχουμε την παρακάτω διάταξη αλλά και τις ακόλουθες σχέσεις:



$$s(n) = -0.23 u(n) + 0.67 u(n-1) - 0.18 u(n-2) + 0.39 u(n-3)$$

$$u(n) = -0.87 u(n-1) - 0.12 u(n-2) - 0.032 u(n-3) + v_1(n)$$

$$x(n) = -0.57 x(n-1) - 0.16 x(n-2) - 0.08 x(n-3) + v_2(n)$$

$$d(n) = s(n) + x(n)$$

όπου $v_1(n)$ και $v_2(n)$ είναι λευκός θόρυβος με μηδενική μέση τιμή και διακύμανση $\sigma_{v_1}^2 = 0.73$ και $\sigma_{v_2}^2 = 0.39$

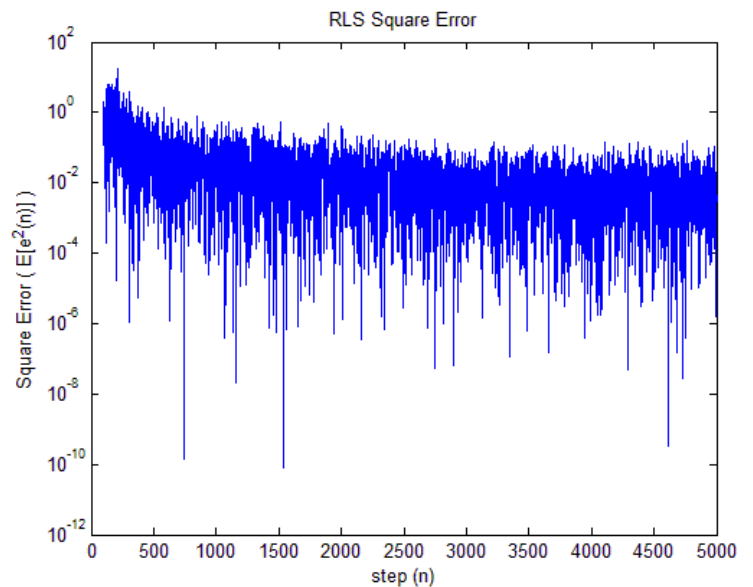
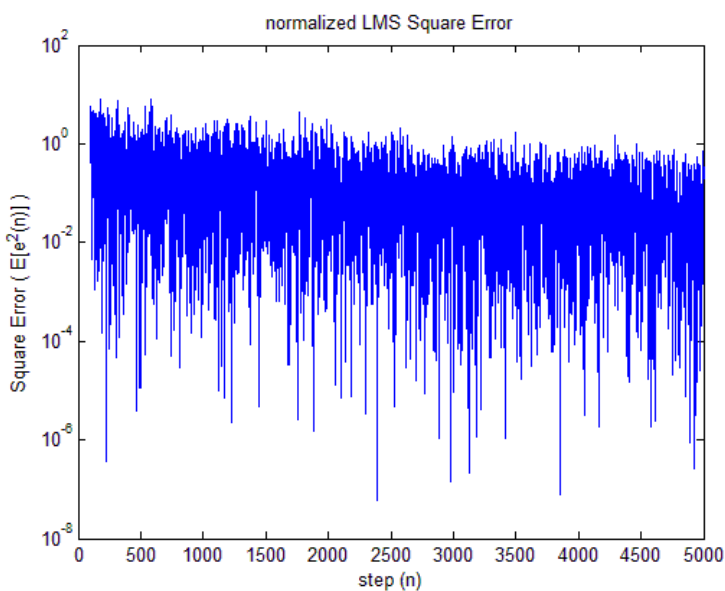
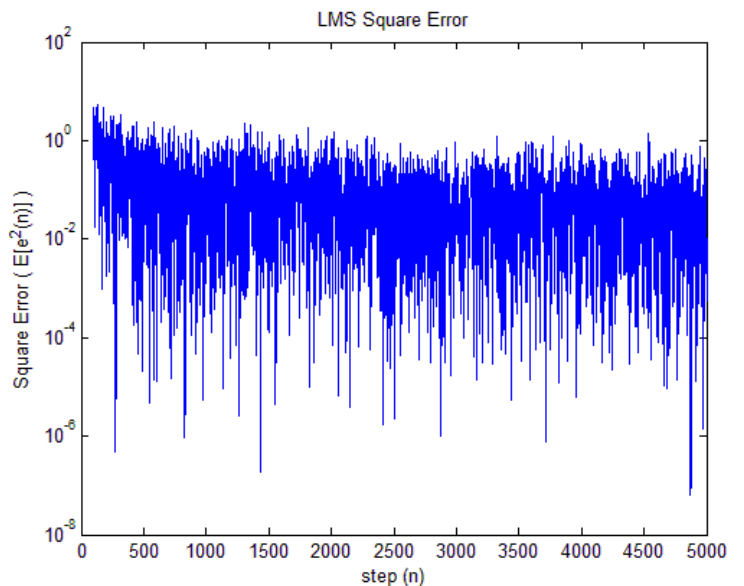
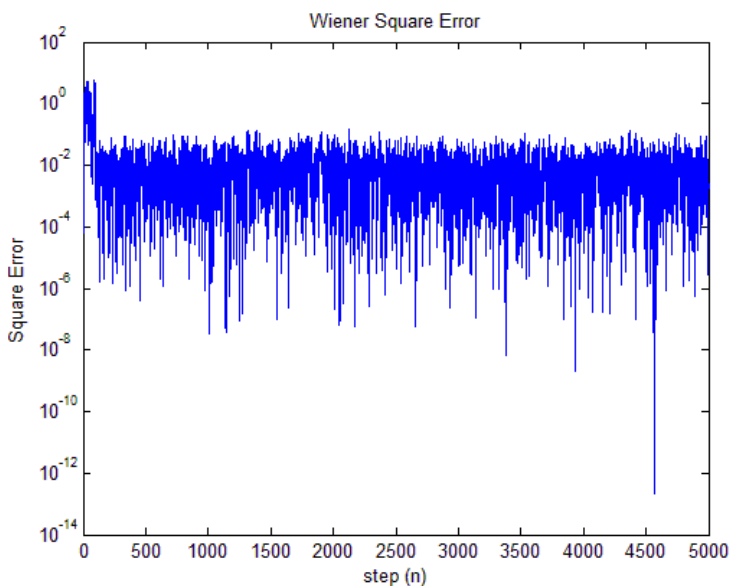
A) Υπολογισμός των R, p, w_o του φίλτρου Wiener

Ο υπολογισμός των ζητούμενων πινάκων, έγινε στο MATLAB στο αρχείο *merosB_ab.m* στις γραμμές 55-60, με αντίστοιχο τρόπο όπως και στο μέρος A.

B) Προσέγγιση βέλτιστων συντελεστών Wiener

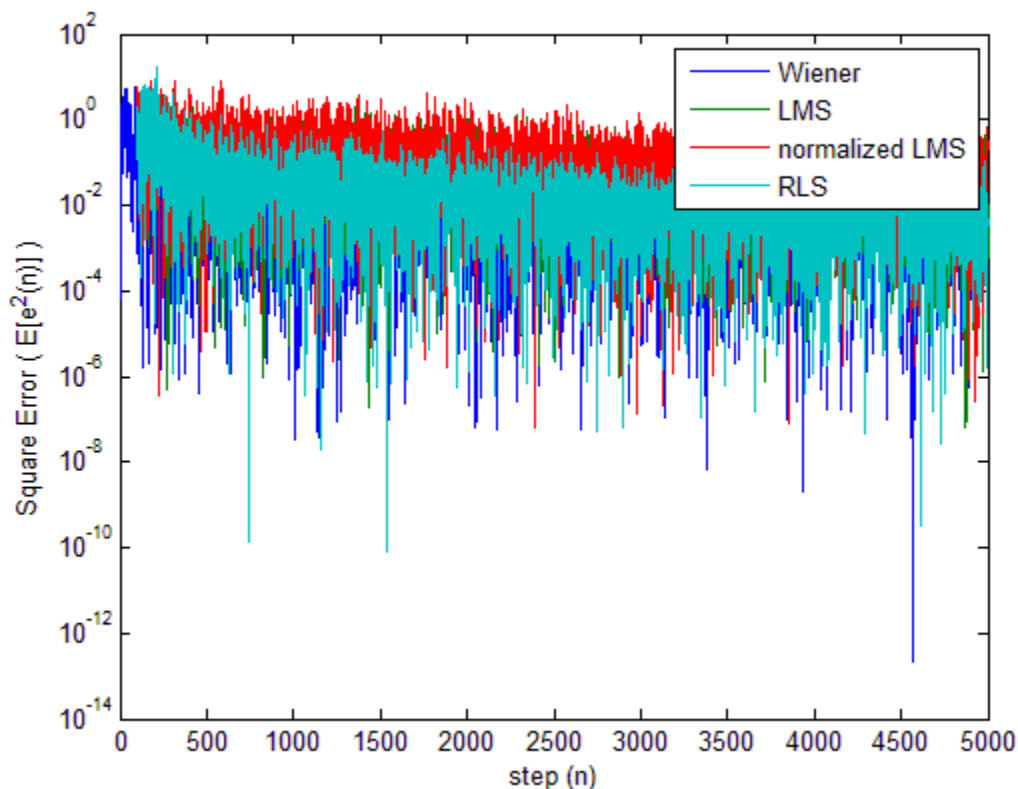
Σε αυτό το υποερώτημα ζητήθηκε η προσέγγιση των συντελεστών *Wiener* χρησιμοποιώντας τους αλγόριθμους προσαρμογής *LMS*, *normalized LMS* και *RLS*. Οι αλγόριθμοι υλοποιήθηκαν με την προαναφερθείσα σειρά στο αρχείο *merosB_ab.m*.

Για κάθε αλγόριθμο, δημιουργήσαμε το αντίστοιχο διάγραμμα του τετραγωνικού σφάλματος της εκτίμησης του αρχικού σήματος και τέλος, δημιουργήσαμε ένα συγκριτικό διάγραμμα με το τετραγωνικό σφάλμα από κάθε αλγόριθμο.



Αρχικά, στο 1^ο διάγραμμα βλέπουμε το τετραγωνικό σφάλμα που προκύπτει από την προσέγγιση του σήματος με το φίλτρο *Wiener*. Ακολούθως, στα υπόλοιπα διαγράμματα, βλέπουμε το τετραγωνικό σφάλμα προσέγγισης του σήματος χρησιμοποιώντας τους αλγόριθμους προσαρμογής *LMS*, *normalized LMS* και *RLS*, αντίστοιχα.

Παρακάτω, παρουσιάζεται το συγκριτικό διάγραμμα με το τετραγωνικό σφάλμα από κάθε αλγόριθμο.



Από τα παραπάνω διαγράμματα, μπορούμε να συμπεράνουμε ότι και οι 3 αλγόριθμοι προσεγγίζουν το βέλτιστους συντελεστές Wiener με αρκετά ικανοποιητικά αποτελέσματα. Ο αλγόριθμος όμως που ξεχωρίζει είναι ο RLS, εφόσον έχει μικρότερο σφάλμα συγκριτικά με τους άλλους 2 και επίσης καταφέρνει να προσεγγίσει τους συντελεστές Wiener σε μικρότερο αριθμό βημάτων.

Παράλληλα, χρησιμοποιώντας το χρονόμετρο του Matlab (tic/toc), μετρήθηκαν οι χρόνοι εκτέλεσης κάθε αλγόριθμου. Τα αποτελέσματα είναι τα ακόλουθα:

Elapsed times	
Wiener:	0.194628
LMS:	0.042793
normalized LMS:	0.088797
RLS:	1.118616

Η γρηγορότερη εκτέλεση ήταν του αλγόριθμου LMS. Παρόλα αυτά, δεν τον καθιστά απαραίτητα γρηγορότερο, εφόσον ο αλγόριθμος RLS προσεγγίζει γρηγορότερα τους συντελεστές Wiener και επομένως θα μπορούσε να χρησιμοποιηθεί μικρότερης τάξης φίλτρο για γρηγορότερη εκτέλεση.

Γ) Απομάκρυνση παρεμβολών συνομιλητή

Στο συγκεκριμένο υποερώτημα, για τον καθαρισμό της ηχούς του δεύτερου συνομιλητή, χρησιμοποιήθηκε αρχικά το φίλτρο Wiener. Τα αποτελέσματα ήταν πολύ καλά μιας και μπορούσαμε να διακρίνουμε την φωνή του κ. Πιτσιάνη χωρίς θόρυβο.

Ακολούθως, για τον καθαρισμό της ηχούς χρησιμοποιήθηκαν οι αλγόριθμοι προσαρμογής ούτως ώστε να πάρουμε μια γενική εικόνα για την αποτελεσματικότητα του κάθε αλγόριθμου και να είμαστε σε θέση να απαντήσουμε στο επόμενο υποερώτημα.

Όταν χρησιμοποιήθηκαν οι αλγόριθμοι LMS και normalized LMS, η ηχώ είχε απομακρυνθεί σε τέτοιο σημείο που μπορούσε να γίνει αντιληπτή η ομιλία του κ. Πιτσιάνη. Παρόλα αυτά, και με τον LMS αλλά και με τον normalized LMS, το σήμα δεν είχε καθαρίσει τόσο καλά όσο είχε καθαρίσει με το φίλτρο Wiener.

Τέλος, χρησιμοποιήθηκε ο αλγόριθμος RLS, οποίος ήταν υπερβολικά αργός με αποτέλεσμα να διακόψουμε την διαδικασία καθαρισμού του σήματος και να δοκιμάσουμε εκ νέου με μικρότερο αριθμό συντελεστών. Δοκιμάζοντας λοιπόν διάφορους αριθμούς συντελεστών, παρατηρήσαμε ότι ο αλγόριθμος RLS με μικρότερο αριθμό συντελεστών (π.χ. 500) μπορεί να καθαρίσει το σήμα τόσο καλά όσο και οι αλγόριθμοι LMS. Αυξάνοντας τον αριθμό συντελεστών, αυξάνεται και ποιότητα του ήχου που παράγεται από το καθαρό σήμα, ωστόσο αυξάνεται και ο χρόνος εκτέλεσης του αλγόριθμου.

Η υλοποίηση του υποερωτήματος αυτού, βρίσκεται στο αρχείο *merosB_c.m*.

Δ) Επιλογή αλγόριθμου με βάση την ποιότητα

Αν καλούμασταν να επιλέξουμε κάποιο αλγόριθμο λαμβάνοντας υπόψη αποκλειστικά και μόνο την ποιότητα του ήχου και όχι τον χρόνο εκτέλεσης, τότε βάση θεωρίας θα επιλέγαμε τον αλγόριθμο RLS. Αυτό όμως, γίνεται αντιληπτό και στην πράξη. Όπως είδαμε στα προηγούμενα υποερωτήματα αλλά και στα διαγράμματα που παρουσιάσαμε, ο αλγόριθμος RLS προσεγγίζει τους βέλτιστους συντελεστές Wiener γρηγορότερα και με μεγαλύτερη ακρίβεια από τους αλγόριθμους LMS. Έτσι, σε μεγαλύτερα χρονικά διαστήματα η ακρίβεια στην προσέγγιση θα είναι ακόμη πιο μεγάλη, παρέχοντας μας ποιοτικότερα αποτελέσματα.