Εργασία 2
Προσαρμογή στο Πεδίο της Συχνότητας

Μπεκιάρης Θεοφάνης ΑΕΜ:8200

Ερώτημα A

Ο ορισμός του διακριτού μετασχηματισμου Fourier θα μετασχηματιστεί σύμφωνα με τα παρακάτω.

$$\hat{x}_k = \sum_{j=0}^{2^q-1} x_j \overbrace{e^{-2\pi i \frac{jk}{2^q}}}^{\omega_n^{jk}} = \sum_{j=0}^{2^{q-1}-1} x_{(2j)}\, e^{-2\pi i \frac{2jk}{2^q}} + \sum_{j=0}^{2^{q-1}-1} x_{(2j+1)}\, e^{-2\pi i \frac{(2j+1)k}{2^q}}$$

$$= \underbrace{\sum_{j=0}^{2^{q-1}-1} x_{(2j)}\, \overbrace{e^{-2\pi i \frac{jk}{2^{q-1}}}}^{\omega_{\frac{n}{2}}^{jk}}}_{\hat{x}_k^{(1)}} + \overbrace{e^{-2\pi i \frac{k}{2^q}}}^{\omega_n^{k}} \underbrace{\sum_{j=0}^{2^{q-1}-1} x_{(2j+1)}\, \overbrace{e^{-2\pi i \frac{jk}{2^{q-1}}}}^{\omega_{\frac{n}{2}}^{jk}}}_{\hat{x}_k^{(2)}}, \; k = 0, 1, \ldots, \frac{n}{2} - 1$$

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}^{(1)} + \Omega_n\, \hat{\mathbf{x}}^{(2)} \\ \hat{\mathbf{x}}^{(1)} - \Omega_n\, \hat{\mathbf{x}}^{(2)} \end{bmatrix}, \; \Omega_n = \begin{bmatrix} \omega_n^0 & 0 & \ldots & 0 \\ 0 & \omega_n^1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \omega_n^{k-1} \end{bmatrix}$$

Στο τέλος του script fftproof έχει συπληρωθεί το αντίστοιχο κομμάτι κωδικά που ζητείται έχοντας χρησιμοποιήσει την παραπάνω τελική σχέση που δίνει τον FFT με την χρήση πινάκων.

Ερώτημα B

<u>Για τον FFT</u>

Η αναδρομική συνάρτηση που ζητείται για τον υπολογισμό FFT είναι το αρχείο anadromikhFFT.m. Για να υπολογίσουμε τον αριθμό των flops του fft θα βασιστούμε στην αναδρομική συνάρτηση που έχουμε φτιάξει και θα υπολογίσουμε απο αυτήν μία αναδρομική σχέση για τον αριθμό των flops.Όπως αναφέρεται και στην εκφώνηση ισχύει οτι :πρόσθεση μιγαδικών: 2 flop και πολλαπλασιασμός μιγαδικών: 6 flop.Το κύριο κομμάτι του κώδικα μας που παράγει αριθμό flops είναι το εξής.

```
fe = x(1:2:n);
fo = x(2:2:n);
if n~=2
  X1 = anadromikhFFT(fe);
  X2 = anadromikhFFT(fo).*w(n);
else
  X1 = fe;
  X2 = fo;
end
F1 = X1 + X2;
F2 = X1 - X2;
y = [F1;F2];
```

Στο συνολικό κώδικα χωρίζουμε το αρχικό σήμα μήκους n στην μέση και καλούμε 2 φορές την αναδρομική συναρτησή για n/2 μήκους σήματα.Άρα έστω οτι θεωρούμαι ότι ο αριθμός flops της συνάρτησης για σήμα μήκους n είναι P(n) τότε η αναδρομικές συναρτήσεις που καλούμε προσθέτουν η κάθε μία τους P(n/2) flops(οι γραμμές 4 και 5 στο παραπάνω κώδικα).Η γραμμή 5 εκτελεί πολλαπλασιασμό n/2 στοιχείων άρα έχουμε απο αυτήν (n/2)*6 flops.Τέλος οι γραμμές 10 και 11 εκτελούν n/2 προσθέσεις η κάθε μια τους δηλαδή συνολικά n προσθέσεις άρα απο αυτές προκύπτουν n*2 flops.Επομένως σύμφωνα με τα παραπάνω ο αναδρομικός τύπος που δίνει των αριθμό των flops είναι

P(n)=n*2+6*n/2+P(n/2)

Η σχέση ισχύει μέχρι και για n=4 καθώς η P(2) είναι η πιο μικρή τιμή και είναι ίση με P(2)=4.Η τιμή P(2)=4 προκύπτει πολύ εύκολα αν παρατηρήσουμε ξανά τον παραπάνω κώδικα.Για n=2 ο κώδικας δεν ξανα κάνει αναδρομή αλλά θέτει τα X1 και X2 ισα με fe και fo αφού είναι τα τελευταία στοιχεία.Τα μοναδικά flops που προκύπτουν παράγονται απο 2 προσθέσεις (γραμμές 10 και 11) και είναι ίσα με 2*2=4 άρα T(2)=4.Τελικά η παραπάνω σχέση μετά και απο τα τελευταία μπορεί να γραφεί σαν.

P(n)=P(2)*n/2+6*n/2+P(n/2) όπου **P(2)=4** και **n>2**

<u>Για τον DFT</u>

Για τον απλό αλγόριθμο DTF απο την αντίστοιχει σχέση απο την οποία προκύπτει

$$\hat{x}_k = \sum_{j=0}^{2^q-1} x_j \overbrace{e^{-2\pi i \frac{jk}{2^q}}}^{\omega_n^{jk}}, \; k = 0, 1, \ldots, n-1$$

μπορούμε να διακρίνουμε ότι για κάθε k (δηλαδή n φορές) το Xk προκύπτει απο n πολλαπλασιασμούς μεταξύ μιγαδικών και n-1 προσθέσεις μιγαδικών.Άρα συνολικά ο αριθμός των flops για n σήμα μήκους n είναι

P(n)=n*(n*6+(n-1)*2)
P(n)=(n^2)*8 +n*2

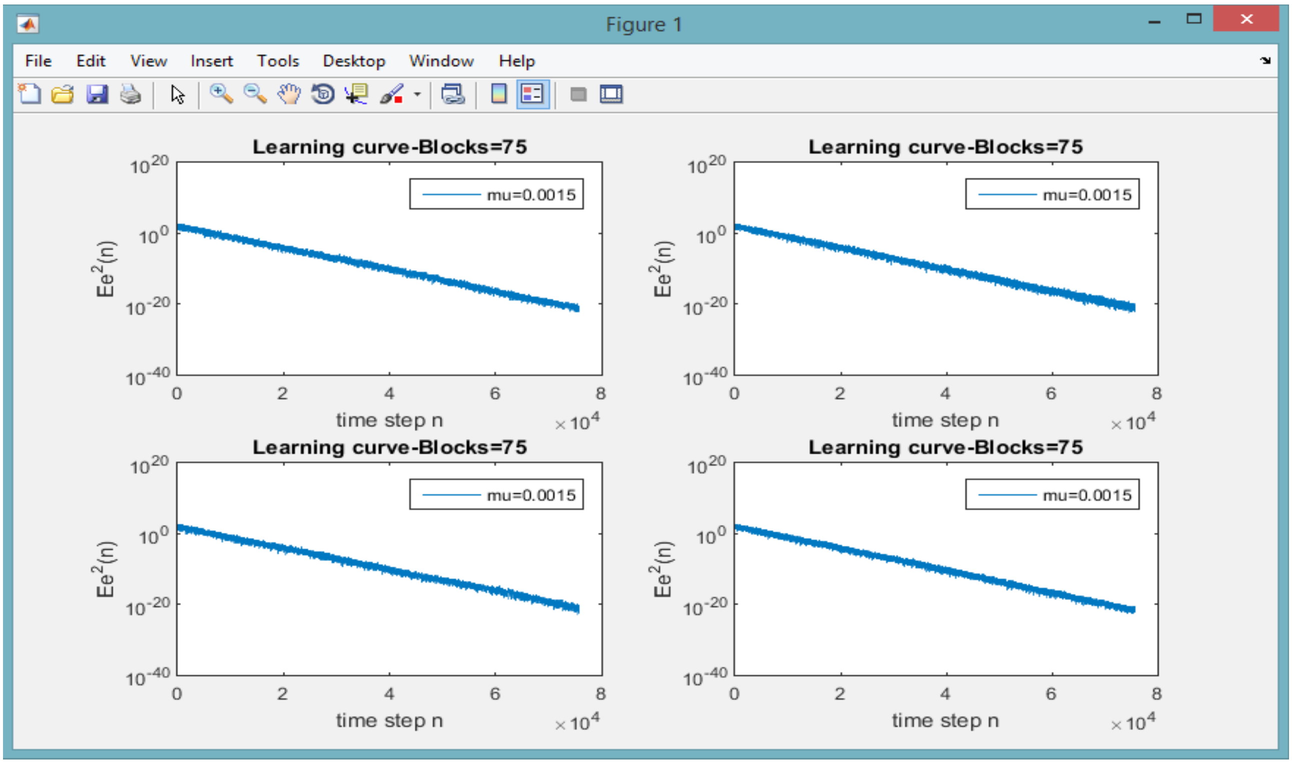
Για τους 2 αλγόριθμους αυτό που βλέπουμε απο τις παραπάνω σχέσεις είναι ότι για τον DFT ο αριθμός των flops είναι συνάρτηση του τεγραγώνου του n ενώ για τον FFT απο την αναδρομική σχέση φαίνεται ότι ο αριθμός των flops του FFT είναι πρώτου βαθμόυ σχέση σε συνάρτηση του n. Αυτό σημαίνει ότι για σήμα μήκους n ο αριθμός των flops ο FFT είναι μικρότερος του αριθμού flops του DFT,άρα ο FFT είναι υπολογιστικά καλύτερος του DFT.

Ερώτημα Γ

Για το ερώτημα γ έχω δημιουργήσει το script με όνομα convolution.m στο οποίο δημιουργούνται τυχαία δύο σήματα x και y και στην συνέχεια εκτελώ τις υλοποιήσεις που ζητούνται.Παράλληλα με την εκτέλεση κάθε υλοποίησης το script εκτυπώνει την διαφόρα των αποτελεσμάτων που προκύπτουν με τα αποτελέσματα την συνάρτησης conv(x,y) του MATLAB.

Ερώτημα Δ

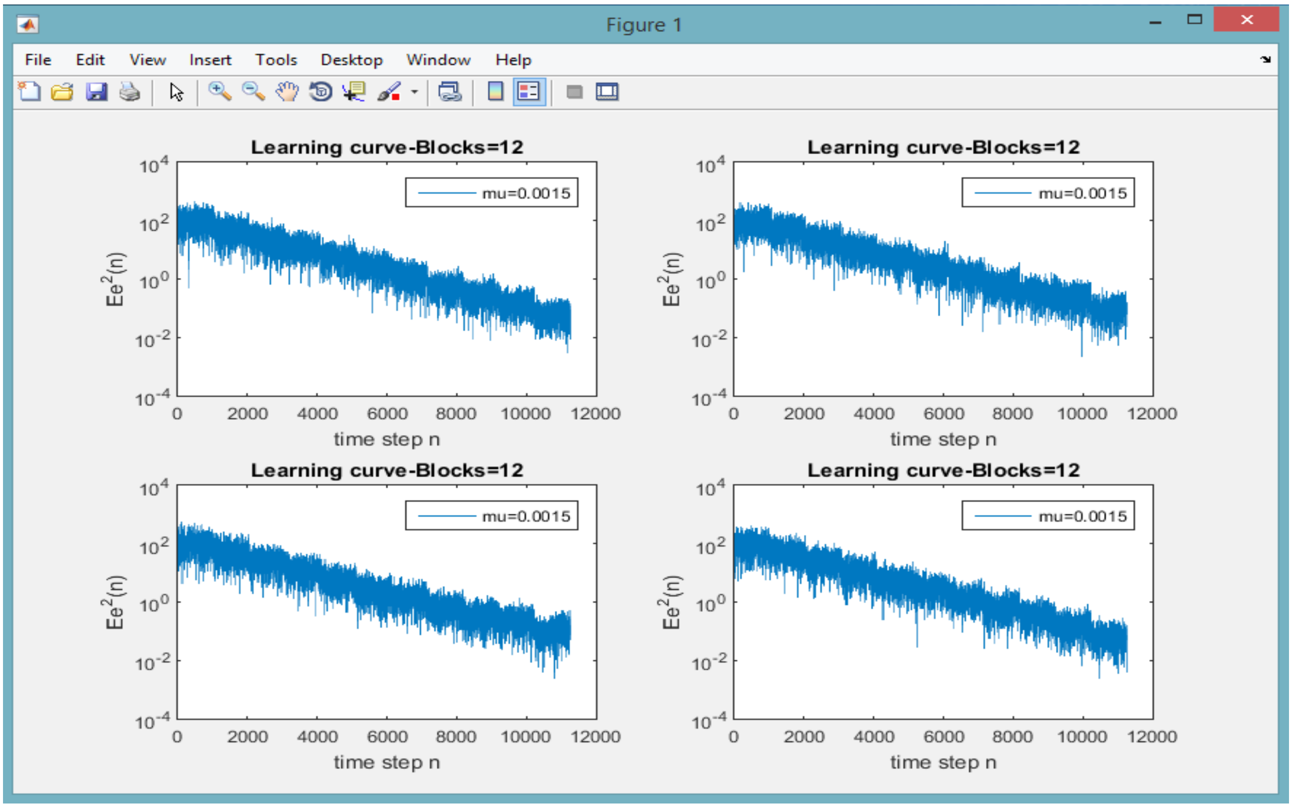
Τα αρχεία block_lms1,2,3,4 περιέχουν τους αντίστοιχους κώδικες των 4 υλοποιήσεων που ζητούνται.Για να εκτελέσουμε και τις 4 υλοποιήσεις πρέπει να τρέξουμε το script run_block_lms. Τα διαγράμματα του τετραγωνικού σφάλματος ή αλλίως εκμάθησης για τις 4 διαφορετικές υλοποιήσεις του αλγόριθμου Block LMS φαινονται παρακάτω.Οι καμπλύλες έχουν σχηματιστεί για 75 blocks,παραμετρο μ=0.0015 και για 5 ανεξάρτητες δοκιμές(independent trials).



Όπως φαίνεται και απο το διάγραμμα οι διαφορετικές υλοποίσεις δεν παρουσιάζουν κάποια διαφορά ως προς την ποιότητα των αποτελεσμάτων.Επίσης βλέπουμε την απόδοση του αλγόριθμου block lms ενεξαρτήτως υλοποιήσεις καθώς το σύστημα plant προσεγγίζεται με πολύ μεγάλη ακρίβεια, με σφάλμα της τάξης του 10^-20.Αν και η ποιότητα των υλοποιήσεων δεν αλλάζει ωστόσο η απόδοσή τους είναι διαφορετική,οι χρόνοι εκτέλεσεις των αλγορίθμων για τα παραπάνω δεδομένα είναι.

	Χρόνος εκτέλεσης
block_lms1	8.6742 seconds
block_lms2	10.8528 seconds
block_lms3	3.229 seconds
block_lms4	3.1295 seconds

Σύμφωνα με τα παραπάνω δεδομένα βλέπουμε ότι η υλοποίσησεις με προσαρμογή στο πεδίο της συχνότητας κάνοντας χρήση του *FFT* είναι πολύ πιο αποδοτικές σε σύγκριση με τις κλασσικές υλοποιήσεις (2 πρώτες) .Η δεύτερη ειναι πιο αργή απο όλες καθώς χρησιμοποιεί μετασχηματισμούς πινάκων που επιβαρύνουν αρκετά το πρόγραμμα,οι δυο τελευταίες με την χρήση του FFT ειναι σχεδόν ίσες στην απόδοση με ένα μικρό προβάδισμα στην *unconstrained* υλοποίηση που χρησιμοποιεί μόνο 3 μετασχηματισμούς FFT.



Για να μπορέσουμε να εκτιμήσουμε την ταχύτητα με την οποία συγκλίνει ο αλγόριθμος και προσεγγίζει το σύστημα plant εκτελούμε ξανα τα προγράμματα για τον αριθμό των 12 blocks και παρατηροούμε ότι για αυτό τον αριθμό block έχουμε καταφέρει να προσεγγίσουμε το σύστημα plant με τετραγωνικό σφάλμα της τάξης του 10^-1=0.1.

Τέλος θα τρέξουμε τον τελευταίο αλγόριθμο για δίαφορες τιμές του μ για να δούμε την σύγκλιση του σε σχέση με το μ.Απο ότι βλέπουμε μέχρι και την τιμή μ=0.002 έχουμε καλή σύγκλιση με μικρό σφάλμα(πιό μεγάλο βέβαια απο ότι για μ=0.0015).Για μεγαλύτερες τιμές βλέπουμε ότι το σύστημα γίνεται ασταθές και το σφάλμα αυξάνει σε πολύ μεγάλο βαθμό.Προφανώς για τιμές μικρότερες απο αυτές και μεγαλύτερες του μηδενός το σύστημα είναι ευσταθές αλλά συγκλίνει με πολύ πιο αργό ρυθμό για όσο μικρένει το μ.