

Министерство науки и высшего образования Российской Федерации

Томский государственный университет систем
управления и радиоэлектроники

Кологривов В. А.

Баранина В.Е.

**ИССЛЕДОВАНИЕ QPSK-МОДЕМА ПРИ МУЛЬТИПЛЕКСИРОВАНИИ
РАЗНОСКОРОСТНЫХ СИГНАЛЬНЫХ ПОТОКОВ ДАННЫХ**

Методические указания по лабораторной работе в среде функционального
моделирования Simulink системы MatLab

2020

УДК 621.376
ББК 32.811.7
К 611

Рецензент:

Захаров Ф.Н., доцент кафедры радиотехнических систем ТУСУР, канд. техн. наук

Авторы:

Кологривов В.А., Баранина В.Е

Кологривов В.А., Баранина В.Е.

К 611 «Исследование QPSK-модема при мультиплексировании разноскоростных сигнальных потоков данных»: методические указания по лабораторной работе для студентов направления «Инфокоммуникационные технологии и системы связи» - Томск: ТУСУР. Образовательный портал, 2020.– 23 с.

Методические указания содержат описание функциональной модели QPSK-модема с мультиплексированием разноскоростных сигнальных потоков данных, разработанном в среде Simulink пакета прикладных программ MatLab.

В данном пособии приведены краткие сведения из теории квадратурной фазовой модуляции и демодуляции, краткая характеристика функциональных блоков библиотеки среды Simulink пакета прикладных MatLab, описание функциональной схемы модели, а также требования к экспериментальному исследованию и контрольные вопросы для допуска к выполнению лабораторной работы.

Одобрено на заседании каф. РТС протокол № 9 от 02.06.2020

УДК 621.376
ББК 32.811.7

© Кологривов В.А., Баранина В.Е., 2020
© Томск. гос. ун-т систем упр. И
радиоэлектроники, 2020

АННОТАЦИЯ

Лабораторная работа «**Исследование QPSK-модема при мультиплексировании разноскоростных сигнальных потоков данных**» посвящена экспериментальному исследованию модели QPSK-модема с совмещенными каналами передачи данных, в среде Simulink пакета прикладных программ MatLab.

В описании сформулирована цель лабораторной работы, приведены необходимые сведения из теории QPSK-модуляции, краткая характеристика среды Simulink пакета прикладных программ MatLab, описание виртуального лабораторного макета и используемых блоков библиотеки Simulink, а также требования к экспериментальному исследованию и контрольные вопросы для допуска к выполнению лабораторной работы.

Исследование проводится по функциональной схеме, построенной на основе блоков базовых разделов библиотеки Simulink.

ОГЛАВЛЕНИЕ

1 ЦЕЛЬ РАБОТЫ, КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ.....	5
2 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK	9
2.1 Общая характеристика пакета SIMULINK.....	9
3 ОПИСАНИЕ ЛАБОРАТОРНОГО МАКЕТА.....	10
4 ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ БЛОКОВ БИБЛИОТЕКИ SIMULINK	16
5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ	21
6 КОНТРОЛЬНЫЕ ВОПРОСЫ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	23

1 ЦЕЛЬ РАБОТЫ, КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Цель работы: изучение структуры и принципа работы варианта QPSK-модема с мультиплексированием разноскоростных сигнальных потоков данных, разработанной в среде Simulink пакета прикладных программ MatLab.

Теоретические сведения

Вариант реализации функциональной Sim-модели квадратурной QPSK (Quadrature Phase Shift Key) квадратурной фазовой манипуляции (модуляции) и кодовым принципом модуляции и демодуляции фазы несущей частоты приведен на рисунке 1.1.

Входной поток битовых импульсов группируется в дибиты (символы). Дибиты, преобразуются вначале в импульсы по амплитуде пропорциональные фазовому состоянию φ_k .

Для обеспечения равного кодового расстояния между составляющими сигнального пространства при кодировании используется код Грея. В результате предварительного этапа QPSK манипуляции реализуется четыре фазовых состояния для текущего символа (дибита). Соответствие дибитов и фазовых состояний с учетом согласованного кодирования по Грею приведено в таблице 1.

Таблица №1 – Кодирование дибитов при QPSK

Модулирующие символы (дибиты) и соответствующие им фазовые состояния			
0_0	0_1	1_0	1_1
$\pi/4$	$3 \cdot \pi/4$	$7 \cdot \pi/4$	$5 \cdot \pi/4$

Далее, фазовые состояния преобразуются в предварительные квадратурные модулирующие импульсы пропорциональные $d_I = \cos(\varphi_k)$ и $d_Q = \sin(\varphi_k)$, совпадающие по длительности с дибитом.

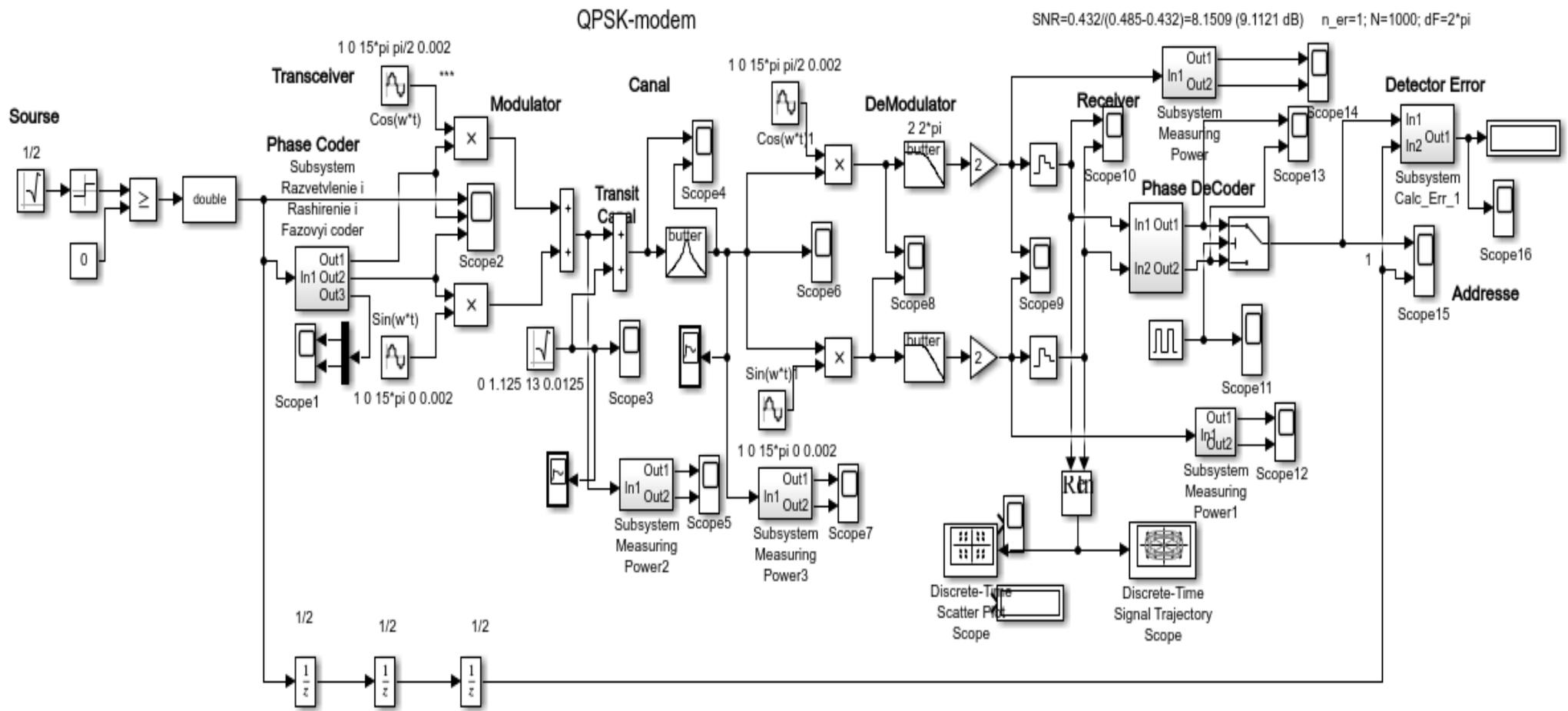


Рисунок 1.1 – Sim-модель модема QPSK квадратурной фазовой манипуляции и кодовым принципом модуляции и демодуляции фазы несущей частоты

Модулятор. Фазовый кодер, как составная часть квадратурного модулятора может быть представлен в виде функциональной схемы (см. рисунок 1.2.).

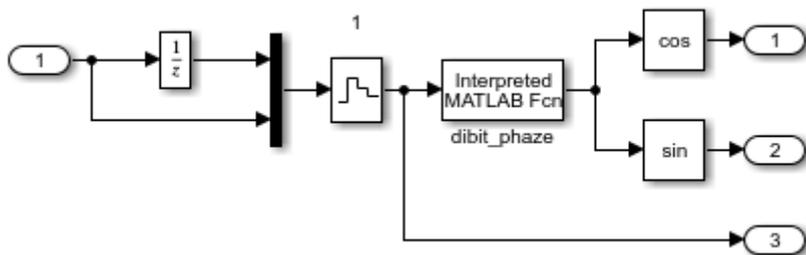


Рисунок 1.2 – Функциональная схема фазового кодера квадратурного модулятора

Здесь преобразование битовой последовательности импульсов в векторное представление дибитов реализовано путем подачи текущего и задержанного на один такт предыдущего импульсов на блок мультиплексора (*Mux*). С выхода мультиплексора векторное представление дибитовых импульсов подается на блок экстраполятора нулевого порядка *Zero-Order Hold*, который выполняет дискретизацию входного сигнала по времени и расширение длительности до дибита. Встроенная М-функция *dibit_phase.m* сопоставляет значениям входного дибита текущее фазовое состояние в соответствии с кодом Грея таблице 1.

Скрипт-код М-функции (фазового кодера QPSK) представлен ниже:

```
function fi=dibit_phase(x)
% Преобразование дибита в фазу
% x- вектор (дибит)
% fi- фаза
if x==[0; 0]; fi=pi/4; end
if x==[0; 1]; fi=3*pi/4; end
if x==[1; 1]; fi=-3*pi/4; end
if x==[1; 0]; fi=-pi/4; end
```

Заключительный этап квадратурной фазовой манипуляции реализован путем умножения текущих значений квадратурных составляющих I и Q несущей частоты $\cos(w \cdot t)$ и $\sin(w \cdot t)$ соответственно на значения $\cos(\varphi_k)$ и $\sin(\varphi_k)$, определяемые текущим фазовым состоянием. Квадратурная модуляция завершается суммированием квадратурных составляющих.

С выхода модулятора модулированный сигнал поступает в канал распространения, где к нему добавляются гауссовские шумы [1]. Выходной сигнал модулятора можно представить в виде:

$$U_k = d_I \cdot \cos(w \cdot t) + d_Q \cdot \sin(w \cdot t). \quad (1.1)$$

Демодулятор. На первые входы умножителей QPSK демодулятора подается принятый сигнал, а на вторые входы умножителей поступают сигналы с опорных генераторов несущей частоты, изменяющиеся по законам $\cos(\omega \cdot t)$ и $\sin(\omega \cdot t)$. Фильтры нижних частот, стоящие на выходах умножителей, отфильтровывают составляющие на второй гармонике несущей частоты и выделяют изменения постоянной составляющей, пропорциональные значениям модулирующих импульсов d_I и d_Q квадратурных каналов. Масштабирование квадратурных выходных сигналов демодулятора осуществляется подключением блоков усиления на **2** [1]. Тогда, после ФНЧ и усиления на **2** согласно простейшим преобразованиям имеем:

$$V_{k1} \approx d_k = \cos(\varphi_k), \quad (1.2)$$

$$V_{k2} \approx g_k = \sin(\varphi_k) \quad (1.3)$$

Далее в блоке восстановления прямоугольные квадратурные модулирующие импульсы) поступают на блоки принятия решений Zero order Hold и фазовый декодер, реализованный программно, согласно, рисунку 1.3.



Рисунок 1.3 – Функциональная схема декодирования квадратурных составляющих принятых импульсов

Скрипт-код М-функции фазового декодера будет иметь следующий вид:

```
function y=phaze_dibit(x);
% Преобразование фазы в виде проекций в дибит
% x- вектор (проекции cos и sin)

if ((x(1)==0) && (x(2)==0)); y=[0 0]; end
if ((x(1)>=0) && (x(2)>=0)); y=[0 0]; end
if ((x(1)<0) && (x(2)>=0)); y=[0 1]; end
if ((x(1)<0) && (x(2)<0)); y=[1 1]; end
if ((x(1)>=0) && (x(2)<0)); y=[1 0]; end
```

Затем, с целью преобразования параллельного представления в последовательное, осуществляется поочередная выборка с выходов фазового декодера и восстанавливается последовательный вид принятой импульсной последовательности [1].

2 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK

Пакет *Simulink* разрабатывается компанией *Mathworks* (www.mathworks.com) и распространяется в составе математического пакета *MatLab*. Пакет основан на графическом интерфейсе и является типичным средством визуально ориентированного программирования. Он обладает обширной библиотекой готовых блоков с модифицируемыми параметрами для построения моделей рассматриваемых систем и наглядными средствами визуализации результатов моделирования.

2.1 Общая характеристика пакета SIMULINK

Для запуска системы Simulink необходимо предварительно выполнить запуск системы **MatLab**. После открытия командного окна системы **MatLab** нужно запустить систему **Simulink**. Это можно сделать одним из трех способов [2]:

- нажать кнопку (**Simulink**) на панели инструментов системы **MatLab**;
- в строке командного окна **MatLab** напечатать **Simulink** и нажать клавишу **Enter**;
- выполнить опцию **Open** в меню **File** и открыть файл модели (**mdl**-файл).

Последний способ предпочтителен при запуске уже готовой и отлаженной модели, когда требуется лишь провести моделирование и не нужно добавлять новые блоки в модель. При применении двух первых способов открывается окно обозревателя библиотеки блоков (**Simulink Library Browser**) [2].

На рисунке 2.1 выведена библиотека системы **Simulink** и 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 задания свойств сигналов;
- **Signal Routing** – блоки маршрутизации сигналов;
- **Sinks** – блоки приема и отображения сигналов;
- **Sources** – блоки источников сигнала;
- **User-Defined Function** – функции, определяемые пользователем.

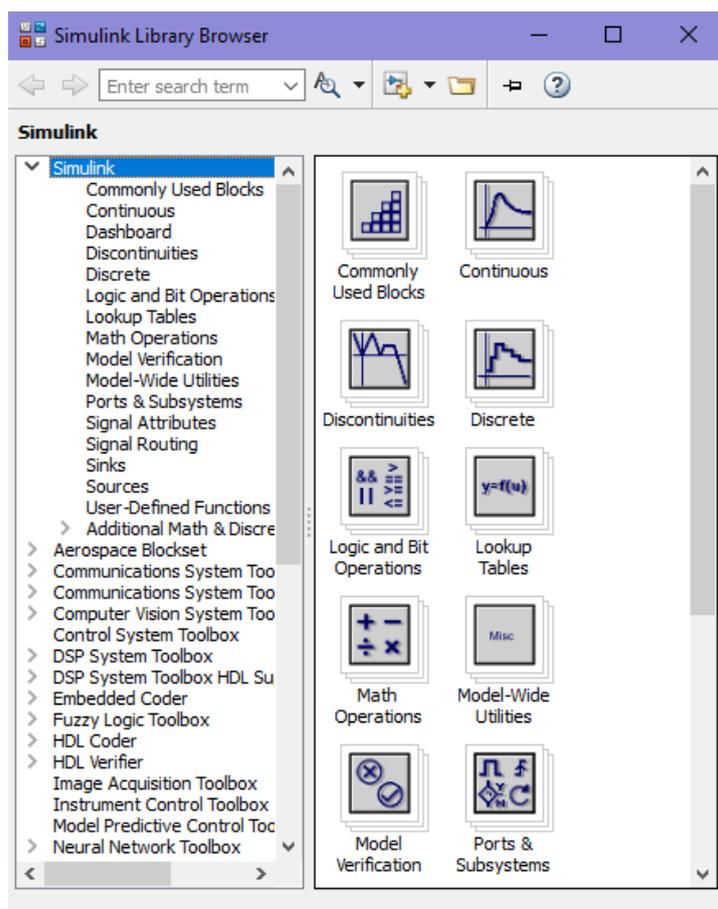


Рисунок 2.1 – Библиотека блоков **Simulink Library Browser**

При работе элементы разделов библиотек "перетаскивают" в рабочую область удержанием левой кнопки мыши на соответствующих изображениях. Для соединения элементов достаточно указать курсором мыши на начало соединения и затем при нажатии левой кнопки мыши протянуть соединение в его конец.

При двойном щелчке левой кнопки мыши на выделенном блоке всплывает меню, в котором задаются параметры блоков.

Работа *Simulink* происходит на фоне открытого окна системы *MatLab*, закрытие которого приведёт к выходу из *Simulink* [2].

3 ОПИСАНИЕ ЛАБОРАТОРНОГО МАКЕТА

В работе предлагается рассмотрение одного из возможных вариантов объединения разноскоростных потоков данных модема – с полосой, определяемой медленным сигнальным потоком [3]. Исследуемая схема представлена на рисунке 3.1. Здесь, для определенности, первый поток имеет в два раза выше скорость, чем второй поток данных.

Процесс мультиплексирования представляет собой процедуру уплотнения потоков данных для передачи их по общей линии связи с целью повышения пропускной способности информационного канала. Предлагается исследовать возможность мультиплексирования разноскоростных потоков

данных, используя структуру квадратурного модулятора на этапе фазового кодирования [3].

Входные потоки данных реализуется блоками источников случайных процессов с нормальным распределением (*Random Number*, *Random Number1*), на выходах которых

включены блоки определения знака сигнала (*Sign, Sign1*). Изменением параметров блоков источников *Initial seed* и *Sample time* задается вид случайной последовательности и скорость потока.

В начало модема включены преобразователи двухполярных импульсных последовательностей в однополярные битовые последовательности импульсов на основе блоков операции сравнения (*Relational Operator*) с блоками констант *Constant=0*.

Квадратурный модулятор. Однополярные битовые последовательности подаются на соответствующие входы фазового кодера, выполненного в виде подсистемы *Subsystem* фазового кодера см. рисунок 3.1.

Multiplexirovanie

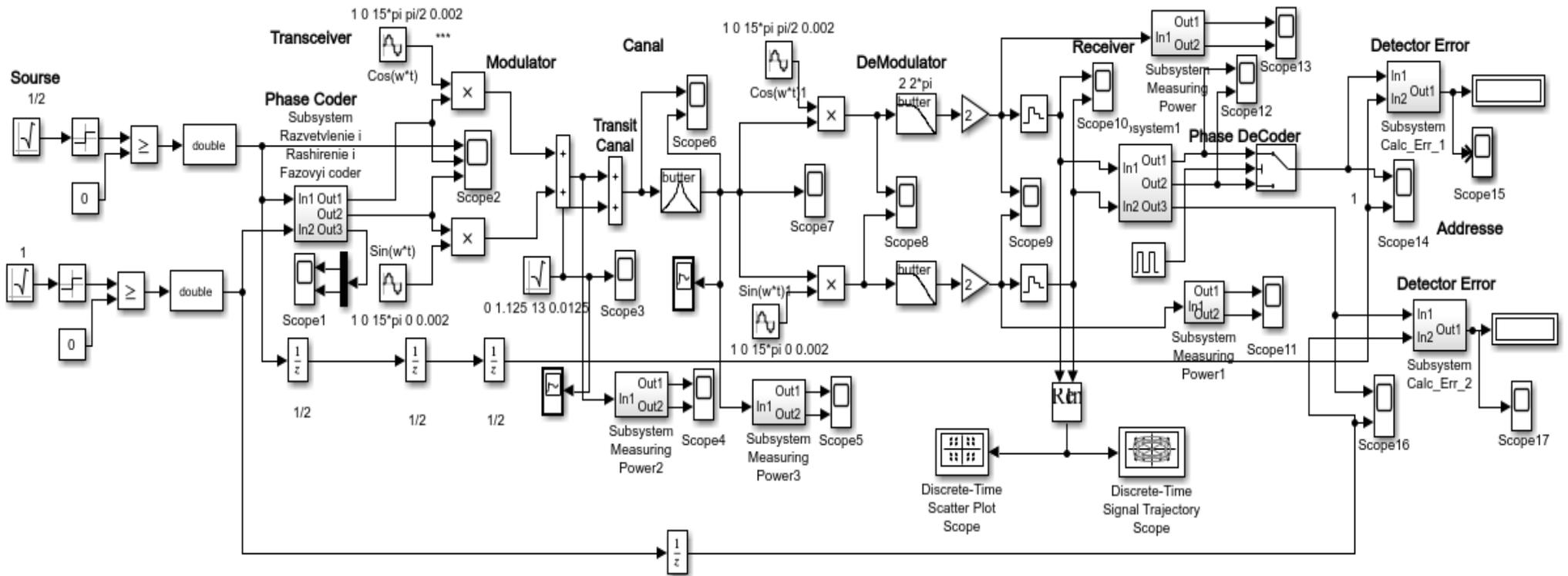


Рисунок 3.1 – Функциональная модель варианта 8PSK-модема с мультиплексированием разноскоростных сигнальных потоков данных

В этой подсистеме фазового кодера (см. рисунок 3.2) преобразование быстрой битовой последовательности импульсов в векторное представление дибитов реализовано путем подачи текущего и задержанного на один такт предыдущего импульсов на блок мультиплексора (*Mux*). На оставшийся вход мультиплексора подается медленная битовая последовательность. С выхода мультиплексора векторное представление уже трёхбитовых импульсов подается на блок экстраполятора нулевого порядка *Zero-Order Hold*, который выполняет дискретизацию входного сигнала по времени и расширение длительности до трибита. Встроенная М-функция *tribit_phase.m* сопоставляет значениям входного трибита текущее фазовое состояние в соответствии с кодом Грея.

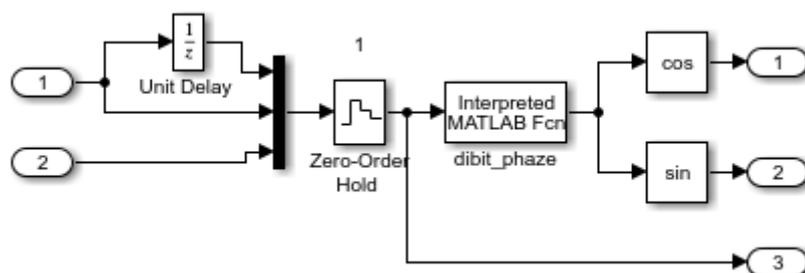


Рисунок 3.2 – Функциональная схема фазового кодера квадратурного модулятора

Скрипт-код М-функции (фазового кодера 8-PSK) представлен ниже по тексту. Скрипт-код в данном случае соответствует 8-PSK модуляции, т.к. соотношение скоростей потоков информации составляет 2:1, а полоса пропускания определяется медленным сигнальным потоком.

```
function fi=tribit_phase(x)
if x==[0; 0; 0]; fi=0; end;
if x==[0; 0; 1]; fi=pi/4; end;
if x==[0; 1; 1]; fi=2*pi/4; end;
if x==[0; 1; 0]; fi=3*pi/4; end;
if x==[1; 1; 0]; fi=4*pi/4; end;
if x==[1; 1; 1]; fi=-3*pi/4; end;
if x==[1; 0; 1]; fi=-2*pi/4; end;
if x==[1; 0; 0]; fi=-pi/4; end;
```

После блока задания функций MatLabFcn модулирующие импульсы через блоки (*Trigonometric Function*) и (*Trigonometric Function1*) в виде импульсов пропорциональных значениям $\cos(\varphi_k)$ и $\sin(\varphi_k)$ поступают на первые входы умножителей (*Product2* и *Product1*). На вторые входы умножителей поступают сигналы с внешних опорных генераторов несущей частоты, изменяющиеся по законам $\cos(\omega \cdot t)$ и $\sin(\omega \cdot t)$. Выходы перемножителей подключены к сумматору (выходной блок модулятора). С выхода модулятора модулированный сигнал поступает в модель канала распространения (*Transit Canal*), где к сигналу добавляются гауссовские шумы n_k . С этой целью на дополнительный вход сумматора подключаем генератор псевдослучайной последовательности с нормальным распределением (*Random Number*), позволяющий изменять соотношение сигнал/шум на входе приемного тракта (демодулятора) при исследовании помехоустойчивости.

Квадратурный демодулятор. Зашумлённый сигнал с имитатора канала распространения подается через полоснопропускающий фильтр (*Analog Filter Design*) на входы демодулятора, реализованного для квадратурного канала обработки двумя умножителями (*Product3* и *Product4*). На первые входы умножителей квадратурного демодулятора подается принятый сигнал, а на вторые входы умножителей поступают сигналы с опорных генераторов несущей частоты, изменяющиеся по законам $\cos(\omega \cdot t)$ и $\sin(\omega \cdot t)$. С выходов умножителей сигналы поступают на ФНЧ (*Analog Filter Design1* и *Analog Filter Design2*), которые отфильтровывают высокочастотные составляющие сигналов умножителей, а низкочастотные составляющие усиливаются блоками *Gain* и *Gain1*. Усиленные низкочастотные составляющие спектра сигналов с ФНЧ, пропорциональные принятым квадратурным модулирующим импульсам вида $\cos(\varphi_k)$ и $\sin(\varphi_k)$ подаются на блоки экстраполяторов нулевого порядка (*Zero-Order Hold* и *Zero-Order Hold1*) (блоки взятия отсчетов квадратурных составляющих). Параметры блоков демодулятора *Analog Filter Design* и *ZeroOrder Hold* соответствуют демодулируемым трибитам.

Составной блок фазового декодера (восстановления) исходных последовательностей битовых импульсов по принятым квадратурным состояниям несущей реализован на основе блока подсистемы (*Subsystem1*) со встроенным блоком задания М-функции (*MatLab Fcn*). В блоке восстановления прямоугольные квадратурные модулирующие импульсы через мультиплексор (*Mux*) поступают на фазовый декодер, реализованный программно согласно рисунку 3.3.

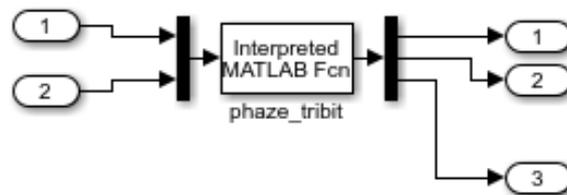


Рисунок 3.3 – Функциональная схема фазового декодера (восстановления) исходных последовательностей битовых импульсов

С выхода мультиплексора квадратурные составляющие в векторной форме поступают на блок задания М-функции (*MatLab Fcn*). Встроенная М-функция *phase_tribit.m* реализует декодирование фазовых состояний в последовательность допустимых значений трибитов. Декодированная блоком М-функции (*MatLab Fcn*) трибитовая последовательность через демультиплексор (*Demux*) поступает на выход подсистемы (*Subsystem1*).

Скрипт-код М-функции (фазового декодера) представлен ниже по тексту.

```
function fi=tribit_phase(x)
if x==[0; 0; 0]; fi=0; end;
if x==[0; 0; 1]; fi=pi/4; end;
if x==[0; 1; 1]; fi=2*pi/4; end;
if x==[0; 1; 0]; fi=3*pi/4; end;
if x==[1; 1; 0]; fi=4*pi/4; end;
if x==[1; 1; 1]; fi=-3*pi/4; end;
if x==[1; 0; 1]; fi=-2*pi/4; end;
if x==[1; 0; 0]; fi=-pi/4; end;
```

Схема восстановления быстрой битовой последовательности импульсов по восстановленным трибитам реализована на основе блока двухпортового переключателя (*Switch*), управляемого генератором прямоугольных импульсов (*Pulse Generator2*).

Первые два значения текущего трибита подаются на входные порты двухпортового переключателя, с целью преобразования параллельного представления в последовательное, которые управляющим генератором прямоугольных импульсов, поочередно (на длительность бита), коммутируются на выход. В результате на выходе переключателя появляется восстановленная быстрая битовая последовательность импульсов.

Медленная последовательность соответствует третьему выходу демультиплексера *Demux*.

Для обеспечения контроля помехоустойчивости реализованы составные блоки (*SubSystem Calc_err_1 u 2*) детекторов ошибок, которые подсчитывают и отображают на блоках (*Display*) количество битовых ошибок (несовпадений импульсов). Схема детекторов ошибок представлены на рисунке 3.4.

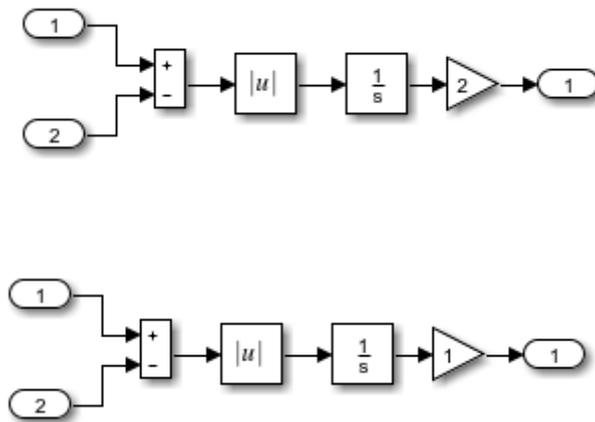


Рисунок 3.4 – Функциональные схемы детекторов ошибок быстрого и медленного каналов приема и обработки

Отличие функциональных схем детекторов битовых ошибок передачи в параметре блока *Gain* обусловлено тем фактом, что число ошибок высчитывается как интеграл модуля разности исходной и принятой битовых последовательностей.

На выходе квадратурного канала для контроля работы включены блоки *Discrete-Time Signal Trajectory Scope* и *Discrete-Time Scatter Plot Scope*, отображающие фазовые переходы и диаграммы фазовых состояний. При воздействии шумов канала происходит размытие фазовых переходов и рассеяние фазовых состояний, что позволяет визуально оценивать качество канала.

4 ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ БЛОКОВ БИБЛИОТЕКИ SIMULINK

Ниже описаны основные блоки базовых разделов библиотеки Simulink, используемые в функциональной схеме.



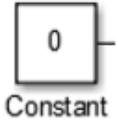
Random Number – блок источника случайного дискретного сигнала с нормальным распределением. Назначение: формирование случайного сигнала с нормальным распределением уровня сигнала. Параметры блока: *Mean* - среднее значение сигнала, *Variance* - дисперсия (среднеквадратическое отклонение), *Initial seed*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



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

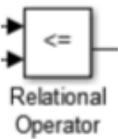
$$\text{sign} = \begin{cases} -1, & \text{где } x < 0; \\ 0, & \text{где } x = 0; \\ 1, & \text{где } x > 0. \end{cases}$$

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



Constant

Constant – блок источника постоянного сигнала. Назначение: задает постоянный по уровню сигнал. Параметры: *Constant value* – постоянная величина, *Interpret vector parameters as 1-D* – интерпретировать вектор параметров как одномерный (при установленном флажке). Значение константы может быть действительным или комплексным числом, вычисляемым выражением, вектором или матрицей.



Relational Operator

Relational Operator – блок вычисления операции отношения. Назначение: блок сравнивает текущие значения входных сигналов. Параметры: *Relational Operator* – тип операции отношения (выбирается из списка): «=» – тождественно равно, «~» – не равно, «<» – меньше, «< =» – меньше или равно, «> =» – больше или равно, «>» – больше.

Data Type Conversion



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* и *boolean*. *Saturate on integer overflow* (флажок) – подавляет переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.



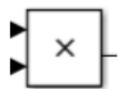
Scope

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.



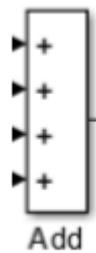
Sine Wave

Sine Wave – блок источника синусоидального сигнала. Назначение: формирование синусоидального сигнала с заданной частотой, амплитудой, фазой и смещением. Параметры блока: *Sine Type* – способ формирования сигнала реализуется двумя алгоритмами: *Time-based* – по текущему времени (для аналоговых систем) или по значению сигнала на предыдущем шаге и величине такта дискретности (для дискретных систем); *Sample-based* – по величине такта дискретности и количеству расчетных шагов на один период синусоидального сигнала. Вид окна задания параметров меняется в зависимости от выбранного способа формирования синусоидального сигнала.

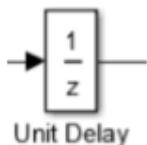


Product1

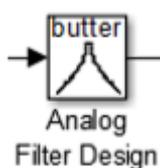
Product – блок умножения и деления. Назначение: вычисление произведения текущих значений сигналов. Параметры блока: *Number of inputs* – количество входов, может задаваться как число или как список знаков. В списке знаков можно использовать знаки: * – умножить и / – разделить. *Multiplication* – способ 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.



Add, Sum – блок сумматора. Назначение: вычисление алгебраической суммы текущих значений входных сигналов. Параметры блока: *Icon shape* – форма блока, выбирается из списка: *round* – круг; *rectangular* – прямоугольник. *List of sign* – список знаков из набора: + - плюс; - - минус, | - разделитель. Флажок *Show additional parameters* – показать дополнительные параметры, при выставленном флажке отображаются окна списка *Output data type mode*, в нашем случае не используется. Количество входов и соответствующие им операции определяются списком знаков *List of sign*. При этом метки входов обозначаются соответствующими знаками. В списке *List of sign* можно также указать число входов, при этом все входы будут суммирующими.

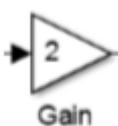


Unit delay – блок единичной дискретной задержки. Назначение: выполняет задержку дискретного сигнала на заданный шаг модельного времени. Параметры блока: *Initial conditions* – начальное значение выходного сигнала; *Sample time* – шаг модельного времени.



Analog Filter Design – блок аналогового фильтра заданного метода проектирования и типа из подраздела *Filter Design*; подраздела *Filtering*, раздела *DSP Blockset* библиотеки *Simulink*. Назначение: аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. Параметры блока: *Design method* – метод проектирования, выбирается из списка: *Butterworth* – фильтр Баттерворта; *Chebyshev I* – фильтр Чебышева 1-го рода; *Chebyshev II* – фильтр Чебышева 2-го рода; *Elliptic* – фильтр эллиптический; *Bessel* – фильтр Бесселя. *Filter type* – тип фильтра, выбирается из списка: *Lowpass* – нижних частот; *Highpass* – верхних частот; *Bandpass* – полосно-пропускающий; *Bandstop* – полосно-заграждающий. Далее для каждого метода проектирования и типа фильтра выдается свой список параметров. Так для фильтра Баттерворта типа нижних частот параметрами являются:

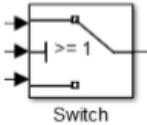
Filter order – порядок фильтра; *Passband edge frequency (rads/sec)* – нижняя граничная частота (радиан в секунду). Для других методик проектирования и типов фильтров определяемые параметры очевидны.



Gain – блок усилителя. Назначение: блок умножает входной сигнал на постоянный коэффициент; Параметры блока: *Multiplication* – способ выполнения операции, значение параметра выбирается из списка: *Elementwise 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.



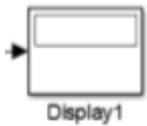
Zero-Order Hold – блок экстраполятора нулевого порядка. Назначение: блок выполняет дискретизацию входного сигнала по времени. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания интервала дискретизации. Затем выходной сигнал изменяется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



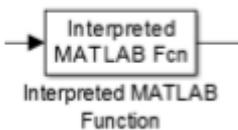
Switch – блок переключателя. Назначение: выполняет переключение входных сигналов по сигналу управления. Параметры: *Threshold* – Порог управляющего сигнала. Если сигнал управления, подаваемый на средний вход меньше, чем величина порогового значения *Threshold*, то на выход блока проходит сигнал с первого (верхнего) входа. Если сигнал управления превысит пороговое значение, то на выход блока будет поступать сигнал со второго (нижнего) входа.



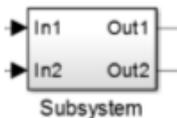
Pulse Generator – блок источника импульсного сигнала. Назначение: Формирование прямоугольных импульсов. Параметры: *Pulse Type* – способ формирования сигнала: *Time-based* – по текущему времени, *Sample-based* – по величине модельного времени и количеству расчетных шагов; *Amplitude* – амплитуда, *Period* – период. Задается в секундах для *Time-based Pulse Type* или в шагах модельного времени для *Sample-based Pulse Type*, *Pulse width* – ширина импульсов. Задается в % по отношению к периоду для *Time-based Pulse Type* или в шагах модельного времени для *Sample-based Pulse Type*, *Phase delay* – фазовая задержка. Задается в секундах для *Time-based Pulse Type* или в шагах модельного времени для *Sample-based Pulse Type*, *Sample time* – шаг модельного времени. Задается для *Sample-based Pulse Type*.



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 ЛВМ 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.

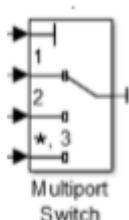


Subsystem – виртуальная и монолитная подсистемы. Доступ к окну параметров подсистемы осуществляется через меню *Edit* командой *Block Parameters*. Параметры: *Show port labels* – показать метки портов, *Treat as atomic unit* (флажок) – считать подсистему монолитной. Таким образом, блоки виртуальной и 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,

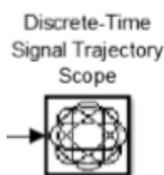
NoReadOrWrite – пользователь не может открывать и изменять подсистему; *Name of error callback function* – имя функции используемой для обработки ошибок возникающих в данной подсистеме.

 *Mux* – блок мультиплексора. Назначение: объединяет входные сигналы в вектор. Параметры блока: *Number of Inputs* – количество входов; *Display option* – способ отображения, выбирается из списка: *bar* – вертикальный узкий прямоугольник черного цвета; *signals* – прямоугольник с белым фоном и отображением меток входных сигналов; *none* – прямоугольник с белым фоном без отображения меток входных сигналов.

 *Demux* – блок демультиплексора. Назначение: разделение входного векторного сигнала на составляющие (последовательного представления в параллельное). Параметры блока: *Number of output* – количество выходов; *Display option* – способ отображения выбирается из списка: *bar* – вертикальный узкий прямоугольник черного цвета; *none* – прямоугольник с белым фоном без отображения меток входных сигналов. Флажок *Bus Selection Mode* – режим разделения векторных сигналов в шине, используется для разделения сигналов, объединенных в шину.



Multiport Switch – блок многовходового переключателя. Назначение: выполняет переключение входных сигналов на выход по сигналу управления, задающему номер активного входного порта. Параметры блока: *Number of inputs* – количество входов; флажок *Show additional parameters* – показать дополнительные параметры, в нашем случае не используется. Блок *Multiport Switch* пропускает на выход сигнал с того входного порта, номер которого равен текущему значению управляющего сигнала.



Discrete-Time Signal Trajectory Scope – блок отображения диаграммы переходов фазовых состояний сигнала из подраздела *Comm Sinks* раздела *Communication Blockset* библиотеки *Simulink*. Назначение: отображение диаграммы рассеяния фазовых состояний за счет влияния полосы пропускания и помех тракта передачи. Параметры блока: Флажок *Show Plotting Properties* – показать графические установки; *Samples per Symbol* – шаг периода символа; *Offset (samples)* – смещение шагов; *Points displayed* – число отсчетов сигнала, начиная с которого отображается диаграмма; *New points per display* – число отсчетов при обновлении отображения; флажки *Show Rendering Properties*, *Show Axes Properties*, *Show Figure Properties* – показать свойства отображения, осей и фигуры в нашем случае не используются [2].

Методика оценки помехоустойчивости. Помехоустойчивость оценивается водопадоподобной зависимостью вероятности битовой ошибки P_b от соотношения сигнал/шум (SNR). Вероятность битовой ошибки определяется по показаниям детектора ошибок как предел частоты появления ошибок при большом числе испытаний.

Подбор мощности шума канала осуществляется таким образом, чтобы число ошибок равнялось примерно 1. После чего производятся измерения SNR. Затем, с целью построения водопадоподобной характеристики, поэтапно увеличивается мощность шума канала, изменяется SNR и фиксируется число ошибок. Измерения мощностей сигнала S и сигнала с шумом SN производится в точке взятия отсчета принятого сигнала, т.е. на входе блока *Zero Order Hold*.

Для измерения отношения сигнал/шум необходимо, отключив шумы канала, измерить мощность сигнала, а потом – сигнала с шумами. Для измерения мощности используются блоки *Subsystem Measuring Power* для каждого из каналов (см. рис. 3.1).

В результате соотношение сигнал/шум n -го канала можно рассчитать по формуле

$$SNR = 10 \cdot \log\left(\frac{S}{SN-S}\right), \text{ дБ.} \quad (4.1)$$

Число ошибок фиксируется блоком детектора ошибок *Detector Error* (см. рис. 3.1).

5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

Перед допуском к работе дать ответы на контрольные вопросы.

В ходе выполнения данной лабораторной работы требуется выполнить следующие пункты:

1. Собрать Sim-модель QPSK- модема (см. рис. 1.1), с параметром Sample Time=1, настроить и найти SNR при вероятности битовой ошибки 0.001.

Преобразовать модель в двухканальную, равноскоростную с параметром Sample Time=1, настроить и найти SNR при вероятности битовой ошибки 0.001.

2. Собрать Sim-модель QPSK- модема (см. рис. 1.1), с параметром Sample Time=1/2, настроить и найти SNR при вероятности битовой ошибки 0.001.

Преобразовать модель в двухканальную, равноскоростную с параметром Sample Time=1/2, настроить и найти SNR при вероятности битовой ошибки 0.001.

3. Собрать Sim-модель, в соответствии с рис. 3.1, для исследования передачи разноскоростных потоков 2:1, настроить и найти SNR при вероятности битовой ошибки 0.001.

4. Свести результаты исследований в таблицу 5.1. Дать интерпретацию полученным результатам по энергетической и спектральным эффективностям.

5. Продумать последовательность и объем представляемого материала и написать отчет о проделанной работе.

6. Защитить проделанную работу.

Таблица 5.1–Результаты модельного исследования

Скорость передачи	Тип модуляции	SNR, дБ, при вероятности битовой ошибки, равной 0.001
1	Однокан. QPSK	
1/1	Двухкан. QPSK	
2	Однокан. QPSK	
2/2	Двухкан. QPSK	
2/1	Двухкан. 8- PSK	

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1 Зачем выполняется мультиплексирование каналов передачи данных в модеме?
- 2 Сравнение манипуляций QPSK, 8-PSK, 16- PSK по помехоустойчивости.
- 3 Сравнение манипуляций QPSK, 8-PSK, 16- PSK по спектральной эффективности.
- 4 Объясните работу QPSK, 8-PSK, 16- PSK модуляторов модема.
- 5 Объясните работу QPSK, 8-PSK, 16- PSK демодуляторов модема.
- 6 Объясните принцип работы формирователя квадратурных модулирующих импульсов фазового манипулятора (фазового кодера).
- 7 Объясните принцип работы схемы восстановления исходной последовательности импульсов по принятым квадратурным модулирующим импульсам (фазового декодера).
- 8 Какую функцию в схеме выполняют полосовые и низкочастотные фильтры?
- 9 Как оценивается помехоустойчивость?
- 10 Как меняется помехоустойчивость модема при мультиплексировании разноскоростных сигнальных потоков данных?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кологривов В.А. Исследование QPSK модема (реализация с фазовым кодером): Учебно-методическое пособие по лабораторной работе / Кологривов В. А. — 2012. 35 с.
2. Черных И.В. SIMULINK: среда создания инженерных приложений / Под ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2003. – 496 с.
3. Кологривов В.А. Энергетическая и спектральная эффективности способов мультиплексирования разноскоростных сигнальных потоков в радиоканале //Тридцатая международная практическая конференция «Электронные средства и системы управления». – 2017, Ч.1.– С. 22-26.