ΨΗΦΙΑΚΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ

2η Εργαστηριακή Άσκηση (προαιρετική)

Ανδρονίκη Πορτοκάλογλου
1067539
4ο Έτος

Περιεχόμενα

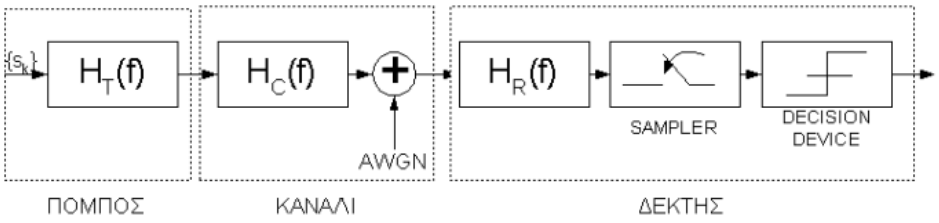
Εισαγωγή

Στη συγκεκριμένη εργαστηριακή εργασία ζητείται η υλοποίηση ενός M-PAM βασικής ζώνης. Στην αναπαράσταση βασικής ζώνης του PAM τα σύμβολα αναπαρίστανται πάνω στον άξονα των πραγματικών αριθμών για να ορισθεί το κάθε σύμβολο χρησιμοποιείται η εξίσωση

$$s_m = (2m - 1 - M) * Es,\ m = 0, ..., M, 0 < t \leq T$$

όπου στην περίπτωση μας η ενέργεια συμβόλου Es είναι 1.

Τα σύμβολα μεταδίδονται με το παρακάτω επικοινωνιακό σύστημα.



Επειδή το κανάλι είναι άγνωστο ,τα φίλτρα πομπού και δέκτη υλοποιούνται ως φίλτρα τετραγωνικής ρίζας ανυψωμένου συνημίτονου με συντελεστή επέκτασης 0.3. Για λογούς υλοποίησης τα φίλτρα αυτά δεν έχουν άπειρη χρονική έκταση ,επομένως γίνεται αποκοπή σε ένα περιορισμένο αριθμό περιόδων σηματοδοσίας 6Ts.Ακόμα , για λόγους καλύτερης ψηφιακής αναπαράστασης ,τα φίλτρα αυτά εφαρμόζονται σε μια υπερδειγματοληψία της ακολουθίας συμβόλων εισόδου. Αυτό όμως προϋποθέτει ότι θα πρέπει και το φίλτρο να είναι υπερδειγματοληπτημένο, έστω κατά 4.

Στα πειράματα που βρίσκονται παρακάτω χρησιμοποιούνται τρία διαφορετικά κανάλια. Αρχικά ,έχουμε το ιδανικό οπού το φίλτρο δέκτη λαμβάνει ως είσοδο την έξοδο του φίλτρου πομπού + θόρυβο. Τα αλλά δυο κανάλια είναι μη ιδανικά και έχουν το καθένα κρουστική απόκριση:

$$h(-5:5) = [0.04 - 0.050.07 - 0.21 - 0.50.720.3600.210.030.07]$$

$$h(-1:1) = [0.4070.8150.407].$$

Στην έξοδο κάθε καναλιού, και πριν την είσοδο στο φίλτρο δέκτη, προστίθεται θόρυβος στην ακολουθία συμβόλων. Ο θόρυβος αυτός είναι λευκός Gaussian θόρυβος, μηδενικής μέσης τιμής. Η ισχύς του καθορίζεται από το SNR το οποίο ορίζεται από την παρακάτω εξίσωση. SNR[db]=10 log10 (Ps/PN)

Η ακολουθία των συμβόλων στην έξοδο του φίλτρου δέκτη υποδειγματοληπτείται στις κατάλληλες χρονικές στιγμές και τα δείγματα που προκύπτουν περνούν από κάποια διάταξη απόφασης οπότε και αποφασίζεται ποια ήταν τα αντίστοιχα σύμβολα που στάλθηκαν. Στη συγκεκριμένη περίπτωση χρησιμοποιείται το κριτήριο ML κατά το οποίο, το σύμβολο που στάλθηκε είναι αυτό που έχει την ελάχιστη Ευκλείδεια απόσταση από το ληφθέν διάνυσμα η οποία ορίζεται ως:

$$D(r, sm) = \sum_{k=1}^{N} (r_k - s_{mk})^2$$

όπου r το ληφθέν διάνυσμα και sm σύμβολα του αστερισμού που χρησιμοποιήθηκε στη διαμόρφωση.

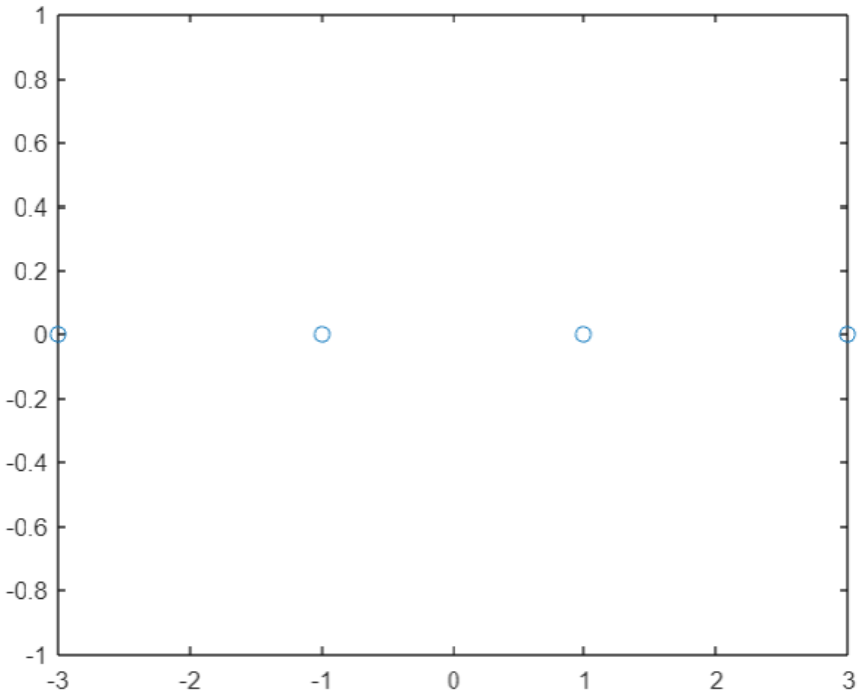
Ζητούμενα

1
Στο ερώτημα αυτό ζητείται να γίνει μια ψευδοτυχαία δυαδική ακολουθία.Στη matlab αυτό υλοποιείται με την συνάρτηση randsrc .

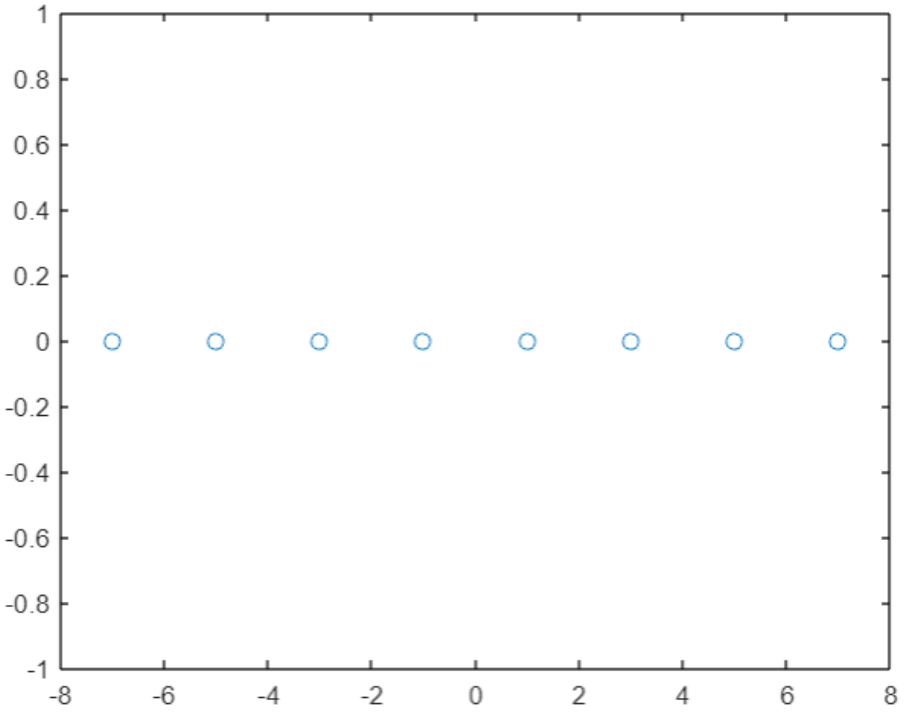
```
Lb=1000
Signal=randsrc(Lb, 1, [0,1])
```

2.
Το σύστημα έχει υλοποιηθεί μέσω μια συνάρτησης .Η συνάρτηση αυτή ονομάζεται MPAM και για την χρησιμοποιήσει κάποιος πρέπει να ορίσει τις εισόδους της. Συγκεκριμένα, ορίζεται το σήμα (Signal) , ο συντελεστής επέκτασης ,η υπερδειγματοληψία (up), η περίοδος συμβόλου T ,το snr ,τα σύμβολα πηγής και το type το οποίο ανάλογα με ποιον αριθμό από το 1 έως το 3(channel_type) δίνεται διαλέγεται το κανάλι .Ως έξοδο δίνεται η ακολουθία που λαμβάνει ο δέκτης. Στη συναρτηση αρχικά γίνεται η υπερδειγματοληψία με την χρήση της έτοιμης συνάρτησης upsample ,έπειτα βάζουμε το φίλτρο και κάνουμε κανονικοποίηση ώστε η νόρμα να είναι 1. Έπειτα μέσω if ανάλογα με την τιμή του channel_type δημιουργείται η έξοδος του καναλιού μαζί με την αντίστοιχη κρουστική απόκριση (αν δεν μιλάμε για το ιδανικό κανάλι). Αφού γίνει αυτό προστίθεται λευκός gaussian θόρυβος στη έξοδο του καναλιού. Επειδή όμως έχουμε Fir φίλτρο υπολογίζεται και το delay. Στη συνέχεια υλοποιείται η διάταξη απόφασης όπου γίνεται με το κριτήριο ML.

$$4 - PAM$$



$8-PAM$



3,4.
Για να μπορέσουμε να έχουμε αποτελέσματα από τον κώδικα πρέπει να τον τρέξουμε για μεγάλο Lb κάτι το οποίο απαιτεί αρκετό χρόνο.Δυστηχώς ,το hardware που διαθέτω δεν ήταν ικανό να μου δώσει τα αποτελέσματα πριν την λήξη της προθεσμίας.Ωστόσο ο κώδικας παρατίθεται στο τελευταίο κομμάτι της αναφοράς.

Ο κώδικας υπολογιζει το SER το οποίο από την στιγμή που ένα σύμβολο αποτελείται απο M bits συντελεί στον υπολογισμό του BER .Συγκεκριμένα, το BER προκύπτει διαιρώντας το SER με τον αριθμό των bits per symbol δηλαδή το M.

Παράρτημα με κώδικα

```
%PAM
Es=1
T=1
roll_off_factor=0.3
up=4
%plot(s_m,[0,0,0,0],'o')
M=4
m=1:M
s_m4=(2*m-1-M)*Es

M=8
m=1:M
s_m8=(2*m-1-M)*Es

Lb=1000
Signal=randsrc(Lb, 1, [0,1])
%Mapper4
M=4
temp = mod(Lb, log2(M))
n = Signal(1 : (Lb - temp), :)
rn = reshape(n, log2(M), (Lb - temp) / log2(M))

Signal_dec4 = bin2dec(num2str(rn'))
Signal4=zeros(length(Signal_dec4),1)

for i=1:length(Signal_dec4)
    Signal4(i,1)=s_m4(Signal_dec4(i)+1)
end


%Mapper8
M=8
temp = mod(Lb, log2(M))
n = Signal(1 : (Lb - temp), :)
rn = reshape(n, log2(M), (Lb - temp) / log2(M))

Signal_dec8 = bin2dec(num2str(rn'))
Signal8=zeros(length(Signal_dec8),1)

for i=1:length(Signal_dec8)
    Signal8(i,1)=s_m8(Signal_dec8(i)+1)
end

%-----------------
SER41=[]
SER42=[]
SER43=[]


SER81=[]
SER82=[]
SER83=[]
SNR=[0:2:30]

for i=1:length(SNR)
        [ output ] = MPAM( Signal4, 4, 0.3, T, '1', SNR(i), s_m4 )
        SER41(i,1) = sum(xor(output,Signal4)) / length(Signal4)
end

```

```
60
61 for i=1:length(SNR)
62         [ output ] = MPAM( Signal4, 4, 0.3, T, '2', SNR(i), s_m4 )
63         SER42(i,1) = sum(xor(output,Signal4)) / length(Signal4)
64 end
65
66 for i=1:length(SNR)
67         [ output ] = MPAM( Signal4, 4, 0.3, T, '3', SNR(i), s_m4 )
68         SER43(i,1) = sum(xor(output,Signal4)) / length(Signal4)
69 end
70
71 for i=1:length(SNR)
72         [ output ] = MPAM( Signal8, 4, 0.3, T, '1', SNR(i), s_m8 )
73         SER81(i,1) = sum(xor(output,Signal8)) / length(Signal8)
74 end
75
76
77 for i=1:length(SNR)
78         [ output ] = MPAM( Signal8, 4, 0.3, T, '2', SNR(i), s_m8 )
79         SER82(i,1) = sum(xor(output,Signal8)) / length(Signal8)
80 end
81
82 for i=1:length(SNR)
83         [ output ] = MPAM( Signal8, 4, 0.3, T, '3', SNR(i), s_m8 )
84         SER83(i,1) = sum(xor(output,Signal8)) / length(Signal8)
85 end
86
87
88 %-----------------
89
90 function [ output ] = MPAM( Signal, up, roll_off_factor, T, channel_type, SNR,
       symbols )
91
92 %Upsampling
93 Signal_up = upsample(Signal,up)
94 Signal_up = Signal_up(1:(length(Signal_up)-(up-1)))
95
96 %Transmission filter
97 t_filter = rcosfir(roll_off_factor  , [-T/2 , T/2] , up , 1 , 'sqrt');
98 t_filter = t_filter./norm(t_filter);
99
100 %Output of transmitter
101 in = conv(t_filter, Signal_up);
102
103
104 %Channel output
105 if  (channel_type=='1')
106     channelled_output=in
107 elseif (channel_type=='2')
108
109     h= [ 0.04 -0.05 0.07 -0.21 -0.5 0.72 0.36 0 0.21 0.03 0.07]
110     h_up = upsample(h,up)
111     h_up = h_up(1:(length(h_up)-(up-1)))
112
113     channelled_output = conv(h_up,in)
114 elseif (channel_type=='3')
115
116     h = [0.407 0.815 0.407]
117     h_up = upsample(h,up);
118     h_up = h_up(1:(length(h_up)-(up-1)))
119
120     channelled_output = conv(h_up,in)
```

```
end
%White noise addition
Ps = sum(abs(channelled_output).^2) / length(channelled_output)
Pn = Ps/(10^(SNR/10))


sigma = sqrt(Pn)
noise = sigma *(randn(length(channelled_output),1))

channel_output=channelled_output+noise



%Reciever filter
r_filter = t_filter

%Filter output
received = conv(r_filter,channel_output)

%FIR DELAY
N_filter = length(r_filter)
N_signal = length(Signal_up)

if  (channel_type=='1')
    delay = 2*floor(N_filter/2)
else
    N_channel=length(h_up)
    delay = 2*floor(N_filter/2) + floor(N_channel/2)
end


s_delayed = received(delay+1:(end-delay))
Signal_down = downsample(s_delayed,up)


N_signal_down = length(Signal_down)
symbols_length = length(symbols)
%Output matrix initialization
output = zeros(N_signal_down,1)
%Distance matrix initialization
D = zeros(symbols_length,1)
% Calculating symbol distance and picking the symbol with min dist.
for i=1:N_signal_down
    % Symbol distance
    for j=1:symbols_length
        D(j) = (Signal_down(i) - symbols(j))^2
    end
    % Sorting D
    [D,I] = sort(D,'ascend')
    % Picking the symbol
    output(i) = symbols(I(1))
end
end
```