ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
Τμήμα Επιστήμης Υπολογιστών

ΗΥ-370: Ψηφιακή Επεξεργασία Σήματος
Χειμερινό Εξάμηνο 2016
Διδάσκοντες: Γ. Στυλιανού - Γ. Καφεντζής
Πέμπτο Εργαστήριο - Ημερομηνία: 12/12/2016

Σημείωση: Για βοήθεια σχετικά με τις εντολές MATLAB, γράψτε doc/help *εντολή*.

Άσκηση 1 - Τεχνητή Αντήχηση

Έχουμε δει ότι τα συστήματα all-pass δε μεταβάλλουν το φάσμα πλάτους της εισόδου αλλά επιφέρουν μόνο μια αλλαγή στη φάση της. Αυτή η ιδιότητα των all-pass συστημάτων έχει πολλές εφαρμογές μεταξύ των οποίων και αυτής σε συστήματα δημιουργίας τεχνητής αντήχησης (reverberation) σε ηχογραφήσεις. Η πρόσθεση αντήχησης μας δίνει την αίσθηση ότι ένα μουσικό κομμάτι έχει παιχτεί σε μεγάλη αίθουσα. Συνήθως χρησιμοποιούμε αρκετά all-pass συστήματα που έχουν συνδεθεί σε σειρά. Τα all-pass συστήματα που χρησιμοποιούνται έχουν τη μορφή

$$H_{ap}(z) = \frac{z^{-m} - a}{1 - az^{-m}} \tag{1}$$

i. Κάντε το διάγραμμα πόλων-μηδενικών του φίλτρου, με χρήση της εντολής `zplane` για $a = 0.7$ και $a = -0.7$, με $m = 10$. Γράψτε `doc zplane` για να δείτε πως συντάσσεται η παραπάνω συνάρτηση. Παραδώστε στην αναφορά σας το διάγραμμα πόλων-μηδενικών.

ii. Δημιουργήστε τεχνητή αντήχηση στο μουσικό κομμάτι Music.wav που σας δίνεται με χρήση δυο all-pass συστημάτων σε σειρά. Φορτώστε το στο MATLAB, ακούστε το, και περάστε το μέσα από ένα φίλτρο που υλοποιεί το πρώτο all-pass σύστημα με τιμές $a_1 = -0.8$ και $m_1 = 1000$. Ακούστε το αποτέλεσμα. Περάστε το σήμα που αποκτήσατε μέσα από ένα δεύτερο φίλτρο που υλοποιεί το δεύτερο all-pass σύστημα με τιμές $a_2 = -0.7$ και $m_2 = 2500$. Παραδώστε πλήρη κώδικα που εκτελεί την εφαρμογή αντήχησης.

Άσκηση 2 - Σχεδίαση Φίλτρων

Στις διαλέξεις είδατε δυο διαφορετικούς τρόπους με τους οποίους μπορείτε να υλοποιήσετε ψηφιακά φίλτρα από προδιαγραφές που σας δίνονται, χρησιμοποιώντας γνώσεις από τον συνεχή χρόνο. Εδώ θα υλοποιήσουμε αυτά τα φίλτρα τόσο με αναλυτικό όσο και με σύντομο τρόπο, και θα εφαρμόσουμε σήματα φωνής στην είσοδό τους, ώστε να ακούσουμε το αποτέλεσμα που αυτά παράγουν.

I. <u>Impulse Invariance Butterworth Filter</u> (p.541, Oppenheim - Schafer)
Η τεχνική του Impulse Invariance (II) περιγράφει το πως από ένα φίλτο συνεχούς χρόνου μπορεί κανείς να υλοποιήσει το αντίστοιχο διακριτού χρόνου. Αν υπάρχει μια εξίσωση στο χώρο του Laplace που περιγράφει ένα φίλτρο $h_c(t)$ ως

$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k} \tag{2}$$

τότε η μέθοδος II μετατρέπει το φίλτρο αυτό σε διακριτού χρόνου $h_c[n]$ με μετασχ. Z ως

$$H(z) = \sum_{k=1}^{N} \frac{T_d A_k}{1 - e^{s_k T_d} z^{-1}} \tag{3}$$

με T_d την περίοδο δειγματοληψίας του φίλτρου.

(α΄) Χρησιμοποιώντας MATLAB, υλοποιήστε με αναλυτικό τρόπο - υλοποιώντας πρώτα το αντίστοιχο φίλτρο συνεχούς χρόνου - το παράδειγμα 7.2 του βιβλίου των Oppenheim - Schafer, το οποίο συζητήσατε και στο μάθημα, παράγοντας το Σχήμα 7.5 (σελ. 544 του βιβλίου). Το φίλτρο αυτό έχει προδιαγραφές

$$0.89125 \leq |H(e^{j\omega})| \leq 1, \quad 0 \leq |\omega| \leq 0.2\pi \tag{4}$$

$$|H(e^{j\omega})| \geq 0.17783, \quad 0.3\pi \leq |\omega| \leq \pi \tag{5}$$

Εφ'όσον η περίοδος δειγματοληψίας T_d δεν έχει σημασία στη μέθοδο αυτή, θεωρήστε ότι $T_d = 1$. Με αυτόν τον τρόπο, η "διακριτή" συχνότητα ω αντιστοιχεί στη "συνεχή" συχνότητα Ω. Άρα οι παραπάνω προδιαγραφές μετατρέπονται ως

$$0.89125 \le |H(e^{j\Omega}) \le 1, \quad 0 \le |\Omega| \le 0.2\pi \tag{6}$$
$$|H(e^{j\Omega})| \ge 0.17783, \quad 0.3\pi \le |\Omega| \le \pi \tag{7}$$

Αυτές είναι λοιπόν οι προδιαγραφές του φίλτρου συνεχούς χρόνου, το οποίο πρέπει να υλοποιήσετε ρητά, και μετά να το μετατρέψετε σε διακριτού χρόνου. Εν συντομία, πρέπει να

- λύστε στο χαρτί σας τις εξισώσεις που υπολογίζουν τις παραμέτρους N, Ω_c του αναλογικού φίλτρου στο χώρο του Laplace: αυτές σας δόθηκαν στο μάθημα, και δεδομένου ότι το Butterworth φίλτρο είναι της μορφής
$$|H_c(\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}} \tag{8}$$
  αυτές οι εξισώσεις είναι
$$1 + \left(\frac{0.2\pi}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.89125}\right)^2 \tag{9}$$
$$1 + \left(\frac{0.3\pi}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.17783}\right)^2 \tag{10}$$
  και προέρχονται από την ισότητα στις προδιαγραφές (6, 7).
  - Λύστε αναλυτικά το παραπάνω σύστημα και βρείτε ότι $\Omega_c = 0.70474$ και $N = 5.8858$. Μεταφέρετε τις λύσεις των εξισώσεων - όχι τους αριθμούς! - στο MATLAB[1].
  - Το N πρέπει να είναι ακέραιος, οπότε $N = 6$. Υπολογίστε ξανά την τιμή της Ω_c, για αυτήν την τιμή του N, και δείξτε ότι αυτή είναι $\Omega_c = 0.7032$.
- από τα παραπάνω, βρείτε τους πόλους του φίλτρου: γνωρίζετε την εξίσωση από τις διαλέξεις. Προγραμματίστε τη στο MATLAB.
- κρατήστε μόνο αυτούς που αντιστοιχούν σε ευσταθές και αιτιατό φίλτρο: από το σύνολο των πόλων που προγραμματίσατε στο MATLAB, επιλέξτε μόνο τους απαραίτητους, δηλ. όσους βρίσκονται στο αριστερό μιγαδικό ημιεπίπεδο του s-χώρου. Δείτε το σχήμα των διαλέξεών σας.
- σχηματίστε το φίλτρο $B(s)/A(s)$ που δημιουργούν οι πόλοι που επιλέξατε: ο λόγος πολυωνύμων του s θα φτιαχτεί ακριβώς όμοια με τα πολυώνυμα του z^{-1} που γνωρίζετε πως κατασκευάζονται στο MATLAB, δηλ. χρησιμοποιώντας τους συντελεστές τους μόνο.
- έχοντας πλέον το φίλτρο συνεχούς χρόνου (τους συντελεστές πολυωνύμων του s αριθμητή και παρονομαστή αντίστοιχα) χρησιμοποιήσετε τις εντολές `impinvar, freqz, grpdelay` για να μετατρέψετε το αναλογικό φίλτρο σε ψηφιακό και να δείτε την απόκριση πλάτους και την καθυστέρηση ομάδας, αντίστοιχα.

Για όλα τα παραπάνω, σας δίνεται ο σκελετός - κώδικας στο αρχείο `Lab5_BW_II.m`, τον οποίο και πρέπει να συμπληρώσετε. Ελέγξτε το σχήμα του φάσματος πλάτους. Τι παρατηρείτε για το $\omega_p = 0.2\pi$ και για το $\omega_s = 0.3\pi$, σε σχέση με τα αρχικά; Κάνετε μεγέθυνση των εικόνων στις κρίσιμες συχνότητες και δείτε/επιβεβαιώσετε την ακρίβεια των υπολογισμών σας. Καταγράψτε στην αναφορά σας τις παρατηρήσεις σας και παραδώστε το γράφημα που παρήγαγε το συμπληρωμένο αρχείο.

(β') Όλα αυτά που κάνατε ρητά στο παραπάνω αρχείο μπορούν να γίνουν και χωρίς όλη την παραπάνω διαδικασία, πιο εύκολα, με χρήση έτοιμων συναρτήσεων. Σας δίνουμε τις εντολές:

```
fs = 8000;  % half the sampling frequency in Hz
Wp = wp*fs;
Ws = ws*fs;
```

[1] Με άλλα λόγια, αν κάποιος θελήσει να αλλάξει τις τιμές των προδιαγραφών, τα νέα Ω_c, N να υπολογίζονται αυτόματα.

```
% Find optimum N, Wc
[Nm, Wcm] = buttord(Wp, Ws, -20*log10(d1), -20*log10(d2), 's');
[Bm, Am] = butter(Nm, Wcm, 's'); % Constructs the filter
[Bzm, Azm] = impinvar(Bm, Am, fs); % Analog -> Digital
[Hm, w] = freqz(Bzm, Azm, 512);
[Gdm, w] = grpdelay(Bzm, Azm, 512);

figure;
subplot(311);plot(w/pi, 20*log10(abs(Hm))); % Divide with pi for easier check
subplot(312);plot(w/pi, abs(Hm));
subplot(313);plot(w/pi, Gdm)
```

Απλά τρέξτε τις και επιβεβαιώστε ότι παίρνετε το ίδιο αποτέλεσμα. Παραδώστε στην αναφορά σας το γράφημα που σας δίνουν.

Περάστε το σήμα φωνής που σας δίνεται (`speech.wav`) μέσα από το παραπάνω φίλτρο (εντολή `filter`). Ακούστε το αποτέλεσμα και καταγράψτε τις παρατηρήσεις σας στην αναφορά σας.

II. <u>Bilinear Transform Butterworth Filter</u> (p.545, Oppenheim - Schafer)

Η τεχνική του Bilinear Transformation (BT) αποφεύγει το εγγενές πρόβλημα του aliasing που υπάρχει στην μέθοδο της Impulse Invariance. Η τεχνική αυτή αντιστοιχεί τη συχνότητα $-\infty \leq \Omega \leq \infty$ του συνεχούς χρόνου με τη συχνότητα $-\pi \leq \omega \leq \pi$ του διακριτού χρόνου, μέσω του μετασχηματισμού

$$s = \frac{2}{T_d}\Big(\frac{1-z^{-1}}{1+z^{-1}}\Big) \tag{11}$$

δηλ.

$$H(z) = H_c\Big(\frac{2}{T_d}\Big(\frac{1-z^{-1}}{1+z^{-1}}\Big)\Big) \tag{12}$$

Οι παραπάνω σχέσεις οδηγούν (δείτε τις διαλέξεις ή το βιβλίο σας) στις σχέσεις

$$\Omega = \frac{2}{T_d}\tan(\omega/2) \tag{13}$$

$$\omega = 2\tan^{-1}(\Omega T_d/2) \tag{14}$$

(α') Χρησιμοποιώντας MATLAB, υλοποιήστε - υλοποιώντας πρώτα το αντίστοιχο φίλτρο συνεχούς χρόνου - το παράδειγμα 7.3 του βιβλίου, το οποίο συζητήσατε και στο μάθημα, παράγοντας το σχήμα 7.11 (σελ. 553 του βιβλίου). Το φίλτρο αυτό έχει προδιαγραφές

$$0.89125 \leq |H(e^{j\omega}) \leq 1, \quad 0 \leq |\omega| \leq 0.2\pi \tag{15}$$

$$|H(e^{j\omega})| \geq 0.17783, \quad 0.3\pi \leq |\omega| \leq \pi \tag{16}$$

Εφ'οσον η περίοδος δειγματοληψίας T_d δεν έχει σημασία και στη μέθοδο αυτή, θεωρήστε ότι $T_d = 1$. Με αυτόν τον τρόπο, η "διακριτή" συχνότητα ω αντιστοιχεί στη "συνεχή" συχνότητα Ω. Άρα οι παραπάνω προδιαγραφές μετατρέπονται ως

$$0.89125 \leq |H(\Omega) \leq 1, \quad 0 \leq \Omega \leq \frac{2}{T_d}\tan(0.2\pi/2) \tag{17}$$

$$|H(\Omega)| \geq 0.17783, \quad \frac{2}{T_d}\tan(0.3\pi/2) \leq |\Omega| \leq \infty \tag{18}$$

Αυτές είναι λοιπόν οι προδιαγραφές του φίλτρου συνεχούς χρόνου, το οποίο πρέπει να υλοποιήσετε ρητά, και μετά να το μετατρέψετε σε διακριτού χρόνου.

Εν συντομία, θα πρέπει να

- λύστε στο χαρτί σας τις εξισώσεις που υπολογίζουν τις παραμέτρους N, Ω_c του αναλογικού φίλτρου στο χώρο του Laplace: αυτές σας δόθηκαν στο μάθημα, και - ξανά - δεδομένου ότι το Butterworth φίλτρο είναι της μορφής

$$|H_c(\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}} \tag{19}$$

  αυτές οι εξισώσεις είναι

$$1 + \left(2\frac{\tan(0.1\pi)}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.89125}\right)^2 \tag{20}$$

$$1 + \left(2\frac{\tan(0.15\pi)}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.17783}\right)^2 \tag{21}$$

  και προέρχονται από την ισότητα στις προδιαγραφές (17, 18).

  – Λύστε αναλυτικά το παραπάνω σύστημα και βρείτε ότι $N = 5.305$. Μεταφέρετε τις λύσεις των εξισώσεων - όχι τους αριθμούς! - στο MATLAB[2].
  – Το N πρέπει να είναι ακέραιος, οπότε $N = 6$. Υπολογίστε ξανά την τιμή της Ω_c, για αυτήν την τιμή του N, και δείξτε ότι αυτή είναι $\Omega_c = 0.766$.

- από τα παραπάνω, βρείτε τους πόλους του φίλτρου: γνωρίζετε την εξίσωση από τις διαλέξεις - είναι άλλωστε ίδια με πριν. Προγραμματίστε τη στο MATLAB.
- κρατήστε μόνο αυτούς που αντιστοιχούν σε ευσταθές και αιτιατό φίλτρο: από το σύνολο των πόλων που προγραμματίσατε στο MATLAB, επιλέξτε μόνο τους απαραίτητους, δηλ. όσους βρίσκονται στο αριστερό μιγαδικό ημιεπίπεδο του s-χώρου. Δείτε ξανά το σχήμα των διαλέξεών σας.
- σχηματίστε το φίλτρο $B(s)/A(s)$ που δημιουργούν οι πόλοι που επιλέξατε: ο λόγος πολυωνύμων του s θα φτιαχτεί ακριβώς όμοια με τα πολυώνυμα του z^{-1} που γνωρίζετε πως κατασκευάζονται στο MATLAB, δηλ. χρησιμοποιώντας τους συντελεστές τους μόνο.
- έχοντας πλέον το φίλτρο συνεχούς χρόνου (τους συντελεστές πολυωνύμων του s αριθμητή και παρονομαστή αντίστοιχα) χρησιμοποιήσετε τις εντολές `bilinear, freqz, grpdelay` για να μετατρέψετε το αναλογικό φίλτρο σε ψηφιακό και να δείτε την απόκριση πλάτους και την καθυστέρηση ομάδας, αντίστοιχα.

Για όλα τα παραπάνω, σας δίνεται ο σκελετός - κώδικας στο αρχείο `Lab5_BW_BT.m`, τον οποίο και πρέπει να συμπληρώσετε. Ελέγξτε το σχήμα του φάσματος πλάτους. Τι παρατηρείτε για το ω_p και για το ω_s, σε σχέση με τα αρχικά; Κάνετε μεγέθυνση των εικόνων στις κρίσιμες συχνότητες και δείτε/επιβεβαιώσετε την ακρίβεια των υπολογισμών σας. Καταγράψτε στην αναφορά σας τις παρατηρήσεις σας και παραδώστε το γράφημα που παρήγαγε το συμπληρωμένο αρχείο.

(β′) Όλα τα παραπάνω μπορούν να γίνουν και χωρίς όλη την παραπάνω διαδικασία, πιο εύκολα, με χρήση έτοιμων συναρτήσεων. Σας δίνουμε τις εντολές:

```
fs = 1;  % sampling frequency in Hz : does not matter
Wp = 2*fs*tan(wp/2);
Ws = 2*fs*tan(ws/2);

% Find optimum N, Wc
[Nm, Wcm] = buttord(Wp, Ws, -20*log10(d1), -20*log10(d2), 's');
[Bm, Am] = butter(Nm, Wcm, 's'); % Construct filter
[Bzm, Azm] = bilinear(Bm, Am, fs); % Analog -> Digital
[Hm, w] = freqz(Bzm, Azm, 512);
[Gdm, w] = grpdelay(Bzm, Azm, 512);

figure;
```

[2] Με άλλα λόγια, αν κάποιος θελήσει να αλλάξει τις τιμές των προδιαγραφών, τα νέα Ω_c, N να υπολογίζονται αυτόματα.

```
subplot(311);plot(w/pi, 20*log10(abs(Hm))); % Divide with pi for easier check
subplot(312);plot(w/pi, abs(Hm));
subplot(313);plot(w/pi, Gdm)
```

Περάστε το σήμα φωνής που σας δίνεται (`speech.wav`) μέσα από το παραπάνω φίλτρο (εντολή `filter`). Ακούστε το αποτέλεσμα και καταγράψτε τις παρατηρήσεις σας στην αναφορά σας. Υπάρχει διαφορά με το προηγούμενο;

III. FIR Design by windowing - Kaiser method (p. 585, Oppenheim - Schafer)
Χρησιμοποιώντας MATLAB, υλοποιήστε το παράδειγμα 7.6.1 στη σελίδα 591 του βιβλιου, το οποίο συζητήσατε και στο μάθημα, παράγοντας το σχήμα 7.34.

Πιο συγκεκριμένα:

(α΄) Δημιουργήστε ένα αρχείο `Lab5_Kaiser.m`.

(β΄) Οι προδιαγραφές του φίλτρου σας είναι

$$\omega_p = 0.4\pi \tag{22}$$
$$\omega_s = 0.6\pi \tag{23}$$
$$\delta = 0.001 \tag{24}$$

(γ΄) Η συχνότητα αποκοπής ω_c είναι

$$\omega_c = \frac{\omega_p + \omega_s}{2} \tag{25}$$

Προγραμματίστε τη στο MATLAB.

(δ΄) Οι παράμετροι[3] του παραθύρου Kaiser δίνονται ως

$$\Delta\omega = \omega_s - \omega_p \tag{26}$$
$$A = -20\log_{10}\delta \tag{27}$$
$$\beta = \begin{cases} 0.1102(A-8.7), & A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21), & 21 \leq A \leq 50 \\ 0.0, & A < 21 \end{cases} \tag{28}$$

και το μήκος του παραθύρου M πρέπει να ικανοποιεί τη σχέση

$$M = \frac{A-8}{2.285\Delta\omega} \tag{29}$$

ενώ

$$\alpha = M/2 \tag{30}$$

Προγραμματίστε τις παραπάνω εξισώσεις στο MATLAB, με την ίδια φιλοσοφία με τις προηγούμενες υλοποιήσεις φίλτρων. Επιβεβαιώστε ότι για τις προδιαγραφές που σας δίνονται οι τιμές που λαμβάνετε είναι

$$\Delta\omega = 0.2\pi,\ \ A = 60,\ \ \beta = 5.653,\ \ M = 37,\ \ \alpha = 18.5 \tag{31}$$

(ε΄) Το φίλτρο Kaiser στο πεδίο του χρόνου δίνεται ως

$$h[n] = \begin{cases} \dfrac{\sin(\omega_c(n-a))}{\pi(n-\alpha)} \dfrac{I_0[\beta(1-[(n-\alpha)/\alpha]^2)^{1/2}]}{I_0(\beta)}, & 0 \leq n \leq M \\ 0, & \text{αλλού} \end{cases} \tag{32}$$

Χρησιμοποιήστε την εντολή `besseli` για να παράγετε την τροποποιημένη συνάρτηση Bessel μηδενικού βαθμού και πρώτου είδους $I_0(\cdot)$, με ορίσματα τα α, β που βρήκατε.

[3] Σεβασμός σε αυτούς που έβγαλαν αυτές τις εξισώσεις! :-)

Χρησιμοποιήσετε τις εντολές `freqz, grpdelay` για να δείτε την απόκριση πλάτους και την καθυστέρηση ομάδας, αντίστοιχα. Τυπώστε επίσης την κρουστική απόκριση που φτιάξατε (εντολή `stem`). Παραδώστε τα γραφήματα που προέκυψαν. Τι Τύπου φίλτρο γραμμικής φάσης φτιάξατε;

Συγκρίνετε τη μέθοδο του παραθύρου με τετραγωνικό παράθυρο (διάρκειας $M+1$, δηλ. η κρουστική του απόκριση αποτελείται από $M+1$ άσσους) και με το Kaiser παράθυρο. Χρησιμοποιήστε την εντολή stem για να σχεδιάσετε τη μοναδιαία απόκριση κάθε φορά (δηλ. για κάθε ένα παράθυρο). Σχολιάστε την απόκριση σε συχνότητα του φίλτρου χρησιμοποιώντας τα δυο αυτά παράθυρα.

Περάστε το σήμα φωνής που σας δίνεται (`speech.wav`) μέσα από το παραπάνω φίλτρο (εντολή `filter`). Ακούστε το αποτέλεσμα και καταγράψτε τις παρατηρήσεις σας στην αναφορά σας. [4]

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.

Προθεσμία: 27/12/2016, timestamp: 23:59:59

[4] Υπάρχει φυσικά η εντολή kaiser η οποία σας επιστρέφει το παράθυρο Kaiser, με τα χαρακτηριστικά που θέλετε.