

ΗΥ-370: Ψηφιακή Επεξεργασία Σήματος
Χειμερινό Εξάμηνο 2016
Διδάσκοντες: Γ. Στυλιανού - Γ. Καφεντζής
Πρώτο Εργαστήριο

Σημείωση: Για βοήθεια σχετικά με τις εντολές/συναρτήσεις MATLAB γράψτε: “*doc εντολή/συνάρτηση*” στο Command Window του MATLAB.

Άσκηση 1 - Υπολογισμός Περιόδου (20 μ.)

Έστω το διακριτό σήμα

$$x[n] = 2 \cos(0.1\pi n) + 3 \sin(0.2\pi n) \quad (1)$$

(Α) Ας δημιουργήσουμε στο MATLAB 50 δείγματα του παραπάνω διακριτού σήματος και ας εμφανίσουμε το αποτέλεσμα με την εντολή `stem`, ορίζοντας κατάλληλα το διάνυσμα του διακριτού χρόνου n , να είναι από μηδέν έως 49 (σύνολο 50 χρονικές στιγμές).

```
n = 0:49; % Discrete time indices
x = 2*cos(0.1*pi*n) + 3*sin(0.2*pi*n); % Signal
stem(n,x); xlabel('Samples'); % Visualize
title('Synthetic signal');
```

- Από το σχήμα, μπορείτε να βρείτε ποιά είναι περίπου η περίοδος του σήματος; (μην κλείσετε το γράφημα)
- Υπολογίστε με ακρίβεια την περίοδο του σήματος σε αριθμό δειγμάτων στο χαρτί, και επιβεβαιώστε το αποτέλεσμα ελέγχοντας στο MATLAB τις τιμές του σήματος.
- Υπολογίσατε σωστά την περίοδο αρχικά “με το μάτι”;
- Από ποιά συχνότητα του σήματος προέρχεται η περίοδος του σήματος;
- Αν αυτό το σήμα έχει προέλθει από δειγματοληψία ενός σήματος συνεχούς χρόνου με συχνότητα δειγματοληψίας 200 Hz, τότε πόση είναι η περίοδος του σήματος συνεχούς χρόνου;

(Β) Ας υποθέσουμε ότι προσθέτουμε άλλη μια συνιστώσα στο παραπάνω σήμα:

$$x[n] = 2 \cos(0.1\pi n) + 2 \cos(0.15\pi n) + 3 \sin(0.2\pi n) \quad (2)$$

- Δημιουργήστε 150 δείγματα στο MATLAB. Εκτελώντας ξανά την εντολή `stem` για το παραπάνω σήμα, εμφανίστε το.
- Έχει αλλάξει η περίοδος του σήματος σε σχέση με το προηγούμενο σήμα;
- Αν ναι, από ποιά συχνότητα προέρχεται η νέα περίοδος;
- Πόση είναι η περίοδος σε αριθμό δειγμάτων;
- Πόση είναι η περίοδος σε δευτερόλεπτα, αν το σήμα συνεχούς χρόνου από το οποίο προήλθε δειγματοληψήθηκε με συχνότητα δειγματοληψίας 200 Hz;

(Γ) Έστω το διακριτό σήμα

$$x[n] = \cos(0.1\pi + 0.4n) \quad (3)$$

όπου $n = 0, 1, \dots, 99$.

(α) Σχεδιάστε με την εντολή `stem` το σήμα.

(β) Ελέγξτε το γράφημα. Είναι το σήμα περιοδικό; Ποιά περίοδο βρίσκετε “με το μάτι”;

(γ) Επιβεβαιώστε (ή διαψεύστε) την απάντησή σας υπολογίζοντας θεωρητικά την περίοδο.

Άσκηση 2 - Διόρθωση σφαλμάτων (30 μ.)

Τα συστήματα DVD αποθηκεύουν τις εικόνες που αποτελούν ένα video με μορφή σειρών από ψηφιακούς αριθμούς. Αν υπάρχει κάποιο πρόβλημα στο δισκάκι (ελαφρά γρατζουνιά, φθορά, σκόνη), το σύστημα του DVD προσπαθεί να “καθαρίσει” το δίσκο από το πρόβλημα χρησιμοποιώντας φίλτρα (δηλ. συστήματα διακριτού χρόνου που περιγράφονται ως εξισώσεις διαφορών!). Εδώ θα εξετάσουμε δυο τέτοια απλά συστήματα, τα οποία θα ονομάζουμε στο εξής *φίλτρα*.

- Το πρώτο είναι το φίλτρο κυλιόμενης μέσης τιμής (moving average). Συγκεκριμένα, θα χρησιμοποιήσουμε ένα τέτοιο φίλτρο τριών σημείων (3-point Moving Average), το οποίο είδατε και στις διαλέξεις:

$$y[n] = \frac{1}{3} \sum_{k=0}^2 x[n-k] \quad (4)$$

- Το δεύτερο είναι ένα φίλτρο μεσαίας τιμής (median filter), επίσης τριών σημείων:

$$y[n] = \text{median}(x[n+1], x[n], x[n-1]) \quad (5)$$

Παρατηρήστε ότι και τα δυο φίλτρα απαιτούν μόνο 3 τιμές της εισόδου για να παράξουν μια τιμή της εξόδου.

i. Χρησιμοποιώντας την εντολή

```
A = imread('Saturn.tif');
```

διαβάστε την εικόνα Saturn.tif. Δείτε τη με την εντολή

```
figure; imshow(A);
```

Υποθέστε ότι αυτή είναι μια καθαρή εικόνα από ένα DVD δίσκο.

ii. Χρησιμοποιώντας την εντολή

```
B = imread('NoiseSaturn.tif');
```

διαβάστε την εικόνα NoiseSaturn.tif. Δείτε τη με την εντολή

```
figure; imshow(B);
```

Ο θόρυβος που βλέπετε να έχει προστεθεί στην εικόνα ονομάζεται *salt and pepper noise* (θόρυβος “αλάτι και πιπέρι”, λόγω της μορφής του). Υποθέστε ότι προήλθε από σφάλμα ανάγνωσης του δίσκου DVD λόγω φθοράς.

iii. Προσπαθήστε να εφαρμόσετε τα παραπάνω φίλτρα για να καθαρίσετε την εικόνα B. Σκεφτείτε ότι η εικόνα είναι ένας πίνακας $M \times N$, που σας είναι γνωστά (γράψτε `[M, N] = size(B)` αν θέλετε να τα έχετε σε μεταβλητές). Εφαρμόστε τα φίλτρα σας κατά γραμμές ή κατά στήλες του πίνακα. Θεωρήστε ότι κάθε γραμμή (ή στήλη) αποτελεί ένα μονοδιάστατο σήμα $x[n]$ για τις παραπάνω εξισώσεις, το οποίο πρέπει να διατρέχετε σε τριάδες, με μετατόπιση ένα δείγμα κάθε φορά. Η έξοδος $y[n]$ θα είναι μια αριθμητική τιμή την οποία θα αποθηκεύσετε σε έναν πίνακα C, στην κατάλληλη θέση. Προσέξτε το type casting μεταξύ των εικόνων όταν τις απεικονίζετε. Αν θέλετε, εμπνευστείτε ☺ από τον παρακάτω κώδικα που αφορά το φίλτρο μεσαίας τιμής της Σχέσης (4) εφαρμοσμένο στις γραμμές τις εικόνας:

```

% Initialize C
[M,N] = size(B);
C = zeros(M,N);

% Cast to double for processing
B = double(B);

% Apply a moving average (MA) filter in every row
for i = 1:M
    for j = 1:N-2 % Why N-2 and not N?
        x = % INSERT CODE HERE - use B matrix to get
              % triads of data of row i
        y = % INSERT CODE HERE - use x vector to implement
              % your filter
        C(i,j) = y; % Store result in another matrix
    end
end

% Cast it to unsigned int
C = uint8(C);

% Show the result
figure; imshow(C); title('MA-filtered image in rows');

```

Μπορείτε αν θέλετε να χρησιμοποιήσετε τις συναρτήσεις `sum`, `mean`, `median`, αλλά μόνο αυτές.

Για κάθε ένα από τα δυο φίλτρα:

- iv. Παραδώστε κώδικα που παράγει δυο εικόνες του πίνακα C, μια που έχει προέλθει από φιλτράρισμα των γραμμών του πίνακα B, και μια που έχει προέλθει από φιλτράρισμα των στηλών του πίνακα B. Σχολιάστε τις παρατηρήσεις σας. Συμπεριλάβετε τις παραγόμενες εικόνες στην αναφορά σας. Εν γένει, θα χαρακτηρίζατε και τα δυο φίλτρα ως αποδοτικά;
- v. Παραδώστε κώδικα που εφαρμόζει τα φίλτρα σας διαδοχικά (π.χ. εφαρμόστε ένα φίλτρο πρώτα σε γραμμές, και στο αποτέλεσμα που θα πάρετε εφαρμόστε το ίδιο φίλτρο ξανά κατά στήλες). Υλοποιήστε αυτή την προσέγγιση και με τα δυο φίλτρα. Συμπεριλάβετε τις παραγόμενες εικόνες στην αναφορά σας. Παρέχει καλύτερο αποτέλεσμα από πριν; Σχολιάστε.

Άσκηση 3 - Ασφαλής Επικοινωνία (30 μ.)

Για την ασφαλή επικοινωνία δεδομένων μεταξύ δυο σταθμών, χρησιμοποιούμε ένα πρότυπο σήμα μήκος 50 δειγμάτων, για να το προσθέσουμε ως θόρυβο σε τυχαίες χρονικές στιγμές στα σήματα που θέλουμε να στείλουμε. Το σήμα-πρότυπο βρίσκεται στο αρχείο `Pat.wav`, και το έχουν τόσο ο πομπός όσο και ο δέκτης. Το σήμα που εμείς λάβαμε σε κάποια χρονική στιγμή στο δέκτη είναι το σήμα `noisy.wav`. Ας τα φορτώσουμε και ας τα ακούσουμε:

```

[pattern, fs] = audioread('Pat.wav');
soundsc(pattern, fs);

figure(1); plot(pattern); xlabel('Samples');

[noisy, fs] = audioread('noisy.wav');

```

```
soundsc(noisy, fs);

figure(2); plot(noisy); xlabel('Samples');
```

Η εντολή `plot` είναι βολικότερη εδώ για την απεικόνιση, αλλά στην πραγματικότητα τα σήματά μας είναι διακριτού χρόνου (μπορείτε να το δείτε με την εντολή `stem`, όπως την γνωρίζετε).

Θα θέλαμε να ανακτήσουμε το καθαρό σήμα. Για το σκοπό αυτό, μπορούμε να χρησιμοποιήσουμε την πράξη της *ετεροσυσχέτισης*, η οποία ορίζεται ως

$$\phi_{xy}[l] = \sum_{n=-\infty}^{+\infty} x[n]y[n+l] \quad (6)$$

Όμοια με την πράξη της ετεροσυσχέτισης στο συνεχή χρόνο, μας αποκαλύπτει ομοιότητες και διαφορές μεταξύ των σημάτων που συσχετίζονται. Ο δείκτης l ονομάζεται *lag*, που σημαίνει *καθυστέρηση*, και έχει ακριβώς όμοιο ρόλο με τη μεταβλητή τ στο συνεχή χρόνο¹. Η συνάρτηση `[cc, lags] = xcorr(...)` του MATLAB θα σας βοηθήσει να συσχετίσετε τα σήματα. Δείτε προσεκτικά (!) το *documentation* της, καθώς η υλοποίηση του MATLAB διαφέρει λίγο από τον παραπάνω ορισμό, και όπως ίσως θυμάστε και από το συνεχή χρόνο, η ετεροσυσχέτιση είναι διαφορετική όταν συσχετίσουμε το $x[n]$ με το $y[n]$ απ' όταν συσχετίσουμε το $y[n]$ με το $x[n]$. Στις χρονικές τιμές των *lags* όπου η τιμή της ετεροσυσχέτισης είναι μεγάλη, σημαίνει ότι εκεί έχει βρεθεί μεγάλη ομοιότητα του ενός σήματος με το άλλο.

Προς βοήθειά σας, πληροφορείστε ότι το πρότυπο σήμα έχει προστεθεί σε 36 θέσεις στο σήμα `noisy.wav`. Αν βρείτε αυτές τις θέσεις, μπορείτε απλά να το αφαιρέσετε με μια απλή πράξη αφαίρεσης του σήματος προτύπου από το θορυβώδες σήμα. Επίσης, το σήμα που έχει σταλεί στο δέκτη περιέχει ομιλία πολύ καλής ποιότητας. Αυτό θα σας βοηθήσει να ελέγξετε τα αποτελέσματά σας.

Ο παρακάτω κώδικας θα σας βοηθήσει. Χρησιμοποιήστε το *documentation* του MATLAB όπου έχετε αμφιβολία.

```
[cross_corr, lags] = % INSERT CODE HERE; % Compute cross-correlation
figure(3); plot(lags, cross_corr); % Plot it and check it carefully
xlabel('Lags'); title('Cross-correlation');

% Using information from the plot, define a good threshold T
% which separates peaks from other -not useful- data
T = % INSERT CODE HERE ;

% Use findpeaks to easily detect instants of high correlation
% T will be used as the minimum peak height threshold
[pks, locs] = findpeaks(cross_corr, lags, 'MinPeakHeight', T);

% Check your selected peaks - did your threshold work? Did you select 36 peaks?
plot(lags, cross_corr); hold on; plot(locs, pks, 'o'); hold off;

% Check without counting peaks :)
size(locs) % is it 1 x 36 ?
```

¹Θυμηθείτε:

$$\phi_{xy}(\tau) = \int_{-\infty}^{+\infty} x(t)y(t+\tau)dt \quad (7)$$

```
% Copy noisy signal to process it
clean = noisy;

% Subtract pattern from noisy signal - use "clean" vector
% For every lag index present in vector locs, you have
% to subtract the pattern signal from the received signal
% Use operator ":" to choose pieces of the signal in MATLAB
% as follows:
% signal(start:end) = signal(start:end) - another_signal(start:end);
for i = 1:length(locs)
    %%%%%%%%%%%%%%%
    % INSERT CODE HERE
    %%%%%%%%%%%%%%%
end

% Listen! It should sound perfectly clear...
soundsc(clean, fs);
```

Η συνάρτηση `findpeaks` του MATLAB είναι πολύ βολική για να βρίσκετε κορυφές σε μια συνάρτηση, ανάλογα με το πόσο απέχουν μεταξύ τους, με το αν θέλετε να υπερβαίνουν κάποια τιμή, κλπ. Ρίξτε απλά μια ματιά στο `documentation` της συνάρτησης στο MATLAB για να δείτε τις δυνατότητές της, μια και η εντολή όπως δίνεται στον παραπάνω κώδικα είναι πλήρης. Ελέγξτε προσεκτικά τις θέσεις των καθυστερήσεων (`lags`) που σας επιστρέφει η `findpeaks`. Δείτε τι υπάρχει στο θορυβώδες σήμα σε αυτές τις χρονικές στιγμές.

Για την παράδοση της άσκησης, γράψτε **πλήρη** αναφορά, συμπεριλαμβάνοντας απαντήσεις σε όλα τα ερωτήματα του εργαστηρίου, καθώς και διαγράμματα/γραφήματα/εικόνες με τα αποτελέσματά σας, και συμπεριλάβετε τον κώδικα MATLAB σε *ξεχωριστά* .m files.

Η παράδοση γίνεται **αποκλειστικά** με το πρόγραμμα TURNIN.

Ανάθεση: 23/09/2016

Προθεσμία: 02/10/2016, 23:59:59 (TURNIN timestamp)