Modulatia QAM

Semnalele de intrare: $I(n), Q(n)$

Frecventa de esantionare: f_s

Frecventa purtatoare: f_p

Semnalul modulat QAM: $s(n)$

$$s(n) = \cos(2\pi \frac{f_p}{f_s} n) I(n) + \cos(2\pi \frac{f_p}{f_s} n + \frac{\pi}{2}) Q(n) =$$

$$= \cos(2\pi \frac{f_p}{f_s} n) I(n) - \sin(2\pi \frac{f_p}{f_s} n) Q(n)$$

Semnalele refacute (dupa demodulare, inainte de filtrarea trece jos): $r_I(n)$, $r_Q(n)$

$$r_I(n) = s(n) \cos(2\pi \frac{f_p}{f_s} n) =$$

$$= \cos(2\pi \frac{f_p}{f_s} n) \cos(2\pi \frac{f_p}{f_s} n) I(n) - \cos(2\pi \frac{f_p}{f_s} n) \sin(2\pi \frac{f_p}{f_s} n) Q(n)$$

$$r_Q(n) = s(n) \cos(2\pi \frac{f_p}{f_s} n + \frac{\pi}{2}) =$$

$$= -s(n) \sin(2\pi \frac{f_p}{f_s} n) =$$

$$= -\sin(2\pi \frac{f_p}{f_s} n) \cos(2\pi \frac{f_p}{f_s} n) I(n) + \sin(2\pi \frac{f_p}{f_s} n) \sin(2\pi \frac{f_p}{f_s} n) Q(n)$$

Relatii trigonometrice:

$$\cos(2\pi \frac{f_p}{f_s} n) \cos(2\pi \frac{f_p}{f_s} n) = \frac{1}{2} [1 + \cos(2\pi \frac{2f_p}{f_s} n)]$$

$$\sin(2\pi \frac{f_p}{f_s} n) \sin(2\pi \frac{f_p}{f_s} n) = \frac{1}{2} [1 - \cos(2\pi \frac{2f_p}{f_s} n)]$$

$$\cos(2\pi \frac{f_p}{f_s} n) \sin(2\pi \frac{f_p}{f_s} n) = \frac{1}{2} \sin(2\pi \frac{2f_p}{f_s} n)$$

Semnalele demodulate (inainte de filtrarea trece jos)

$$r_I(n) = \frac{1}{2} I(n) + c\,\mathrm{os}(2\pi \frac{2f_p}{f_s} n) I(n) - \frac{1}{2}\sin(2\pi \frac{2f_p}{f_s} n) Q(n)$$

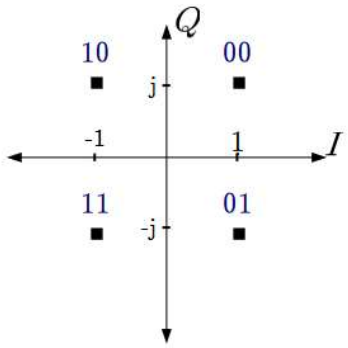
$$r_Q(n) = -\frac{1}{2}\sin(2\pi \frac{2f_p}{f_s} n) I(n) + \frac{1}{2} Q(n) - \frac{1}{2}\cos(2\pi \frac{2f_p}{f_s} n) Q(n)$$

Semnalele refacute (dupa filtrarea trece jos , cu frecventa de taiere f_p

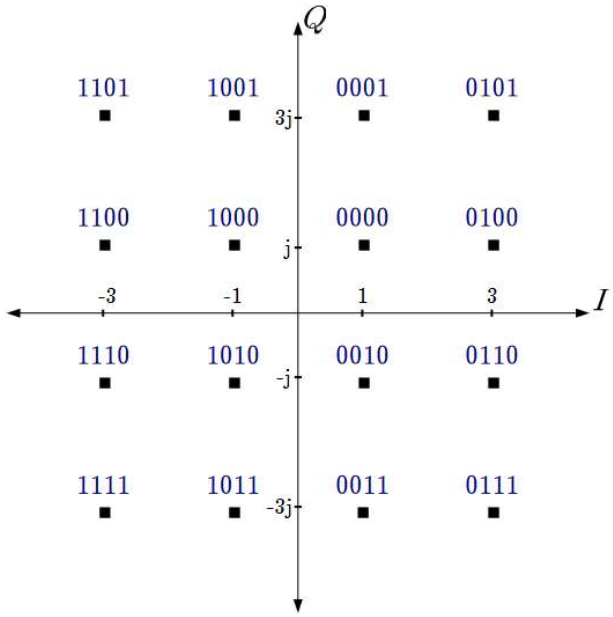
$$r^*_I(n) = \frac{1}{2} I(n)$$

$$r^*_Q(n) = \frac{1}{2} Q(n)$$

Constelatiile QAM



4-QAM



16-QAM

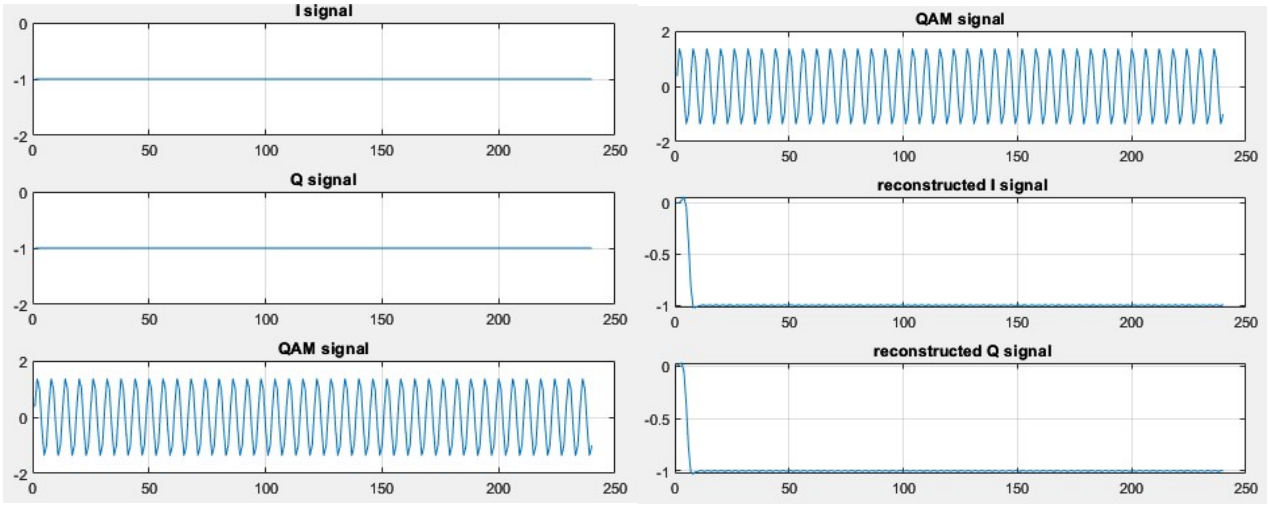
Rezultate

Frecventa purtatoare 8000 Hz; Frecventa de simbol 200 Hz; Frecventa de esantionare 48000 Hz

4-QAM

Sirul de biti : 1, 1

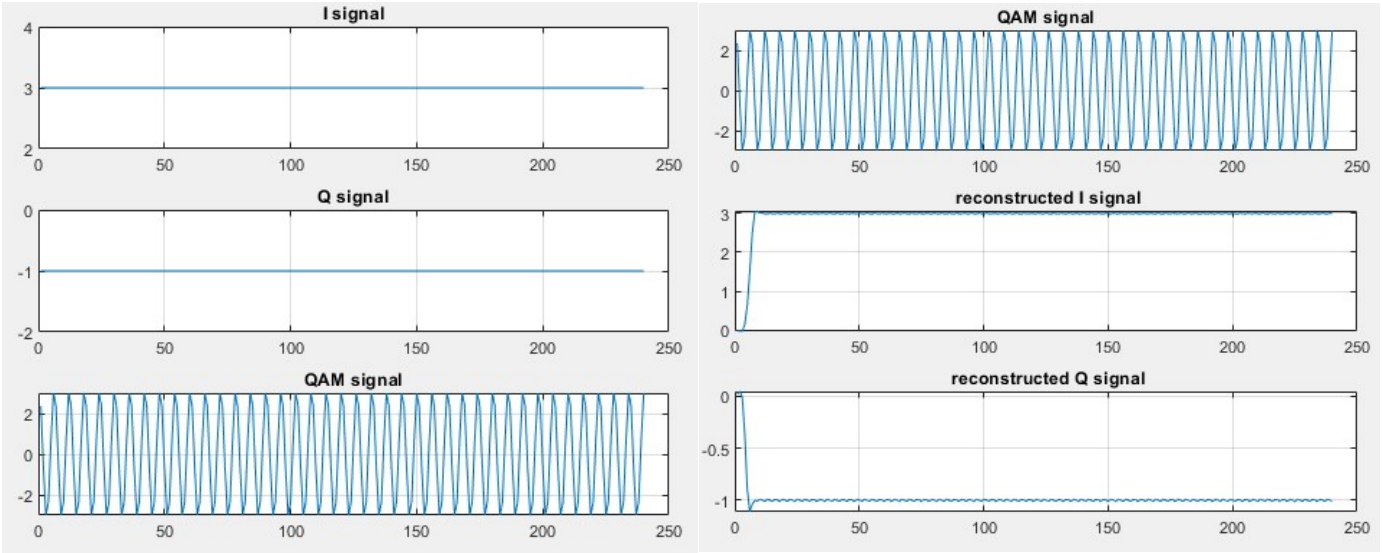
I = -1; Q = -1



16-QAM

Sirul de biti : 0, 1, 1, 0

I = 3; Q = -1



Codul de test in Matlab (Octave)

```
% test digital QAM modulation
% pkg load signal
clear all

fs = 48000; % sampling frequency
fp = 8000;  % carrier frequency

fd = 200; % symbol rate
R = fs / fd;  % number of samples per symbol

N = 1;    % number of sybmols

%%
%% p1 and  p2 between  0  and (f2/fp - 1)
%carrier phase at transmission p1
p1 = 0;
%%carrier phase at reception p2
p2 = 0;
%%


%% sample time tsample between 1 and R
tsample = R/2;


% number of bits per symbol
%B = 2;
%B = 4;
B = input("Number of bits per symbol (2 or 4): ");

if (B~=2 && B~=4)
display(strcat("Wrong value!"));
return;
end
M = N*B;    % number of bits

% input
in = zeros(M,1);
for k=1:M
  if (rand >0.5)
      in(k) = 1;
  end
end

% mapping
% 2 bits per simbol:
% 00 - I = 1, Q = 1
% 01 - I = 1, Q = -1
% 10 - I = -1, Q = 1
% 11 - I = -1, Q = -1

% 4 bits per simbol:
% 0000 - I = 1, Q = 1
% 0001 - I = 1, Q = -1
% 0010 - I = 1, Q = 1
% 0011 - I = 1, Q = 1
% 0100 - I = 3, Q = 1
```

```
% 0101 - I = 3, Q = -1
% 0110 - I = 3, Q = 1
% 0111 - I = 3, Q = 1
% 1000 - I = -1, Q = 1
% 1001 - I = -1, Q = -1
% 1010 - I = -1, Q = 1
% 1011 - I = -1, Q = 1
% 1100 - I = -3, Q = 1
% 1101 - I = -3, Q = -1
% 1110 - I = -3, Q = 1
% 1111 - I = -3, Q = 1

I = zeros(N*R,1);
Q = zeros(N*R,1);

if (B==2)
imap = [ 1, 1, -1, -1];
qmap = [ 1,-1,  1, -1];

q = 1;
for k=1:N
  b1 = in(q);
  b0 = in(q+1);
  q = q + 2;
  c = 2*b1 + b0;

  I((k-1)*R+1:k*R ) = imap(c+1);
  Q((k-1)*R+1:k*R ) = qmap(c+1);

end
end

if (B==4)
imap = [ 1,1, 1, 1,  3,3, 3, 3,  -1,-1,-1,-1,  -3,-3,-3,-3];
qmap = [ 1,3,-1,-3,  1,3,-1,-3,   1, 3,-1,-3,   1, 3,-1,-3];

q = 1;
for k=1:N
  b3 = in(q);
  b2 = in(q+1);
  b1 = in(q+2);
  b0 = in (q+3);
  q = q + 4;
  c = 8*b3 + 4*b2 + 2*b1 + b0;

  I((k-1)*R+1:k*R ) = imap(c+1);
  Q((k-1)*R+1:k*R ) = qmap(c+1);

end
end

%modulation

% QAM signal

s = zeros(N*R,1);
for k=1:N*R
s(k)= I(k)*cos(2*pi*(k+p1)*fp/fs) + Q(k)*cos(pi/2 + 2*pi*(k+p1)*fp/fs);
end

%plot
```

```
figure(1);
subplot(3,1,1);
plot(I); grid; title ("I signal");
subplot(3,1,2);
plot(Q); grid; title ("Q signal");
subplot(3,1,3);
plot(s); grid; title ("QAM signal");

% demodulation

% prepare low pass filter, cutt-of at fp
H = fir1(10, 2*fp/fs, 'low');
A = 1;

I1 = zeros(N*R,1);
Q1 = zeros(N*R,1);

for k=1:N*R
I1(k)= s(k)*cos(2*pi*(k+p2)*fp/fs);
end
Ir = 2*filter(H,A,I1);

for k=1:N*R
Q1(k)= s(k)*cos(pi/2 + 2*pi*(k+p2)*fp/fs);
end
Qr = 2*filter(H,A,Q1);


%plot
figure(2);
subplot(3,1,1);
plot(s); grid; title ("QAM signal");
subplot(3,1,2);
plot(Ir); grid; title (" reconstructed I signal");
subplot(3,1,3);
plot(Qr); grid; title (" reconstructed Q signal");


% demapping
err = 0.1;
out = zeros(M,1); % output

if (B==2)
q = 1;
for k=1:N
sampledI = Ir((k-1)*R + tsample);
sampledQ = Qr((k-1)*R + tsample);
for n=1:size(imap,2)
if ( abs(sampledI-imap(n)) < err && abs(sampledQ-qmap(n)) <err )
  p = n-1;
  v = bitshift(p,-1);
  out(q)= bitand(v,1);
  out(q+1)= bitand(p,1);
  q = q+2;
  break;
end
end
end
end

if (B==4)
```

```
q = 1;
for k=1:N
sampledI = Ir((k-1)*R + tsample);
sampledQ = Qr((k-1)*R + tsample);
for n=1:size(imap,2)
if ( abs(sampledI-imap(n)) < err && abs(sampledQ-qmap(n)) <err )
  p = n-1;
  v = bitshift(p,-3);
  out(q)= bitand(v,1);
  v = bitshift(p,-2);
  out(q+1)= bitand(v,1);
  v = bitshift(p,-1);
  out(q+2)= bitand(v,1);
  out(q+3)= bitand(p,1);
  q = q+4;
  break;
end
end
end
end


ERROR = abs(in-out);
BER=0;
for k=1:M
  BER = BER+ERROR(k);
end
BER = BER/M;
display(strcat("BER = ", num2str(BER)));
```