

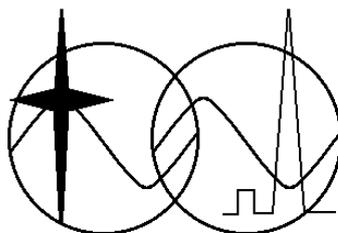
Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра средств радиосвязи (СРС)

Утверждаю:

Зав. кафедрой СРС, проф., д.т.н.

_____ Мелихов С.В.

_____ 2015 г.



ИССЛЕДОВАНИЕ ЗАВИСИМОСТИ ПОМЕХОУСТОЙЧИВОСТИ
РАЗНОСКОРОСТНЫХ КАНАЛОВ С РАЗНЫМИ ТИПАМИ
МАНИПУЛЯЦИИ ОТ СКОРОСТИ ПЕРЕДАЧИ ДАННЫХ

Учебно-методическое пособие

по лабораторной работе для студентов радиотехнических специальностей

Разработчики:

доц. каф. СРС

_____ Кологривов В. А.

_____ Олчейбен Д-Х. Н.

Томск 2015

Кологривов В. А, Олчейбен Д-Х. Н.

«Исследование зависимости помехоустойчивости разноскоростных каналов с разными типами манипуляции от скорости передачи данных». Учебно-методическое пособие по лабораторной работе для студентов радиотехнических специальностей. – Томск: ТУСУР. Образовательный портал, 2015. – 38 с.

Учебно-методическое пособие содержит описание функциональных моделей двухканальных модемов для передачи данных с беспилотного летательного аппарата (БПЛА) на наземный комплекс управления (НКУ), выполненной в среде функционального моделирования *Simulink* системы для инженерных и научных расчетов *MatLab*.

В пособии приведены сведения из теории фазовой модуляции и демодуляции, краткая характеристика пакета *Simulink* системы *MatLab*, подробное описание виртуального лабораторного макета и используемых блоков библиотеки *Simulink*, а также требования к экспериментальному исследованию и контрольные вопросы, ответы на которые необходимы для успешной защиты лабораторной работы.

Пособие предназначено для студентов радиотехнических специальностей по направлениям: «*Радиотехника*», «*Инфокоммуникационные технологии и системы связи*» и др.

© Кологривов В. А, Олчейбен Д-Х. Н., 2015 г.

© ТУСУР, РТФ, каф. СРС, 2015 г.

Аннотация

Лабораторная работа «Исследование зависимости помехоустойчивости разноскоростных каналов с разными типами манипуляции от скорости передачи данных» посвящена экспериментальному исследованию двухканального PSK-модема для передачи данных с БПЛА на НКУ с использованием пакета функционального моделирования *Simulink*.

В описании сформулирована цель лабораторной работы, приведены необходимые сведения из теории PSK модуляций, краткая характеристика пакета *Simulink* системы *MatLab*, подробное описание виртуального лабораторного макета и используемых блоков библиотеки *Simulink*, а также требования к экспериментальному исследованию и контрольные вопросы, ответы на которые необходимы для успешной защиты лабораторной работы.

Исследование зависимости помехоустойчивости разноскоростных каналов с разными типами манипуляции от скорости передачи данных проводится по функциональной схеме, построенной на основе блоков базовых разделов библиотеки *Simulink*.

Оглавление

1	Цель работы. Краткие сведения из теории.....	5
1.1	Модуляция BPSK	5
1.2	Модуляция QPSK.....	6
1.3	Модуляция 8-PSK.....	7
2	Описание лабораторного макета	9
2.1	Описание работы BPSK/QPSK модема.....	9
2.2	Описание работы BPSK/8-PSK модема	18
3	Краткое описание пакета Simulink.....	23
3.1	Запуск и работа с пакетом Simulink	23
3.2	Описание используемых блоков библиотеки Simulink.....	25
4	Экспериментальное задание	35
5	Контрольные вопросы	37
	Список использованных источников	38

1 Цель работы. Краткие сведения из теории

Цель работы: исследовать зависимость помехоустойчивости разноскоростных каналов с разными типами манипуляций (BPSK, QPSK, 8-PSK) от скорости передачи данных для передачи данных с БПЛА.

1.1 Модуляция BPSK

Одна из простейших разновидностей цифровой модуляции двухпозиционная или двоичная фазовая модуляция (BPSK). Фазы несущего колебания принимают значение 0° и 180° [1]. Модулированный сигнал записывается выражением (1):

$$e(t) = A_c \cos(\omega_c t + \varphi(t)), \quad (1)$$

где $\varphi(t)$ – мгновенная фаза, зависящая от модулирующего сигнала;

A_c – амплитуда сигнала.

Фазовую модуляцию можно осуществить различными способами. Как уже упоминалось выше, начальная фаза $\varphi(t)$ может принимать два значения 0° и 180° . Тогда для реализации фазовой модуляции достаточно биполярную информационную последовательность перемножить с колебанием несущей.

В этом случае фазовая модуляция будет рассматриваться как вариант амплитудной модуляции с активной паузой, где сигнал амплитуды A_c принимает два значения -1 и $+1$. Такое изменение значения амплитуды эквивалентно изменению фазы на 180° . Так как значения амплитуды сигнала -1 и $+1$ можно рассматривать как особый вариант логического нуля и единицы, то с использованием бинарной фазовой модуляции BPSK можно передавать обычный двоичный сигнал. Символьная скорость этого вида модуляции составляет 1 бит на символ.

На комплексной плоскости на рисунке 1.1 показано, что при формировании сигнала двухпозиционной фазовой модуляции квадратурная компонента комплексного сигнала I принимает два значения: -1 и $+1$ [1].

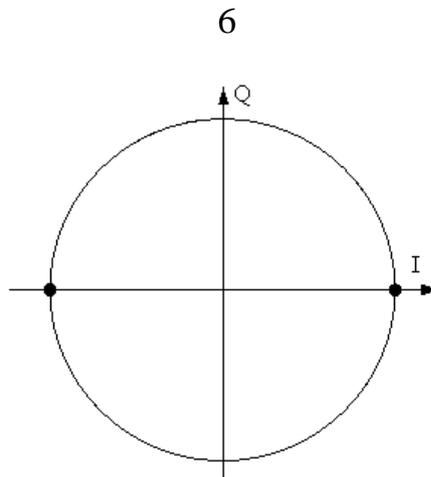


Рисунок 1.1 – Диаграмма сигнала BPSK

1.2 Модуляция QPSK

Из теории связи известно, что наивысшей помехоустойчивостью обладает двоичная фазовая модуляция BPSK. Однако в ряде случаев за счет уменьшения помехоустойчивости канала связи можно увеличить его пропускную способность.

В четырехпозиционной фазовой модуляции используются четыре значения фазы несущего колебания. В этом случае фаза $\varphi(t)$ сигнала должна принимать четыре значения: 0° , 90° , 180° и 270° . Однако чаще используются другие значения фаз: 45° , 135° , 225° и 315° . Такой вид представления квадратурной фазовой модуляции приведен на рисунке 1.2 [2].

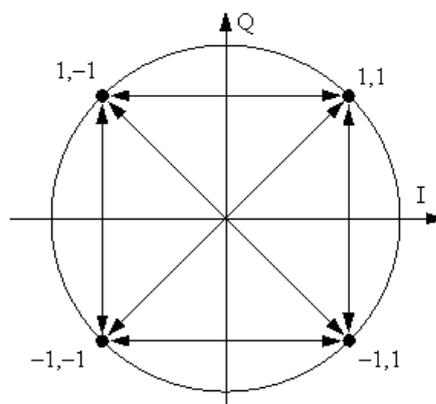


Рисунок 1.2 – Диаграмма сигнала QPSK

На этом же рисунке представлены значения бит, которые передаются каждым состоянием фазы несущего колебания, закодированные кодом Грея. Каждое состояние осуществляет передачу сразу двух бит полезной информации. При этом содержимое бит выбрано таким образом, чтобы переход к соседнему состоянию фазы несущего колебания за счет ошибки приема приводил не более чем к одиночной битовой ошибке.

Так как при этом виде модуляции в течение одного символического интервала передается сразу два бита входного битового потока, то символьная скорость этого вида модуляции составляет 2 бита на символ. При реализации модулятора следует разделять входной поток на две составляющих — синфазную составляющую I и квадратурную составляющую Q [2].

1.3 Модуляция 8-PSK

В этом виде фазовой модуляции существуют восемь значений фазы несущего колебания. Так как $2^3 = 8$, то каждым символом одновременно передается сразу 3 бита. Это означает, что битовая скорость при этом виде модуляции в три раза выше символьной скорости передачи данных.

Диаграмма переходов состояний сигнала при восьмипозиционной фазовой модуляции приведена на рисунке 1.3 [3].

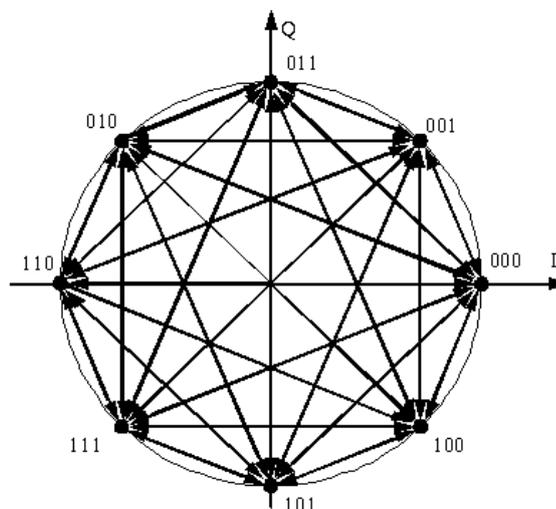


Рисунок 1.3 – Диаграмма сигнала 8-PSK

На этой же диаграмме приведены значения трибитов, передаваемых каждым состоянием символа. Как видно из рисунка 1.3, фазовые состояния закодированы кодом Грея, который позволяет минимизировать количество битовых ошибок при неправильном приеме символьной комбинации на приемном конце радиоканала. Соседние фазовые состояния символа отличаются друг от друга не более чем на один бит [3].

2 Описание лабораторного макета

Работа заключается в сравнении зависимости помехоустойчивости разноскоростных каналов с разными типами манипуляции от скорости передачи. Сигналы двух источников с разными типами модуляции совмещены в одном приеме-передающем блоке. Один из сигналов – это BPSK модулированный, второй – QPSK либо 8-PSK модулированный сигнал. Для каждого из источников используется своя несущая частота. Разнос частот несущих выбирается из условия ортогональности:

$$\int_0^T f_1(t) \cdot f_2(t) = 0,$$

где T – длительность символа;

$f_1(t)$, $f_2(t)$ – несущие частоты;

$\Delta f = f_1 - f_2$ – разнос между несущими.

При когерентном приеме $\Delta f = k/2T$, а при некогерентном – $\Delta f = k/T$, где $k = 1, 2, \dots$.

В целом работа состоит из двух частей. В первой части исследуется пара BPSK и QPSK модулированных сигналов, а во второй части пара BPSK и 8-PSK модулированных сигналов. В итоге производится сравнение помехоустойчивости от скорости передачи.

2.1 Описание работы BPSK/QPSK модема

Вариант реализации функциональной модели двухканального модема при модуляциях BPSK и QPSK приведен на рисунке 2.1.

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 длительность импульсов.

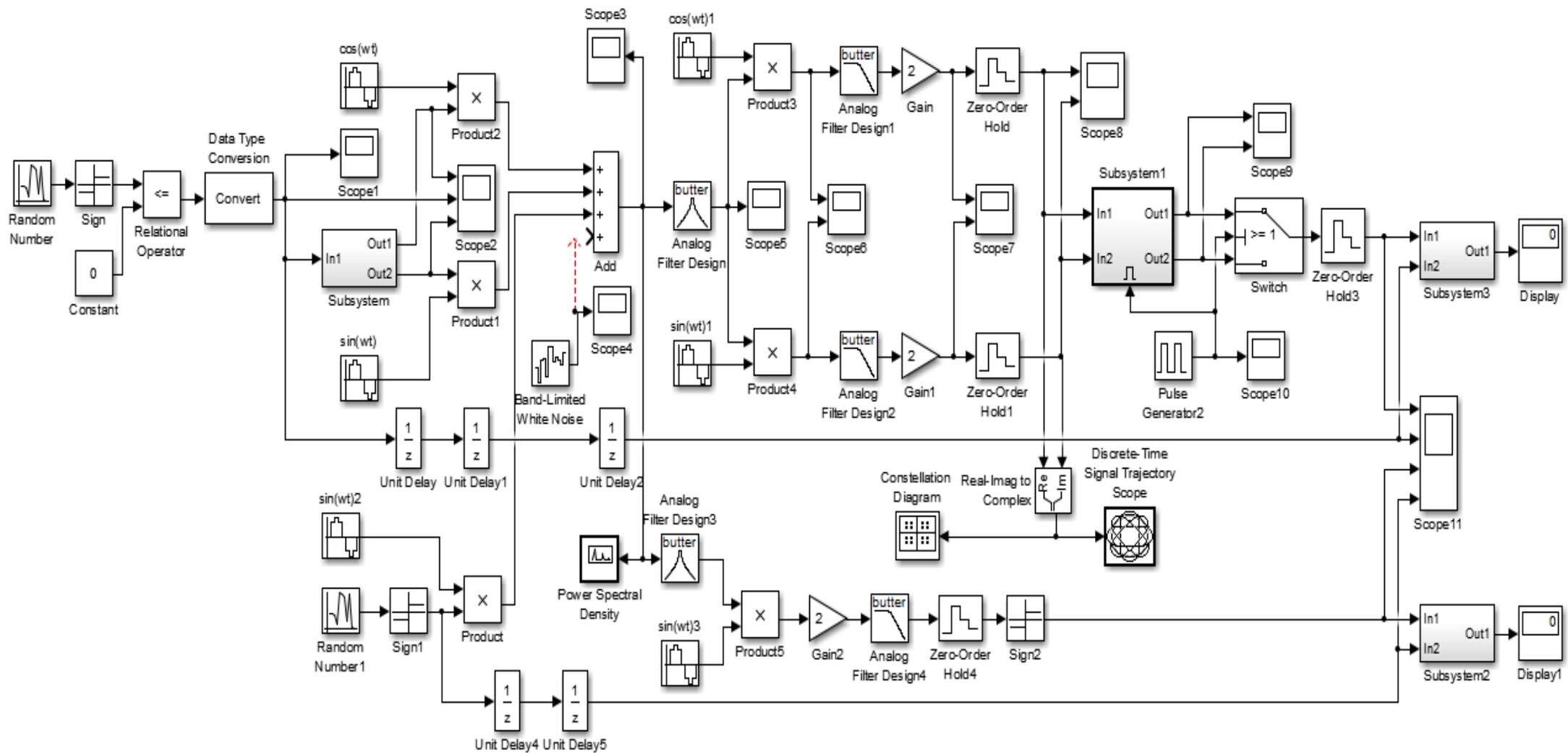


Рисунок 2.1 – Функциональная модель двухканального модема передачи данных между БПЛА и НКУ

В начало блока QPSK-модулятора включен преобразователь двухполярной импульсной последовательности в однополярную битовую последовательность импульсов на основе блока операции сравнения (*Relational Operator*) с блоком источника постоянного сигнала $Constant=0$.

Далее однополярная битовая последовательность подается на блок QPSK-модулятора, представленная в виде блока подсистемы *Subsystem* на рисунке 2.1. В этой подсистеме на рисунке 2.2 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 *M-функция* *dibit_phase.m* сопоставляет значениям входного дибита текущее фазовое состояние в соответствии с кодом Грея согласно рисунку 1.2.

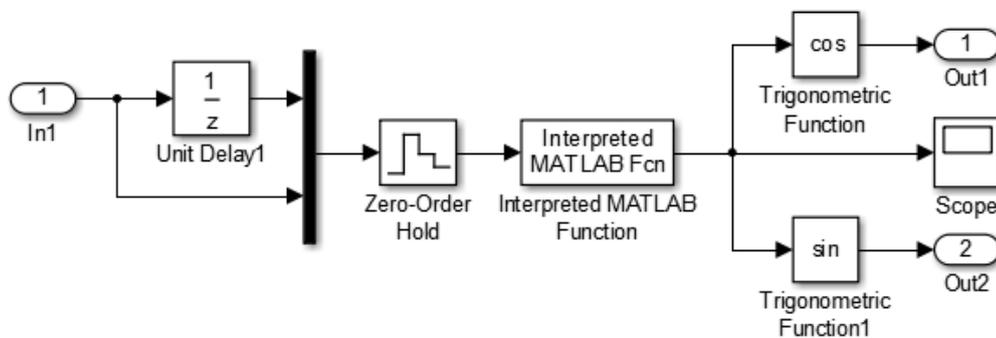


Рисунок 2.2 – Функциональная схема QPSK-модулятора

Скрипт-код *M-функции* (фазового кодера QPSK) представлен ниже по тексту.

```
function fi = dibit_phase(x)
%x-vector
%fi-phase
if all (x==[0; 0]); fi=pi/4;           end;
if all (x==[0; 1]); fi=3*pi/4;        end;
if all (x==[1; 1]); fi=-3*pi/4;       end;
if all (x==[1; 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 $\cos(\omega t)$ и $\sin(\omega t)$.

BPSK модулятор реализован на основе блока умножителя (*Product*), на первый вход которого подается биполярная последовательность сформированная блоками (*Random Number1*, *Sign1*), а на второй вход гармонический сигнал с опорного генератора на ортогональной несущей $\sin(\omega_1 t)$. На выходе перемножителя (преобразователя) получаем BPSK-модулированный сигнал.

После умножителей радиосигналы QPSK канала вместе со вторым каналом BPSK поступают на внешний блок сумматора (*Add*). Этот же блок сумматора выступает как имитатор шумов канала распространения сигнала. С этой целью на дополнительный вход сумматора подключаем генератор белого шума (*Band-Limited White Noise*), позволяющий изменять соотношение сигнал/шум на входе приемного тракта (демодулятора) при исследовании помехоустойчивости. В результате с выхода сумматора получаем манипулированные (QPSK и BPSK) сигналы в аддитивной смеси с шумами канала распространения.

Суммарный сигнал с имитатора канала передачи подается через полосно-пропускающие фильтры (*Analog Filter Design*) и (*Analog Filter Design1*) на входы демодуляторов, реализованного для QPSK канала двумя умножителями (*Product3* и *Product4*), а для BPSK канала – умножителем (*Product5*).

На первые входы умножителей QPSK демодулятора подается принятый сигнал, а на вторые входы умножителей поступают сигналы с опорных генераторов несущей частоты, изменяющиеся по законам $\cos(\omega t)$ и $\sin(\omega t)$. 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 для преобразования в прямоугольную форму (регенерацию) на блоки экстраполяторов нулевого порядка (*Zero-Order Hold* и *Zero-Order Hold1*). Параметры блоков демодулятора *Analog Filter Design* и *Zero-Order Hold* соответствуют демодулируемым дибитам.

Составной блок восстановления исходной последовательности битовых импульсов по принятым фазовым состояниям несущей реализован на основе блока *E-подсистемы (Subsystem1)* со встроенным блоком задания *M-функции (MatLab Fnc)* и управляемого опорным генератором прямоугольных импульсов (*Pulse Generator2*). В блоке восстановления прямоугольные квадратурные модулирующие импульсы через мультиплексор (*Mux*) поступают на блок принятия решений (фазовый декодер), реализованный программно согласно рисунку 2.3. С выхода мультиплексора квадратурные составляющие в векторной форме поступают на блок задания *M-функции (MatLab Fnc)*. Встроенная *M-функция phase_dibit.m* реализует декодирование фазовых состояний в последовательность допустимых значений дибитов. Декодированная блоком *M-функции (MatLab Fnc)* дибитовая последовательность через демультимплексор (*Demux*) поступает на выход *E-подсистемы (Subsystem1)*.

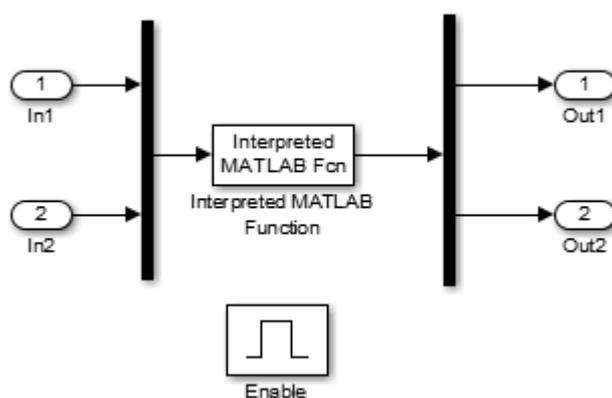


Рисунок 2.3 – Функциональная схема восстановления исходной последовательности битовых импульсов

Скрипт-код *M-функции* (фазового декодера QPSK) представлен ниже по тексту.

```
function y = phase_dibit(x)
%x-vector
%fi-phase
%fi=atan2(x(2),x(1));
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;
```

Схема восстановления исходной битовой последовательности импульсов по восстановленным дибитам реализована на основе блока двухпортового переключателя (*Switch*), управляемого генератором прямоугольных импульсов (*Pulse Generator2*). На выходе схемы восстановления для исключения нежелательных скачков включен блок экстраполятора нулевого порядка (*Zero-Order Hold3*). Схемой восстановления дибиты переводятся демультиплексором в параллельное (векторное) представление. Значения текущего дибита подаются на входные порты двухпортового переключателя, которые управляющим генератором прямоугольных импульсов, поочередно (на длительность бита), коммутируются на выход. На выходе экстраполятора в результате появляется восстановленная исходная битовая последовательность импульсов.

Функциональная схема демодулятора BPSK по структуре аналогична одному из каналов демодулятора QPSK, с разницей, что используется опорный генератор на ортогональной несущей $\sin(\omega_1 t)$, и составной блок восстановления исходной последовательности битовых импульсов по принятым фазовым состояниям заменяется блоком определения знака сигнала (*Sign2*).

Обе принятые последовательности подаются на осциллограф *Scope11*, на дополнительные входы которого, с целью обеспечения визуального контроля, подаются задержанные блоками (*Unit Delay*) исходные последовательности.

Для обеспечения контроля помехоустойчивости реализованы составные блоки (*SubSystem*) детекторов ошибок, которые подсчитывают и отображают на

блоках (Display) количество битовых ошибок (несовпадений импульсов). Схема детекторов ошибок представлены на рисунке 2.4.

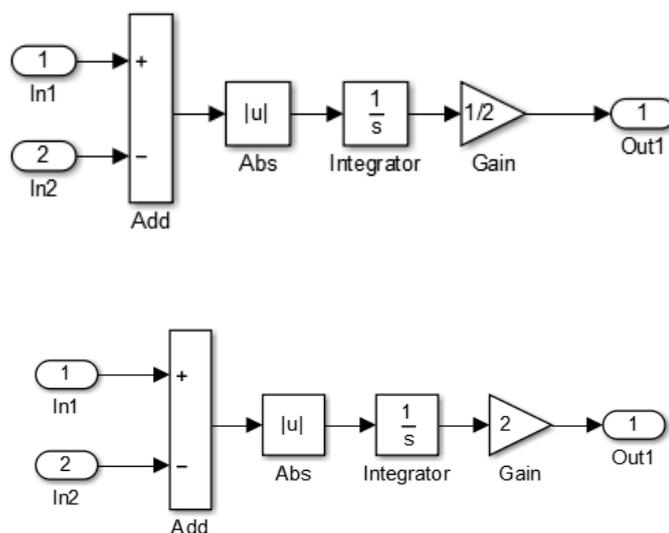


Рисунок 2.4 – Функциональные схемы детекторов ошибок BPSK и QPSK каналов приема и обработки

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. То есть фактически высчитывается площадь под модулем разности функций времени. В связи с этим при однополярной битовой последовательности с длительностью импульсов $\tau = 1$ параметр $Gain = 1$. Тогда, если последовательность становится биполярной при $\tau = 1$, то при том же количестве ошибок площадь удвоится, следует брать параметр $Gain = 1/2$. В том случае, если у однополярной последовательности длительность импульсов сокращается вдвое $\tau = 1/2$, следовательно, и площадь уменьшается в два раза, то следует брать параметр $Gain = 2$.

На выходе QPSK канала для контроля работы включены блоки *Time Signal Trajectory Scope* и *Constellation Diagram* (см. рисунок 2.1), 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.

На рисунке 2.5 приведена диаграмма фазовых состояний сигнала на выходе QPSK канала без подключения шумов.

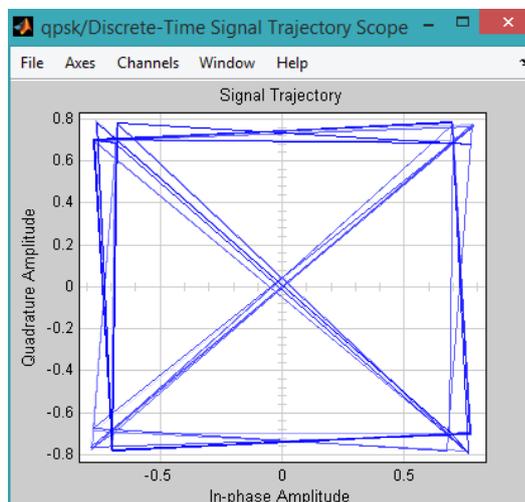


Рисунок 2.5 – Диаграмма фазовых переходов сигнала на выходе QPSK канала без подключения шумов

На рисунке 2.6 приведена диаграмма фазовых состояний сигнала на выходе QPSK канала.

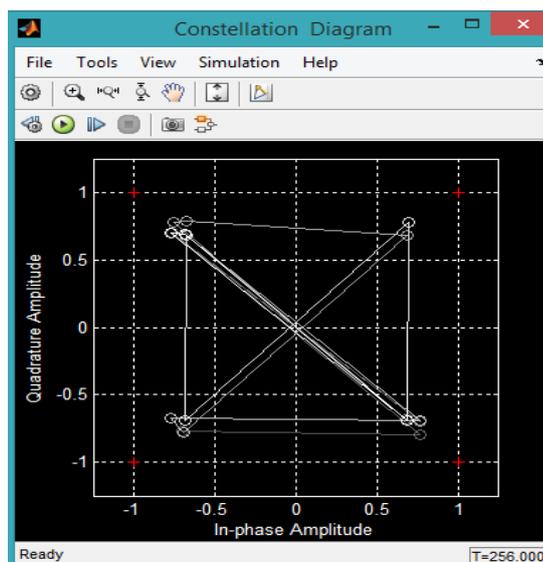


Рисунок 2.6 – Диаграмма фазовых состояний сигнала на выходе QPSK канала без подключения шумов

На рисунке 2.7 приведены осциллограммы переданных и принятых последовательностей.

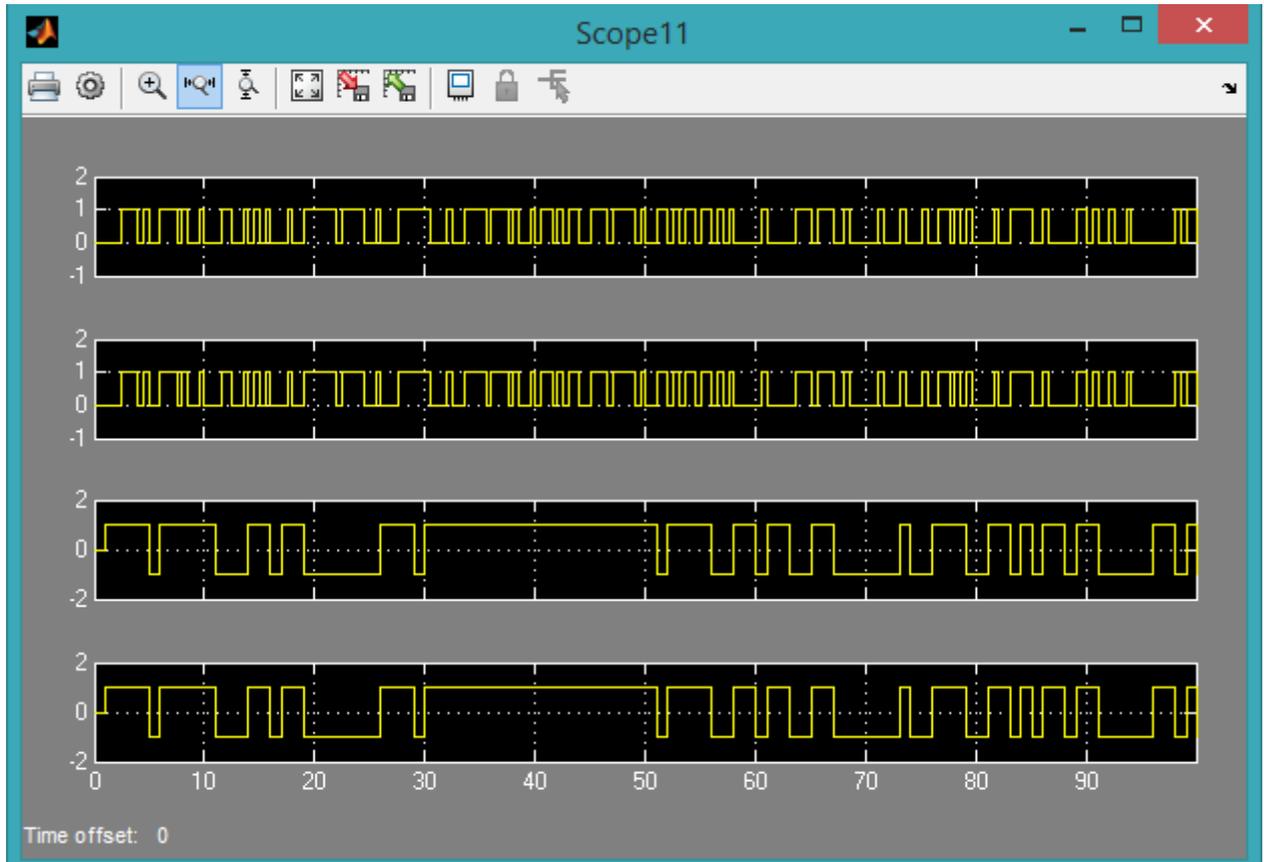


Рисунок 2.7 – Осциллограммы переданных и принятых последовательностей

2.2 Описание работы BPSK/ 8-PSK модема

Как говорилось выше, работа состоит из двух частей. Первая часть описана в предыдущем пункте, в этом пункте речь пойдет о второй части, о BPSK/8-PSK модеме.

Модель реализации функциональной модели передачи/приема двух BPSK и 8-PSK модулированных сигналов почти идентична предыдущей модели на рисунке 2.1 с тем отличием, что в первом канале реализуется не QPSK модулированный сигнал, а 8-PSK, во втором канале, как и было, – BPSK модулированный сигнал.

Функциональная схема фазового кодера модулятора 8-PSK приведена на рисунке 2.8. Для формирования 8-ми фазовых состояний из потока бит с помощью блоков задержек *Unit Delay* и мультиплексора *Mux* формируются трибиты, которые подаются на блок экстраполятора нулевого порядка *Zero-Order Hold*, который выполняет дискретизацию входного сигнала по времени и расширение длительности до трибита. Расширенные трибиты подаются на блок задания *M-функции* (*MatLab Fcn*). Встроенная *M-функция* *tribit_phase.m* сопоставляет амплитудам входного импульса текущее фазовое состояние в соответствии с кодом Грея, согласно рисунку 1.3.

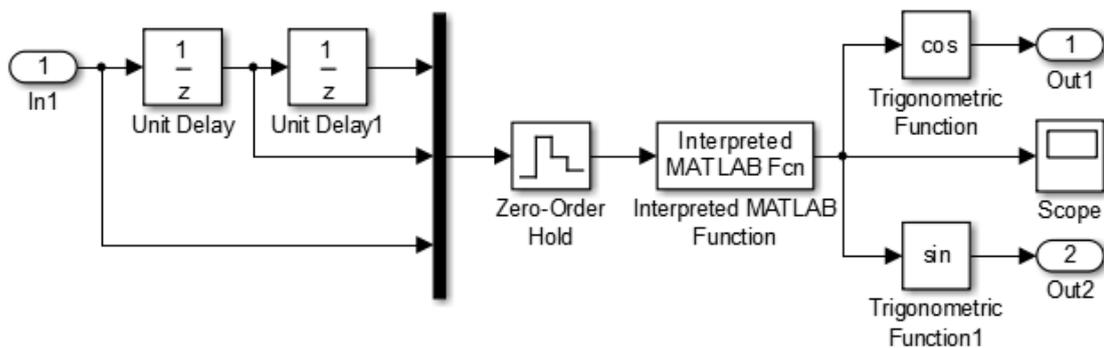


Рисунок 2.8 – Функциональная схема фазового кодера модулятора 8-PSK

Скрипт-код *M-функции* (фазового кодера 8-PSK) представлен ниже по тексту.

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

После блока задания функций *MatLab Fcn* модулирующие импульсы через блоки (*Trigonometric Function*) и (*Trigonometric Function1*) в виде импульсов пропорциональных значениям $\cos(\varphi_k)$ и $\sin(\varphi_k)$ поступают на квадратурный модулятор, аналогичный модулятору QPSK.

Параметры блоков демодулятора *Analog Filter Design* и *Zero-Order Hold* соответствуют демодулируемым трибитам.

Составной блок восстановления исходной последовательности битовых импульсов по принятым фазовым состояниям несущей, представленный на рисунке 2.9, реализован на основе блока *E-подсистемы (Subsystem1)* со встроенным блоком задания *M-функции (MatLab Fcn)* и управляемого опорным генератором прямоугольных импульсов (*Pulse Generator1*). В блоке восстановления прямоугольные квадратурные модулирующие импульсы через мультиплексор (*Mux*) поступают на блок принятия решений (фазовый декодер), реализованный программно согласно рисунку 2.10. С выхода мультиплексора квадратурные составляющие в векторной форме поступают на блок задания *M-функции (MatLab Fcn)*. Встроенная *M-функция phase_tribit.m* реализует декодирование фазовых состояний в последовательность допустимых значений трибитов. Декодированная блоком *M-функции (MatLab Fcn)* трибитовая последовательность через демультимплексор (*Demux*) поступает на выход *E-подсистемы (Subsystem1)*.

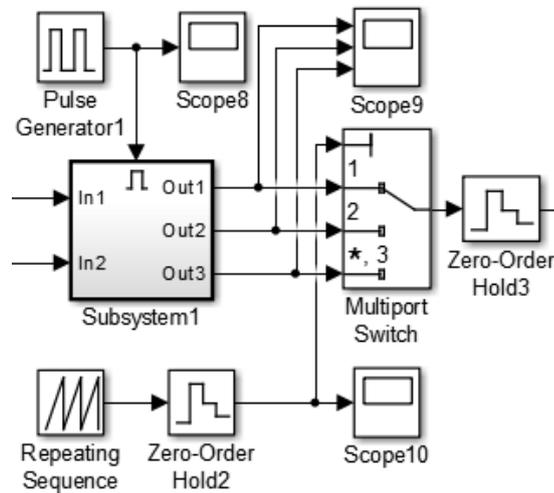


Рисунок 2.9 – Составной блок восстановления исходной последовательности битовых импульсов

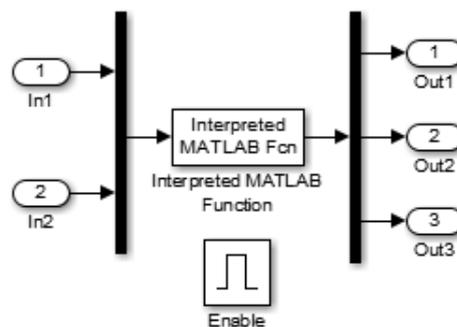


Рисунок 2.10 – Функциональная схема восстановления исходной последовательности битовых импульсов в E-подсистеме

Скрипт-код *M-функции* (фазового декодера 8-PSK) представлен ниже по тексту.

```
function y = phase_tribit(x)
%x-vector
%fi-phase
fi=atan2(x(2),x(1));
if (fi>=-pi/8)    &&    (fi<pi/8);    y=[0 0 0];    end;
if (fi>=pi/8)    &&    (fi<3*pi/8);    y=[0 0 1];    end;
if (fi>=3*pi/8)  &&    (fi<5*pi/8);    y=[0 1 1];    end;
if (fi>=5*pi/8)  &&    (fi<7*pi/8);    y=[0 1 0];    end;
if (fi>=7*pi/8)  &&    (fi<8*pi/8) || ((fi>=-8*pi/8) && (fi<-
7*pi/8));    y=[1 1 0];    end;
if (fi>=-7*pi/8) &&    (fi<-5*pi/8);    y=[1 1 1];    end;
if (fi>=-5*pi/8) &&    (fi<-3*pi/8);    y=[1 0 1];    end;
if (fi>=-3*pi/8) &&    (fi<-pi/8);    y=[1 0 0];    end;
```

Схема восстановления исходной битовой последовательности импульсов по восстановленным трибитам, представленная на рисунке 2.9, реализована на основе блока мультипортового переключателя (*Multiport Switch*), управляемого генератором треугольных импульсов (*Repeating Sequence*). На выходе схемы восстановления для исключения нежелательных скачков включен блок экстраполятора нулевого порядка (*Zero-Order Hold3*). Схемой восстановления дибиты переводятся демультимплексором в параллельное (векторное) представление. Значения текущего трибита подаются на входные порты мультипортового переключателя, которые управляющим генератором треугольных импульсов, поочередно (на длительность бита), коммутируются на выход. На выходе экстраполятора в результате появляется восстановленная исходная битовая последовательность импульсов.

На рисунке 2.11 приведена диаграмма фазовых состояний сигнала на выходе 8-PSK канала без подключения шумов.

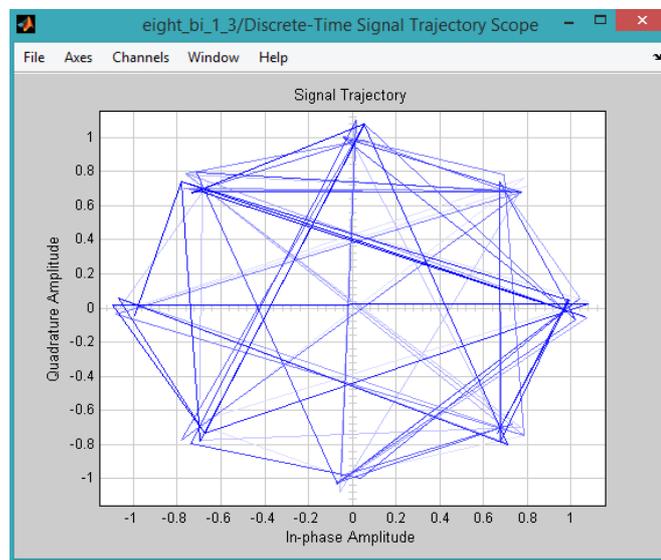


Рисунок 2.11 – Диаграмма фазовых переходов сигнала на выходе 8-PSK канала без подключения шумов

На рисунке 2.12 приведена диаграмма фазовых состояний сигнала на выходе 8-PSK канала.

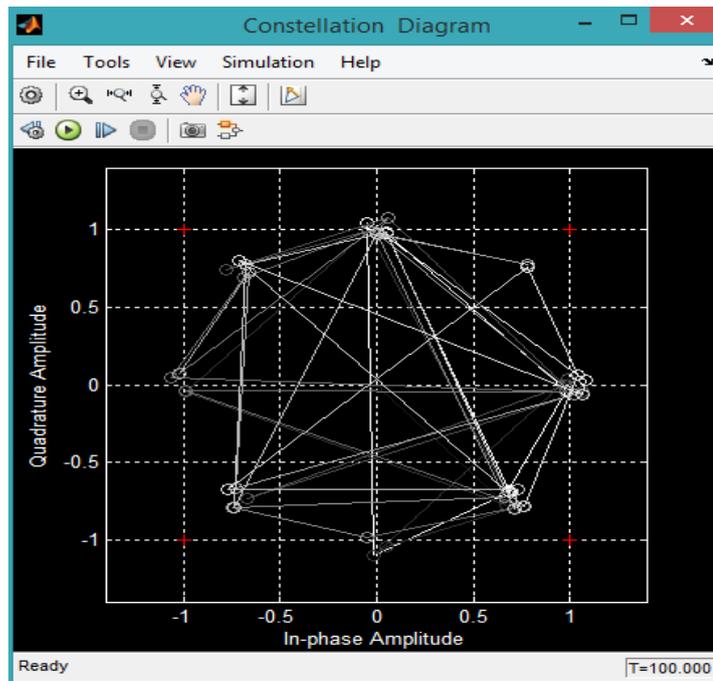


Рисунок 2.12 – Диаграмма фазовых состояний сигнала на выходе 8-PSK канала без подключения шумов

На рисунке 2.13 приведены осциллограммы переданных и принятых последовательностей.

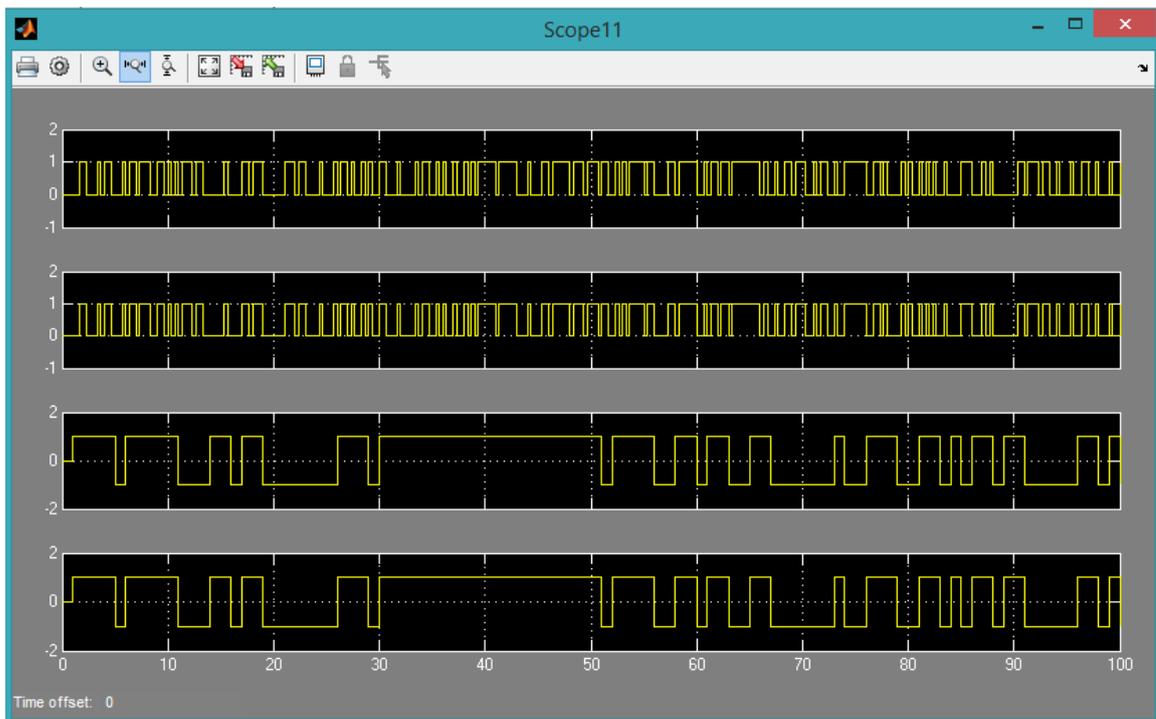


Рисунок 2.13 – Осциллограммы переданных и принятых последовательностей

3 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.

3.1 Запуск и работа с пакетом Simulink

Для запуска системы *Simulink* необходимо предварительно выполнить запуск системы *MatLab*. После открытия командного окна системы *MatLab* нужно запустить систему *Simulink* [4-7]. Это можно сделать одним из трех способов:

- нажать кнопку  (*Simulink Library*) на панели инструментов системы *MatLab*;
- в строке командного окна *MatLab* напечатать *Simulink* и нажать 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*), представленное на рисунке 3.1.

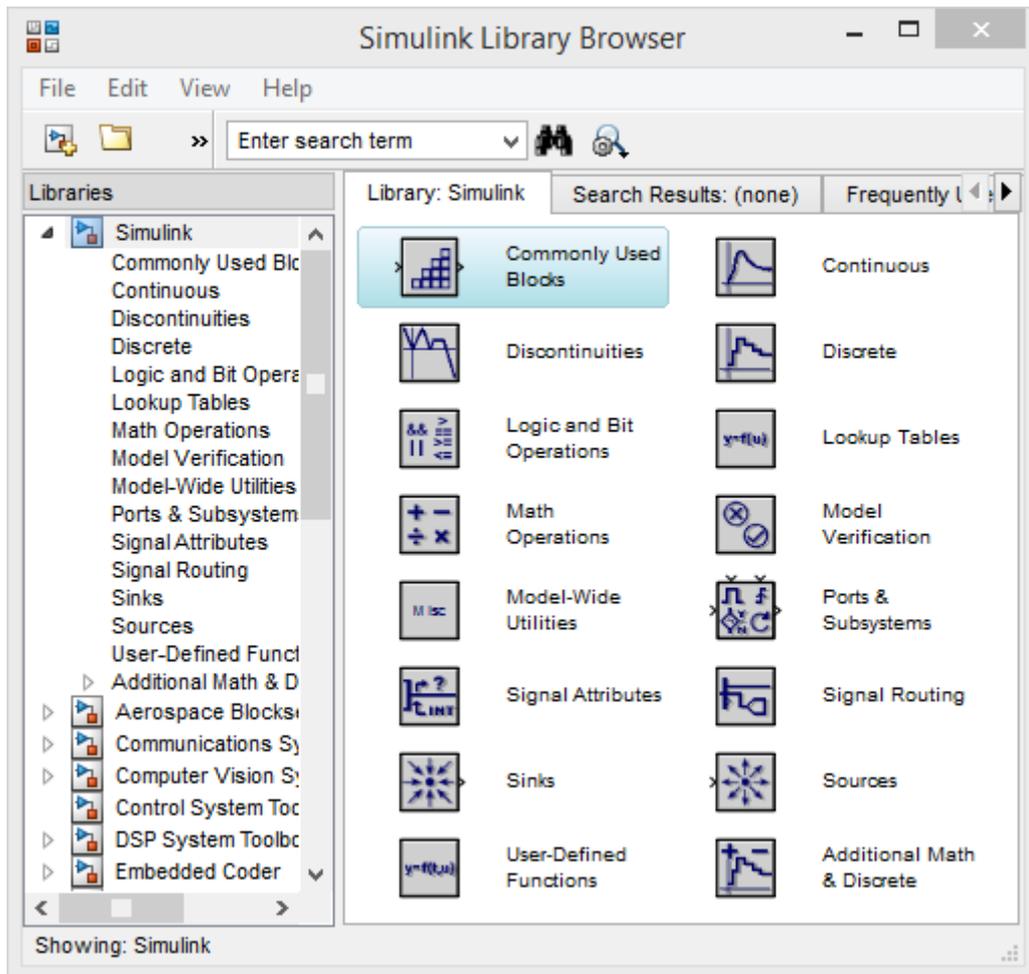


Рисунок 3.1 – Библиотека блоков *Simulink Library Browser*

На рисунке 3.1 выведена библиотека системы *Simulink* (в левой части окна) и показаны ее разделы (в правой части окна). Основная библиотека системы содержит следующие разделы:

- *Commonly Used Blocks* – часто используемые блоки;
- *Continuous* – блоки аналоговых элементов;
- *Discontinuous* – блоки нелинейных элементов;
- *Discrete* – блоки дискретных элементов;
- *Logic and Bits Operations* – блоки элементов, определяющие логические и битовые операции;
- *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.

Список разделов библиотеки представлен в виде дерева, и правила работы с ним являются общими для списков такого вида: пиктограмма свернутого узла дерева содержит символ «▸», а пиктограмма развернутого – символ «▹».

Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши (ЛКМ). При выборе соответствующего раздела библиотеки его содержимое отображается в правой части окна.

При работе элементы разделов библиотек "*перетаскивают*" в рабочую область удержанием ЛКМ 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 ЛКМ на выделенном блоке всплывает меню, в котором задаются параметры блоков.

Работа *Simulink* происходит на фоне открытого окна системы *MatLab*, закрытие которого приведет к выходу из *Simulink*.

3.2 Описание используемых блоков библиотеки Simulink

Ниже описаны основные блоки базовых разделов библиотеки *Simulink*, используемые в функциональной схеме двухканального модема передачи данных между БПЛА и НКУ [4-7].



Random Number

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.

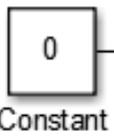


Sign

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

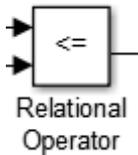
$$\text{sign} = \begin{cases} -1, & \text{где } x < 0; \\ 0, & \text{где } x = 0; \\ 1, & \text{где } x > 0. \end{cases}$$

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.

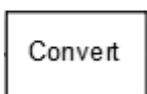


Relational Operator

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



Convert

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 – 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.

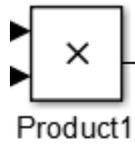
Выходной сигнал источника в режиме *Time-based* (по текущему значению времени для аналоговых систем) определяется выражением: $y = Amplitude \cdot \sin(frequency \cdot time + phase) + bias$. Параметры блока в режиме *Time-based*: *Amplitude* – амплитуда; *Bias* – постоянная составляющего сигнала (смещение); *Frequency (rads/sec)* – частота (rad/c); *Phase (rads)* – начальная фаза (rad); *Sample time* – такт дискретности, используется для получения дискретной выборки из непрерывного сигнала и может принимать следующие значения: 0 (по умолчанию) – используется при моделировании непрерывных систем; > 0 (положительное значение) – задается при моделировании дискретных систем; -1 (минус один) – такт дискретности устанавливается таким же, как и в предшествующем блоке. Данный параметр может задаваться для многих блоков библиотеки *Simulink*. Флажок *Interpreted vector parameters as 1 – D* – интерпретировать вектор как массив скаляров. Для очень больших значений времени точность вычисления значений сигнала падает.

Выходной сигнал источника в режиме *Time-based* (по текущему значению времени для дискретных систем) определяется матричным выражением:

$$\begin{bmatrix} \sin(t + \Delta t) \\ \cos(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \cos(\Delta t) & \sin(\Delta t) \\ -\sin(\Delta t) & \cos(\Delta t) \end{bmatrix} \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix},$$

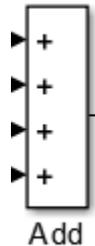
где Δt - постоянная величина, равная значению *Sample time*.

В данном режиме ошибка округления для больших значений времени меньше, чем в предыдущем режиме.



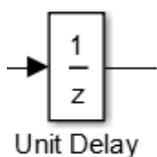
Product – блок умножения и деления. Назначение: вычисление произведения текущих значений сигналов. Параметры блока: *Number of inputs* – количество входов, может задаваться как число или как список

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.

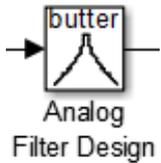


Unit delay – блок единичной дискретной задержки. Назначение: выполняет задержку дискретного сигнала на заданный шаг модельного времени. Параметры блока: *Initial conditions* – начальное значение

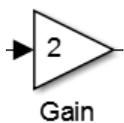
выходного сигнала; *Sample time* – шаг модельного времени.



Band-Limited White Noise – блок генератора белого шума. Назначение: создает сигнал заданной мощности, равномерно распределенной по частоте. Параметры: *Noise Power* – мощность шума, *Sample Time* – модельное время, *Seed* – число, необходимое для инициализации генератора случайных чисел.

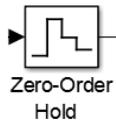


Analog Filter Design – блок аналогового фильтра заданного метода проектирования и типа из подраздела *Filter Design*; подраздела *Filtering*, раздела *DSP Blockset* библиотеки *Simulink*. Назначение: аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. Параметры блока: *Design 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: *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*.

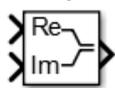
Saturate on integer – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.



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.

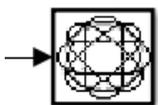
Real-Imag to
Complex



Real-Imag to Complex – блок вычисления комплексного числа по его действительной и мнимой частям. Назначение: вычисление комплексного значения по его проекциям на действительную и мнимую

оси. Параметры блока: *Input* – входной сигнал, значение выбирается из списка: *Real* – действительная часть; *Image* – мнимая часть; *RealAndImage* – действительная и мнимая часть. *Image part* – мнимая часть, параметр доступен, если *Input* определен как *Real*. *Real part* – действительная часть, параметр доступен, если *Input* определен как *Image*. Входные сигналы могут быть скалярными, векторными или матричными. Параметры *Real part* и *Image part* при векторном или матричном входном сигнале должны задаваться как векторы или матрицы.

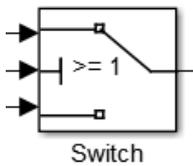
Discrete-Time
Signal Trajectory
Scope



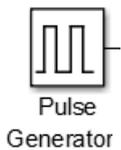
Discrete-Time Signal Trajectory Scope – блок отображения диаграммы переходов фазовых состояний сигнала из подраздела *Comm Sinks* раздела *Communication Blockset* библиотеки *Simulink*. Назначение: отображение диаграммы рассеяния фазовых состояний за счет влияния

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*

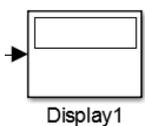
Properties, Show Axes Properties, Show Figure Properties – показать свойства отображения, осей и фигуры в нашем случае не используются.



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.

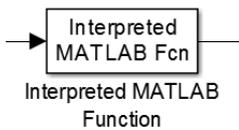


Pulse Generator – блок источника импульсного сигнала. Назначение: Формирование прямоугольных импульсов. Параметры: *Pulse Type* – способ формирования сигнала: *Time-based* – по текущему времени, *Sample-based* – по величине модельного времени и количеству расчетных шагов; *Amplitude* – амплитуда, *Period* – 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*.



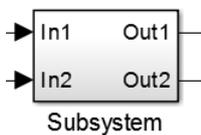
Display – блок цифрового дисплея. Назначение: отображает значение сигнала в виде числа. Параметры: *Format* – формат отображения данных. Параметр *Format* может принимать следующие значения: *short* – 5 значащих десятичных цифр, *long* – 15 значащих десятичных цифр, *short_e* – 5 значащих десятичных 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 *LBM* 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.



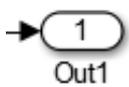
Subsystem – виртуальная и монолитная подсистемы. Доступ к окну параметров подсистемы осуществляется через меню *Edit* командой *Block Parameters*. Параметры: *Show port labels* – показать метки

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.



Inport – блок входного порта. Назначение: создает входной порт для подсистемы или модели верхнего уровня иерархии. Параметры: *Port number* – номер порта, *Port dimensions* – размерность входного сигнала. Если этот

параметр равен -1 , то размерность входного сигнала будет определяться автоматически, *Sample time* – шаг модельного времени, *Data type* – тип данных входного сигнала: *auto*, *double*, *single*, *int8*, *uint8*, *int16*, *uint16*, *int32*, *uint32* или *Boolean* \$ *Signal type* – Тип входного сигнала: *auto* – автоматическое определение типа, *real* – действительный сигнал, *complex* – комплексный сигнал; *Interpolate data* (флажок) – интерполировать входной сигнал. В случае, если временные отсчеты входного сигнала считываемого из рабочей области *MatLab* не совпадают с модельным временем, то блок будет выполнять интерполяцию входного сигнала. При использовании блока *Inport* в подсистеме данный параметр не доступен.

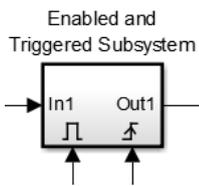


Outport – блок выходного порта. Назначение: создает выходной порт для подсистемы или для модели верхнего уровня иерархии. Параметры: *Port number* – номер порта; *Output when disabled* – вид сигнала на выходе подсистемы, в случае если подсистема выключена. Используется для управляемых подсистем. Может принимать значения (выбираются из списка): *held* – выходной сигнал подсистемы равен последнему рассчитанному значению, *reset* – выходной сигнал подсистемы равен значению задаваемому параметром *Initial output*; *Initial output* – значение сигнала на выходе подсистемы до начала ее работы и в случае, если подсистема выключена. Используется для управляемых подсистем.

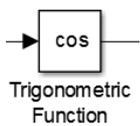
Mux – блок мультиплексора. Назначение: объединяет входные сигналы в вектор. Параметры блока: *Number of Inputs* – количество входов; *Display option* – способ отображения, выбирается из списка: *bar* – вертикальный узкий прямоугольник черного цвета; *signals* – прямоугольник с белым фоном и отображением 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.



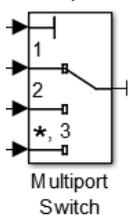
Enabled and Triggered Subsystem (ET-подсистема) – управляемая уровнем и фронтом сигнала подсистема. Подсистема включается фронтом сигнала поступающего на Т-вход системы при наличии положительного сигнала на Е-входе системы. Выполняет вычисления только на том шаге моделирования, где произошло изменение управляющего сигнала на Т-входе. Параметр *States when enabling* блока *Enable* не оказывает влияния на работу ET-подсистемы. Оба управляющих сигнала могут быть векторными.



Trigonometric Function – блок вычисления тригонометрических функций.

Назначение: выполняет вычисление тригонометрической функции.

Параметры: *Function* – вид вычисляемой функции (выбирается из списка): *sin, cos, tan, asin, acos, atan, atan2, sinh, cosh u tanh*; *Output signal type* – тип выходного сигнала (выбирается из списка): *auto* – автоматическое определение типа, *real* – действительный сигнал, *complex* - комплексный сигнал. При векторном или матричном входном сигнале блок выполняет поэлементное вычисление заданной функции.



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.

4 Экспериментальное задание

- 1 Собрать Sim-модель для исследования в соответствии с рисунком 2.1.
- 2 Выставить параметры блоков источника сигналов Sim-модели, для *Random Number1* $Mean=0$, $Variance=1$, $Seed=11$, $Sample\ time=0,5$ и для *Random Number2* $Mean=0$, $Variance=1$; $Seed=13$, $Sample\ time=1$.
- 3 Выставить параметры остальных блоков, согласовать с исходными параметрами блока источника случайного сигнала с нормальным распределением.
- 4 Пронаблюдать и зафиксировать основные осциллограммы, иллюстрирующие работу каналов BPSK и QPSK манипуляций.
- 5 Модифицировать функциональную схему QPSK в 8-PSK, пронаблюдать диаграммы, сделать выводы.
- 6 Зафиксировать все результаты работы в таблице 1.
- 7 Проанализировать зависимости помехоустойчивости от скорости передачи, соотношения сигнал/шум, сделать выводы.
- 8 Ответить на контрольные вопросы.
- 9 Написать отчет по проделанной работе.

Т а б л и ц а 1 – Результаты работы

Параметры	Первая часть. Вид модуляции		Вторая часть. Вид модуляции	
	BPSK	QPSK	BPSK	8-PSK
Длительность бита (Sample time)	1	0,5	0,5	0,5
Скорость передачи	1	2	1	2
Время моделирования 1000 с				
Количество ошибок без шумов				
Количество ошибок с шумами при: <i>Noise power=0,05</i> <i>Noise power=0,1</i> <i>Noise power=0,3</i>				
Время моделирования 10000 с				
Количество ошибок без шумов				
Количество ошибок с шумами при: <i>Noise power=0,05</i> <i>Noise power=0,1</i> <i>Noise power=0,3</i>				

5 Контрольные вопросы

- 1 Что такое BPSK, QPSK, 8-PSK?
- 2 Объясните работу BPSK, QPSK, 8-PSK модуляторов модема.
- 3 Объясните работу BPSK, QPSK, 8-PSK демодуляторов модема.
- 4 В чем преимущества манипуляций BPSK, QPSK, 8-PSK?
- 5 В чем основной недостаток манипуляций BPSK, QPSK, 8-PSK?
- 6 Изобразите упрощенные функциональные схемы BPSK, QPSK, 8-PSK модулятора.
- 7 Изобразите упрощенные функциональные схемы BPSK, QPSK, 8-PSK демодулятора.
- 8 Объясните принцип работы формирователя квадратурных модулирующих импульсов фазового манипулятора (фазового кодера).
- 9 Объясните принцип работы схемы восстановления исходной последовательности импульсов по принятым квадратурным модулирующим импульсам (фазового декодера).
- 10 Объясните принцип работы детектора ошибок. Каким образом выбирается значение параметра блока усилителя *Gain*?
- 11 Какую функцию в схеме выполняют блоки фильтров?
- 12 Опишите изменения в спектре сигнала при расширении его длительности во временной области в два раза.

Список использованных источников

- 1 Цифровая техника в радиосвязи: Двухпозиционная фазовая модуляция (BPSK). [Электронный ресурс]. – Режим доступа: <http://digteh.ru/WLL/BPSK.php>.
- 2 Цифровая техника в радиосвязи: Четырехпозиционная фазовая модуляция (QPSK). [Электронный ресурс]. – Режим доступа: <http://digteh.ru/WLL/QPSK.php>.
- 3 Цифровая техника в радиосвязи: Восьмипозиционная фазовая модуляция (8-PSK). [Электронный ресурс]. – Режим доступа: <http://digteh.ru/UGFSvSPS/modul/8PSK/>.
- 4 Гультияев А.К. MatLab 5.3. Имитационное моделирование в среде Windows: Практическое пособие. – СПб.: КОРОНА принт, 2001. - 400 с.
- 5 Черных И.В. Simulink: среда создания инженерных приложений. / Под общ. ред. В.Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2003. - 496 с.
- 6 Дьяконов В.П. MatLab 6.5 SP1/7 + Simulink 5/6. Основы применения. Сер. Библиотека профессионала. - М.: СОЛОН-Пресс, 2005. - 800 с.
- 7 Дьяконов В.П. MatLab 6.5 SP1/7 + Simulink 5/6 в математике и моделировании. Сер. Библиотека профессионала. - М.: СОЛОН-Пресс, 2005. - 576 с.