Project Σήματα ΙΙ

Κων/νος Αραβανής ΑΜ: 3628
mail-to: arabanis@ceid.upatras.gr

27 Αυγούστου 2009

1 Ερώτημα

1.1 Ανάκτηση των $x_\alpha(t)$ και $\tau_\alpha(t)$ από την υπέρθεσή τους

Από τη θεωρία μας είναι γνωστό ότι το οποιοδήποτε σήμα μπορεί να αναλυθεί σε άθροισμα δύο διαφορετικών σημάτων με πολλούς τρόπους, και συγκεκριμένα μπορούμε να βρούμε άπειρα ζευγάρια σήματων που να καταλήγουν στο ίδιο άθροισμα. Επιπλέον αν δεν μας είναι γνωστή κάποια επιπλέον πληροφορία, τότε οποιοδήποτε συνδιασμό και να επιλέξουμε είναι το ίδιο. Για να καταλήξουμε βέβαια σε μία μοναδική ανάλυση του δοθέντος σήματος απαιτείται ο προσδιορισμός κάποιων επιπλέον χαρακτηριστικών των σημάτων πληροφορίας και θορύβου. Η μέθοδος βέβαια που πάντα προτιμάμε να βρούμε καλό θα είναι να είναι γενική και να εφαρμόζεται σε πληθώρα σημάτων.

Γνωρίζουμε ότι ένα σήμα μπορεί να περιγραφεί πέρα από το πεδίο του χρόνου και από το συχνοτικό του περιεχόμενο. Αν τώρα το σήμα πληροφορίας και ο θόρυβος δεν περιέχουν κοινές συχνότητες (από Βασική Υπόθεση), τότε πληροφορία και θόρυβος μπορούν να διαχωριστούν μέσω των συχνοτήτων τους. Επιπλέον αν τα $x(t)$ και $\tau(t)$ έχουν διαφορετικό συχνοτικό περιεχόμενο[1] τότε και αυτά με τη σειρά τους μπορούν να διαχωριστούν σε δύο διαφορετικά σήματα από το $y(t)$, και να πάρουμε με άλλα λόγια τα δύο ξεχωριστά $x(t)$ και $\tau(t)$.

1.2 Τρόπος διαχωρισμού των $x_\alpha(t)$ και $\tau_\alpha(t)$

Ο διαχωρισμός αυτός μπορεί να επιτευχθεί με τη χρήση των κατάλληλων φίλτρων. Τα κλασσικά-ιδανικά φίλτρα από τη θεωρία μας είναι τα ακόλουθα:

1. **κατωπερατά ή κατωδιάβατα**

[1] δηλαδή δεν έχουν αλληλοεπικαλυπτόμενες συχνότητες.

Η ιδανική απόκριση συχνότητας ορίζεται:

$$D(e^{j\omega}) = \begin{cases} 1, 0 \preceq \omega \preceq \omega_c \\ 0, \text{αλλού} \end{cases} \quad (1)$$

όπου η συχνότητα ω_c καλείται συχνότητα αποκοπής.

2. **ανωπερατά ή ανωδιάβατα**

   Η ιδανική απόκριση συχνότητας ορίζεται:

$$D(e^{j\omega}) = \begin{cases} 1, \omega_c \preceq \omega \preceq \pi \\ 0, \text{αλλού} \end{cases} \quad (2)$$

   όπου η συχνότητα ω_c καλείται συχνότητα αποκοπής.

3. **ζωνοπερατά ή ζωνοδιάβατα**

   Η ιδανική απόκριση συχνότητας ορίζεται:

$$D(e^{j\omega}) = \begin{cases} 1, \omega_{c_1} \preceq \omega \preceq \omega_{c_2} \\ 0, \text{αλλού} \end{cases} \quad (3)$$

   όπου $\omega_{c_1}, \omega_{c_2}$ συχνότητες αποκοπής.

4. **αποκοπής ζώνης**

   Η ιδανική απόκριση συχνότητας ορίζεται:

$$D(e^{j\omega}) = \begin{cases} 0, \omega_{c_1} \preceq \omega \preceq \omega_{c_2} \\ 1, \text{αλλού} \end{cases} \quad (4)$$

   όπου $\omega_{c_1}, \omega_{c_2}$ συχνότητες αποκοπής.

5. **πολυπέρατα ή πολυδιάβατα**

   Η ιδανική απόκριση συχνότητας ορίζεται:

$$D(e^{j\omega}) = \begin{cases} 1, \omega_{l_i} \preceq \omega \preceq \omega_{u_i} \\ 0, \text{αλλού} \end{cases} \quad (5)$$

   όπου $\omega_{l_i}, \omega_{u_i}$ η κάτω και άνω συχνότητα αποκοπής της i-οστής ζώνης διάβασης.

6. **ολοπέρατα ή ολιδιάβατα**

   Η ιδανική απόκριση συχνότητας ορίζεται:

$$D(e^{j\omega}) = 1 \quad (6)$$

   δηλαδή τα φίλτρα αυτά δεν απομακρύνουν κάμμιά συχότητα[2].

[2]αυτό στη περίπτωση μας σίγουρα δεν μας κάνει

1.3 Για $f_s = 2f_M$ διατηρείται η πληροφορία από το x_n στο $x_\alpha(t)$;

Από το γνωστό **θεώρημα δειγματοληψίας του Shannon** ξέρουμε ότι:

Ένα σήμα $x_\alpha(t)$ συνεχούς χρόνου το οποίο δεν περιέχει συχνότητες μεγαλήτερες της f_M μπορεί να ανακατασκευαστεί ακριβώς από τα δείγματα $x_n = x_\alpha(nT_s)$, εάν η συχνότητα δειγματοληψίας ικανοποεί $f_s \succeq 2f_M$

Το θεώρημα δειγματοληψίας ισχύει και στη περίπτωση μας για $f_s = 2f_M$, εφόσον το σήμα $x_\alpha(t)$ δεν περιέχει ημιτονικό σήμα συχνότητας f_M (μπορεί φυσικά να περιέχει ημιτονικά σήματα με μικρότερες συχνότητες). Η τιμή $f_s = 2f_M$ γνωστή και ως **όριο Nyquist** πρόκειται για την μικρότερη δυνατή συχνότητα δειγματοληψίας που δίνει την ακριβή ανακατασκευή ενός αναλογικού σήματος, πεπερασμένου εύρους ζώνης, από τα δείγματα του. Έτσι η πληροφορία του x_α από το x_n μπορούμε να πούμε με βεβαιότητα ότι διατηρείται.

1.4 Μπορεί να επιτευχθεί διαχωρισμός των $x_\alpha(t)$ και $\tau_\alpha(t)$;

Όταν στο σύστημα μας επιλέξουμε σαν συχνότητα δειγματοληψίας $f_s = 2f_M$, δηλαδή δύο φορές την μεγαλύτερη απο την μέγιστη συχνότητα του σήματος βασικής ζώνης, τότε δεν μπορούμε με την ίδια συχνότητα δειγματοληψίας να δειγματοληπτήσουμε και το ASCII κείμενο, γιατί τότε ο ήχος και το κείμενο δεν θα έχουν διαφορετικό συχνοτικό περιεχόμενο. Αυτό γιατί η συχνότητα δειγματοληψίας είανι δύο φορές μεγαλύτερη μόνο της μιας συνιστώσας και όχι διπλάσια από τη μέγιστη συχνότητα του συνολικού σήματος.

2 Ερώτημα

2.1 Κατασκευή του transmitter

Παρακάτω παρατίθεται ο κώδικας σε Matlab που εξομειώνει την λειτουργία του πομπούτου συστήματος μας. Η συνάρτηση αυτή δέχεται σαν είσοδο ένα διακριτού χρόνου σήμα βασικής ζώνης, ένα αλφαρηθμιτικό, τη συχνότητα δειγματοληψίαςf_s, καθώς και τις παραμέτρους $f = [f_0; f_1]$, T_b (της BFSK διαμόρφωσης) και E_b και παράγει στην έξοδο το σήμα y_n.

Για την επεξήγηση του κώδικα παρακαλώ ανατρέξτε στα σχόλια του κώδικα που εξηγούν βήμα προς βήμα τι συμβαίνει. Ο κώδικας βασίστηκε στους τύπους και τις υποδείξεις που δόθηκαν στην εκφώνηση.

```
function [ y_n ] = transmitter( voice, text, f_s, f, T_b, E_b)
% The function that calculates the y_n signal that is the addition of both
% the (sampled) voice signal and the encoded text signal
% [ y_n ] = transmitter( voice, text, f_s, f, T_b, E_b)
%
% Arguments:
%   voice: the signal that is already sampled and must be added with the
%          signal of the text
```

```
%   text:  a char array with the text for encoding
%   f_s:   frequency of sampling
%   f:     array with length 2 that keeps the frequences of the sines
%   T_b:   Period of each bit’s transmition
%   E_b:   Energy of signal for each bit

% period of sampling
T_s = 1 / f_s;

% the length of the text (actually the total number of samples needed)
text_length = num_samples(text, T_b, T_s);

% the length of the signal that we want to transmit with a text
voice_length = length(voice);

% the difference between the length of the text signal and the voice signal
diff = abs(text_length - voice_length);

% pad the smaller signal with the method of post null ascii  charachters
% for the text or with zero for the voice
if (text_length > voice_length)
    % produce the signal of the text without padding anything to it
    text = encoder(text, f_s, f, T_b, E_b, 0);
    voice = padarray(voice, [diff], ’post’);
else
    % produce the signal of the text after padding it with the missing null
    % ascii charachters
    text = encoder(text, f_s, f, T_b, E_b, diff);
end

% the digital signal to be transmitted (it is the addition of both
% the (sampled) voice signal and the encoded text signal)
y_n = voice + text;

end
```

2.1.1 Βοηθητικές Συναρτήσεις

1. Κωδικοποιητής κειμένου

```
function text_encoded = encoder(text, f_s, f, T_b, E_b, l_expand)
% text_encoded = encoder(text, f_s, f, T_b, E_b)
%
% Arguments:
%   text: a char array with the text for encoding
%   f_s:  frequency of sampling
%   f:    array with length 2 that keeps the frequences of the sines
%   T_b:  Period of each bit’s transmition
%   E_b:  Energy of signal for each bit

% period of sampling
T_s = 1 / f_s;

% convert the text - charachters to ascii decimal code
dec_symbol_array = text - 0;

% convert each decimal to a row of 8-bit’s binary numbers
bin_symbol_matrix = dec2bin(dec_symbol_array, 8);

% concatenate the rows to a single column
bin_symbol_array = bin2dec(reshape(bin_symbol_matrix, [], 1));
```

```
% the total number of samples that must be used for the signal to transmit
% a bit
num_samples_per_bit = ceil(T_b / T_s);

% pad the bit-ascii text with null characters in order to add it then with
% the voice signal (http://www.asciitable.com/)
bin_symbol_array = padarray(bin_symbol_array, [l_expand], 'post');

% the length of the the text represented with bits in ascii code
M = length(bin_symbol_array);

% the total number of samples that must be used for the signal to
% transmit the text
num_samples = M * num_samples_per_bit;

% initialize the text_encoded array with zeros
text_encoded = zeros(num_samples, 1);

for n = 1: num_samples
     for m = 0: M-1
            text_encoded(n, 1) = text_encoded(n,1) + phi_l(bin_symbol_array(floor((n - 1)/num_samples_pe
     end
end

% Debugging
% plot that represents the digital signal of text (encoded)
plot([1:num_samples] , text_encoded);
% title of the plot
title('Text Encoding');
% title of the x axis
xlabel('time');
% title of the y axis
ylabel('signal of text');

end
```

2. Συνάρτηση ϕ_l

```
function [ phi_l ] = phi_l(l, t, f, T_b, E_b)
% PHI_L Function
% [ phi_l ] = phi_l(l, t, f, T_b, E_b)
%
% Arguments:
%   l:   the value of the current bit (0 or 1)
%   t:   time
%   f:   array with length 2 that keeps the frequences of the sines
%   T_b: Period of each bit's transmition
%   E_b: Energy of signal for each bit


% calculation of phi_l from the function that is given from the exercise
if ( (l == 0) | (l==1) )
    phi_l = sqrt(2 * E_b / T_b) * cos(2 * pi * f(l + 1, 1) * t);
else
    phi_l = 0;
end

end
```

3. Εύρεση των ελάχιστων δειγμάτων που χρειαζόμαστε για να στείλουμε το σήμα μας με βάση τα δεδομένα T_b και T_s

```
function [ num_samples ] = num_samples(text, T_b, T_s)
% Function that calculates the lesser ammount of samples that must be used
% to encoded the given text with period of bit T_b and Period of sampling
% T_s
% [ num_samples ] = num_samples(text, T_b, T_s)
%
% Arguments:
%   text: a char array with the text for encoding
%   T_b:  Period of each bit's transmition
%   T_s:  period of sampling

% the length of the the text represented with bits in ascii code (each
% character in ascii is represented with a byte)
M = length(text) * 8;

% the total number of samples that must be used for the signal to transmit
% a bit
num_samples_per_bit = ceil(T_b / T_s);

% the total number of samples that must be used for the signal to
% transmit the text
num_samples = M * num_samples_per_bit;

end
```

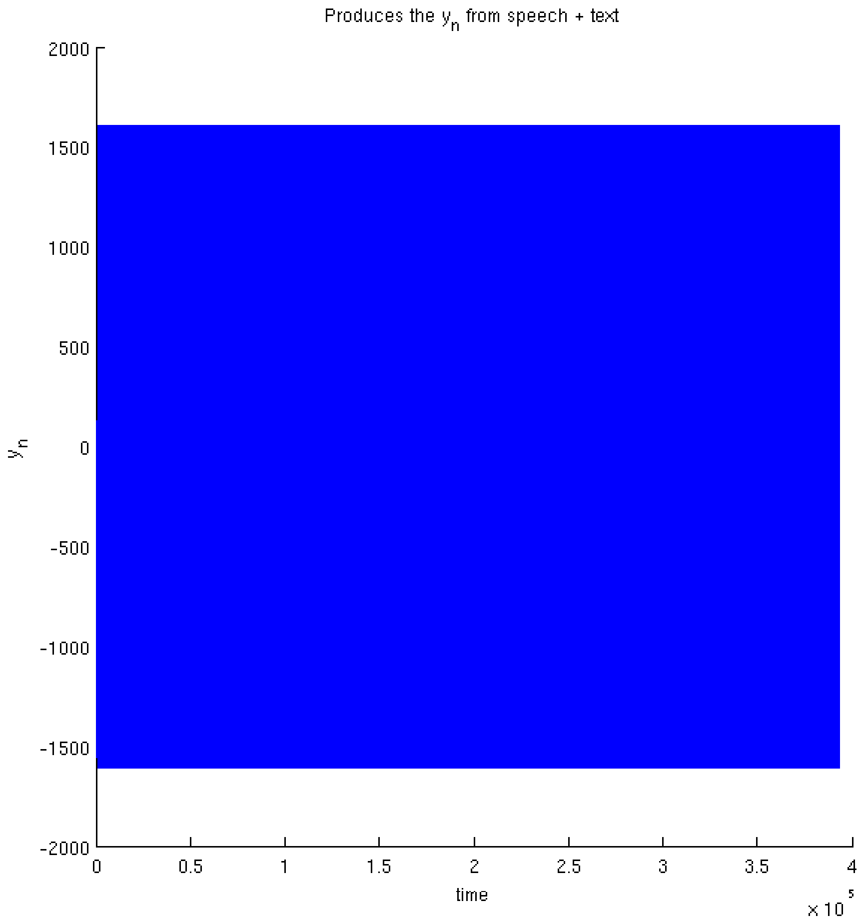
2.2 'I will never cheat again during the exams'

Στο σημείο αυτό τρέξαμε το κωδικά μας για το δωθέν σήμα βασικής ζώνης (από το σπεεςη.ματ) και λόγο τού ότι το f_s ήταν $48kHz$ σαν f_0 και f_1 επιλέξαμε συχνότητες μεγαλύτερες από το $f_s/2$ για να μην έχουμε αλληλοεπικαλήψεις. Επίσης σαν T_s επιλέξαμε τα 0.025 sec και σαν $E_b = 0.3$

Η εντολή λοιπόν που εκτελέσαμε ήταν η ακόλουθη:
y_n = transmitter(speech, 'I will never cheat again during the exams',fs,[25000;30000],0.025,0.3);

Ενώ η γραφική παράσταση που πήραμε τρέχοντας τα παρακάτω σε Command Window

```
>> hold on;
>> plot([1:length(y_n)], y_n)
>> xlabel('time')
>> ylabel('y_n')
>> title('Produces the y_n from speech + text')
>> hold;
```



ήταν η παραπάνω εικόνα.

Μπορεί να φαίνεται περίεργο το αποτέλεσμα αλά στη πραγματικότητα δεν είναι λόγο της πυκνότητας του σήματος σε τόσο μικρό χρονικό διάστημα. Αν επιχειρήσουμε να κάνουμε zoom-in στην εικόνα μπορούμε να παρατηρήσουμε ότι πρόκειται πράγματι για το σήμα που περιμέναμε να δούμε.

Όσον αφορά τον υπολογισμό του dft αυτό που κάναμε είναι να εκτελέσουμε την ακόλουθη εντολή:

```
>> y_n_fft = fft(y_n);
```

Όλα τα παραπάνω αποτελέσματα βρίσκονται στο παραδοτέο.

2.3 Παραγωγή δικού μας σήματος βασικής ζώνης - y_n - dft

Λόγω έλλειψης χρόνου αυτό το ερώτημα δεν έγινε παρόλαυτά να αναφέρουμε ότι η διαδικασία για τη παραγωγή ενός δικού μας σήματος βασικής ζώνης θα ήταν μέσω του συνδιασμού των idinput με type = 'rgs' για να είναι γκαουσιανό και ίσως της iddata.

3 Ερώτημα

3.1 Gaussian διαδικασία $\bar{w}(t)$, μέση τιμή, διασπορά

Λόγω του ότι το $\bar{w}(t)$ προκύπτει από το γραμμικό (πράξεις πρόσθεσης και πολλαπλασιασμού) συνδιασμό επιμέρος γκαουσιανών διαδικασιών (δηλαδή των $w_l\bar{\phi}_l(t)$, που με τη σειρά τους είναι γκαουσιανές διαδικασίες αφού πρόκειται για τις προβολές γκαουσιανών διαδικασιών ($w_l, \bar{\phi}_l(t)$)), όπως βλέπουμε και από τους τύπους της αναφοράς, συνεπάγεται ότι και το $\bar{w}(t)$ πρόκειται για μία Gaussian διαδικασία. Ως γνστών οι μέση τιμή κάθε Gaussian διαδικασίας είναι 0 και η διασπορά της $\sqrt{\frac{N_0}{2}}$

3.2 Τι παριστάνει η παραπάνω διαδικασία;

Η παραπάνω διαδικασία παριστάνει τη διαφορά μεταξύ της αρχικής διαδικασίας θορύβου $w(t)$ και της προβολής της αρχικής διαδικασίας στις συναρτήσεις βάσης $\phi_l(t)$

3.3 Τα w_l πρόκειται για Gaussian τυχαίες μεταβλητές - αναγκαίες στατιστικές

Τα w_l όπως και αποδείξαμε και στο 3.1 έτσι και εδώ για τον ίδιο λόγο αποτελούν γκαουσιανές τυχαίες μεταβλητές αφού γνωρίζουμε ότι ο γραμμικός συνδιασμός γκαουσιανών τυχαίων μεταβλητών (οι προβολές γκαουσιανών τυχαίων μεταβλητών είναι και αυτές γκαουσιανές τυχαίες μεταβλητές) έχει σαν αποτέλεσμα επίσης γκαουσιανές τυχαίες μεταβλητές. Οι μέσες τιμές τους είναι

$$E[w_l] = \int_0^T E[w(t)]\psi_l(t)dt = 0 \tag{7}$$

για όλα τα l.
και οι διακυμάνσεις τους $\frac{N_0}{2}$.

3.4 Ετεροσυσχετίσεις των w_l

Οι συμμεταβολές των συνιστωσών θορύβου είναι

$$\begin{aligned} E[w_l w_m] &= \int_0^T \int_0^T E[w(t)w(\tau)]\psi_l(t)\psi_m(\tau)dtd\tau && (8)\\ &= \int_0^T \int_0^T \frac{N_0}{2}\delta(t-\tau)\psi_l(t)\psi_m(t)dtd\tau && (9)\\ &= \frac{N_0}{2}\int_0^T \psi_l(t)\psi_m(t)dt && (10)\\ &= \frac{N_0}{2}\delta_{ml} && (11) \end{aligned}$$

όπου $\delta_{ml} = 1$ όταν $m = l$ και μηδέν σε αντίθετη περίπτωση. Συνεπώς οι 2 συνιστώσες θορύβου w_l είναι ασυσχέτιστες Gaussian τυχαίες μεταβλητές μηδενικής μέσης τιμής με την ίδια διακύμανση $\sigma_n^2 = \dfrac{N_0}{2}$ και επομένως

$$f(w) = \Pi_{i=0}^{1} f(w_i) = \frac{1}{\pi N_0} e^{-\Sigma_{i=0}^{1}\frac{n_i^2}{N_0}} \tag{12}$$

3.5 Χαρακτηρισμός των r_l

Από την προηγούμενη ανάπτυξη προκύπτει ότι υπό την προϋπόθεση ότι μεταδόθηκε το m-οστό σήμα, οι έξοδοι των συσχετιστών r_l είναι γκαουσιανές τυχαίες μεταβλητές με μέση τιμή

$$E[r_l] = E[s_{ml} + w_l] = s_{ml} \tag{13}$$

και με την ίδια διακύμανση

$$\sigma_r^2 = \sigma_n^2 = N_0/2 \tag{14}$$

3.6 Υπολογισμός της δεσμευμένης από κοινού PDF

Γνωρίζουμε ότι οι συνιστώσες θορύβου είναι ασυσχέτιστες Gaussian τυχαίες μεταβλητές, είναι στατιστικά ανεξάρτητες. Συνεπώς οι έξοδοι των συσχετίσεων είναι στατιστικά ανεξάρτητες Gaussian μεταβλητές. Επομένως, οι υποσυνθήκη συναρτήσεις πυκνότητας πιθανότητας PDFs των τυχαίων μεταβλητών $(r_0, r_1) = r$, δεδομένου ότι μεταδόθηκε το σήμα είναι απλά

$$f(r|s_m) = \Pi_{l=0}^{1} f(r_l|s_{ml}), m = 1, 2, ..., M \tag{15}$$

όπου

$$f(r|s_m) = \frac{1}{\sqrt{\pi N_0}} e^{\frac{-(r_l - s_{ml})^2}{N_0}}, m = 1, 2, ..., M \tag{16}$$

Αντικαθηστώντας τώρα την 16 εξίσωση στην 15, λαμβάνουμε τη συνδιασμένη υποσυνθήκη συνάρτηση πυκότητα πιθανότητας

$$f(r|s_m) = \frac{1}{(\pi N_0)^{\frac{N}{2}}} exp[-\Sigma_{l=0}^{1} \frac{(r_l - s_{ml})^2}{N_0}], m = 1, 2, ..., M \tag{17}$$

δηλαδή

$$f(r|s_m) = \frac{1}{(\pi N_0)^{\frac{N}{2}}} exp[-\frac{\|r_l - s_{ml}\|^2}{N_0}], m = 1, 2, ..., M \tag{18}$$

3.7 Στατιστική ανεξαρτησία των $\bar{w}(t)$ και r_0, r_1

Γνωρίζουμε ότι η διαδικασία $\bar{w}(t)$ είναι ασυσχέτιστη προς τις δύο εξόδους r_l, αφού

$$E[\bar{w}(t)r_l] = E[\bar{w}(t)]s_{ml} + E[\bar{w}(t)w_l] = E[\bar{w}(t)w_l] \tag{19}$$

$$= E[(w(t) - \Sigma_{j=0}^{1} w_j \psi_j(t))w_l] \tag{20}$$

$$= \int_T^0 E[w(t)w(\tau)]\psi_l(\tau)d\tau - \Sigma_{j=0}^{1} E[n_j n_l]\psi_j(t) \tag{21}$$

$$= \frac{N_0}{2}\psi_l(t) - \frac{N_0}{2}\psi_l(t) = 0 \tag{22}$$

Επειδή η διαδικασία $\bar{w}(t)$ και οι τυχαίες μεταβλητές r_l είναι Gaussian και ασυσχέτιστες, είναι και **στατιστικά ανεξάρτητες**.

4 Εργαλεία ανάπτυξης

Η συγγραφή και ανάπτυξη του κώδικα έγινε σε GNU/Linux, Debian Squeze .
Χρησιμοποιήθηκαν:

- **Matlab R2009a**
- **LaTeX** για την συγγραφή της αναφοράς.