



Универзитет у Нишу
Електронски факултет у Нишу



Реализација кодера и декодера за 16 DQAM модулацију

Семинарски рад из Телекомуникација

Предметни наставник:
Ненад Милошевић

Студенти:
Јована Голијанин, 16038
Горан Димчић, 16055
Ненад Ђорђевић, 16080

Ниш, фебруар 2021.

САДРЖАЈ

1. Обрада сигнала и мултиплексни системи	1
1.1. Модулација аналогним сигналом	1
1.2. Квадратурна амплитудна модулација	1
1.3. Дигитални QAM	2
1.4. 16-то квадратурна амплитудска диференцијална модулација	3
Пример 1.	4
Пример 2.	7
1.5. Програм за 16 QAM модулацију	8
Генерисање случајних бинарних токова података	8
Претварање бинарних сигнала у сигнал целобројне вредности	9
Модулирање помоћу 16 QAM	10
Додавање <i>White Gaussian Noise</i>	10
Прављење дијаграма сазвежђа	10
Демодулирање 16-QAM	11
Претварање целобројних вредности сигнала у бинарни сигнал	11
Рачунање системског БЕР-а	11
1.6. МАТЛАБ програм за анализирање сазвежђа 16-QAM	12
Литература	14

1. Обрада сигнала и мултиплексни системи

1.1. Модулација аналогним сигналом

Модулација је поступак обраде сигнала којим се у преносни сигнал утискује сигнал информације. На пријемној страни се врши обрнути поступак – демодулација, како би се поново добила информација. Преносни сигнал има већу фреквенцију те има боља својства ширења преносним медијумом. Сигнал информације зовемо још и модулацијски сигнал, док као резултат модулације добијамо модулирани сигнал.

Постоји више врста модулација:

- Аналогна модулација сигнала, код које се мења један од параметара синусног сигнала: амплитуда, фреквенција или фаза. Стога разликујемо амплитудну модулацију (АМ), фреквенцијску модулацију (ФМ) и фазну модулацију (ПМ).
- Дискретна модулација или дигитална модулација синусног сигнала
- Модулација импулсних сигнала
- Дигитални модулацијски поступци, од којих су најпознатији: Импулсно кодна модулација (ПЦМ) и делта модулација (ДМ).
- Модулацијски поступци за пренос података у радио-дифузији.

У историји комуникација значајну улогу је одиграла амплитудна модулација, која је данас у свом изворном облику готово напуштена. Међутим њена примена је и даље присутна у сложеним модулацијским поступцима. Посебно је погодна за разумевање модулације уопштено.

Фреквенцијска модулација је данас најзаступљенија у радио-дифузији, премда су развијене и користе се и друге модулације односно модулацијски поступци који у себи садрже већи број поступака и фаза обраде сигнала.

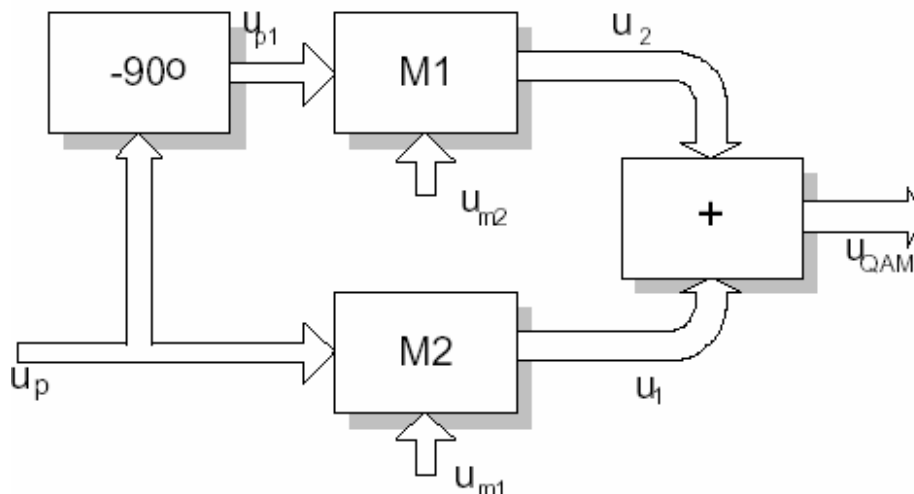
Најзаступљеније модулације су:

- амплитудна модулација (АМ)
- фреквенцијска модулација (ФМ)
- квадатурна амплитудна модулација (QAM)
- импулсно кодна модулација (ПЦМ) и
- делта модулација (ДМ)

1.2. Квадатурна амплитудна модулација

Квадатурна амплитудна модулација или QAM је сложенији модулацијски поступак, који се користи код преноса ТВ сигнала и код преноса дигиталних сигнала.

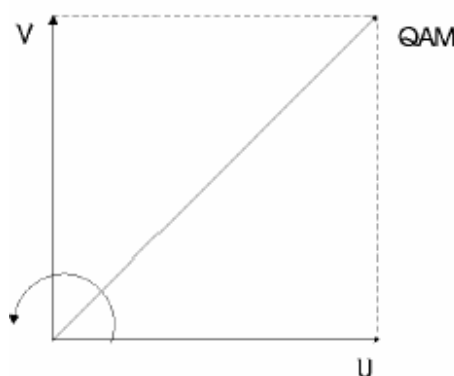
Овом модулацијом постиже се то да се истом преносном сигналу, помоћу амплитудне модулације, утисну два модулацијска сигнала (Слика 1.)



Слика 1. Блок шема добијања QAM сигнала

Модулациски сигнал U_{m1} модулира амплитуду преносног сигнала U_p , док истовремено модулациски сигнал U_{m2} , модулира амплитуду сигнала, који је релативно фазно померен за 90° у односу на сигнал U_p .

QAM сигнал се добије сабирањем та два сигнала дакле синусоиде и косинусоиде истих фреквенција. Резултантни модулирани сигнал U_{qam} , ће дакле имати промењиву амплитуду, али и фазу у односу на немодулирани сигнал (Слика 2.).



Слика 2. Векторски приказ QAM сигнала

U и V су тренутне вредности амплитуде појединих модулираних сигнала, а QAM је амплитуда резултантног сигнала.

1.3. Дигитални QAM

Као у многим шемама дигиталне модулације, дијаграм сазвежђа је користан за QAM. У QAM су тачке сазвежђа обично распоређене у квадратну мрежу са једнаким вертикалним и хоризонталним размаком, мада су могуће и друге конфигурације (нпр. Cross-QAM). Будући да су у дигиталним телекомуникацијама подаци обично бинарни, број тачака у мрежи обично је степен двојке (2, 4, 8, 16, ...). Будући да је QAM обично квадратни, неки од њих су ретки, док су најчешћи облици 16-QAM, 64-QAM и 256-QAM. Преласком у сазвежђе вишег реда могуће је пренети више бита по симболу. Међутим, ако ће средња енергија сазвежђа остати иста (путем поштеног поређења), тачке морају бити ближе једна према другој и тако су подложније шуму и другој корупцији сигнала што резултира већом стопом грешака у битовима и тако QAM вишег реда може испоручити више података мање поуздано од QAM-а нижег реда, за константну средњу констелациону енергију.

Коришћење QAM вишег реда без повећања стопе грешке у биту захтева већи однос сигнал-шум и то се постиже повећањем енергије сигнала, смањењем шума или коришћењем обе технике.

Ако су потребне брзине преноса података изнад оних које нуди 8-PSK, уобичајније је прећи на QAM јер се постиже већа удаљеност између суседних тачака у I-Q равни равномернијом дистрибуцијом тачака. Оно што компликује ствар је што тачке више нису исте амплитуде, тако да демодулатор сада мора тачно да детектује и фазу и амплитуду, а не само фазу.

64-QAM и 256-QAM се често користе у апликацијама дигиталне кабловске телевизије и кабловског модема. У Сједињеним Државама, 64-QAM и 256-QAM су обавезне шеме модулације за дигитални кабл које је SCTE стандардизовао у стандарду ANSI/SCTE 07 2013.

Комуникациони системи дизајнирани да постигну врло високе нивое спектралне ефикасности обично користе врло густа QAM сазвежђа. На пример, тренутни Homeplug AV2 500-Mbit/s мрежни Етернет уређаји користе 1024-QAM и 4096-QAM, као и будући уређаји који користе ITU-T G.hn стандард за умрежавање преко постојећих кућних ожичења (коаксијални кабл, телефонски водови и далеководи); 4096-QAM пружа 12 бита/симбол. Други пример је АДСЛ технологија за бакарне увијене парице, чија се величина сазвежђа креће до 32768-QAM (у АДСЛ терминологији ово се назива учитавање бита или бит по тону, 32768-QAM је еквивалентно 15 битова по тону).

Системи микроталасних повратних веза ултра великог капацитета такође користе 1024-QAM. Уз 1024-QAM, адаптивно кодирање и модулацију (ACM) и XPIС, добављачи могу добити гигабитни капацитет у једном каналу од 56 Mhz.

1.4. 16-то квадратурна амплитудска диференцијална модулација

Типично коло за 16 QAM диференцијални модулатор приказано је на слици 3. Битови А, Б, Ц и Д заједно са битовима а, б, ц и д чине адресу за ЕПРОМ. Битови А, Б, Ц и Д су улазни битови, при чему је бит А бит највеће тежине (MSB), а бит Д бит најмање тежине (LSB). Излаз ЕПРОМ-а производи битове а (MSB), б, ц и д (LSB). Ови битови се затим примењују на 16 QAM модулатор.

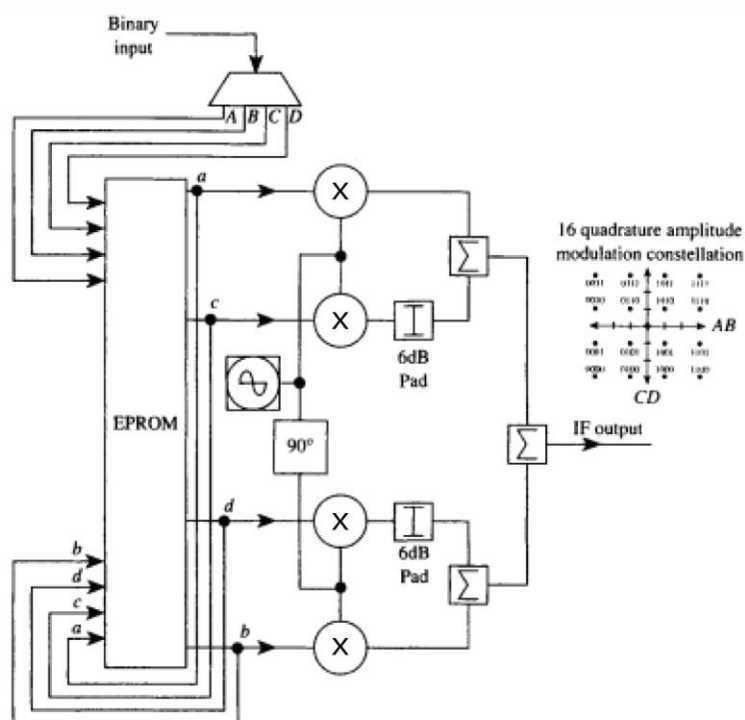


Fig. 4.20 A 16 QAM differential modulator

Слика 3. 16 QAM диференцијални модулатор

Метода диференцијалне модулације дата у решењу примера 1 ослања се на могућност постављања фазора у тачан квадрант у сазвежђу. Позивајући се на слику 4, може се видети да X-X оса садржи бинарне кодове за АБ, који су $00_2, 01_2, 10_2, 11_2$ с лева на десно. Исто тако Y-Y оса садржи бинарне кодове за ЦД који су $00_2, 01_2, 10_2, 11_2$ одоздо према горе.

Користећи ове информације прилично је једноставно поставити сваки фазор у исправан квадрант. Као што је приказано, АЦ = 11, за квадрант 1; АЦ = 01, за квадрант 2; АЦ = 00 за квадрант 3; и АЦ = 10 за квадрант 4. Отуда логички услови за АЦ одређују квадрант у коме лежи фазор. Логички услови за БД одређују положај у квадранту, јер у сваком квадранту логичка стања БД узимају вредности су $00_2, 01_2, 10_2, 11_2$.

Као пример, фазор који има код 0010_2 значи да је АЦ = 01 и БД = 00. Бинарни код 1000_2 доводи до тога да фазор лежи у квадранту 4 јер су АЦ = 10 и БД = 00.

Диференцијално кодирање се одвија према табели на слици 5.

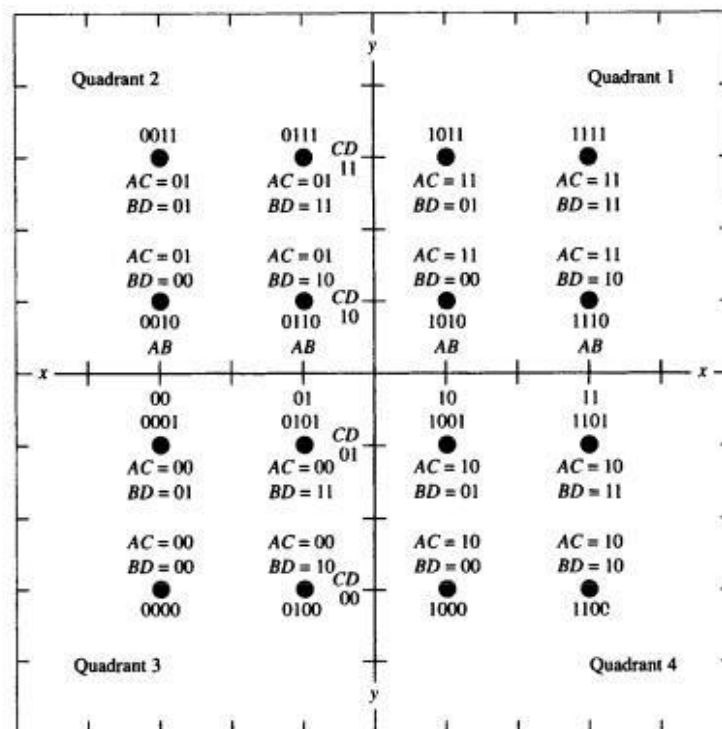


Fig. 4.21 Constellation for 16 QAM showing positioning of phasors

Слика 4. 16 QAM фазе

Пример 1.

Диференцијални модулатор од 16 QAM ствара сазвежђе као што је приказано на слици 4.11. Следећи ток бита примењује се на улаз модулатора. Диференцијално кодирање се одвија према табели на слици 5.

Бинарни подаци = (0110) 1101 1000 0001 1100 1010₂

Одредити кодове за кодиране податке на излазу из ЕПРОМ-а. Претпоставимо да је подразумевани код 0110_2 .

Table 4.10 Differential encoding data

Bin code		Phase angle	Bin code		Amplitude	
A/a	C/c		B/b	D/d	b/B	d/D
0	0	0°	0	0	1	1
0	1	-90°	0	1	1	0
1	0	+90°	1	0	0	1
1	1	±180°	1	1	0	0

Слика 5. Диференцијални подаци кодирања

Решење

Диференцијално кодирање се врши према подацима датим у табели 4.11. на слици 6.

- Пронаћи положај на дијаграму сазвежђа за $0110_2 \rightarrow 2$. квадрант.
 Пронаћи положај на дијаграму сазвежђа за $1101_2 \rightarrow 4$. квадрант.
 У односу на 0110_2 , 1101_2 се налази померањем 0110_2 за $\pm 180^\circ \rightarrow$ битови ац = 11.
 Битови БД = 11, дакле битови бд = 00.
 Подаци су 1010_2 .
- Послати 1010_2 назад на улаз.
 Пронаћи положај на дијаграму сазвежђа за $1010_2 \rightarrow 1$. квадрант.
 Пронаћи положај на дијаграму сазвежђа за $1000_2 \rightarrow 4$. квадрант.
 У односу на 1010_2 , 1000_2 се проналази померањем 1010_2 за $-90^\circ \rightarrow$ битови ац = 01.
 Битови БД = 00, дакле битови бд = 11.
 Подаци су 0111_2 .
- Послати 0111_2 назад на улаз.
 Пронаћи положај на дијаграму сазвежђа за $0111_2 \rightarrow 2$. квадрант.
 Пронаћи положај на дијаграму сазвежђа за $0001_2 \rightarrow 3$. квадрант.
 У односу на 0111_2 , 0001_2 се проналази померањем 0001_2 за $+90^\circ \rightarrow$ битови ац = 10.
 Битови БД = 01, дакле битови бд = 10.
 Подаци су 1100_2 .
- Послати 1100_2 назад на улаз.
 Пронаћи положај на дијаграму сазвежђа за $1100_2 \rightarrow 4$. квадрант.
 Пронаћи положај на дијаграму сазвежђа за $1100_2 \rightarrow 4$. квадрант.
 У односу на 1100_2 , 1100_2 се проналази померањем 1100_2 за $0^\circ \rightarrow$ битови ац = 00. Битови БД = 10, дакле битови бд = 01.
 Подаци су 0001_2 .
- Послати 0001_2 назад на улаз.
 Пронаћи положај на дијаграму сазвежђа за $0001_2 \rightarrow 3$. квадрант.
 Пронаћи положај на дијаграму сазвежђа за $1010_2 \rightarrow 1$. квадрант.
 У односу на 0001_2 , 1010_2 се налази померањем 1010_2 за $180^\circ \rightarrow$ битови ац = 11. Битови БД = 00, дакле битови бд = 11.
 Подаци су 1111_2 .

Table 4.11 Differential encoding for 16 QAM modulator

Present				Quad	Next				Quad	Phase			Amplitude				Output data			
A	B	C	D		A	B	C	D		Change	a	c	B	D	b	d	a	b	c	d
0	1	1	0	2nd	1	1	0	1	4th	180°	1	1	1	1	0	0	1	0	1	0
1	0	1	0	1st	1	0	0	0	4th	-90°	0	1	0	0	1	1	0	1	1	1
0	1	1	1	2nd	0	0	0	1	3rd	+90°	1	0	0	1	1	0	1	1	0	0
1	1	0	0	4th	1	1	0	0	4th	0°	0	0	1	0	0	1	0	0	0	1
0	0	0	1	3rd	1	0	1	0	1st	180°	1	1	0	0	1	1	1	1	1	1

Слика 6. Диференцијално кодирање за 16 QAM модулатор

Ово даје следеће токове података:

АБЦД АБЦД АБЦД АБЦД АБЦД

Улазни подаци = 1101 1000 0001 1100 1010₂

абцд абцд абцд абцд абцд абцд

Диференцијално кодирани подаци = (0110) 1010 0111 1100 0001 1111₂

Једном када се подаци диференцијално кодирају, на крају ће се пренети на удаљени пријемник. Пријемник мора бити у стању да правилно декодира податке да би добио оригиналне податке.

Типично коло приказано је на слици 7. IF улаз је подељен на четири миксера где је сваки улаз демодулисан са истом носећом фреквенцијом. Осцилатор носача је повезан директно са горња два миксера и носећа фреквенција се помера за 90° пре него што се примени на доња два миксера. Горња два миксера производе битове а и ц, док доња два миксера производе битове б и д. Ова четири тока бита се затим примењују на ЕПРОМ као део адресе.

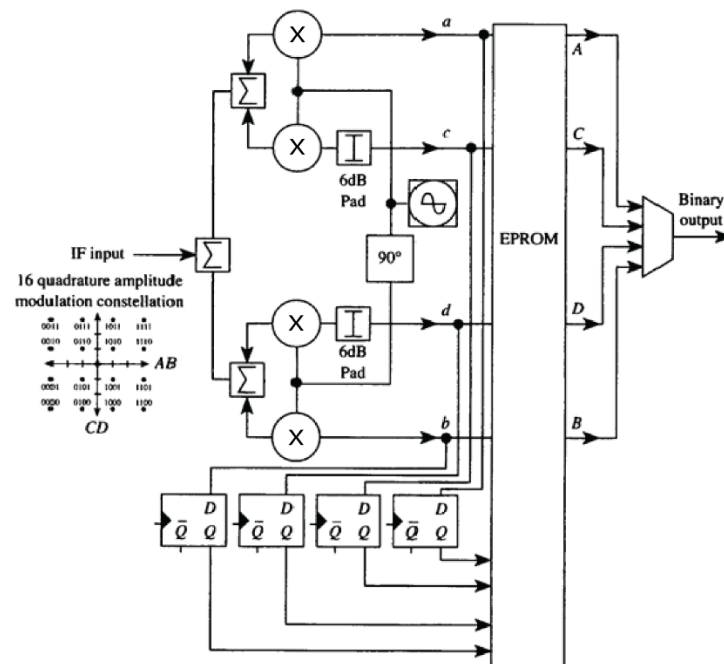


Fig. 4.22 A 16 QAM differential demodulator

Слика 7. 16 QAM диференцијални демодулатор

Они се такође примењују на Д улазе четири флип-флопа и један циклус такта касније исти логички услови се појављују на четири излаза и чине други део ЕПРОМ адресе. Излаз ЕПРОМ-а производи токове битова на водовима А, Б, Ц и Д. Ова четири тока бита се затим мултиплексирају заједно да би формирали серијски излазни ток битова.

Пример 2.

Диференцијални демодулатор од 16 QAM ствара сазвежђе као што је приказано на слици 4.11. Следећи ток битова се прима са удаљеног предајника. Диференцијално декодирање се одвија према подацима датим у табели 4.10. на слици 5.

Бинарни подаци = (0110) 1010 0111 1100 0001 1111₂

Одредити излазне податке декодера. Претпоставити да је подразумевани код 0110₂.

Решење

Декодирање се врши како је приказано у табели 4.12. на слици 8.

1. Пронаћи 0110₂ на сазвежђу → 2. квадрант.
Одредити где се премешта 0110₂.
Следећи улазни подаци су 1010₂ → бд = 00 и БД = 11 → ац = 11.
11₂ = +/- 180°.
0110₂ се премешта се у четврти квадрант.
АЦ = 10.
Подаци = 1101₂.
2. Вратити 1010₂ назад на улаз.
Пронаћи 1010₂ у сазвежђу → 1. квадрант.
Одредити где се премешта 1010₂.
Следећи улазни подаци су 0111₂ → бд = 11 и БД = 00 → ац = 01.
01₂ = - 90°.
1010₂ се премешта се у четврти квадрант.
АЦ = 10.
Подаци = 1000₂.

Table 4.12 Differential decoding for 16 QAM demodulator

Present				Quad	Next				Phase				Amplitude				Output data					
a	b	c	d		a	b	c	d	a	c	Change	Quad	A	C	b	d	B	D	A	B	C	D
0	1	1	0	2nd	1	0	1	0	1	1	180°	4th	1	0	0	0	1	1	1	0	1	
1	0	1	0	1st	0	1	1	1	0	1	-90°	4th	1	0	1	1	0	0	1	0	0	0
0	1	1	1	2nd	1	1	0	0	1	0	+90°	3rd	0	0	1	0	0	1	0	0	0	1
1	1	0	0	4th	0	0	0	1	0	0	0°	4th	1	0	0	1	1	0	1	1	0	0
0	0	0	1	3rd	1	1	1	1	1	1	180°	1st	1	1	1	1	0	0	1	0	1	0
↑					↓																	

Слика 8. Диференцијално декодирање за 16 QAM демодулатор

3. Послати 0111₂ назад на улаз.
Пронаћи 0111₂ у сазвежђу → 2. квадрант.
Одредити где се премешта 0111₂.

Следећи улазни подаци су $1100_2 \rightarrow бд = 10$ и $БД = 01 \rightarrow ац = 10$.

$10_2 = +90^\circ$.

0111_2 се премешта се у трећи квадрант.

$АЦ = 00$.

Подаци = 0001_2 .

4. Послати 1100_2 назад на улаз.

Пронаћи 1100_2 у сазвежђу \rightarrow 4. квадрант.

Одредити где се премешта 1100_2 .

Следећи улазни подаци су $0001_2 \rightarrow бд = 01$ и $БД = 10 \rightarrow ац = 00$.

$00_2 = 0^\circ$.

1100_2 се остаје у четвртом квадранту.

$АЦ = 10$.

Подаци = 1100_2 .

5. Послати 0001_2 назад на улаз.

Пронаћи 0001_2 у сазвежђу \rightarrow 3. квадрант.

Одредити где се премешта 0001_2 .

Следећи улазни подаци су $1111_2 \rightarrow бд = 11$ и $БД = 00 \rightarrow ац = 11$.

$11_2 = +/- 180^\circ$.

0001_2 се премешта се у трећи квадрант.

$АЦ = 11$.

Подаци = 1010_2 .

Ово даје следеће токове података:

	АБЦД	АБЦД	АБЦД	АБЦД	АБЦД
Излазни подаци =	1101	1000	0001	1100	1010 ₂

Ово је исти скуп података као и улазни подаци који су унети у предајник.

1.5. Програм за 16 QAM модулацију

Генерисање случајних бинарних токова података

Првенствено треба генерисати случајни бинарни ток података.

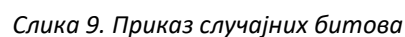
Конвенционални формат за представљање сигнала у МАТЛАБ-у је вектор или матрица. Функција *randi* креира вектор колоне који садржи вредности бинарног тока података. Дужина бинарног тока података (односно број редова у вектору колоне) произвољно се поставља на 30 000. Затим се дефинишу параметри.

```
M = 16; % Modulation order (alphabet size or number of points in signal constellation)
k = log2(M); % Number of bits per symbol
n = 30000; % Number of bits to process
sps = 1; % Number of samples per symbol (oversampling factor)
```

Након тога треба се поставити *rng* функција на њено подразумевано стање или било коју статичку вредност, тако да пример даје поновљиве резултате. Затим користити функцију *randi* за генерисање случајних бинарних података.

```
rng default;
dataIn = randi([0 1],n,1); % Generate vector of binary data
```

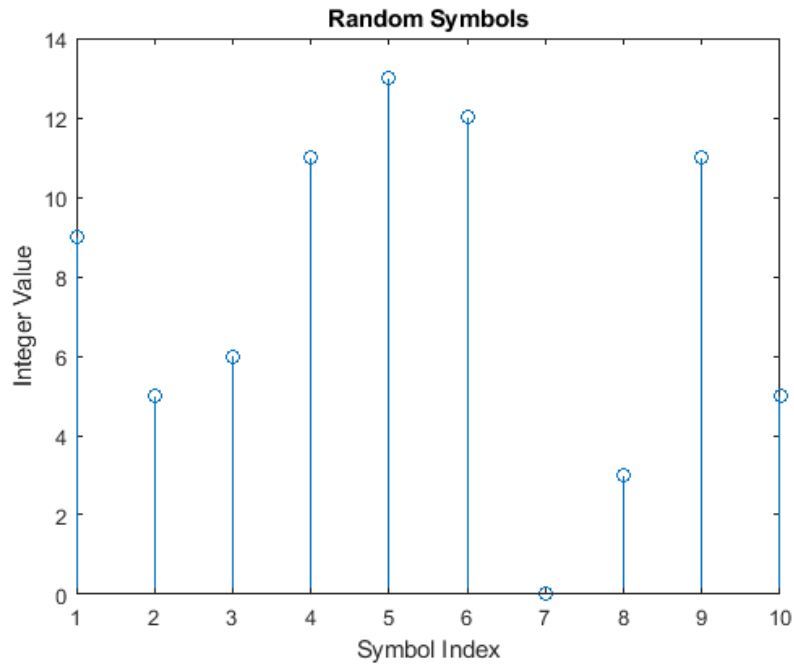
```
stem(dataIn(1:40), 'filled');
title('Random Bits');
xlabel('Bit Index');
ylabel('Binary Value');
```



Подразумевана конфигурација функције *qammod* очекује да подаци са целобројном вредношћу као улазни симболи модулирају. У овом примеру, бинарни ток података се претходно обрађује у целобројне вредности пре употребе функције *qammod*. Конкретно, функција *bi2de* претвара сваку четворку података у одговарајући цели број у опсегу $[0, (M - 1)]$. Редослед модулације, M , је 16 у овом примеру.

```
dataInMatrix = reshape(dataIn,length(dataIn)/k,k);
dataSymbolsIn = bi2de(dataInMatrix);
```

```
figure; % Create new figure window.
stem(dataSymbolsIn(1:10));
title('Random Symbols');
xlabel('Symbol Index');
ylabel('Integer Value');
```



Слика 10. Приказ случајних симбола

Модулирање помоћу 16 QAM

Користимо функцију *qammod* за примену 16-QAM модулације на *dataSymbolsIn* вектор колоне за бинарно кодирано и кодирано Грејевим кодом бинарно пресликавање бит-у-симбол.

```
dataMod = qammod(dataSymbolsIn,M,'bin'); % Binary coding with phase offset of zero
dataModG = qammod(dataSymbolsIn,M); % Gray coding with phase offset of zero
```

Операција модулације даје сложене векторе колоне који садрже вредности које су елементи сазвежђа 16-QAM сигнала. Касније у овом примеру дијаграми сазвежђа показују мапирање природних и Грејових бинарних симбола.

Додавање White Gaussian Noise

Модулисани сигнал пролази кроз канал помоћу функције *awgn* са наведеним односом сигнал-шум (SNR). Претварамо однос енергије по биту према спектралној густини снаге шума (E_b/N_0) на SNR вредност коју користи функција *awgn*. У овом примеру променљива *sps* није значајна, али олакшава проширавање примера да би се користило обликовање импулса.

Затим рачунамо SNR када канал има E_b/N_0 од 10 децибела.

```
EbNo = 10;
snr = EbNo+10*log10(k)-10*log10(sps);
```

Пренесимо сигнал кроз AWGN канал за бинарно и Грејево кодирано пресликавање симбола.

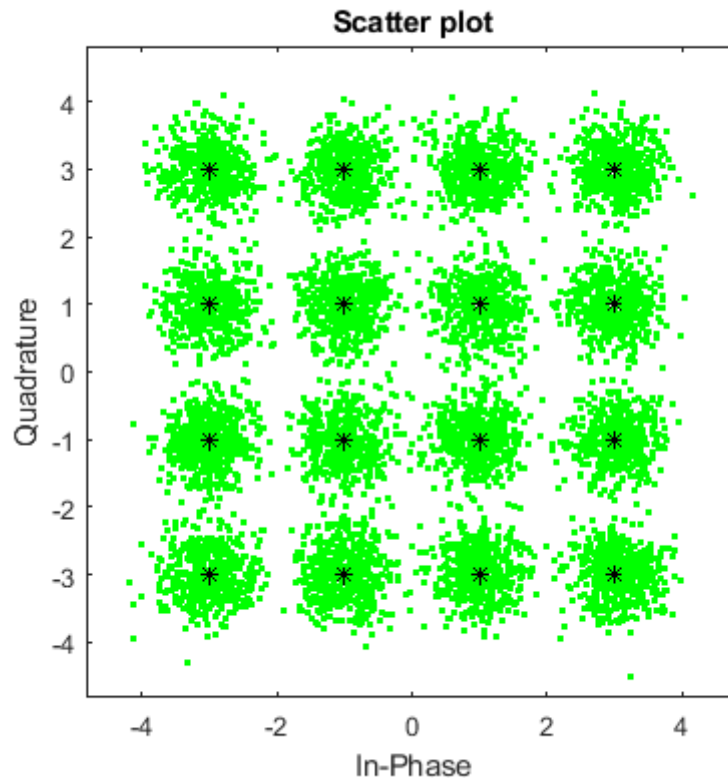
```
receivedSignal = awgn(dataMod,snr,'measured');
receivedSignalG = awgn(dataModG,snr,'measured');
```

Прављење дијаграма сазвежђа

Искористићемо функцију *scatterplot* за приказ фазних и квадратурних компоненти модулисаног сигнала, *dataMod* и шума сигнала примљеног након канала. Ефекти AWGN функције су присутни на дијаграму сазвежђа.

```
sPlotFig = scatterplot(receivedSignal,1,0,'g. ');
hold on
```

```
scatterplot(dataMod,1,0,'k*',sPlotFig)
```



Слика 11. Приказ података на дијаграму

Демодулирање 16-QAM

Користимо функцију *qamdemod* да бисмо демодулирали примљене податке и избацили целобројне вредности података.

```
dataSymbolsOut = qamdemod(receivedSignal,M,'bin');
dataSymbolsOutG = qamdemod(receivedSignalG,M);
```

Претварање целобројних вредности сигнала у бинарни сигнал

Помоћу функције *de2bi* претварамо симболе података из QAM демодулатора *dataSymbolsOut* у бинарну матрицу *dataOutMatrix*, димензија $N_{\text{sym}} \times N_{\text{bits/sym}}$. У матрици N_{sym} је укупан број QAM симбола а $N_{\text{bits/sym}}$ је број бита по симболу. За 16-QAM, $N_{\text{bits/sym}} = 4$. Затим претварамо матрицу у вектор колоне дужине једнаком броју улазних бита, 30.000. Поновљамо поступак за Грејево кодиране симболе података, *dataSymbolsOutG*. Враћамо мапиране битове у симбол.

```
dataOutMatrix = de2bi(dataSymbolsOut,k);
dataOut = dataOutMatrix(:); % Return data in column vector
dataOutMatrixG = de2bi(dataSymbolsOutG,k);
dataOutG = dataOutMatrixG(:); % Return data in column vector
```

Рачунање системског БЕР-а

Функција *biterr* израчунава статистику грешака бита из оригиналног бинарног тока података, *dataIn* и примљених токова података, *dataOut* и *dataOutG*. Грејево кодирање значајно смањује БЕР (Bit Error Rate - Стопа грешака у битовима).

За израчунавање статистике грешака користићемо функцију стопе грешака, а функцију *fprintf* за приказ резултата.

```
[numErrors,ber] = biterr(dataIn,dataOut);
fprintf('\nThe binary coding bit error rate is %5.2e, based on %d errors.\n',
ber,numErrors)
```

Стопа грешке бинарног кодираног бита је $2,40\text{e-}03$, на основу 72 грешке.

```
[numErrorsG,berG] = biterr(dataIn,dataOutG);
fprintf('\nThe Gray coding bit error rate is %5.2e, based on %d errors.\n',
berG,numErrorsG)
```

Стопа грешака битова Грејевог кода је $1,33\text{e-}03$, на основу 40 грешака.

1.6. МАТЛАБ програм за анализирање сазвежђа 16-QAM

Програм за анализу сазвежђа написан од стране *Dr. Vinoth Babu Kumaravelu*, из Електронске школе Техничког института у Валуру у Индији.

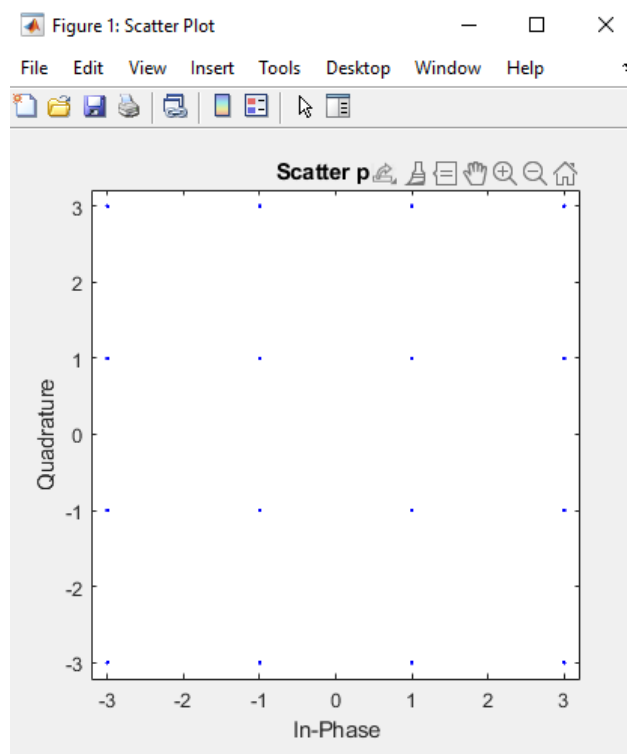
```
clc; clear; close all;
M=16; %modulation order
bb=log2(M); %bits per symbol
N=4000; %input bits size
x=randi([0,1],1,N); %random input bits
yy=[];
for i=1:4:length(x)
    if x(i)==0 && x(i+1)==0 && x(i+2)==1 && x(i+3)==0
        y=-3+1j*3;
    elseif x(i)==0 && x(i+1)==1 && x(i+2)==1 && x(i+3)==0
        y=-1+1j*3;
    elseif x(i)==0 && x(i+1)==0 && x(i+2)==1 && x(i+3)==1
        y=-3+1j*1;
    elseif x(i)==0 && x(i+1)==1 && x(i+2)==1 && x(i+3)==1
        y=-1+1j*1;
    elseif x(i)==1 && x(i+1)==1 && x(i+2)==1 && x(i+3)==0
        y=1+1j*3;
    elseif x(i)==1 && x(i+1)==0 && x(i+2)==1 && x(i+3)==0
        y=3+1j*3;
    elseif x(i)==1 && x(i+1)==1 && x(i+2)==1 && x(i+3)==1
        y=1+1j*1;
    elseif x(i)==1 && x(i+1)==0 && x(i+2)==1 && x(i+3)==1
        y=3+1j*1;
    elseif x(i)==0 && x(i+1)==0 && x(i+2)==0 && x(i+3)==1
        y=-3-1j*1;
    elseif x(i)==0 && x(i+1)==1 && x(i+2)==0 && x(i+3)==1
        y=-1-1j*1;
    elseif x(i)==0 && x(i+1)==0 && x(i+2)==0 && x(i+3)==0
        y=-3-1j*3;
    elseif x(i)==0 && x(i+1)==1 && x(i+2)==0 && x(i+3)==0
        y=-1-1j*3;
    elseif x(i)==1 && x(i+1)==1 && x(i+2)==0 && x(i+3)==1
        y=1-1j*1;
    elseif x(i)==1 && x(i+1)==0 && x(i+2)==0 && x(i+3)==1
        y=3-1j*1;
    elseif x(i)==1 && x(i+1)==1 && x(i+2)==0 && x(i+3)==0
        y=1-1j*3;
```

```

elseif x(i)==1 && x(i+1)==0 && x(i+2)==0 && x(i+3)==0
    y=3-1j*3;
end
yy = [yy y];
end
scatterplot(yy)
EbN0dB=10;
EbN0=10^(EbN0dB/10);
n=(1/sqrt(2))*[randn(1,length(yy))+sqrt(-1)*randn(1,length(yy))];
sigma=sqrt(1/((log2(M))*EbN0));
r=yy+sigma*n;
scatterplot(r)

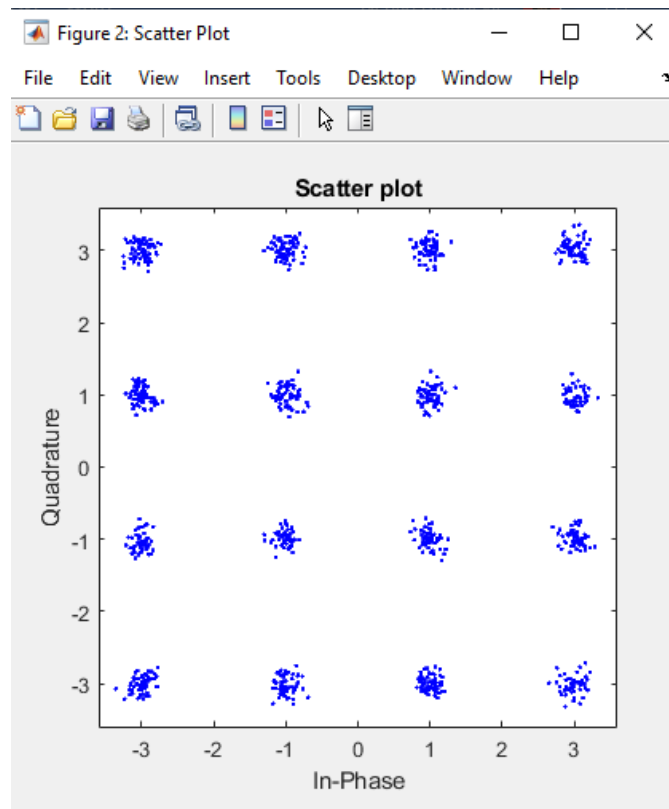
```

Фигура 1 на слици 12. испод показује распоред података који се добија уносом ових вредности пре додавања *White Gaussian Noise*, а затим и покретањем овог програма.



Слика 13. Приказ података на дијаграму

Фигура 2 на слици 13. испод показује распоред података који се добија уносом ових вредности након додавања *White Gaussian Noise*, а затим и покретањем овог програма.



Слика 13. Приказ података на дијаграму

Литература

1. *Obrada signala i multipleksni sustavi, одломак*
2. https://en.wikipedia.org/wiki/Quadrature_amplitude_modulation#Digital_QAM
3. <https://www.mathworks.com/help/comm/gs/examine-16-qam-using-matlab.html>
4. <https://www.youtube.com/watch?v=NvUU8FDuTP4>