

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

Кологривов В. А.
Крупина В. В.

**ИССЛЕДОВАНИЕ ЧЕТЫРЕХКАНАЛЬНОЙ МОДЕЛИ СИСТЕМЫ ПЕРЕДАЧИ
ДААННЫХ НА ОСНОВЕ ТЕХНОЛОГИИ LFM_16PSK**

Методические указания по лабораторной работе в среде функционального моделирования
Simulink системы MatLab для студентов радиотехнических специальностей

Томск 2022

УДК 621.396.61+621.376.3

ББК 32.811.7

К 61

Рецензент:

Мещеряков А.А., доцент кафедры радиотехнических систем ТУСУР, канд. техн. наук

Кологривов В. А., Крупина В. В.

К 61 Исследование четырехканальной модели системы передачи данных на основе технологии *LFM_16PSK*: методические указания по лабораторной работе для студентов радиотехнических специальностей / Кологривов В. А., Крупина В. В. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2022 – 27 с.

Настоящие методические указания по лабораторной работе составлены с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО).

Лабораторная работа «Исследование четырехканальной модели системы передачи данных на основе технологии *LFM_16PSK*» посвящена модельному исследованию четырехканальной системы передачи данных на основе технологии *LFM_16PSK* -модема с использованием пакета функционального моделирования *Simulink* системы для инженерных и научных расчетов *MatLab*.

Работа «Исследование четырехканальной модели системы передачи данных на основе технологии *LFM_16PSK*» относится к циклу лабораторных работ по разделу «Модуляция/Демодуляция», входящему в дисциплины радиотехнических специальностей.

В описании сформулирована цель лабораторной работы, приведены краткие теоретические сведения о *LFM* –модуляции и *16PSK* манипуляции, краткая характеристика пакета *Simulink* системы *MatLab*, описание виртуального лабораторного макета и используемых блоков библиотеки *Simulink*, а также требования к модельному исследованию и контрольные вопросы, ответы на которые необходимы для успешной защиты лабораторной работы.

Одобрено на заседании каф. РТС протокол № 3 от 27.10.22

УДК 621.396.61→621.376.3

ББК 32.811.7

© Кологривов В. А., Крупина В.В., 2022

© Томск. гос. ун-т систем упр. и радиоэлектроники, 2022

ОГЛАВЛЕНИЕ

1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	4
1.1 Теоретические сведения о <i>LFM</i> -модуляции	4
1.2 Теоретические сведения о <i>16PSK</i> -манипуляции	6
1.3 Кратко о технологии <i>LFM_MPSK</i>	7
2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ <i>SIM</i>-МОДЕЛИ <i>LFM_16PSK</i>-МОДЕМА...	10
3 ПРИНЦИП РАБОТЫ ФУНКЦИОНАЛЬНОЙ <i>SIM</i>-МОДЕЛИ <i>LFM_16PSK</i>-МОДЕМА	16
4 КРАТКОЕ ОПИСАНИЕ ПАКЕТА <i>SIMULINK</i> И ИСПОЛЬЗУЕМЫХ БЛОКОВ	20
4.1 Запуск и работа с пакетом <i>Simulink</i>	20
4.2 Описание используемых блоков библиотеки <i>Simulink</i>	22
5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ	25
6 КОНТРОЛЬНЫЕ ВОПРОСЫ	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27

1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Цель работы: Разработка модели и сравнительное исследование с точки зрения помехоустойчивости и спектральной эффективности четырехканального модема, реализованного на основе технологии *LFM_16PSK* с использованием пакета функционального моделирования *Simulink*.

1.1 Теоретические сведения о *LFM*-модуляции

Линейная частотная модуляция (*ЛЧМ*; *LFM*) сигнала – это вид частотной модуляции, при которой частота несущего сигнала изменяется по линейному закону [1-6] (см. рис. 1.1).

На практике используются радиоимпульсы, частота которых меняется по линейному закону:

$$\omega(t) = \omega_0 + \alpha t, \quad (1.1)$$

где $\alpha = \frac{2\Delta\omega}{\tau}$, $\Delta\omega = 2\pi\Delta f$ – девиация частоты;

τ – длительность импульса.

Частота изменяется в пределах от $\omega_{\min} = \omega_0 - \Delta\omega$ до $\omega_{\max} = \omega_0 + \Delta\omega$.

Фаза сигнала модуляции

$$\vartheta(t) = \int \alpha t dt = \frac{1}{2} \alpha t^2, |t| \leq \frac{\tau}{2}. \quad (1.2)$$

Период колебания средней частоты

$$T_0 = \frac{2\pi}{\omega_0} = \frac{1}{f_0}. \quad (1.3)$$

Число периодов

$$N_0 = \frac{\tau}{T_0}. \quad (1.4)$$

Глубина модуляции частоты

$$m = \frac{\Delta\omega}{\omega_0} = \frac{\Delta f}{f_0}. \quad (1.5)$$

Основным параметром *ЛЧМ*-сигнала является его база *B*

$$B = \Delta f \tau = N_0 m. \quad (1.6)$$

На рисунке 1.1 представлен *ЛЧМ*- сигнал.

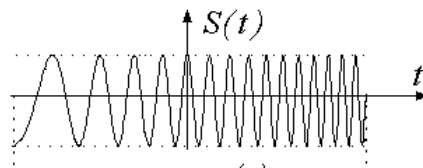


Рисунок 1.1 - *ЛЧМ*- сигнал

На рисунке 1.2 представлен спектр колебания. Он выражается через специальные функции – интегралы Френеля. Благодаря тому, что согласно (1.2) фаза $\mathcal{Y}(t)$ здесь функция четная, все составляющие спектра имеют четное распределение относительно частоты ω .

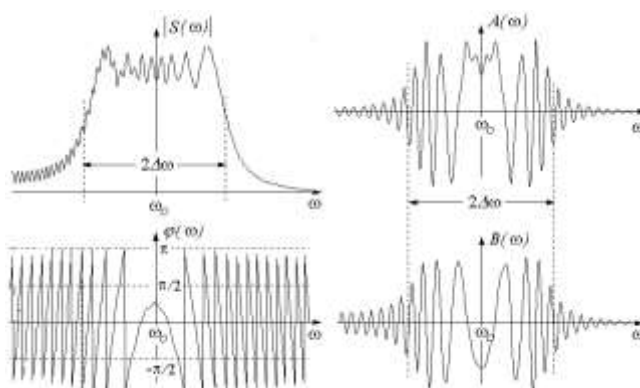


Рисунок 1.2 – Спектр колебания ЛЧМ- сигнала

ЛЧМ-сигналы широко используются в системах оптимального приема дальней радиолокации, и в этой связи важную роль играет их автокорреляционная функция, огибающая которой по форме близка к функции синуса, с шириной лепестков

$$\Delta t \approx \frac{1}{2\Delta f} = \frac{\tau}{2B}. \quad (1.7)$$

В пределах центрального лепестка шириной $2\Delta t$ укладывается N периодов T_0 колебаний несущей частоты:

$$N = \frac{2\Delta t}{T_0} \approx \frac{\tau/T_0}{B} = \frac{N_0}{B} = \frac{1}{m}. \quad (1.8)$$

При $B \gg 1$ центральный лепесток имеет вид узкого корреляционного пика. Коэффициент сжатия

$$B = \frac{\tau}{2\Delta T}. \quad (1.9)$$

На рисунке 1.3 представлена автокорреляционная функция ЛЧМ-сигнала и ее спектр

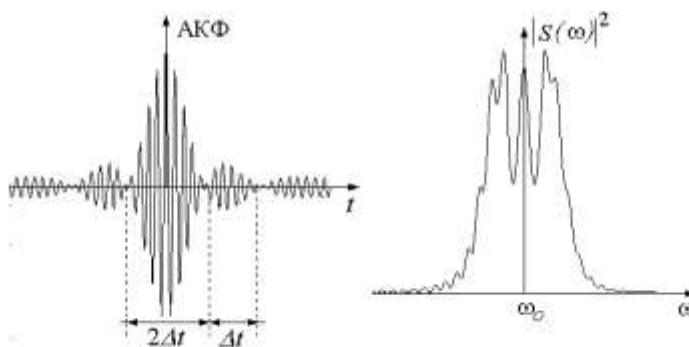


Рисунок 1.3 - Автокорреляционная функция ЛЧМ-сигнала и спектр сигнала

1.2 Теоретические сведения о 16PSK-манипуляции

Одним из основных параметров модема и вида модуляции при цифровой передаче является число вариантов состояний сигнала на выходе или, другими словами, число вариантов модулируемого параметра выходного сигнала. Так, если m вариантов сигнала равновероятны, то производительность модулятора как источника информации на входе непрерывного канала связи, т.е. количество информации, вырабатываемое им в единицу времени, прямо пропорционально двоичному логарифму числа m :

$$N = \log_2 m, \quad (1.10)$$

где m – кратность модуляции,

N – количество бит информации.

В рассматриваемом случае для 16PSK-модема согласно формуле 1.10 четырех кратная фазовая модуляция ($N = 4$) означает, что в каждом элементарном сигнале на выходе модулятора содержится N бит информации, а фаза имеет шестнадцать допустимых значения ($m = 16$).

Таким образом, шестнадцатипозиционная фазовая модуляция (*16-Phase Shift Keying* – 16PSK), применяемая в цифровой системе передачи, является шестнадцатипозиционной фазовой манипуляцией, при которой шестнадцати символам цифрового сообщения {0000, 1000, 1001, 1011, 1010, 1110, 1111, 1101, 1100, 0100, 0101, 0111, 0110, 0010, 0011, 0001} соответствует шестнадцать значений фазы модулированного сигнала с шагом, кратным $\pi/8$, например, $\{\pi/8, \pi/4\}$, $\{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8} \dots\}$ или любого другого множества.

На рисунке 1.4 представлено созвездие 16PSK

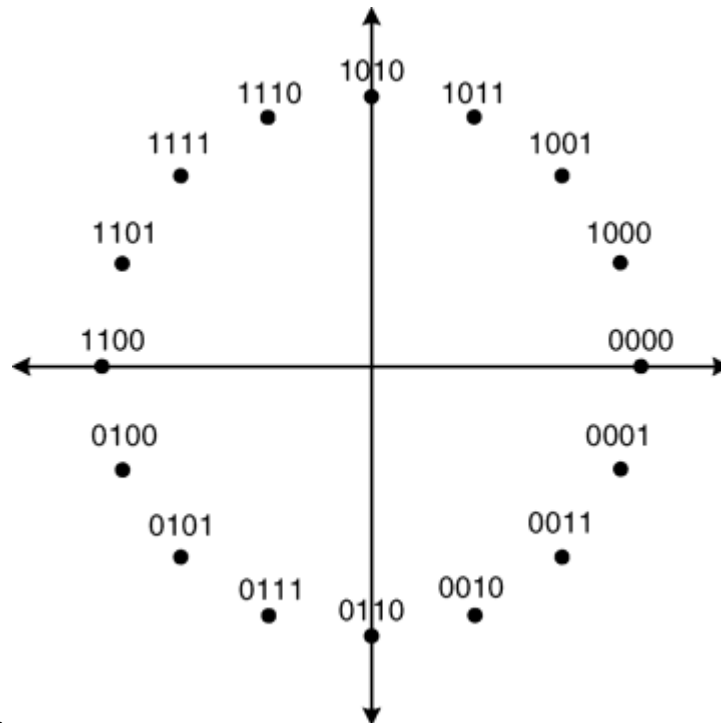


Рисунок 1.4 – Фазовая диаграмма 16PSK-сигнала

Используемый метод модуляции так же является и основным фактором, определяющим помехоустойчивость, спектральную и энергетическую эффективности

цифровой системы передачи. Улучшение одной из таких характеристик может привести к ухудшению остальных, поэтому необходимо исследование и установление взаимосвязи перечисленных характеристик при изменении определяющих параметров системы передачи, например, отношения сигнал/шум.

Под помехоустойчивостью понимают свойство системы противодействовать влиянию помех. Ее принято оценивать зависимостью вероятности битовой ошибки P_b от удельного расхода энергии на передачу бита данных E_b/N_0 . Такая вероятность – основополагающее качество модема, зависящее от статистических характеристик сигнала и параметров, действующих в канале связи.

Под спектральной эффективностью подразумевают отношение скорости передачи данных на 1 Гц используемой полосы частот (бит/с/Гц) [2]:

$$\gamma = \frac{R_b}{W}, \quad (1.11)$$

где R_b – скорость передачи информации, бит/с ;

W – полная полоса частот канала, Гц .

В свою очередь, энергетическая эффективность определяется мощностью, необходимой для передачи единицы информации с заданной достоверностью. Для цифровых систем используется нормированное значение соотношения сигнал/шум, представляющее удельный расход энергии на передачу бита данных [2]:

$$\frac{E_b}{N_0} = \frac{S}{N} \cdot \frac{W}{R_b}, \quad (1.12)$$

где $E_b = \frac{S}{R_b}$ – энергия бита данных;

$N_0 = \frac{N}{W}$ – односторонняя спектральная плотность мощности шума.

1.3 Кратко о технологии *LFM_MPSK*

LFM_MPSK технология представляет собой объединение широкополосных сигналов с традиционными методами модуляции узкополосных сигналов. Эта технология сродни технологии прямого расширения спектра, только для расширения спектра информационной последовательности используются *LFM* импульсы. Другими словами, в *LFM_MPSK* технологии вместо монохромных опорных гармонических генераторов используются повторяющихся *LFM* косинусоидальные либо синусоидальные импульсы. Длительность повторяющихся *LFM* импульсов совпадает с длительностью битов, либо модулирующих символов. Соответственно спектр таких сигналов будет определяться параметрами *LFM* импульсов. Для приёма *LFM_MPSK* сигналов используется коррелятор.

Остановимся на формировании повторяющихся *LFM* импульсов на модельном уровне. Простейшая модель формирования включает два генератора повторяющейся последовательности *Repeating Sequence* с двумя параметрами – *Time values* и *Output values*. *Time values* задаётся массивом значений определяющих точки временного интервала. *Output values* также задаётся массивом значений определяющих форму импульса в заданном временном интервале. Интервал времени, задаваемый в обоих генераторах, совпадает с длительностью модулирующего символа. *Output values* первого генератора в форме линейной пилы (возрастающей либо убывающей) задаёт девиацию частоты, например, от ω_{min} до ω_{max} с необходимым числом промежуточных точек. *Output values* второго генератора просто повторяет *Time values*. *Output values* первого генератора можно представить как $\omega_i \cdot t_i$. *Output values* второго генератора можно представить как t_i . Тогда их

произведение представляется как $\omega_i \cdot t_i^2$. Теперь, если подать, сформированные таким образом импульсы в качестве аргумента блока *sin* или *cos*, то на выходе получим повторяющиеся синусоидальные либо косинусоидальные *LFM* импульсы.

Наиболее простая модуляция *LFM_BPSK* представляет собой манипуляцию знака *LFM* импульса. Биполярная информационная последовательность имеет два состояния ± 1 , что соответствует межсигнальному расстоянию равного 2. Для реализации *LFM_BPSK* достаточно перемножить псевдослучайную биполярную информационную последовательность и периодически повторяющиеся *LFM* импульсы. Ширина спектра *LFM_BPSK* сигнала определяется заданной девиацией частоты *LFM* импульса и скоростью перестройки частоты. Для приёма *LFM_BPSK* сигнала достаточно использовать коррелятор. В плечах коррелятора в пределах бита запоминаются отсчеты синхронизированного опорного *LFM* импульса и принятого зашумлённого *LFM_BPSK* модулированного сигнала. Затем накопленные отсчеты скалярно перемножаются (перемножаются и суммируются), масштабируются ограничителем до ± 1 и тем самым восстанавливаются принятые биты.

Квадратурная *LFM_QPSK* модуляция. В фазовом кодере псевдослучайный информационный битовый поток преобразуется в пары (дибиты). Дибиты могут принимать 4-е состояния (00 01 10 11) и каждому из них в соответствии с кодом Грея приписываются фазовые состояния, например, $\varphi_k = \pm \pi/4$, равномерное распределение которых на единичной окружности даёт межсигнальное расстояние $\sqrt{2}$. Вычисляя значения $\cos(\varphi_k)$ и $\sin(\varphi_k)$, получаем амплитуды квадратурных модулирующих импульсов (символов) равные $\pm 1/\sqrt{2}$ длительностью $\tau_s = 2 \cdot \tau_b$. Квадратурные модулирующие импульсы поступают на первые входы умножителей квадратурного модулятора, а на вторые входы умножителей с формирователей поступают ортогональные косинусоидальные и синусоидальные *LFM* импульсы. Квадратурный модулятор заканчивается сумматором.

Прием *LFM_QPSK* модулированных сигналов осуществляется двумя корреляторами для разделения квадратурных потоков. На одном плече модулятора накапливаются отсчеты соответствующего синхронизированного *LFM* импульса, а на другом плече коррелятора запоминаются отсчеты принятого зашумлённого *LFM_QPSK* модулированного сигнала. По окончании модулирующего символа отсчеты скалярно перемножаются (перемножаются и суммируются) и подаются на вход подсистемы фазового декодера. Фазовый декодер, выполненный программно, по принятым квадратурным проекциям восстанавливает квадранты комплексной плоскости, соответствующие фазовые состояния и дибиты в параллельной (векторной форме), а преобразователь из параллельного представления в последовательное – выдаёт последовательно биты принятого дибита.

2-х канальная передача на основе *LFM_QPSK*. Так как фазовый кодер квадратурный *LFM_QPSK* модулятора образует дибиты из информационной последовательности, то можно организовать формирование дибитов из двух независимых информационных потоков. При этом дибиты имеют длительность битов $\tau_s = \tau_b$. Это означает на основе *LFM_QPSK* модуляции можно организовать передачу двух независимых информационных каналов в виде квадратурных потоков длительностью в бит.

В приёмной части с помощью корреляторов квадратурные потоки разделяются и подаются на фазовый декодер. Фазовый декодер, по принятым квадратурным проекциям восстанавливает фазовые состояния и дибиты в параллельной (векторной форме). Первая компонента вектора дибита соответствует восстановленным битам первого информационного источника. Вторая компонента вектора дибита соответствует восстановленным битам второго информационного источника.

Квадратурная *LFM_16PSK* модуляция. В фазовом кодере псевдослучайный информационный битовый поток преобразуется в квадробиты. Квадробиты могут принимать 16 состояний и каждому из них в соответствии с кодом Грея приписываются фазовые

состояния, равномерное распределение которых на единичной окружности даёт межсигнальное расстояние 0.3902 . Вычисляя значения $\cos(\varphi_k)$ и $\sin(\varphi_k)$, получаем амплитуды квадратурных модулирующих импульсов (символов) длительностью $\tau_s = 4 \cdot \tau_b$. Квадратурные модулирующие импульсы поступают на первые входы умножителей квадратурного модулятора, а на вторые входы умножителей с формирователями поступают ортогональные косинусоидальные и синусоидальные *LFM* импульсы. Квадратурный модулятор заканчивается сумматором.

Прием *LFM_16PSK* модулированных сигналов осуществляется двумя корреляторами для разделения квадратурных потоков. На одном плече модулятора накапливаются отсчеты соответствующего синхронизированного *LFM* импульса, а на другом плече коррелятора запоминаются отсчеты принятого зашумлённого *LFM_16PSK* модулированного сигнала. По окончании модулирующего символа отсчеты скалярно перемножаются (перемножаются и суммируются) и подаются на вход подсистемы фазового декодера. Фазовый декодер, выполненный программно, по принятым квадратурным проекциям восстанавливает области комплексной плоскости, соответствующие фазовые состояния и квадробиты в параллельной (векторной форме), а преобразователь из параллельного представления в последовательное – выдаёт последовательно биты принятого квадробита.

4-х канальная передача на основе *LFM_16PSK*. Так как фазовый кодер квадратурный *LFM_16PSK* модулятора образует квадробиты из информационной последовательности, то можно организовать формирование квадробитов из четырёх независимых информационных потоков. При этом квадробиты имеют длительность битов $\tau_s = \tau_b$. Это означает на основе *LFM_16PSK* модуляции можно организовать передачу четырёх независимых информационных каналов в виде квадратурных потоков длительностью в бит.

В приёмной части с помощью корреляторов квадратурные потоки разделяются и подаются на фазовый декодер. Фазовый декодер, по принятым квадратурным проекциям восстанавливает фазовые состояния и квадробиты в параллельной (векторной форме). Номер компоненты вектора квадробита соответствует восстановленным битам соответствующего информационного источника.

2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ LFM_16PSK-МОДЕМА

Источники биполярных информационных последовательностей реализованы на основе генераторов псевдослучайной последовательности с гауссовским распределением **Random Number** и блоков двухстороннего ограничителя на основе функции **sign(x)**, которые нормирует амплитуду сигнала к ± 1 или 0 в зависимости от поступающего сигнала. Длина бита информационной последовательности задаётся параметром **Sample Time=1** генератора **Random Number**. Блоки **Data Type Conversion** осуществляют преобразование типа с **boolean** в тип **double**.

На рисунке 2.1 представлена подсистема **16PSK Code**, которая образует с помощью блока **Mux** вектор дибитов, которые через блок **Zero Order Hold** поступают на вход блока **MatLab function**, содержащего **MatLab** функцию фазового кодера **quadrobit_phase**, представленную на рисунке 2.2, которая назначает дибитам фазовые состояния типа ($\pi/8$; $\pi/4$...). С выхода **MatLab** функции состояния поступают на блоки **cos** и **sin**, далее на умножители квадратурного модулятора.

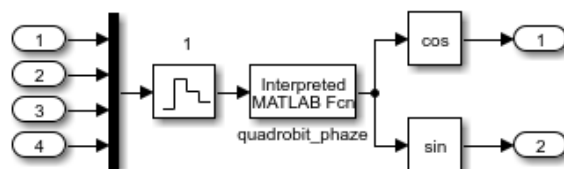


Рисунок 2.1 – Функциональная модель подсистемы **16PSK Code**

```

1 function fi = quadrobit_phase( x )
2
3     if x==[0; 0; 0; 0]; fi=0; end
4     if x==[0; 0; 0; 1]; fi=pi/8; end
5     if x==[0; 0; 1; 1]; fi=2*pi/8; end
6     if x==[0; 0; 1; 0]; fi=3*pi/8; end
7     if x==[0; 1; 0; 0]; fi=4*pi/8; end
8     if x==[0; 1; 0; 1]; fi=5*pi/8; end
9     if x==[0; 1; 1; 1]; fi=6*pi/8; end
10    if x==[0; 1; 1; 0]; fi=7*pi/8; end
11    if x==[1; 0; 0; 0]; fi=pi; end
12    if x==[1; 0; 0; 1]; fi=-7*pi/8; end
13    if x==[1; 0; 1; 1]; fi=-6*pi/8; end
14    if x==[1; 0; 1; 0]; fi=-5*pi/8; end
15    if x==[1; 1; 0; 0]; fi=-4*pi/8; end
16    if x==[1; 1; 0; 1]; fi=-3*pi/8; end
17    if x==[1; 1; 1; 1]; fi=-2*pi/8; end
18    if x==[1; 1; 1; 0]; fi=-pi/8; end
19    end

```

Рисунок 2.2 – Функция **quadrobit_phase**

Формирователи **LFM**-импульсов, изменяющиеся по косинусоидальному и синусоидальному законам выполнены на основе генераторов **Repeating Sequense**. Каждый из формирователей содержит по два генератора: первый задаёт девиацию частоты во время **LFM** импульса, а второй – дополнительное модельное время. После блока **Product** имеем аргумент **LFM** импульса и подаём его на блоки **cos** и, соответственно **sin**, с выходов которых получаем квадратурные **LFM**-импульсы.

Квадратурный *LFM_16PSK* модулятор выполнен на основе умножителей *Product*. На первые входы умножителей поступают опорные косинусоидальные и синусоидальные *LFM* импульсы, а на вторые входы подаются импульсы $\cos(\varphi_k)$ и $\sin(\varphi_k)$ фазовых состояний дибитов. На выходах умножителей после блока *Sum* получаем квадратурно фазоманипулированный *LFM*-импульс.

Простейшая модель канала распространения собрана на основе блока *Sum*, на второй вход которого поступает шумовая псевдослучайная последовательность с гауссовским распределением *Random Number*. Параметр генератора *Sample Time* позволяет реализовать необходимую широкополосность шумов канала распространения, а параметр *Variance* регулировать мощность шумов в процессе измерения помехоустойчивости (зависимости вероятности битовой ошибки P_b от отношения сигнал/шум *SNR*).

Корреляционный прием и обработка сигнала реализованы на основе простейшего коррелятора при сдвиге $\tau = 0$. На каждый информационный канал реализуется свой корреляционный приёмник. Плечи корреляционного приёмника состоят из экстраполяторов нулевого порядка *Zero Order Hold* и блока *Buffer*, которые накапливают в течение бита отсчеты опорного колебания (опорного *LFM* импульса) и принятого в шумах *LFM_16PSK* модулированного радиосигнала. Накопленные отсчеты, как вектора скалярно перемножаются и суммируются блоками *Product* и *Sum*. Блок *Frame Status Conversion* призван преобразовать фреймовый тип данных в тип *double*. В данном случае реализуется накопление 64 отсчетов в течение бита.

На рисунке 2.3 представлен классический фазовый декодер подсистемы *16PSK Decode*, куда поступают квадратурные сигналы с выходов корреляторов каналов. С выходов декодера имеем демодулированные биты информационных каналов в виде вектора. На рисунке 2.4 представлена функция блока *Matlab function*, которая по принятым проекциям определяет области фазовой плоскости и восстанавливает квадробиты.

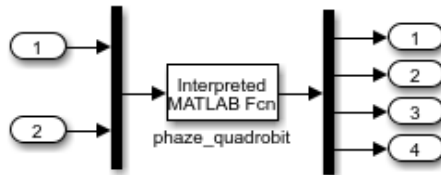


Рисунок 2.3 – Функциональная модель подсистемы *16PSK Decode*

```

1 function [ y ] = phase_quadrobit( x )
2
3 f1=atan2(x(2),x(1));
4 if (f1>=pi/16) && (f1<pi/16); y=[0 0 0 0]; end
5 if (f1>=pi/16) && (f1<3*pi/16); y=[0 0 0 1]; end
6 if (f1>=3*pi/16) && (f1<5*pi/16); y=[0 0 1 1]; end
7 if (f1>=5*pi/16) && (f1<7*pi/16); y=[0 0 1 0]; end
8 if (f1>=7*pi/16) && (f1<9*pi/16); y=[0 1 0 0]; end
9 if (f1>=9*pi/16) && (f1<11*pi/16); y=[0 1 0 1]; end
10 if (f1>=11*pi/16) && (f1<13*pi/16); y=[0 1 1 1]; end
11 if (f1>=13*pi/16) && (f1<15*pi/16); y=[0 1 1 0]; end
12 if ((f1>=15*pi/16) && (f1<16*pi/16)) || ((f1>=-16*pi/16) && (f1<=-15*pi/16)); y=[1 0 0 0]; end
13 if (f1>=-15*pi/16) && (f1<=-13*pi/16); y=[1 0 0 1]; end
14 if (f1>=-13*pi/16) && (f1<=-11*pi/16); y=[1 0 1 1]; end
15 if (f1>=-11*pi/16) && (f1<=-9*pi/16); y=[1 0 1 0]; end
16 if (f1>=-9*pi/16) && (f1<=-7*pi/16); y=[1 1 0 0]; end
17 if (f1>=-7*pi/16) && (f1<=-5*pi/16); y=[1 1 0 1]; end
18 if (f1>=-5*pi/16) && (f1<=-3*pi/16); y=[1 1 1 1]; end
19 if (f1>=-3*pi/16) && (f1<=-pi/16); y=[1 1 1 0]; end
20 end

```

Рисунок 2.4 – Функция *phase_quadrobit*

Блоки *Display* и *Scope* призваны отображать число ошибок передачи и фиксации момента их возникновения.

Блок *Scope* призваны вычислять и отображать изменение уровня мощности сигнала

или смеси сигнал+шум в процессе измерения помехоустойчивости. Точка подключения измерителя мощности обусловлена блоком принятия решений, который соответствует блоку ***Zero Order Hold***.

Блоки ***Spectrum Scope*** и ***Averaging Power Spectral Density*** призваны отображать спектры сигнальных и шумовых потоков.

Такая упрощенная реализация корреляционной обработки позволяет найти отсчет автокорреляционной функции опорного колебания и зашумленного радиосигнала при смещении $\tau = 0$ и в зависимости от знака корреляции принять решение о приеме ***1*** или ***-1***.

На рисунке 2.5 представлена имитационная четырехканальная модель ***LFM_16PSK*** модема с корреляционными приёмниками.

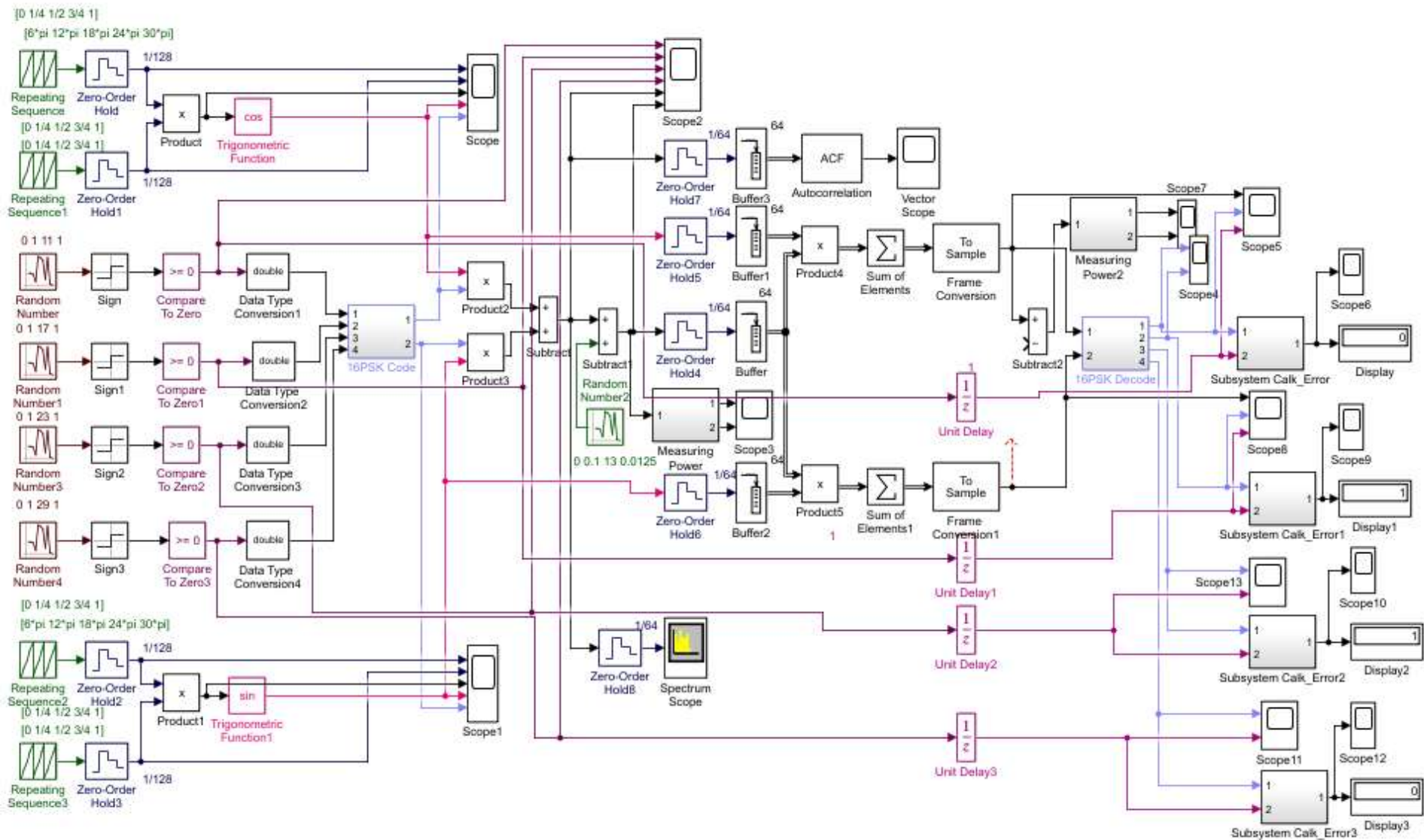


Рисунок 2.5 – Четырехканальная модель *LFM_16PSK* модема с корреляционными приёмниками

Модель измерителя мощности. На рисунке 2.6 представлена подсистема измерителя мощности сигнала и шума *Subsystem Measuring Power*.

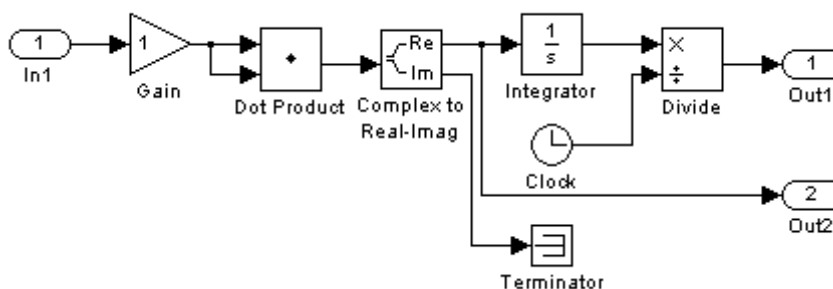


Рисунок 2.6 – Подсистема измерителя мощности

На вход подсистемы поступает сигнал, блок *Dot Product* скалярно перемножает входной вектор на сопряженный. Блок *Complex to Real-Imag* выделяет вещественную часть. Блок *Integrator* вычисляет энергию процесса, а деление блоком *Product* энергии на время (блок *Clock*) вычисляет мощность как скорость поступления энергии.

Модель подсчета ошибок. На рисунке 2.7 представлена подсистема подсчета ошибок *Subsystem Calc Err*.

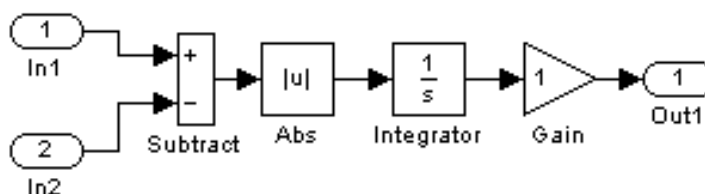


Рисунок 2.7 – Подсистема подсчета ошибок

На первый вход подается сигнал с декодера, на второй вход подан сигнал с генератора входной информационной последовательности с необходимой задержкой. В детекторе ошибок вычисляется интеграл от модуля разности входных последовательностей. Затем, в зависимости от длительности битов и одно- или биполярности подбирается множитель *Gain*, переводящий интеграл разности в эквивалентное количество битов (ошибок).

Модель коррелятора. Корреляционный приемник реализован следующим образом (см. рис. 2.8):

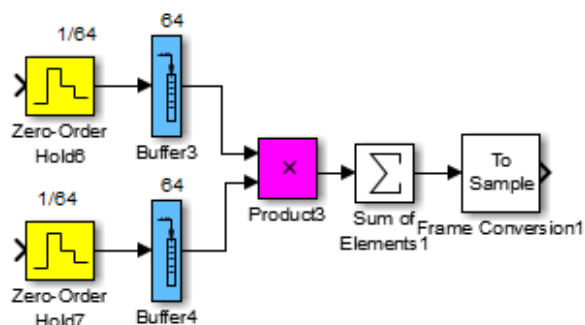


Рисунок 2.8 – Реализация корреляционного приемника

При реализации функциональной модели корреляционного приема решено ограничиться значением функции корреляции при $\tau = 0$.

Т.к. используются дискретные (цифровые) сигналы, то значение функции корреляции вычисляется через сумму произведений отсчетов принятого сигнала и колебаний опорного формирователя *LFM* импульсов. Модель коррелятора содержит два плеча накопления отсчетов. В каждом плече с помощью блоков *Zero-Order Hold* сигнал дискретизируется на отсчеты (в данном случае на **64**) и накапливается в блоках *Buffer*. Далее накопленные отсчеты попарно перемножаются в блоке *Product*. Затем вычисляется сумма этих произведений блоком *Sum*, т.е. реализуется скалярное произведение векторов накопленных отсчетов.

Измерение SNR. Отношение сигнал/шум (*SNR*) измеряется по следующей методике: для того, чтобы определить мощность сигнала без шумовой составляющей **S** необходимо отсоединить генератор шума от модели канала распространения. Для того чтобы определить мощность смеси полезного сигнала с шумом **SN** генератор шумов должен быть подключен к каналу распространения. Изменение отношения сигнал/шум **SNR** достигается вариацией параметра дисперсии генератора псевдослучайной гауссовской последовательности модели канала распространения.

SNR рассчитывается по формуле, представленной ниже:

$$\frac{S}{(SN-S)} = \mathbf{SNR} \text{ (дБ)}. \quad (2.1)$$

При большом числе испытаний, частота появлений битовых ошибок, т.е. отношение числа ошибок к общему числу битов, стремится к вероятности битовых ошибок **P_b**.

Таким образом, измерение **SNR** при соответствующем числе ошибок позволяет определить точки водопадоподобной характеристики, позволяющей оценить помехоустойчивость модема.

3 ПРИНЦИП РАБОТЫ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ LFM_16PSK-МОДЕМА

Опишем принцип работы *LFM-16PSK* модема по функциональной модели, представленной на рис. 2.5. Источники на основе генераторов *Random Number* формируют псевдослучайный информационные однополярные битовые потоки.

Фазовый кодер *16PSK Code* (см. рис. 2.1) формирует дибиты, назначает им фазовые состояния φ_k (см. текст на рис. 2.2), вычисляет значения $\cos(\varphi_k)$ и $\sin(\varphi_k)$ создаёт управляющие квадратурные символы.

На основе генераторов *Repeating Sequence* выполнены формирователи *LFM*-импульсов, изменяющиеся по косинусоидальному и синусоидальному законам. Каждый из формирователей содержит по два генератора: первый задаёт девиацию частоты, во время *LFM* импульса, а второй – дополнительное модельное время. Блоки *Product* реализуют аргумент *LFM* импульса и подают его на блоки *cos* и, соответственно *sin*, с выходов которых получаем квадратурные *LFM*-импульсы.

Квадратурный *LFM_16PSK* модулятор в каждом канале перемножает опорные косинусоидальные и синусоидальные *LFM* импульсы, и импульсы $\cos(\varphi_k)$ и $\sin(\varphi_k)$ фазовых состояний дибитов. После суммирования сигналов плеч модулятора получаем квадратурно фазоманипулированный *LFM*-импульс.

Простейшая модель канала распространения добавляет в канал передачи шумовую псевдослучайную последовательность с гауссовским распределением. Параметр генератора *Sample Time* позволяет реализовать необходимую широкополосность шумов канала распространения, а параметр *Variance* регулировать мощность шумов в процессе измерения помехоустойчивости (зависимости вероятности битовой ошибки P_b от отношения сигнал/шум *SNR*).

Корреляционный прием и обработка сигнала реализованы на основе простейшего коррелятора при сдвиге $\tau = 0$. На каждый информационный канал реализуется свой корреляционный приёмник. Плечи корреляционного приёмника накапливают в течение бита отсчеты опорного колебания (опорного *LFM* импульса) и принятого в шумах *LFM_16PSK* модулированного радиосигнала. Накопленные отсчеты, как вектора скалярно перемножаются и суммируются блоками *Product* и *Sum*. Блок *Frame Status Conversion* призван преобразовать фреймовый тип данных в тип *double*. В данном случае реализуется накопление **64** отсчетов в течение бита.

Принятые сигналы квадратурных каналов приёмника с выходов корреляторов подаются на классический фазовый декодер (см. рис. 2.3). С выходов декодера имеем демодулированные биты информационных каналов в виде вектора. На рисунке 2.4 представлена функция блока *Matlab function*, которая по принятым проекциям определяет области фазовой плоскости и восстанавливает дибиты.

Блок *Scope*, установленный на выходе, позволяет наблюдать и сравнивать принятый информационный поток и подаваемый на второй вход передаваемый информационный поток.

Для иллюстрации работы модема на рисунке 3.1 приведены фрагменты осциллограмм формирователя косинусоидальных *LFM* импульсов и импульсов на выходе плеча квадратурного модулятора.

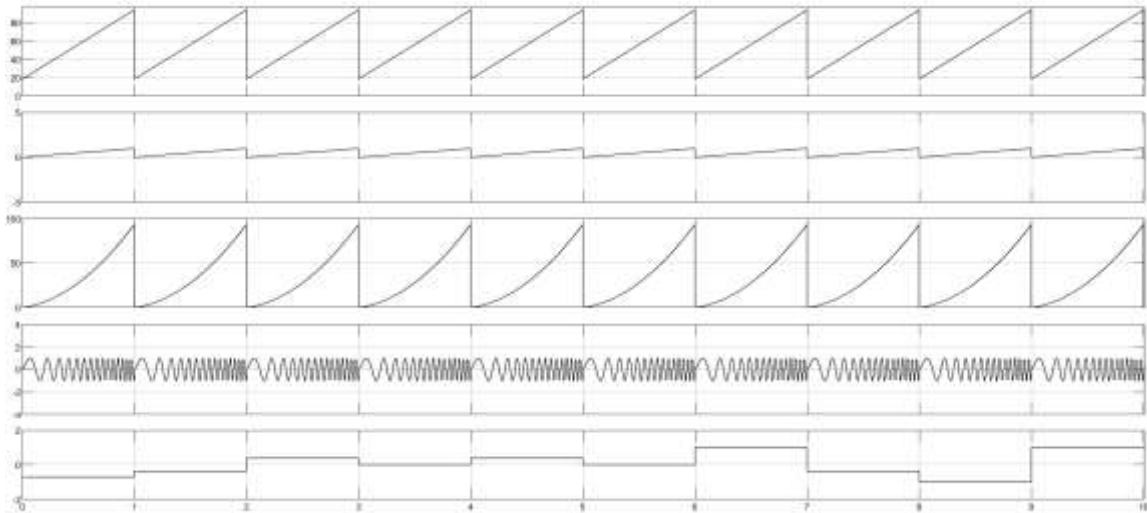


Рисунок 3.1 - Фрагменты осциллограмм формирователя косинусоидальных *LFM* импульсов и импульсов на выходе плеча квадратурного модулятора (1 - импульсы девиации частоты; 2 - импульсы модельного времени; 3 - импульсы аргумента косинусоидального *LFM* импульса; 4 - косинусоидальные *LFM* импульсы; 5 - импульсы на выходе плеча квадратурного модулятора)

На рисунке 3.2 приведены фрагменты осциллограмм информационных импульсов каналов передачи и *LFM* импульсов до и после канала распространения.

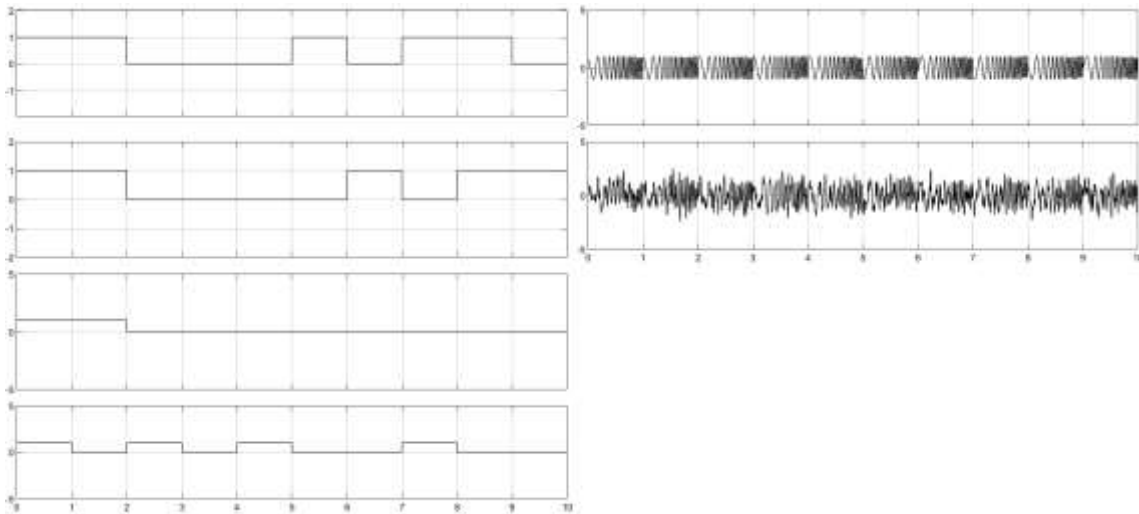


Рисунок 3.2 - Фрагменты осциллограмм информационных импульсов каналов передачи и *LFM* импульсов до и после канала распространения

На рисунке 3.3 приведены фрагменты осциллограмм принятых коррелятором сигналов первого квадратурного канала, принятые и передаваемые биты первого канала.

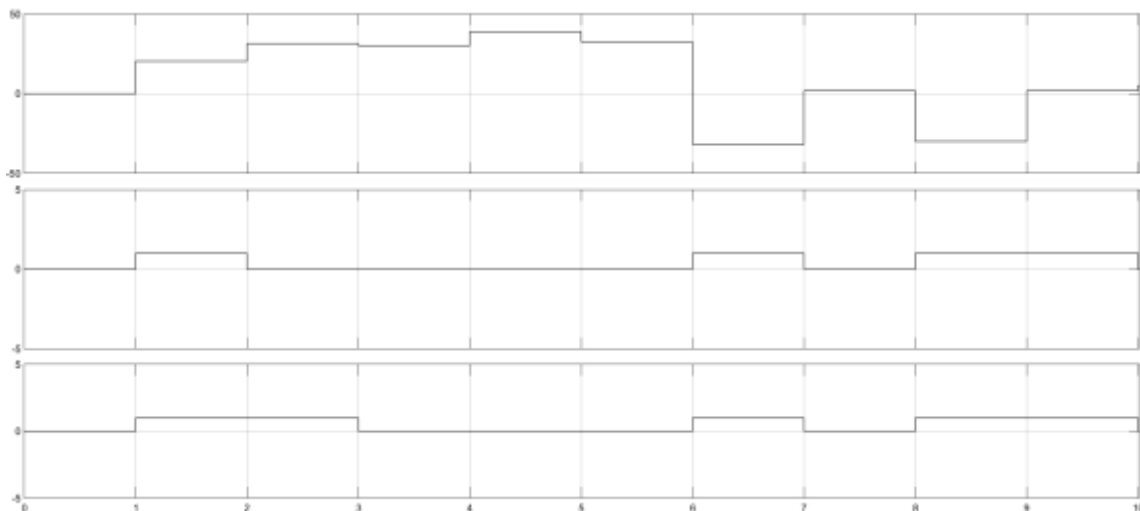


Рисунок 3.3 - Фрагменты осциллограмм принятых коррелятором сигналов первого квадратурного канала, принятые и передаваемые биты первого канала

На рисунке 3.4 приведена автокорреляционная функция косинусоидальных *LFM* импульсов.

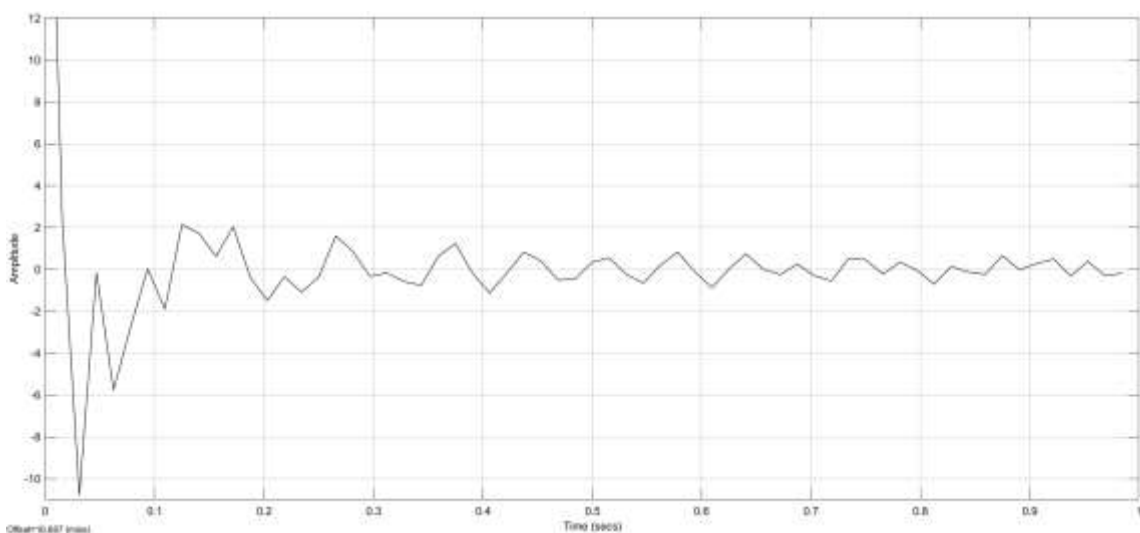


Рисунок 3.4 – Автокорреляционная функция косинусоидальных *LFM* импульсов

На рисунке 3.5 приведен спектр **LFM-16PSK** модулированного сигнала.

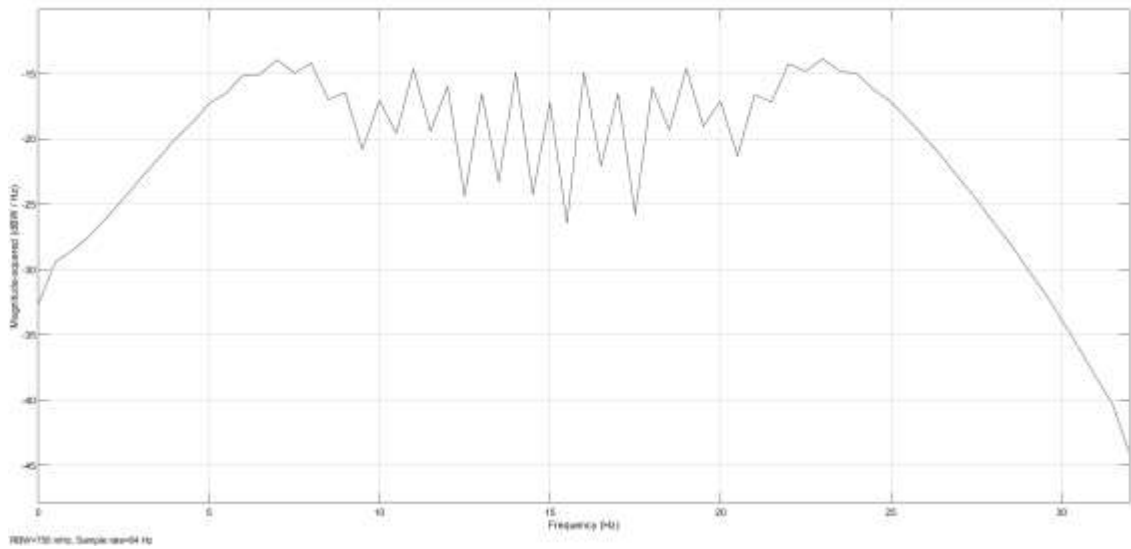


Рисунок 3.5 – Спектр *LFM-16PSK* модулированного сигнала

Приведённые осциллограммы модельного исследования *LFM-16PSK* четырехканального модема призваны ориентировать студента в процессе выполнения лабораторной работы.

4 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK И ИСПОЛЬЗУЕМЫХ БЛОКОВ

Пакет *Simulink* разработан компанией *Mathworks* и распространяется в составе математического пакета *MatLab*. *Simulink* – это основной инструмент для модельно-ориентированного проектирования. При помощи *Simulink* можно понимать и анализировать сложные системы путем симуляции блочных диаграмм. Из модели можно генерировать код для прототипирования и тестирования в реальном времени и развертывания на встраиваемых системах.

Simulink обладает обширной библиотекой готовых блоков с модифицируемыми параметрами для построения моделей рассматриваемых систем и наглядными средствами визуализации результатов моделирования.

4.1 Запуск и работа с пакетом *Simulink*

Чтобы начать работу в *Simulink*, нужно запустить систему *MatLab*. Затем нажать



кнопку *(Simulink)* на панели инструментов системы *MatLab*; Либо, открыть уже готовую модель, выполнив опцию *Open* в меню *File* и открыть файл модели (*mdl*- файл). На рисунке 4.1 представлено окно обозревателя библиотеки блоков *Simulink Library Browser*.

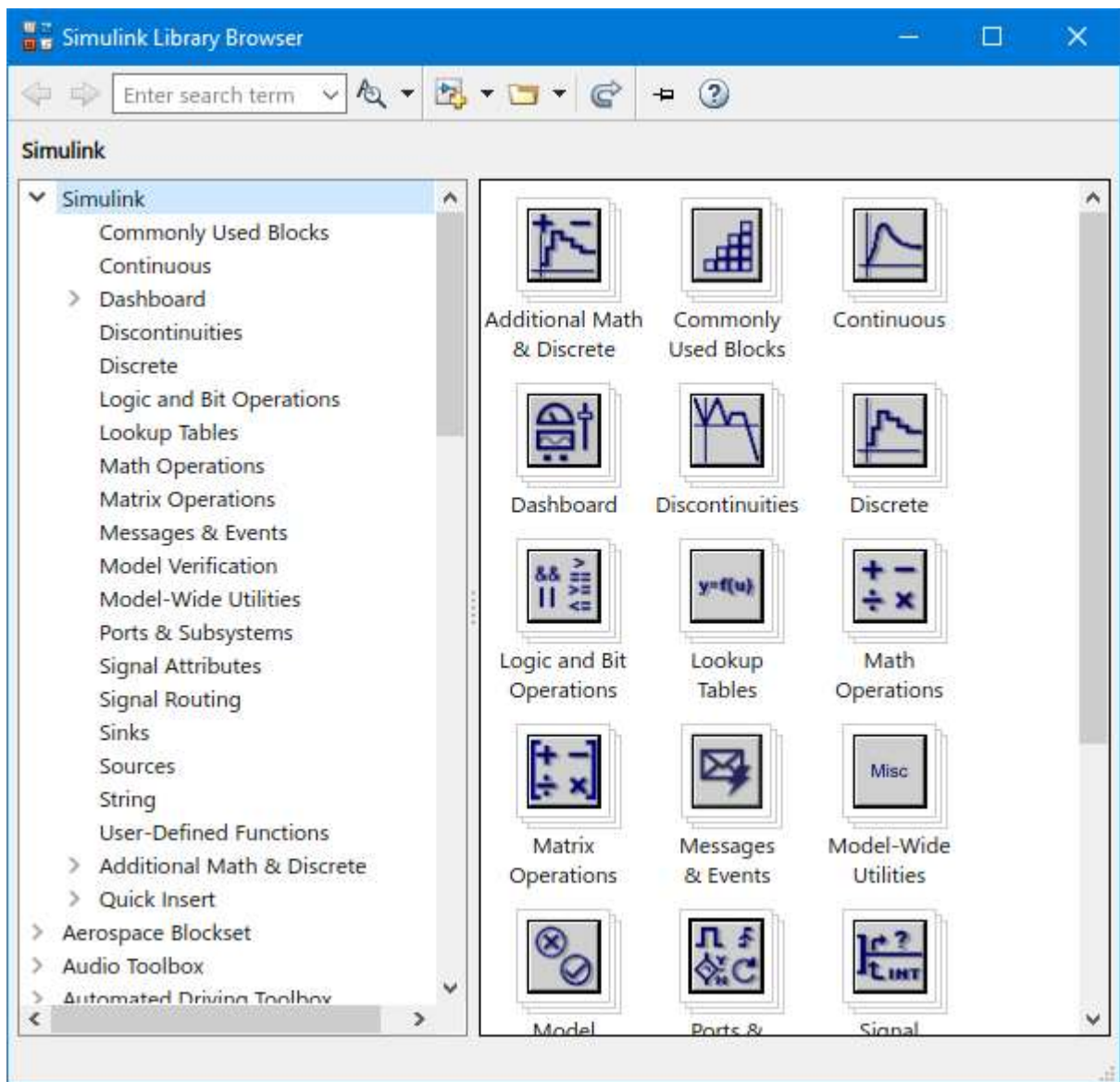


Рисунок 4.1 – Библиотека блоков *Simulink Library Browser*

В левой части окна выведена библиотека системы *Simulink*, а в правой части показаны ее разделы. Основная библиотека системы содержит следующие разделы:

- ***Continuous*** – блоки аналоговых элементов;
- ***Discontinuous*** – блоки нелинейных элементов;
- ***Discrete*** – блоки дискретных элементов;
- ***Look-Up Tables*** – блоки таблиц;
- ***Math Operations*** – блоки элементов, определяющие математические операции;
- ***Model Verification*** – блоки проверки свойств сигнала;
- ***Model-Wide Utilities*** – раздел дополнительных утилит;
- ***Port & Subsystems*** – порты и подсистемы;
- ***Signal Attributes*** – блоки маршрутизации сигналов;
- ***Signal Routing*** – блоки маршрутизации сигналов;

- *Sinks* – блоки приема и отображения сигналов;
- *Sources* – блоки источников сигнала;
- *User-Defined Function* – функции, определяемые пользователем.

4.2 Описание используемых блоков библиотеки Simulink

Ниже описаны основные блоки базовых разделов библиотеки *Simulink*, используемые в функциональной схеме *LFM_16PSK*-модема:



Repeating Sequence - выводит периодический скалярный сигнал, имеющий форму волны, что вы задаете использование параметры **Output values** и **Time values**.

Параметр **Time values** задает вектор выходных времен. Параметр **Output values** задает вектор из амплитуд сигнала в соответствующие выходные времена. Вместе, эти два параметра задают выборку выходной формы волны в точках, измеренных с начала интервала, на котором форма волны повторяется (период сигнала).



Random Number – блок источника случайного дискретного сигнала с нормальным распределением. Назначение: формирование случайного сигнала с нормальным распределением уровня сигнала. Параметры блока: **Mean** - среднее значение сигнала, **Variance** - дисперсия (среднеквадратическое отклонение), **Initial seed** – начальное значение.

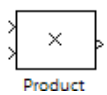
Zero-Order Hold – экстраполятор нулевого порядка. Назначение: экстраполяция входного сигнала на интервале дискретизации. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания интервала дискретизации. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге дискретизации. Параметры блока: **Sample time** – такт дискретности. Блок экстраполятора нулевого порядка может использоваться также для согласования работы дискретных блоков, имеющих разные такты дискретности.



Sign – блок определения знака сигнала. Назначение: определяет знак входного сигнала, при этом, если x – входной сигнал, то сигнал на выходе определяется выражением:

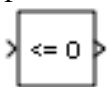
$$\text{sign} = \begin{cases} -1, & \text{где } x < 0; \\ 0, & \text{где } x = 0; \\ 1, & \text{где } x > 0. \end{cases}$$

Параметры блока: флажок **Enable zero crossing detection** позволяет фиксировать прохождение сигнала через нулевой уровень.



Product – блок умножения и деления. Назначение: вычисление произведения текущих значений сигналов. Параметры блока: **Number of inputs** – количество входов, может задаваться как число или как список знаков. В списке знаков можно использовать знаки: * – умножить и / – разделить. **Multiplication** – способ выполнения операции, может принимать значения из списка: **Element-wise** – поэлементный; **Matrix** – матричный. Флажок **Show additional parameters** – показать дополнительные параметры. При выставленном флажке отображается окно списка **Output data type mode**, в нашем случае флажок не используется.

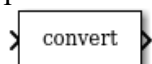
Compare To Zero - Блок **Compare To Zero** сравнивает входной сигнал, чтобы обнулить. Задайте, как вход сравнивается с нулем параметром **Operator**. Выходом является 0 если сравнение является ложным, и 1 если это верно.



Trigonometric Function - выполняет общие тригонометрические функции и выводит результат в рад или версии. Параметры: **Function** — тригонометрическая функция, **Approximation method** — **CORDIC**, поиск или ни один, **Angle unit** — угловой модуль, **Number of data points** — количество точек данных для интерполяционной

sin

таблицы, **Output signal type** — сложность выходного сигнала, **Remove protection against out-of-range input** — удалите защиту от входа из области значений, **Sample time** — задайте шаг расчета как значение кроме **-1**.



Data Type Conversion – блок преобразования типа сигнала. Назначение:

блок преобразует тип входного сигнала. Параметры: **Data type** – тип данных выходного сигнала. Может принимать значения (выбираются из списка): **auto, double, single, int8, int16, int32, uint8, uint16, uint32 u boolean**. **Saturate on integer overflow** (флажок) – подавляет переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.



Subsystem – виртуальная и монолитная подсистемы. Доступ к окну

параметров подсистемы осуществляется через меню **Edit** командой **Block Parameters**. Параметры: **Show port labels** – показать метки портов, **Treat as atomic unit** (флажок) – считать подсистему монолитной. Таким образом, блоки виртуальной и монолитной подсистем – это один и тот же блок, отличающийся значением данного параметра. **Access** – доступность подсистемы для изменений. Выбирается из списка: **ReadWrite** – пользователь может открывать и изменять подсистему, **ReadOnly** – пользователь может открывать подсистему только для просмотра, **NoReadOrWrite** – пользователь не может открывать и изменять подсистему; **Name of error callback function** – имя функции, используемой для обработки ошибок, возникающих в данной подсистеме.



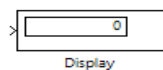
Unit delay – блок единичной дискретной задержки. Назначение: выполняет

задержку дискретного сигнала на заданный шаг модельного времени. Параметры блока: **Initial conditions** – начальное значение выходного сигнала; **Sample time** – шаг модельного времени.



Scope – блок осциллографа. Назначение: построение графиков исследуемых

сигналов как функций времени. Открытие окна осциллографа производится двойным щелчком **ЛКМ** на пиктограмме блока. В случае векторного сигнала каждая компонента вектора отображается отдельным цветом. Настройка окна осциллографа выполняется с помощью панелей инструментов, позволяющих: осуществить печать содержимого окна осциллографа; установить параметры, в частности, **Number of axes** – число входов осциллографа, **Time range** – отображаемый временной интервал и другие; изменить масштабы графиков; установить и сохранить настройки; перевести в плавающий режим и так далее.



Display – блок цифрового дисплея. Назначение: отображает значение

сигнала в виде числа. Параметры: **Format** – формат отображения данных. Параметр **Format** может принимать следующие значения: **short** – 5 значащих десятичных цифр, **long** – 15 значащих десятичных цифр, **short_e** – 5 значащих десятичных цифр и 3 символа степени десяти, **long_e** – 15 значащих десятичных цифр и 3 символа степени десяти, **bank** – "денежный" формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа; **Decimation** – кратность отображения входного сигнала, при **Decimation = 1** отображается каждое значение входного сигнала, при **Decimation = 2** отображается каждое второе значение, при **Decimation = 3** – каждое третье значение и т.д.; **Sample time** – шаг модельного времени. Определяет дискретность отображения данных; **Floating display** (флажок) – перевод блока в "свободный" режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения выполняется щелчком **ЛКМ** на соответствующей линии связи. В этом режиме для параметра расчета **Signal storage reuse** должно быть установлено значение **off** (вкладка **Advanced** в окне диалога **Simulation parameters...**).



MatLab Fcn – блок задания функции. Назначение: задает выражение в

стиле языка программирования **MatLab**. Параметры: **MatLab function** –

Выражение на языке *MatLab*. **Output dimensions** – размерность выходного сигнала. Значение параметра минус *1* предписывает блоку определять размерность автоматически. **Output signal type** – тип выходного сигнала. Выбирается из списка: *real* – действительный сигнал, *complex* – комплексный сигнал, *auto* – автоматическое определение типа сигнала; **Collapse 2-D results to 1-D** – преобразование двумерного выходного сигнала к одномерному.



Demux - извлекает компоненты сигнала входного вектора и выходных параметров отдельные сигналы. Порты выходного сигнала упорядочены сверху донизу.



Mux - комбинирует входные параметры с совпадающим типом данных и сложностью в векторный выход. Выходной сигнал мультиплексора является плоским, даже если вы создаете сигнал мультиплексора из других сигналов мультиплексора. Однако можно использовать несколько блоков **Mux**, чтобы создать сигнал мультиплексора шаг за шагом. Сигнал мультиплексора упрощает общий вид модели путем объединения двух или больше сигнальных линий в одну линию. Сигналы мультиплексора не влияют на симуляцию или генерацию кода.



Gain – блок усилителя. Назначение: блок **Gain** умножает входной сигнал на постоянный коэффициент; Параметры блока: **Multiplication** – способ выполнения операции, значение параметра выбирается из списка: **Element-wise $K*u$** – поэлементный; **Matrix $K*u$** – матричный, коэффициент усиления является левосторонним оператором; **Matrix $u*K$** – матричный, коэффициент усиления является правосторонним оператором; **Matrix $K*u$** (*u*-вектор) – векторный, коэффициент усиления является левосторонним оператором. Флажок **Show additional parameters** – показать дополнительные параметры, при выставленном флажке отображаются окна списков **Parameter data type mode**, **Output data type mode**. **Saturate on integer** – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.



Clock – Источник времени. Назначение: Формирует сигнал, величина которого на каждом шаге равна текущему значению моделирования. Параметры блока: **Decimation** - Шаг, с которым обновляются показания времени на изображении источник. Параметр задается как количество шагов расчета.

5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

Исходные данные.

В источниках биполярных информационных последовательностей в блоках **Random Number** заданы параметры (в порядке расположения окон) **0 1 11 1**, **0 1 17 1**, **0 1 23 1** и **0 1 29 1** соответственно, длина бита информационной последовательности задаётся параметром **Sample Time=1**. В формирователе **LFM**-импульсов для блока **Repeating Sequence** заданы параметры **Time values** [**0 1/4 1/2 3/4 1**] – как модельного времени и **Output values** [**6*pi 12*pi 18*pi 24*pi 30*pi**] - как девиации частоты и второго генератора **Repeating Sequence** с параметрами **Time values** [**0 1/4 1/2 3/4 1**] и **Output values** [**0 1/4 1/2 3/4 1**] – как дополнительного модельного времени, а для блока **Zero Order Hold** параметр **Sample Time** равен **1/128**. В плечах корреляционного приёмника параметр **Sample Time** равен **1/64** блока **Zero Order Hold** и **Output buffer size** равен **64** блока **Buffer**. В шумовой псевдослучайной последовательности с гауссовским распределением **Random Number** заданы параметры **0 0.1 13 0.0125**.

Экспериментальное задание.

1. Написать **Matlab**-функции, показанные на рисунках 2.2 и 2.4.
2. Собрать **Sim**-модель **LFM_16PSK** модема в соответствии с рисунком 2.5.
3. Определить отношение сигнал/шум (**SNR**) в одном из каналов при вероятности битовой ошибки $P_b = 10^{-3}$ и оценить помехоустойчивость модема.
4. Сравнить полученные результаты с результатами **LFM_QPSK** модема, у которого $SNR \sim 10.9 \text{ dB}$ при $P_b = 10^{-3}$ и сделать выводы.
5. Составить отчёт по проделанной работе.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как реализуется **16PSK** манипуляция?
2. Как реализуется **LFM**-модуляция?
3. В чём суть **LFM_16PSK** технологии?
4. Как формируются **LFM** импульсы?
5. Чем отличается **QPSK** от **16PSK**?
6. Как устроен квадратурный модулятор?
7. Как устроен корреляционный приёмник?
8. Назначение фазового кодера?
9. Назначение фазового декодера?
10. Опишите методику измерения отношения сигнал/шум.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Антипенский Р. Моделирование источников сложных сигналов // Журнал «Современная электроника», 2007, №9.– С. 47-49.
2. Баранина В.Е. Модельное исследование вариантов организации разноскоростных каналов передачи в QPSK-модеме. – Томск: ТУСУР, 2020.- 53 с.
3. Бронштейн И.Н., Семендяев К.А. Б 88. «Справочник по математике для инженеров и учащихся втузов» Учебное пособие. — СПб.: Издательство «Лань», 2010. 608 с.
4. Леонидов В.В. Конспект лекций «Модуляция и демодуляция цифровых сигналов». Учебно-методический комплект по дисциплине «Цифровая обработка». МГТУ имени Н.Э. Баумана. Электронный адрес: <https://leonidov.su/wp-content/uploads/2020/04/Modulation-and-Demodulation-of-Digital-Signals-Lecture-V.V.-Leonidov.pdf> (дата обращения 7.11.22).
5. Лукашин И.В. Исследование применения ЛЧМ-сигналов для передачи данных по радиоканалу. Электронный адрес: // <http://omoled.ru/publications/view/414> (дата обращения 7.11.22).
6. Речкина Д.А. Модем на основе LFM-BPSK сигналов. ВКР бакалавра. - Томск: ТУСУР, 2022.- 55 с.