ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΨΗΦΙΑΚΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ

1Η ΑΣΚΗΣΗ

Ιανουάριος 2017

Διδάσκων: Κ. Μπερμπερίδης

Φώτιος Διονυσόπουλος ΑΜ:5753
Email: dionys@ceid.upatras.gr

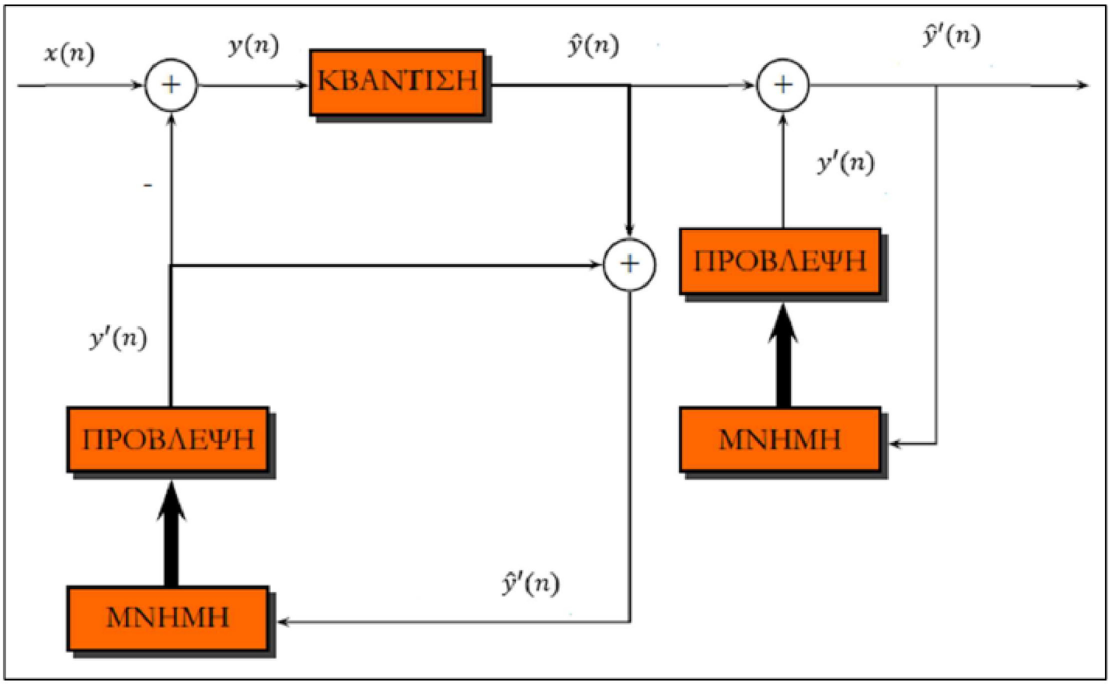
Περιεχόμενα

Μέρος Α

Κωδικοποίηση/αποκωδικοποίηση DPCM

Η κωδικοποίηση DPCM είναι μια γενίκευση της κωδικοποίησης Δέλτα όπου το σήμα που κβαντίζεται και αποστέλλεται στο δέκτη. Προκειμένου να κωδικοποιήσουμε την τιμή του τρέχοντος δείγματος υπολογίζουμε αρχικά μια πρόβλεψη για την τιμή του βασιζόμενοι σε κωδικοποιημένες τιμές προηγούμενων δειγμάτων.

Κωδικοποιητής και Αποκωδικοποιητής DPCM:



Σκοπός μας είναι να ελαχιστοποιήσουμε τη διασπορά του σήματος σφάλματος:
$y(n) = x(n) - y'(n)$

Αρχικά, η πρόβλεψη y'(n) είναι ίση με μηδέν αφού δεν έχουμε κάποια πρόβλεψη για το σήμα.

Υπολογισμός του Φίλτρου Πρόβλεψης

Σε ένα σύστημα DPCM η πρόβλεψη του δείγματος δίνεται ως ένας γραμμικός συνδυασμός των p προηγούμενων τιμών. Ο υπολογισμός του φίλτρου γίνεται με βάση τους τύπους που δίνονται στην εκφώνηση.
Ο υπολογισμός του διανύσματος α γίνεται μόνο μια φορά.

Ομοιόμορφος Κβαντιστής

Υλοποιήθηκε ένας κβαντιστής N δυαδικών ψηφίων, 2^N επιπέδων ο οποίος κβαντίζει το σφάλμα πρόβλεψης το οποίο έχει μικρότερη δυναμική περιοχή σε σχέση με το σήμα εισόδου. Δηλαδή, μετατρέπει το αναλογικό σήμα, σε ψηφιακό.

Χαρακτηριστικά κβαντιστή:

$$[\hat{y}(n)] \;=\; \boldsymbol{my\_quantizer}(\boldsymbol{y}(\boldsymbol{n}), \boldsymbol{N}, \boldsymbol{min\_value}, \boldsymbol{max\_value});$$

$y(n)$: το τρέχον δείγμα του σφάλματος πρόβλεψης ως είσοδος του κβαντιστή

N: ο αριθμός των δυαδικών ψηφίων που θα χρησιμοποιηθούν

max_value: η μέγιστη αποδεκτή τιμή του σφάλματος πρόβλεψης

min_value: η ελάχιστη αποδεκτή τιμή του σφάλματος πρόβλεψης

$\hat{y}(n)$: το κβαντισμένο δείγμα του τρέχοντος δείγματος του σφάλματος πρόβλεψης

Για τον υπολογισμό του διανύσματος α χρησιμοποιήθηκε ο κβαντιστής με min_value = -2, max_value = 2 και N = 8 bits.

Αποτελέσματα- Μετρήσεις

Ερώτημα 1

Υλοποίηση του κωδικοποιητή DPCM:

```
function [ ] = main ( )
   load source.mat
   N = length(x);


   %Number of bits we are going to use
   NoB = 1
   p = 4
   %Allocating memory
   y = zeros(N,1);
   y_hat_t = zeros(N,1);
   y_hat = zeros(N,1);
   y_t = zeros(N,1);
   r = zeros(p, 1);
   R = zeros(p, p);


   %Calculating r
   for i = 1:p
     r(i) = 1/(N - p) * sum(x(p+1:N).*x(p+1-i:N-i));
   end

   %Calculating R
   for i = 1:p
     for j = 1:p
       R(i, j) =  1/(N - p + 1) * sum(x(p+1-j:N-j).*x(p+1-i:N-i));
     end
   end

   a = R\r;
   aq = my_quantizer(a, 8, -2, 2);

   y(1:p) = x (1:p);
   y_hat(1:p) = my_quantizer(y(1:p), NoB, -3, 3);
   y_hat_t(1:p) = y_t(1:p) + y_hat(1:p);


   for i = p+1:N
      y_t(i) = sum(aq(1:p).*y_hat_t(i-1:-1:i-p));
      y(i) = x(i) - y_t(i);
      y_hat(i) = my_quantizer(y(i), NoB, -3, 3);
      y_hat_t(i) = y_t(i) + y_hat(i);
   end
   y_hat_t
   %Calculating E_Y_2
   E_y_2 = mean((x - y_t).^2)

end
```

Ο κβαντιστής:

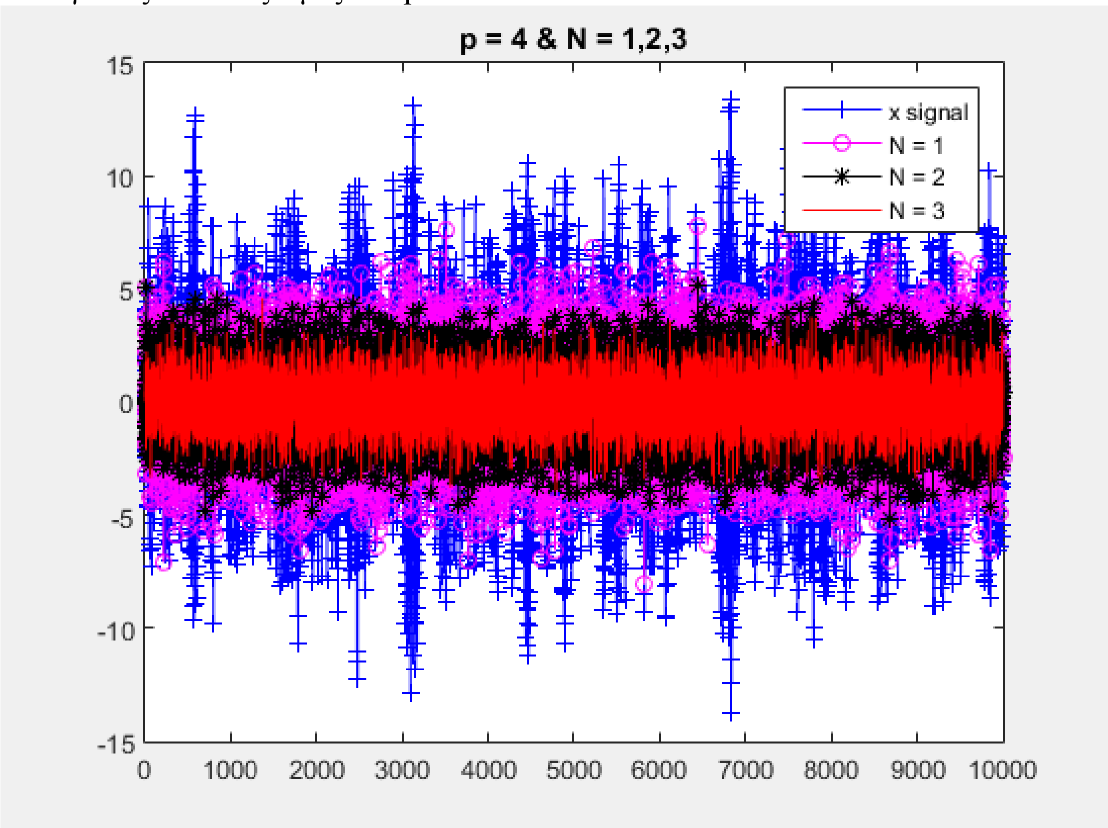
```
function [ x_q ] = my_quantizer( x, N, min_value, max_value )

         if (x > max_value)
                  x = max_value;
         elseif (x < min_value)
                  x = min_value;
         end

         levels = 2^N;

         D = ( max_value - min_value )/levels;
         x_q = D * floor((x/D) + 0.5);

end %End of Function
```

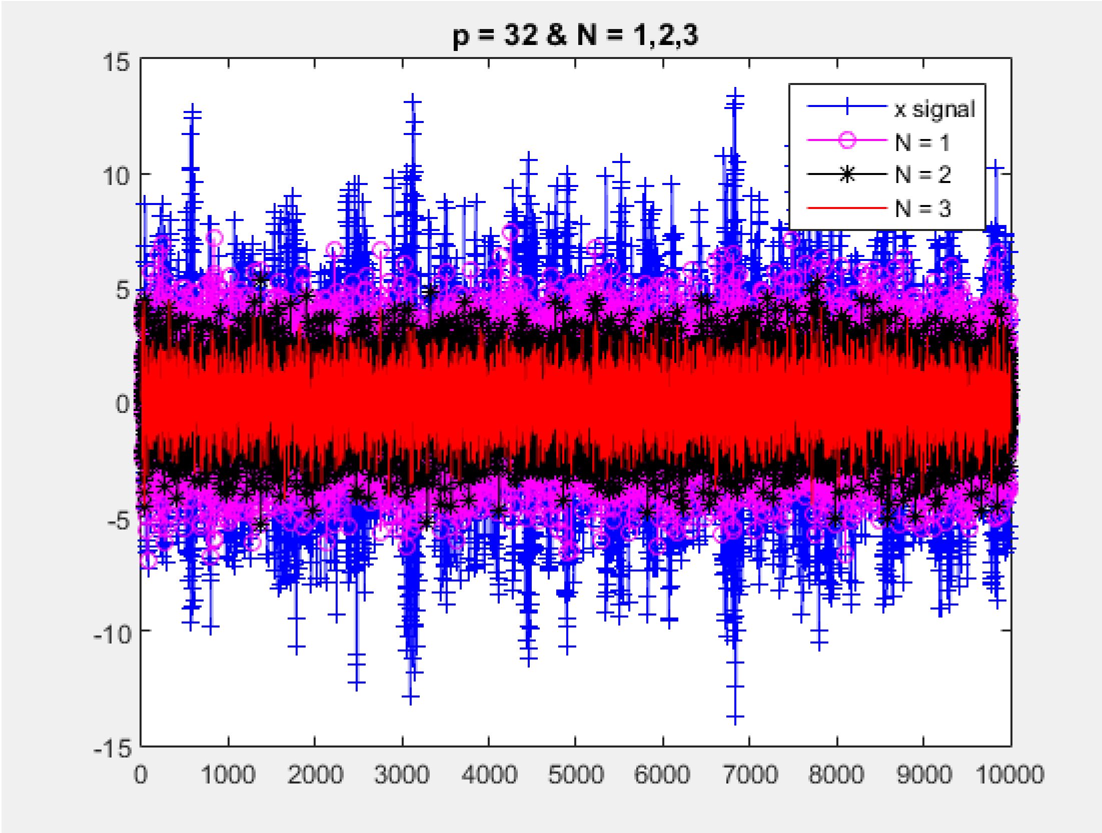
Υλοποίηση του αποκωδικοποιητή DPCM:

```
function [ ] = main2 ( aq , y_hat)
   N = length ( y_hat ) ;
    p = length (aq) ;

   %Allocating memory
   y_hat = zeros(N,1);
   y_hat_t( 1 :p ) = y_hat(1:p)


   for i = p+1:N
      y_t(i) = aq(1:p).*y_hat_t(i-1:-1:i-p);
      y_hat_t(i) = y_t(i) + y_hat(i);
   end
end
```

Ερώτημα 2

Σκοπός του ερωτήματος ήταν να επιλεγούν δύο τιμές για το p (4 και 32) και να σχεδιαστούν τα διαγράμματα πρόβλεψης y σε σύγκριση με το αρχικό σήμα x. Καθώς επίσης και να αποτυπωθεί το μέσο τετραγωνικό σφάλμα πρόβλεψης ως προς το N για τις δοθείσες τιμές του p.

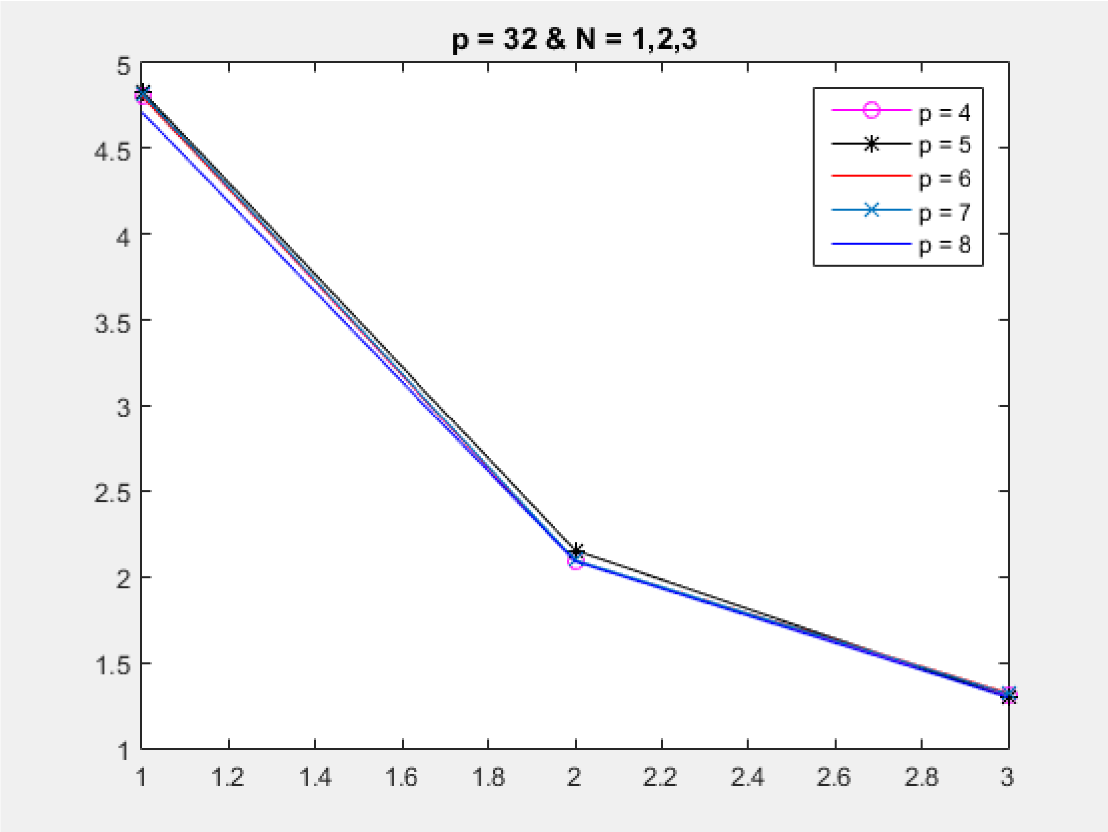




Γενικά, παρατηρείται πως για μεγαλύτερο p υπάρχει μια μικρή διαφορά στο σφάλμα y, η μεγαλύτερη διαφορά φαίνεται για N = 1.
Πιο συγκεκριμένα, σε κάθε σχήμα ξεχωριστά, παρατηρείται πως για μεγαλύτερο N μειώνεται η δυναμική περιοχή του σφάλματος y.

Ερώτημα 3

	N=1	N=2	N=3
P=4	4.8069	2.0907	1.3129
P=5	4.8362	2.1539	1.2979
P=6	4.8069	2.0956	1.3221
P=7	4.8198	2.0967	1.3169
P=8	4.7128	2.0894	1.2975



Στο παραπάνω σχήμα παρατηρούμε ότι για μεγαλύτερο N, το σφάλμα είναι μικρότερο, λογικό, διότι αυξάνεται η ακρίβεια.
Όσο αυξάνονται τα p, μειώνεται και το σφάλμα κατά μια μικρή τιμή.

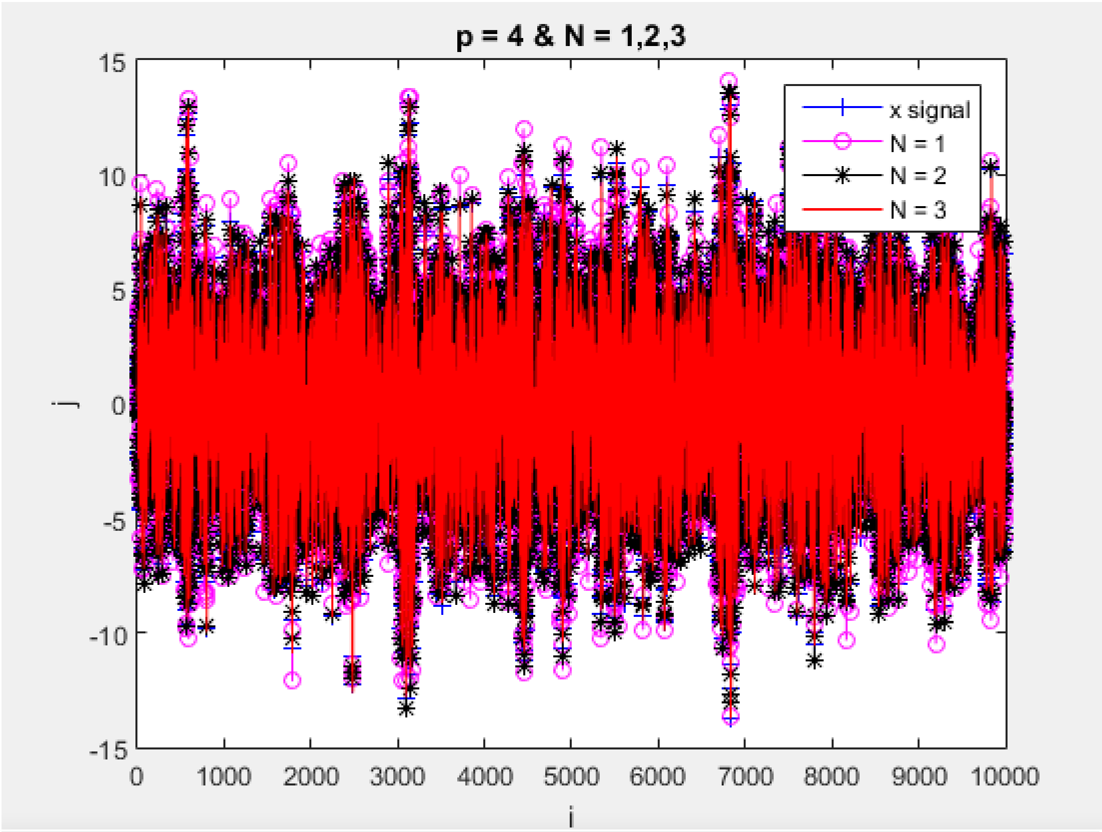
Οι συντελεστές α :

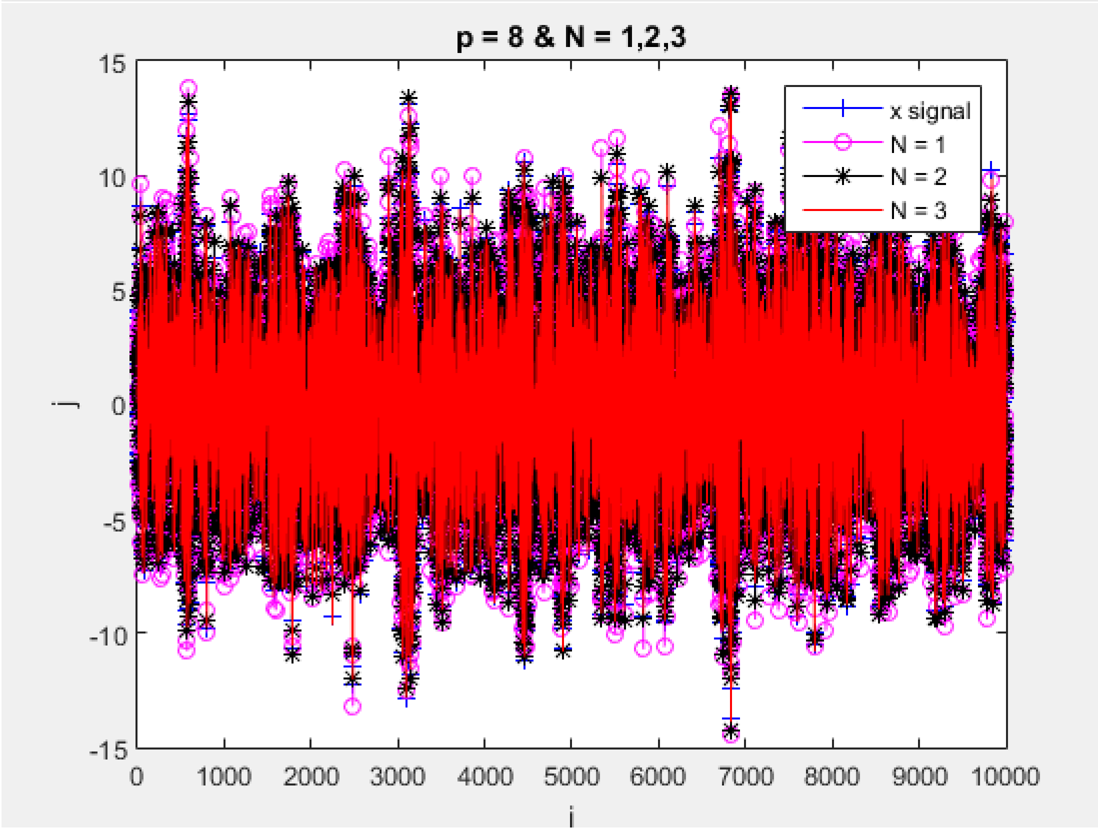
p = 4	p = 5	p = 6	p = 7	p = 8
1.3889	1.3883	1.3882	1.3881	1.3880
-1.5214	-1.5186	-1.5193	-1.5195	-1.5193
1.2124	1.2088	1.2118	1.2110	1.2108
-0.3017	-0.2984	-0.3023	-0.2975	-0.2998
0	-0.0025	0.0011	-0.0051	0.0050
0	0	-0.0026	0.0033	-0.0095
0	0	0	-0.0045	0.0073
0	0	0	0	-0.0085

Σχετικά με τους συντελεστές, παρατηρείται ότι όσο με μεγαλύτερο είναι το p, μειώνονται οι μηδενικές τιμές του διανύσματος ενώ οι ήδη υπάρχουσες τιμές διαφοροποιούνται ελάχιστα.

Ερώτημα 4

Στο 4ο και τελευταίο ερώτημα του Μέρους Α, ζητούμενο ήταν να απεικονιστεί το αρχικό και το ανακατασκευασμένο σήμα για p = 4, 8 και N = 1, 2, 3 :





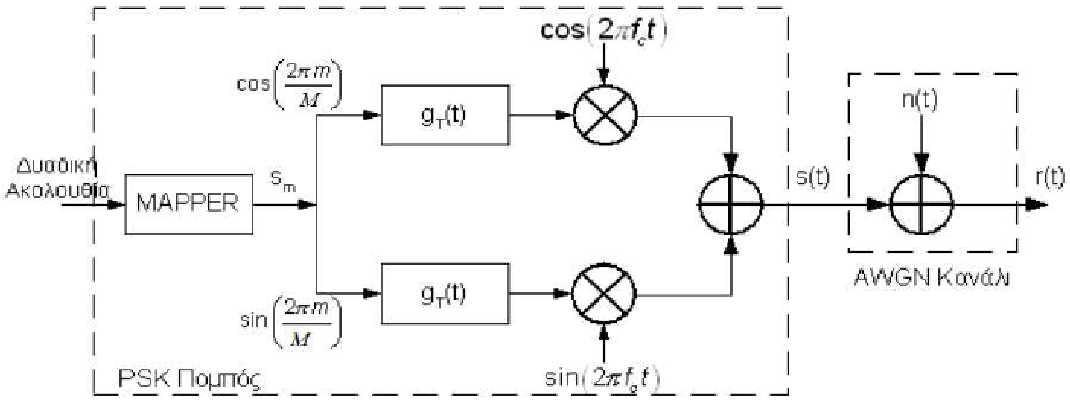
Κάνοντας μεγέθυνση, παρατηρήθηκε ότι οι διαφορές μικραίνουν όσο μεγαλώνει το N, ενώ για τις δύο τιμές του p η διαφορά είναι πολύ μικρή διότι τοι τιμές έχουν μικρή διαφορά.

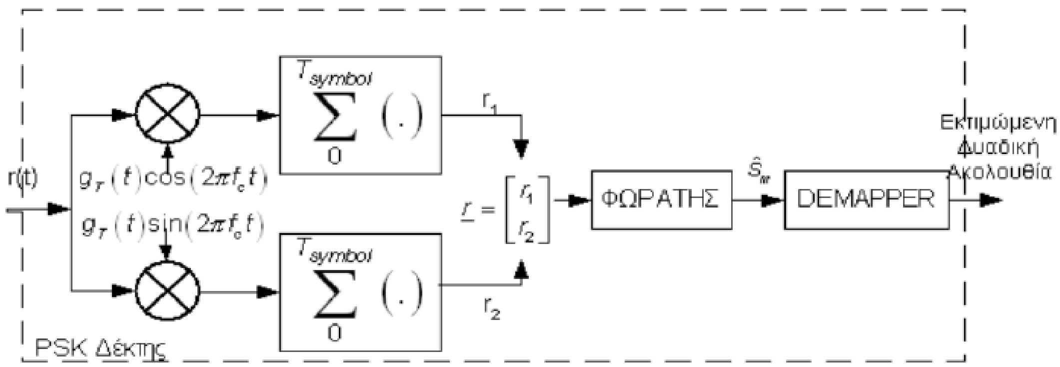
Μέρος B

Εισαγωγή – Περιγραφή Συστήματος PSK

Στο 2ο μέρος της εργασίας ζητήθηκε η μελέτη της απόδοσης της διαμόρφωσης M-PSK. Πιο συγκεκριμένα, η μελέτη έγινε πάνω σε μετρήσεις πιθανότητας σφάλματος οι οποίες πραγματοποιήθηκαν σε ένα ομόδυνο ζωνοπερατό σύστημα με ορθογώνιο παλμό.

Σύστημα PSK:





Στο παραπάνω σχήμα απεικονίζεται ένα κύκλωμα PSK, πιο συγκεκριμένα, αποτελείται από τον πομπό και δέκτη καθώς και το κανάλι agwn.
Τα module του πομπού είναι ο Mapper και ο Modulator ενώ του δέκτη ο Detector, Demodulator και Demapper. Επίσης, το κανάλι μετάδοσης αποτελείται από τον Noise Generator.

Αρχικά, δημιουργείται η τυχαία δυαδική ακολουθία η οποία περνά στον Mapper, ο οποίος κάνει την αντιστοίχιση της σε σύμβολα. Ο Mapper κάνει κωδικοποίησή κατά Grey με σκοπό τη δημιουργία γειτονικά σύμβολα στον δισδιάστατο χώρο.
Στη συνέχεια, η έξοδος του Mapper περνά στον Modulator του οποίου η δουλειά είναι να διαμορφώνει την κάθε συνιστώσα, δηλαδή πολλαπλασιάζει με ορθογώνιο παλμό και τη διαμορφώνει γύρω από τη φέρουσα συχνότητα ώστε να προκύψει το επιθυμητό ζωνοπερατό σήμα. Προστίθενται θόρυβος μέσω του AWGN καναλιού και

έπειτα αποστέλλονται στον δέκτη.

Ο αποδιαμορφωτής του συστήματος M-PSK συσχετίζει (δηλαδή πολλαπλασιάζει και ολοκληρώνει-αθροίζει) το ληφθέν σήμα με τη φέρουσα και τον ορθογώνιο παλμό. Η συσχέτιση γίνεται στα χρονικά πλαίσια μιας περιόδου συμβόλου. Κατά την προσομοίωση υποθέτουμε ότι το M-PSK είναι ομόδυνο. Αυτό σημαίνει ότι ο δέκτης γνωρίζει τη φάση της φέρουσας και τα χρονικά πλαίσια κάθε συμβόλου, δηλαδή είναι πλήρως συγχρονισμένος με τον πομπό. Ο αποδιαμορφωτής συσχετίζει το ληφθέν σήμα με τις δύο συνιστώσες της φέρουσας, οπότε προκύπτουν δύο τιμές, δηλαδή ένα διάνυσμα r που αντιστοιχεί στη θέση του ληφθέντος σήματος πάνω στο επίπεδο του αστερισμού του M-PSK.

Ο φωρατής δέχεται ως είσοδο το διάνυσμα r και αποφασίζει σε ποιο σύμβολο βρίσκεται πιο κοντά. Το διάνυσμα S_m που θα έχει τη μικρότερη απόσταση από το r, αντιστοιχεί και στο σύμβολο που στάλθηκε.

Τέλος, ο Demapper κάνει την αντιστοίχιση. Δηλαδή, μετατρέπει τον κωδικό συμβόλου στην εκτιμώμενη δυαδική ακολουθία.

Αποτελέσματα- Μετρήσεις

Ερώτημα 1

Υλοποίηση του συστήματος M-PSK:

```
function [ s_m ] = mapper(x, M, Gray_)

   sizex = length(x);
   b = ceil(log2(M));
   r = mod(sizex, b);
   % Is gray
   if (Gray_ == 1)
      s_m = bin2gray(s_m, 'psk', M);
   end

      % Is not gray
   if (r ~= 0)
     x(sizex-r+1:sizex-r+b) = [zeros(b-r, 1); x(sizex-r+1:sizex)];
   end
   s_m = bi2de(reshape(x, b, [])', 2, 'left-msb');

end
```

```
function [ st ] = modulator(s_m, M, T_symbol, deigmaT, tc, es)

   N = length(s_m);
   samples = T_symbol/deigmaT;
   st = zeros(N, samples);

   res = sqrt(es) * sqrt(2 * es/T_symbol);

   for i = 1:N
      a = s_m(i)/M;
      for t = 1:samples
         st(i, t) = res * cos(2*pi*(a - t/tc));
      end
   end
end
```

```
function [ r_t ] = noise(s_t, M, Es, SNR)

    e_b = Es/ceil(log2(M));
    s = sqrt(e_b/(2 * 10^(SNR/10)));

    [symbols, samples] = size(s_t);

    noise = s * randn(symbols, samples);
    r_t = s_t + noise;
end
```

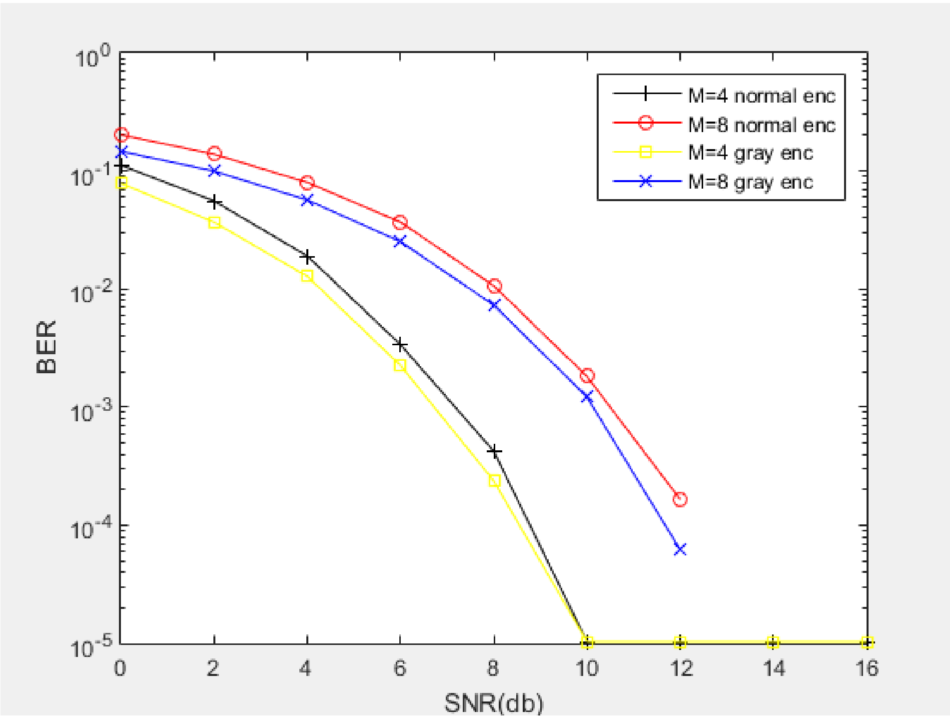
```
function [ x ] = demapper(smh, M, Gry)

    b = ceil(log2(M));

    % Decode from gray to dec
    if (Gr == 1)
        smh = gray2bin(smh, 'psk', M);
    end
    % Serialize dec numbers as a bit vector
    x = reshape(de2bi(smh, b, 2, 'left-msb')', [], 1);
end
```

```
function [ pos_r ] = demodulator(r_t, deigmaT, tc, es)

    [symbols, samples] = size(r_t);
    z = zeros(samples, 2);
    pos_r = zeros(symbols, 2);

    gt = sqrt(2 * es/deigmaT);

    for t = 1:samples
        z(t, 1) = gt * cos(2*pi*t/tc);
        z(t, 2) = gt * sin(2*pi*t/tc);
    end

    for i = 1:symbols
        pos_r(i,:) = [r_t(i,:)*z(:,1), r_t(i,:)*z(:,2)];
    end
end
```

```
function [ smh ] = foratis(r, M)

    [symbols, ~] = size(r);
    smh = zeros(symbols, 1);
    d = zeros(M, 1);
    s_m = zeros(M, 2);

    for m = 1:M
        s_m(m,:) = [cos(2*pi*m/M), sin(2*pi*m/M)];
    end

    for i = 1:symbols
        for m = 1:M
            d(m) = norm(r(i,:) - s_m(m,:));
        end
        [~, smh(i)] = min(d);
    end
    smh = mod(smh, M);
end
```

```
function [ ] = main()
SNR = 0:2:16;

% Eisodos
Lb=randsrc(1, 10000, [0 1; 0.5 0.5]);
x_v = randsrc(Lb, 1, [0 1]);

output = zeros(2, 9);
symbolo = 40;
 tc = 4;
deigma = 1;
E_s = 1;

for i = 0:1
for M = 4:4:8
    for snr = SNR

        s_m = mapper(x_v, M, i);
        s_t = modulator(s_m, M, symbolo, deigma, tc, E_s);
        rt = noise(s_t, M, E_s, snr);
        r = demodulator(rt, symbolo, tc, E_s);
        s_m_hat = foratis(r, M);
        x_hat = demapper(s_m_hat, M, i);

        % Bit error calc
        [~, output(M/4, snr/2+1)] = biterr(x_v, x_hat(1:Lb));
    end
end

end
```

Ερώτημα 2



Από την γραφική παράσταση, παρατηρείται ότι για M = 4 το BER μειώνεται πιο γρήγορα.
Επίσης, και για M = 4 και για M = 8, η ανοχή στα σφάλματα είναι πιο μεγάλη διότι στην κωδικοποίηση Grey τα σύμβολα διαφέρουν μόνο κατά ένα Bit.

Τέλος, στην περίπτωση όπου $M = 4$ με Grey, η γραφική σταματά να υφίσταται για $SNR > 8$, κάτι που σημαίνει ότι τα σφάλματα είναι πολύ μικρά και τείνουν στο μηδέν.