МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

Утверждаю:

Зав. каф. РТС, проф., д.т.н.

_____ Мелихов С.В.

___ _____ 2017 г.

Кологривов В.А., Токбаева А.А.

МОДЕЛЬНОЕ ИССЛЕДОВАНИЕ СИГНАЛЬНО-КОДОВЫХ КОНСТРУКЦИИ ЦИФРОВОЙ РАДИОСВЯЗИ НА ОСНОВЕ BPSK И 8-PSK МОДУЛЯЦИЙ

Учебно-методическое пособие по лабораторной и самостоятельной работе и практическим занятиям
для студентов направления
«Инфокоммуникационные технологии и системы связи»
по дисциплине
«Теоретические основы систем мобильной связи»

Разработчики

доц. каф. РТС

_____ Кологривов В.А.

_____Токбаева А.А

2017

Кологривов В. А., Токбаева А.А.

«Модельное исследование сигнально - кодовых конструкции цифровой радиосвязи на основе BPSK и 8-PSK модуляций»: Учебно-методическое пособие по лабораторной работе и самостоятельной и практическим занятиям для студентов направления «Инфокоммуникационные технологии и системы связи» по дисциплине «Теоретические основы систем мобильной связи» – Томск: ТУСУР. Научно-образовательный портал, 2017.– 41 с.

Учебно-методическое пособие содержит описание функциональной модели для исследования сигнально-кодовых конструкции в цифровой радиосвязи, выполненной в среде функционального моделирования ***Simulink***, системы для инженерных и научных расчетов ***MatLab***.

В пособии приведены краткие теоретические сведения о способах, поиска компромисса между модуляцией и кодированием, краткая характеристика пакета ***Simulink*** системы ***MatLab***, описание виртуальных лабораторных макетов и используемых блоков библиотеки ***Simulink***.

Аннотация

Лабораторная работа **«Модельное исследование сигнально - кодовых конструкции цифровой радиосвязи на основе BPSK и 8-PSK модуляций»** посвящена экспериментальному исследованию сигнально-кодовых конструкции в цифровой радиосвязи, с использованием пакета функционального моделирования ***Simulink***, системы для инженерных и научных расчетов ***MatLab***.

Работа **«Модельное исследование сигнально - кодовых конструкции цифровой радиосвязи на основе BPSK и 8-PSK модуляций»** относится к циклу лабораторных работ, входящему в дисциплины по направлению ***«Инфокоммуникационные технологии и системы связи»***.

В описании сформулирована цель лабораторной работы, приведены краткие теоретические сведения о поиске компромисса между модуляцией и кодированием, краткая характеристика пакета ***Simulink*** системы ***MatLab,*** описание виртуальных лабораторных макетов и используемых блоках библиотеки ***Simulink***, а также требования к экспериментальному исследованию и контрольные вопросы, ответы на которые необходимы для успешной защиты лабораторной работы.

Лабораторная работа рассчитана на два лабораторных задания по 2 часа.

Содержание

1. Цель работы. Краткие сведения из теории

Цель работы: Изучить принцип работы и особенности совместного применения модуляции и кодирования для достижения компромиссных параметров между скоростью передачи и помехоустойчевостью.

1.1. Теоретическая часть

В технике цифровой связи методы модуляции играют весьма значительную роль. Помимо своей основной функции – преобразования символ – сигнал – процесс модуляции является составной частью общего процесса согласования сигнала с характеристиками канала. Современные методы многопозиционной модуляции в полном соответствии с теоремой Шеннона могут рассматриваться и как способ кодирования данных сообщений в символы канала [1].

Для передачи по любому каналу связи цифровое сообщение, представляющее собой последовательность символов (чисел), необходимо преобразовать в видеоимпульс – изменяющуюся во времени физическую величину (например, напряжение). Кроме того, канал связи способен пропускать лишь определенную полосу частот, так что сформированный сигнал должен этой полосе соответствовать. Указанное преобразование осуществляется путем модуляции. Обратный процесс носит название демодуляции [2].

Манипуляция-процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала [3].

Виды манипуляций:

1) FSK - Frequency Shift Keying (Частотная манипуляция)

2) ASK- Amplitude shift keying (Амплитудная манипуляция)

3) QAM- Quadrature amplitude modulation (Квадратурная амплитудная манипуляция),

4) PSK Phase-shift keying (Фазовая манипуляция),

Помехоустойчивое кодирование. Практически важный вывод работ Шеннона состоит в том, что если скорость передачи информации меньше пропускной способности канала, то с использованием кодов, исправляющих ошибки, можно создать систему связи со сколь угодно малой вероятностью ошибки на выходе декодера канала. При этом адекватная система без корректирующего кодирования будет более сложной, дорогой и энергоемкой. Отсюда вывод: система, не имеющая корректирующего кодирования и работающая без ошибок, - это крайне неэффективная система. Наоборот, эффективная система должна иметь возможность работы в режиме с достаточно высокой частотой ошибок в потоке на входе декодера, а сам декодированный поток должен иметь крайне малую вероятность ошибки на бит [1].

Задача декодирования состоит в получении k - элементной комбинации из принятого n - разрядного кодового слова при одновременном обнаружении или исправлении ошибок [4].

Энергетический выигрыш кодирования. Положительным эффектом помехоустойчивого кодирования является либо снижение вероятности ошибки, либо снижение энергетики передачи при той же вероятности ошибки, либо и то, и другое одновременно. Таким образом, кодирование расширяет возможности компромисса между полосой и энергетикой канала, присущего любой системе связи [1].

Выбор вида канальной модуляции находится из компромисса между пропускной способностью системы передачи $\boldsymbol{Rb}$ и достоверностью полученных данных. Существуют разные методы цифровой модуляции, но большое распространение получили такие виды цифровой модуляции как: **BPSK**, **M-PSK**, где **M** может принимать значения от 4 до 256, большее значением **M** не используется на практике. С увеличением позиционности модуляции **M** возрастает спектральная эффективность, то есть увеличивается способность передачи информации с большей скоростью в более узкой полосе частот. При переходе от **BPSK** сигналов к **8-PSK** скорость передачи

возрастает в 3 раза при неизменной занимаемой полосе частот. Но при увеличении позиционности модуляции требуется увлечение отношения сигнал/шум (**SNR)** для сохранения достоверного приема данных. Отношение сигнал/шум **SNR** является основной характеристикой определяющей качество приема данных, и оно влияет на энергетическую эффективность модуляции. Отношение сигнал/шум на входе системы определяется из выражения:

$$\mathrm{SNR} = \frac{S}{N} = \frac{E_b}{N_0 \cdot T_b \cdot W} = \frac{E_b \cdot R_b}{N_0 \cdot W},$$

где S – мощность сигнала;

N – мощность шумов;

E_b – энергия бита;

N_0 – спектральная мощность шума;

W – занимаемая полоса пропускания;

T_b – длительность бита;

R_b – битовая скорость передачи.

Выражение для определения энергетической эффективности модуляции примет вид:

$$\frac{E_b}{N_0} = \frac{S \cdot W}{N \cdot R_b}.$$

В нашем исследовании будем использовать **BPSK** модуляцию, у которой **M=2, 8-PSK**, у которой **M=8**, при этом водопадоподобная характеристика энергетической эффективности модуляции для них разная. На рисунке 1.1 представлены характеристики энергетической эффективности **PSK**-модуляций, в том числе для **BPSK**, **QPSK** и **8-PSK**.

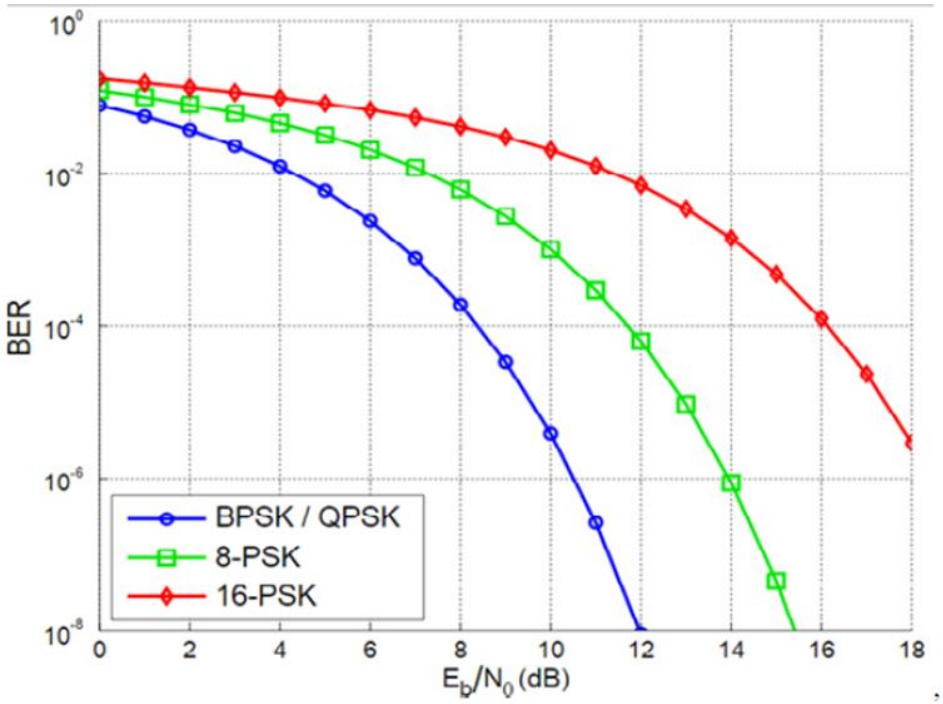


Рисунок 1.1 - Характеристика энергетической эффективности некоторых видов PSK-модуляций

У **8-PSK** полоса пропускания в 3 раза уже чем у **BPSK.** из-за того что символ состоит из трех бит (трибит).

2. Краткое описание пакета Simulink

2.1 Общая характеристика пакета Simulink

Пакет **Simulink** распространяется в составе математического пакета **MatLab.** Пакет основан на графическом интерфейсе и является типичным средством визуально-ориентированного программирования. Пакет **Simulink** обладает обширной библиотекой готовых блоков с модифицируемыми параметрами для построения моделей рассматриваемых систем и наглядными средствами визуализации результатов моделирования [5,6,7].

2.2 Запуск и работа с пакетом Simulink

Для запуска системы **Simulink** необходимо выполнить запуск системы **MatLab**. После открытия командного окна системы **MatLab** нужно запустить систему **Simulink**. Существует три способа запуска системы **Simulink**:

- 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).

При применении двух первых способов открывается окно обозревателя библиотеки блоков (**Simulink Library Browser**). Если нам не требуется добавление новых блоков, а нужно лишь открыть уже готовую модель и провести моделирование, то следует воспользоваться третьим способом.

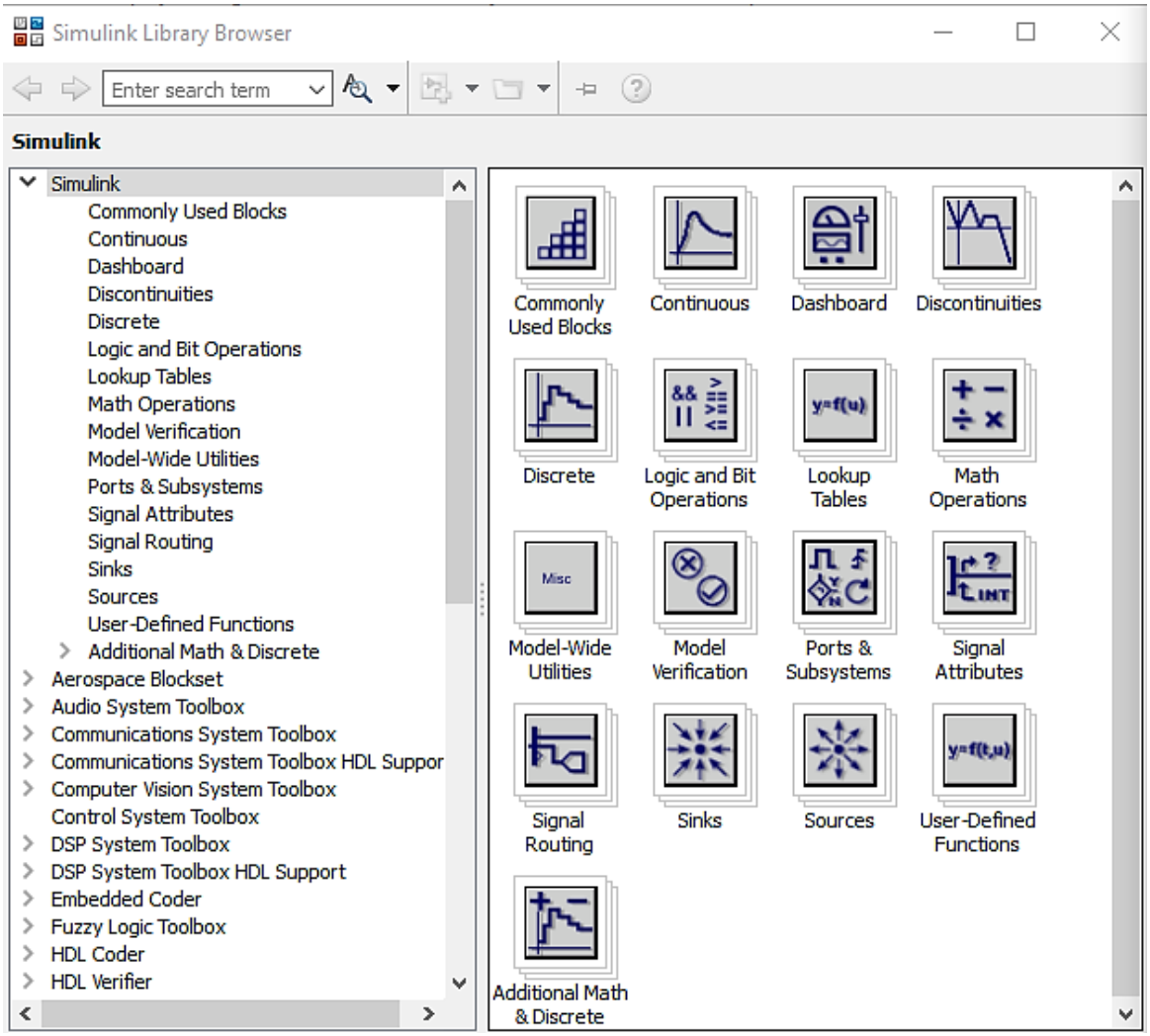


Рисунок 2.1 – Библиотекаблоков**Simulink Library Browser**

На рисунке 2.1 выведена библиотека системы **Simulink**и показаны ее разделы. Основная библиотека системы содержит следующие разделы:

- **Continuous**– блоки аналоговых элементов;
- **Dashboard** – панель приборов
- **Discontinuous** – блоки нелинейных элементов;
- **Discrete** – блоки дискретных элементов;
- **Logic and Bit Operations** – Логические и битовые операции;
- **Lookup Tables** – блокитаблиц;
- **Math Operations** – блоки элементов, определяющие математические операции;
- **Model-Wide Utilities –** раздел дополнительных утилит**;**

- **Model Verification** – блоки проверки свойств сигнала;
- **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-DefinedFunction** – функции, определяемые пользователем.
- **Additional Math & Discrete –** Дополнительная и дискретная математика.

Правила работы со списком разделов библиотеки: пиктограмма свернутого узла содержит символ **«+»,** а пиктограмма развернутого – символ **«-».**

Для того чтобы развернуть или свернуть узел, достаточно щелкнуть на его пиктограмме левой клавишей мыши.

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 приведёт к выходу из **Simulink**.

3. Описание лабораторных макетов

3.1 BPSK модем

Принцип работы функциональной модели при **BPSK–** модуляции:

Передающая часть. Random number – формирует случайный процесс с нормальным распределением уровней, который поступая на блок **Sign,** преобразуется в биполярную псевдослучайную информационную последовательность с нормальным распределением. Дальше, сформированный таким образом цифровой сигнал попадает в блок **Product** где происходит перемножение текущих значений сигнала с синусоидальным колебанием несущей частоты, который формируется блоком **Sine Wave**. Далее с помощью блока **Sum** добавляются широкополосные шумы канала распространения радиосигнала.

Приемная часть. На приемной стороне сигнал поступает на блок канального полосового фильтра (**Analog Fillter Design)**, настроенного на несущую частоту с полосй пропускания порядка $\Delta\omega=4\pi$. Канальные фильтры производят предварительное выделение радиосигнала своего канала, включая шумы канала распространения. Далее предварительно отфильтрованный сигнал и помехи подаются на блок **Product,** где с целью демодуляции происходит перемножение полученного сигнала с колебаниями опорного генератора несущей частоты канала. После этого сигнал фильтруется в блоке **ФНЧ** (**Analog Fillter Design)** с целью подавления высокочастотных составляющих на выходе умножителя-преобразователя. В процессе модельного исследования частоты среза **ФНЧ** принимались равными π либо 2π. Затем **ФНЧ** отфильтрованный сигнал подается на блок **Zero-Order Hold**. В этом блоке происходит принятие решения, поскольку блок фиксирует значение входного сигнала в начале интервала дискретизации и удерживает его до конца интервала. Блок **Sign** завершает процесс регенерации формы принятого сигнала – псевдослучайную биполярную последовательность прямоугольных импульсов с нормальным распределением. Для визуализации принятой и переданной

последовательностей используется блок **Scope**, выполняющий функцию многолучевого осциллографа.

Детектор ошибок. Для отслеживания ошибок при приеме используем детектор ошибок – блок **Subsystem**, который подсчитывает и отображает на блоке **Display** количество битовых ошибок. Функциональная схема детектора ошибок представлена на рисунке 3.1

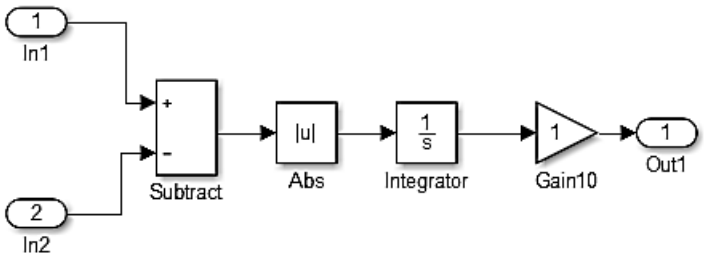


Рисунок 3.1 - Функциональная схема детектора ошибок

На первый вход поступает полученный сигнал, на вход второй поступает исходный сигнал, задержанный сигнала на один такт. В сумматоре (**Sum**) вычисляется разница текущих значений входных сигналов и модуль разности (**Abs**) исходной и принятой последовательности. Модуль разности подается на интегратор (**Integrator**) для накопления информации об ошибках. Блок **Gain** используется для коррекции значения ошибок при использовании последовательностей с разной длиной и/или полярностью импульсов. Для отображения числа ошибок с выхода детектора информация поступает на блок **Display**.

Измеритель мощности. Для проведения оценки помехоустойчивости системы используем измеритель мощности – блок **Subsystem**, функциональная схема, которого приведена ниже на рисунке 3.2.

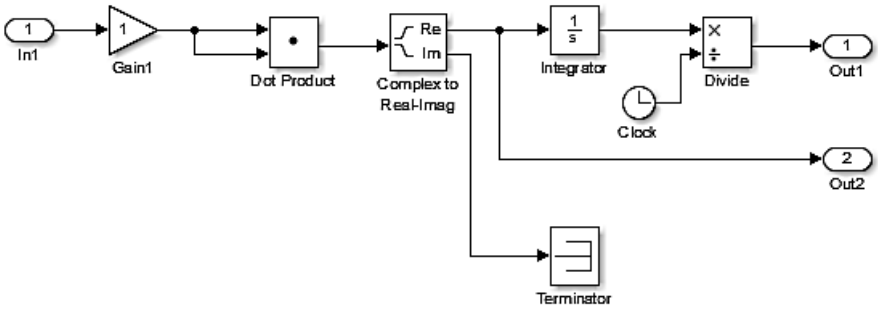


Рисунок 3.2 - Функциональная схема измерителя мощности

На вход один поступает полученный сигнал после фильтрации на блок **Gain** используемый как масштабный множитель. Далее измеряемый сигнал поступает на блок **Dot Product**, где вычисляется скалярное произведение комплексно сопряженных последовательностей. Если входная последовательность является вещественной, то выходной сигнал будет действительным. Если же входная последовательностей будет иметь комплексный вид, то на выходе будет комплексный сигнал. После сигнал поступает на блок **Complex to Real-Image,** который вычисляет действительную и мнимую части комплексного числа. Дальше мнимую часть глушим в блоке **Terminator,** а действительная часть числа поступает на выход два и на вход осциллографа, а также поступает на интегратор, где накапливается энергия входного процесса, как интеграл от квадрата входной функции. После интегрирования сигнал поступает на блок **Product**, где происходит деление время, т.е. вычисление значения текущей мощности входного процесса. Далее информация о мощности поступает на выход **1** и на блок **Scope** для отображения.

Общая функциональная схема **BPSK**-модуляции представлена на рисунке 3.3.

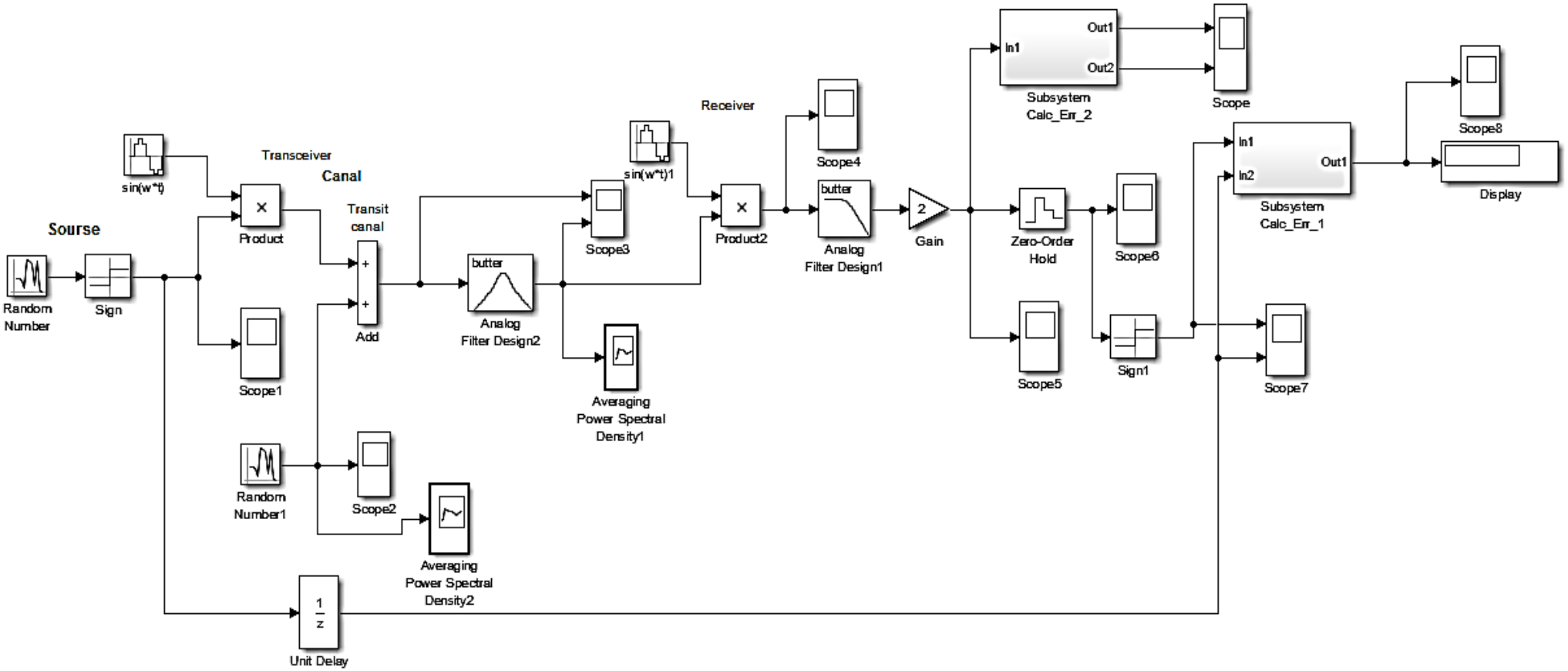


Рисунок 3.3- Функциональная схема **BPSK**-модема

На рисунке 3.4 приведена функциональная модель **BPSK** модема с помехоустойчивым кодером/декодером блочного кода (6,3).

Основное отличие модема с кодером (рисунок 3.4) от модели без кодера (рисунок 3.3) состоит в подсистемах канальных кодера и декодера.

Функциональная схема кодера приведена на рисунке 3.5

Функциональная схема канального кодера включает в себя формирователь кодового символа, построенного на элементах задержки **Unit Delay**,блоках **XOR** и мультиплексоре (**Mux**). С выхода кодера кодовые символы поступают на вход фазового модулятора **BPSK**.

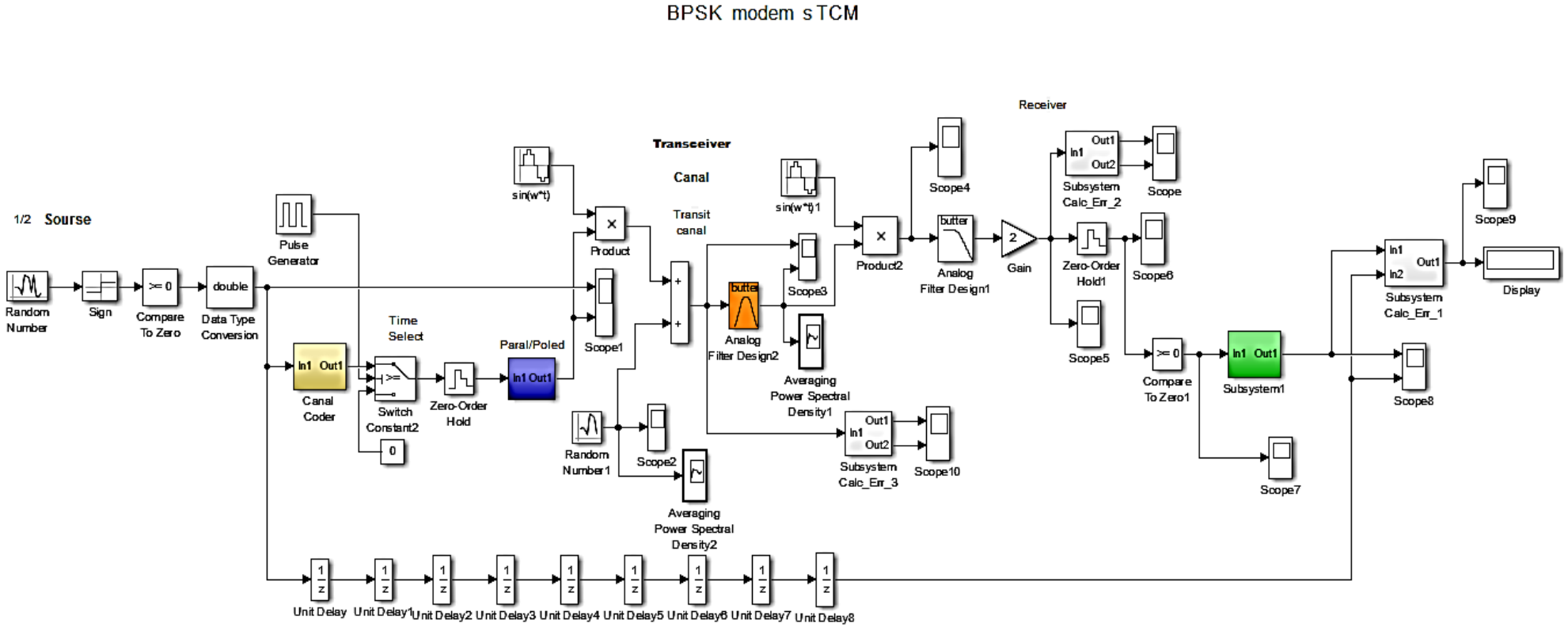


Рисунок 3.4 – Функциональная модель **BPSK** модема с помехоустойчивым кодером/декодером блочного кода (6,3)

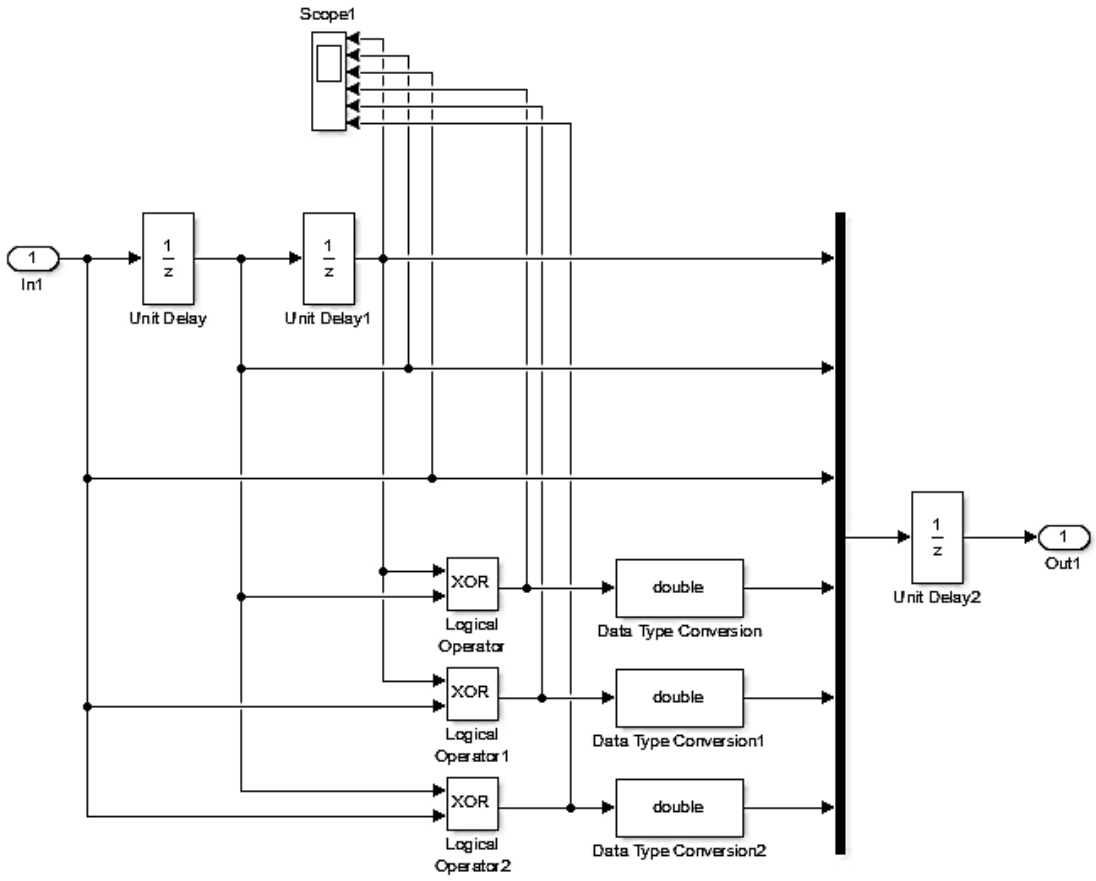


Рисунок 3.5 - Функциональная схема кодера **BPSK** модема

Функциональная схема фазового декодера приведена на рисунке 3.6.

С помощью блоков **Unit Delay** подбраем задержку в начало такта генератора переодической последовательности **Repeating Sequence.** В блоке **Buffer** накапливаем биты кодового символа и с помощью блока **Frame Conversion** подаём на собственно декодер, реализованый блоками **Mux** и **XOR.** С помощью блоков **XOR** определяется вектор синдрома ошибки. Далее систематическую часть кодового символа снимаем блоком **DeMux,** а синдром с помощью блока **DeMux** подаем на блок **Combinatorial Logic** вырабатывающего вектор ошибки. Систематическая часть кодового символа для исправления битовой ошибки блоком **XOR** суммируется с вектором ошибки. С помощью блоков **Repeating Sequence, Zero-Order Hold** и **Multiport Switch** параллельное представление декодированого символа преобразуется в последовательный поток битов.

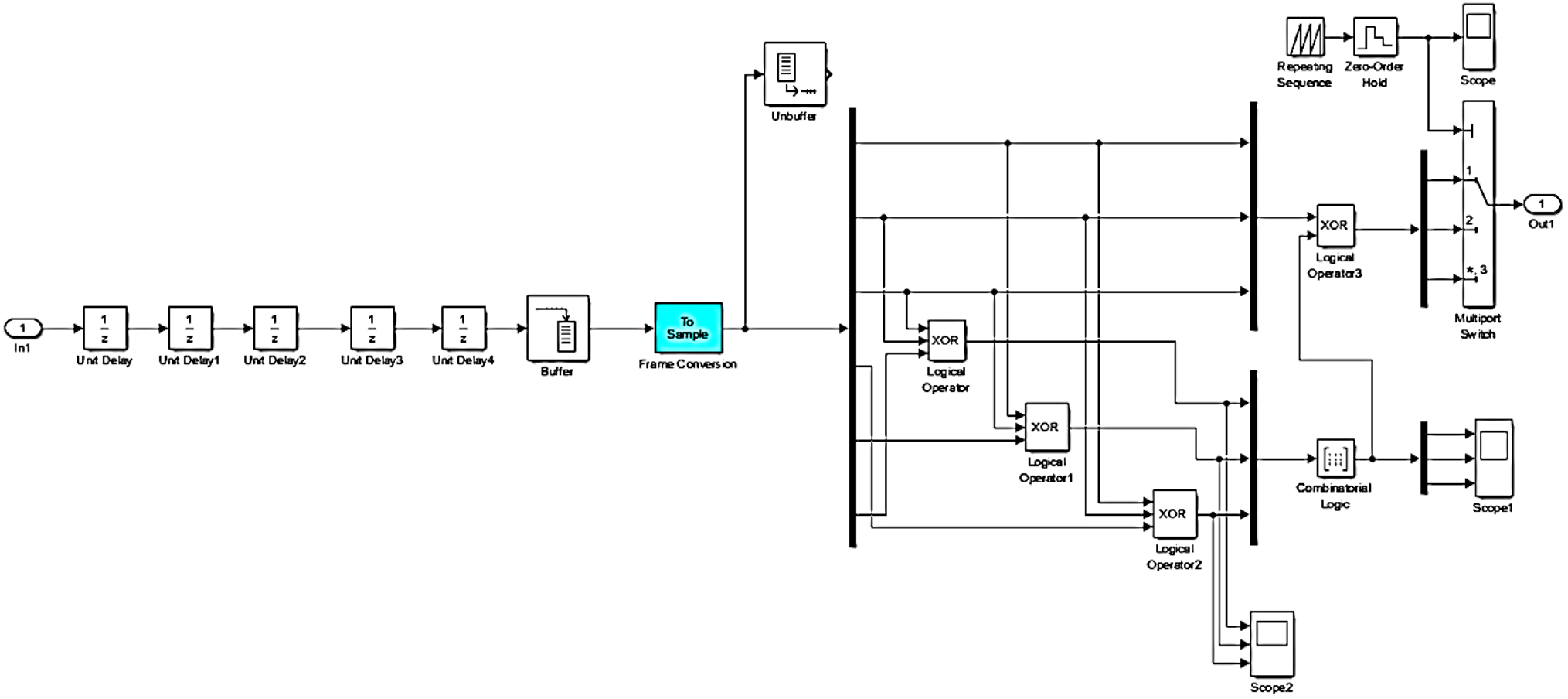


Рисунок 3.6 - Функциональная схема декодера **BPSK** модема

3.2 Элементы теории блочного алгебраического кодирования

Блочное алгебраическое кодирование подразумевает разбиение битовой последовательности на блоки и введение избыточности (бит четности). Пусть размер исходного блока составляет 3 бита (m=3). Нарастим бит 3-мя битами четности (k=3). В итоге кодовый символ будет содержать 6 бит (n=m+k=3+3=6). Такой код принято обозначать как агебраический код (6,3).

Формирование систематического кода (6,3) можно описать следующей системой уравнений:

$$\left\{\begin{array}{l} 1\cdot x_k \oplus 0\cdot x_{k+1} \oplus 0\cdot x_{k+2} = y_k \\ 0\cdot x_k \oplus 1\cdot x_{k+1} \oplus 0\cdot x_{k+2} = y_{k+1} \\ 0\cdot x_k \oplus 0\cdot x_{k+1} \oplus 1\cdot x_{k+2} = y_{k+2} \\ 0\cdot x_k \oplus 1\cdot x_{k+1} \oplus 1\cdot x_{k+2} = y_{k+3} \\ 1\cdot x_k \oplus 0\cdot x_{k+1} \oplus 1\cdot x_{k+2} = y_{k+4} \\ 1\cdot x_k \oplus 1\cdot x_{k+1} \oplus 0\cdot x_{k+2} = y_{k+5} \end{array}\right\}, \qquad (3.1)$$

где x_k - информационный бит; y_k - канальный бит.

Структура уравнения 3.1 описывает структуру алгебраического кодера (6,3). Транспонированная матрица коэффициентов этого уравнения представляет собой генерирующую или порождающую матрицу **G**. Тогда операция кодирования может быть записана в виде:

$$y = x \cdot G, \qquad (3.2)$$

где x - информационный вектор; y - кодовый вектор. В процессе передачи данных кодовые символы (вектора) могут приниматься с ошибками $y \Rightarrow z$.

Строки уравнения 3.1 соответсвуют битам кодового символа. Можно заметить, что в отсутствии ошибок, при сложении по модулю 2, в уравнении 3.1 строк **1**, **2**, **6**; **1**, **3**, **5**; и **2**, **3**, **4** будем получать нули. Этот факт может быть положен в основу проверки правильности передаваемых бит. Из коэффициентов перечисленных строк можно сформировать матрицу H

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Матрица H^T называется проверочной матрицей. Наличие ошибок передачи обнаруживается по синдрому *s*, формируемому уравнением:

$$s = z \cdot H^T. \qquad (3.3)$$

Отличие от нуля синдрома свидетельствует о наличии ошибок. Структура уравнения 3.3 описывает структуру алгебраического декодера. Функциональные модели канального кодера и декодера рисунки 3.5 и 3.6 выполнены в соответствии со структурами кодирующей матрицы **G** и проверочной матрицы H^T.

3.3 8-PSK модем

Рассмотрим принцип работы и структуру модема с **8-PSK**-модуляцией.

Передающая часть. На рисунке 3.7 приведена схема модулятора **8-PSK**. Информационная последовательность, генерированная, блоками **Random Number** и **Sign** подается на подсистему фазового кодера изображенного на рисунке 3.8. Управляющие сигналы с выходов фазового кодера поступают на первые входы квадратурного преобразователя. На вторые входы преобразователя подаются гармонические колебания опорных генераторов, сдвинутые на 90°. С выходов преобразователя сигналы подаются на сумматор.

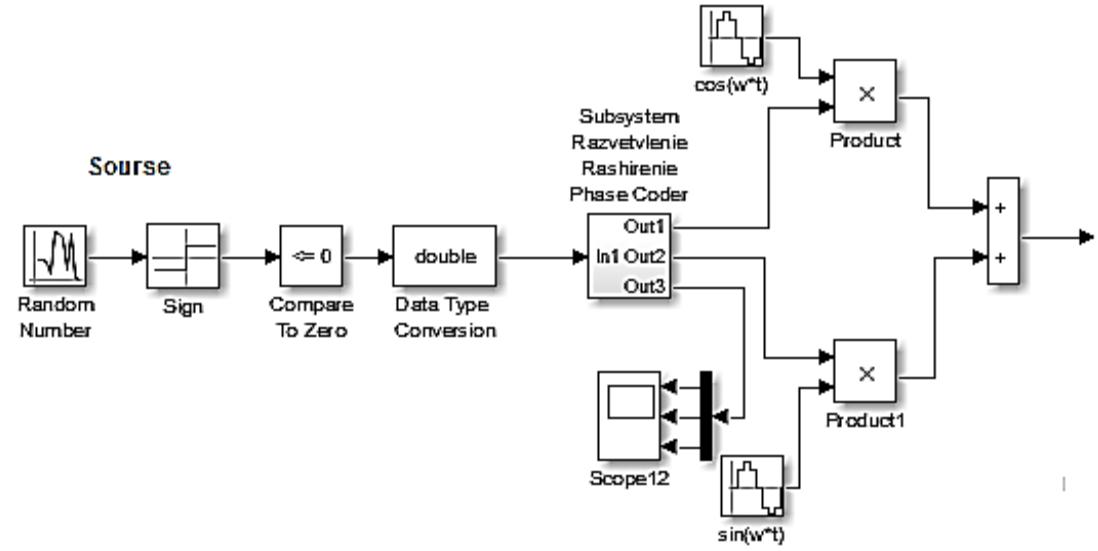


Рисунок 3.7 – Функциональная модель передающей части **8-PSK** модема

Функциональная схема фазового кодера включает в себя формирователь трех битового символа, построенного на элементах задержки **Unit Delay**, и мультиплексоре (**Mux**). С помощью блока **Zero-Order Hold** трёхбитовый символ расширяется до длительности трёх бит. Далее трёхбитовый символ поступает на блок **MATLAB Function**, в котором каждому трёхбитовому символу ставится в соответствие значение фазового состояния. С помощью тригонометрических функций **cos** и **sin** получаем управляющий сигнал квадратурного модулятора **8-PSK.**

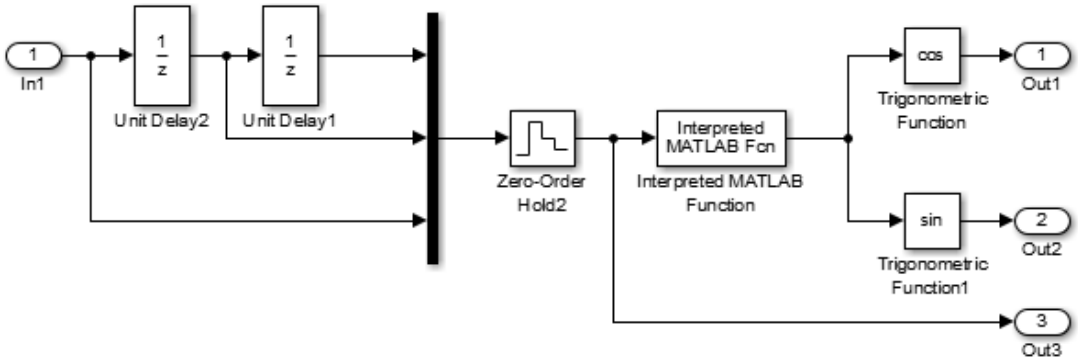


Рисунок 3.8. – Функциональная схема фазового кодера **8-PSK** модема

```
tribit_phaze.m
1   function fi=tribit_phaze(x);
2
3 -   if x ==[0;0;0]; fi=0; end;
4 -   if x == [0;0;1]; fi=pi/4; end;
5 -   if x == [0;1;1]; fi=2*pi/4; end;
6 -   if x == [0;1;0]; fi=3*pi/4; end;
7 -   if x == [1;1;0]; fi=4*pi/4; end;
8 -   if x == [1;1;1]; fi=-3*pi/4; end;
9 -   if x == [1;0;1]; fi=-2*pi/4; end;
10 -  if x == [1;0;0]; fi=-pi/4; end;
```

Рисунок 3.9 - Код функции **tribit_phaze**

Приёмная часть. На рисунке 3.10 приведена функциональная модель **8-PSK** демодулятора.

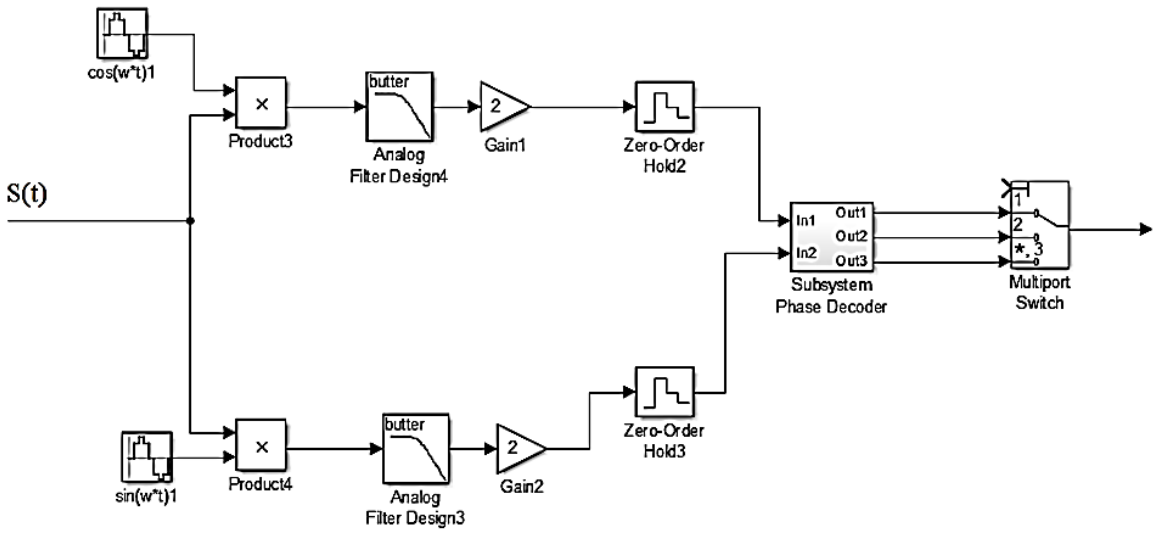


Рисунок 3.10 – Функциональная модель **8-PSK** демодулятора

В демодуляторе принятый зашумлённый сигнал разделяются на два канала обработки и подаются на первые входы преобразователей. На вторые входы подаются колебания опорных генераторов. Далее сигналы поступают на **ФНЧ**, где отфильтровываются высокочастотные составляющие и выделяются управляющая информационная последовательность. С помощью блоков **Zero-Order Hold** берутся отсчеты управляющих символов и подаются на вход подсистемы фазового декодера. В фазовом декодере принятым квадратурным составляющим информационных символов ставится в соответствие трёхбитовый код, который подается на мультипортовый ключ. С помощью управляющего пилообразного сигнала мультипортовый ключ разделяет параллельный трёхбитовый код на последовательность бит. На рисунке 3.11 приведена функциональная модель фазового декодера, выполнена на основе блока **MatLab Function**. Блок **MatLab Fcn** содержит **MatLab**-функцию, которая каждому состоянию фазы φ_k ставит в соответствие значение трибита. Программный код функции **phaze_tribit** представлен на рисунке 3.12

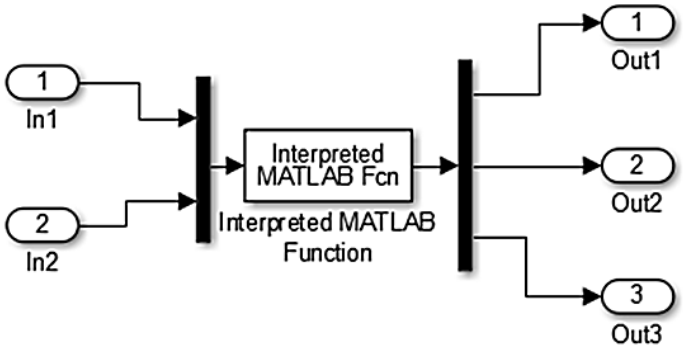


Рисунок 3.11 – Функциональная схема фазового декодера

phaze_tribit.m

```
function y = phaze_tribit(u)
fi=atan2(u(2),u(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;

```

Рисунок 3.12 - Код функции **phaze_tribit**

На рисунке 3.13 представлена функциональная схема **8-PSK** модема, а на рисунке 3.14 представлена функциональная модель **8-PSK** модема с помехоустойчивым кодером/декодером блочного кода (6,3). На входе и выходе этого модема, как и при **BPSK** и **QPSK** модуляциях стоят подсистемные блоки канального помехоустойчивого кодера и декодера (см. рисунки 3.5 и 3.6).

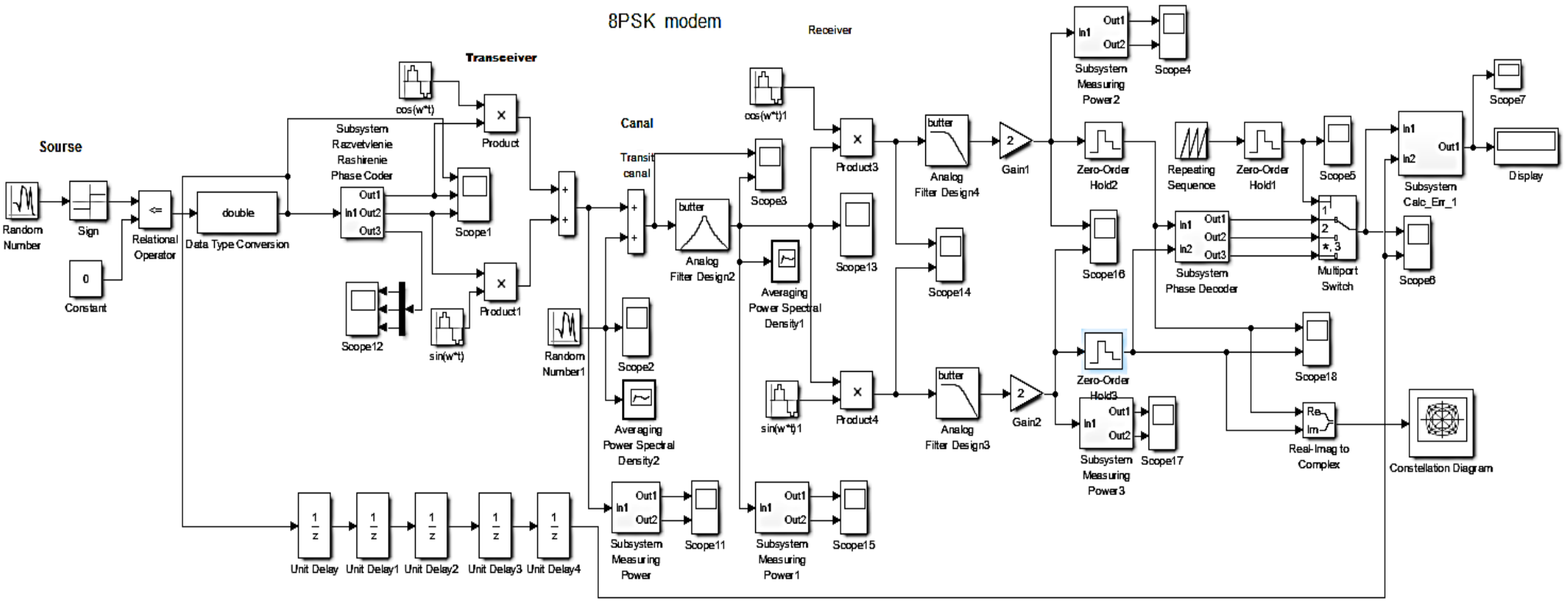


Рисунок 3.13 – Функциональная схема **8-PSK** модема

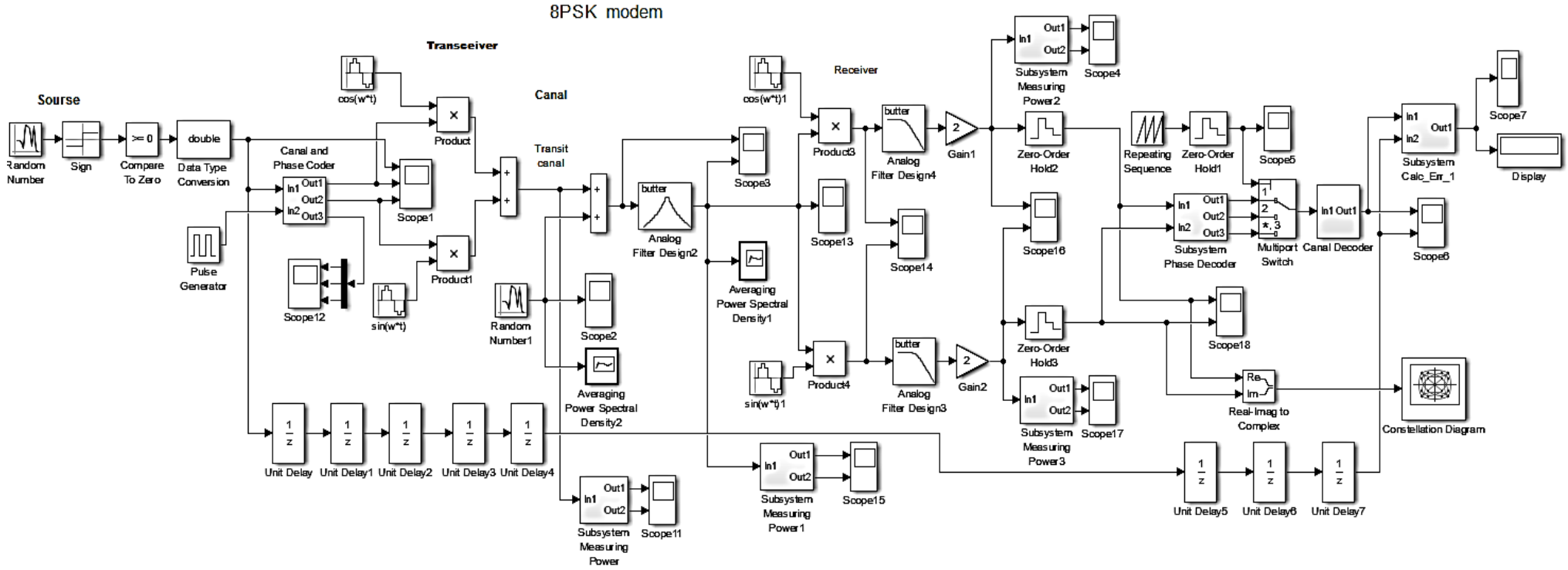


Рисунок 3.13 – Функциональная модель **8-PSK** модема с помехоустойчивым кодером/декодером блочного кода (6,3)

4. Описание используемых блоков библиотеки Simulink

Ниже описаны основные блоки базовых разделов библиотеки **Simulink** [2], используемые в функциональной схеме двухканального модема при **BPSK** модуляции.

Random Number – источник случайного сигнала с нормальным распределением. *Назначение:* формирование случайного сигнала с равномерным распределением уровня сигнала. *Параметры блока :***Mean**– среднее значение сигнала; **Variance**– дисперсия; **Initia lseed** – начальное значение генератора случайного сигнала; **Sample time** – такт дискретности

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

$$sign\,(x) = \begin{cases} -1 \text{ при } x < 0, \\ 0 \text{ при } x = 0, \\ 1 \text{ при } x > 0. \end{cases}$$

Параметры блока: флажок – **Enable Zero Crossing Detection** позволяет фиксировать прохождение сигнала через нулевой уровень.

Switch – блок переключателя. *Назначение:* переключение входных сигналов по сигналу управления. *Параметры блока:***Criteria for Passing Firstinput** – условие прохождения сигнала с первого входа, значение выбирается из списка: **u2 > =Threshold** – сигнал управления больше или равен пороговому значению; **u2 > Threshold** – сигнал управления больше порогового значения; **u2~=Threshold** – сигнал управления не равен

пороговому значению. **Threshold –** порог; флажок **Show Additional Parameters** – показать дополнительные параметры.

Scope – блок осциллографа. *Назначение:* построение графиков исследуемых сигналов как функций времени. Открытие окна осциллографа производится двойным щелчком **ЛКМ** на пиктограмме блока. 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. Для формирования выходного сигнала блоком могут использоваться два алгоритма. Вид алгоритма определяется параметром **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.

Параметры блоков в режиме *Time-based:* Amplitude – амплитуда; **Bias** – постоянная составляющего сигнала (смещение); **Frequency (rads/sec)** – частота (рад/с); **Phase (rads)** – начальная фаза (рад); **Sample time** – такт

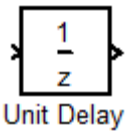
дискретности. Параметр может принимать следующие значения: **0** (по умолчанию) – используется при моделировании непрерывных систем; **> 0** (положительное значение) – задается при моделировании дискретных систем; -1 (минус один) – такт дискретности устанавливается таким же, как и в предшествующем блоке. Флажок **Interpret Vector Parameters As 1 – D** – интерпретировать вектор как массив скаляров. Для очень больших значений времени точность вычисления значений сигнала падает.

Параметры блоков режиме **Sample-based:Amplitude**– амплитуда; **Bias**– постоянная составляющего сигнала (смещение); **Samples Per Period** – количество тактов на один период синусоидального сигнала:

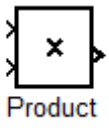
$$N = Samples\ per\ second = \frac{1}{f \cdot T}$$

$$p = Samples\ per\ period = 2 \cdot \pi \cdot N$$

Number Of Offset Samples – начальная фаза сигнала, задается количеством тактов дискретности. **Sample time** – такт дискретности. Флажок **Interpret Vector Varameters As 1 - D** – интерпретировать вектор как одномерный. В данном режиме ошибка округления не накапливается, поскольку **Simulink** начинает отсчет номера текущего шага с нуля для каждого периода.

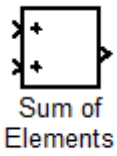


Unit Delay – блок единичной дискретной задержки. *Назначение:* выполняет задержку дискретного сигнала на заданный такт дискретности. *Параметры блока :***Initial Conditions** – начальное значение выходного сигнала; **Sample time** – такт дискретности (при задании значения параметра равного -1 такт дискретности наследуется от предшествующего блока).

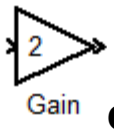


Product – блок умножения и деления. *Назначение:* вычисление произведения текущих значений сигналов. *Параметры блока:* **Number Of**

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.



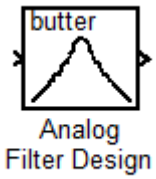
Sum – блок сумматора. *Назначение:* вычисление алгебраической суммы текущих значений входных сигналов. *Параметры блока :***Iconshape** – форма блока, выбирается из списка: **round** – круг; **rectangular** – прямоугольник. **Listofsign** – список знаков из набора: **+ - плюс; – - минус, | - разделитель.** 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.



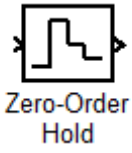
Gain – блок усилителя. *Назначение:* выполняет умножение входного сигнала на постоянный коэффициент; *Параметры блока:* **Gain -** коэффициент усиления. **Multiplication** – способ выполнения операции, значение параметра выбирается из списка: **Element - wise K*u** – поэлементный; **Matrix K*u** – матричный, 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.

Блоки **Gain** и **Matrix Gain** по сути есть один и тот же блок, но с разными начальными установками параметра **Multiplication.**

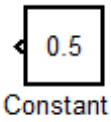


Analog Filter Design – блок аналогового фильтра заданного метода проектирования и типа из подраздела **Filter Design**. *Назначение:* аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. *Параметры блока:* **Design Method** – метод проектирования, выбирается из списка: **Butterworth** – фильтр Баттерворта; **Chebuschev I** – фильтр Чебышева 1-го рода; **Chebuschev II** – фильтр Чебышева 2-го рода; **Elliptic** – фильтр эллиптический; **Bessel** – фильтр Бесселя. **Filter Type** – тип фильтра, выбирается из списка: **Low Pass**– нижних частот; **High pass** – верхних частот; **Band Pass** – полосно-пропускающий; **Band Stop** – 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.

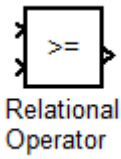


Zero – Order Hold – экстраполятор нулевого порядка. *Назначение:* экстраполяция входного сигнала на интервале дискретизации. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания

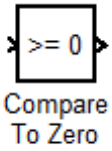
интервала дискретизации. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге дискретизации. *Параметры блока:* **Sample Time** – такт дискретности. Блок экстраполятора нулевого порядка может использоваться также для согласования работы дискретных блоков, имеющих разные такты дискретности.



Constant – блок источника постоянного сигнала. *Назначение* :задает постоянный по уровню сигнал. *Параметры блока:* **Constant Value** – постоянная величина. **Interpret Vector Parameters As 1-D** – интерпретировать вектор как массив скаляров. **Show Additional Parameters** – показать дополнительные параметры. При выставленном флажке появится окно списка Output Data Type Mode. **Output Data Type Mode** – выбор типы выходных данных.**Output Data Type** – тип выходных данных. **Output Scaling Mode –** способ масштабирования выходного сигнала. **Output Scaling Value** – величина масштаба.

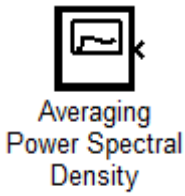


Relational Operator – блок выполнения операций отношения. *Назначение:* блок сравнивает текущие значения входных сигналов. *Параметры блока:* Relational Operator - тип операции отношения, выбирается из списка. **Show Additional Parameters** – показать дополнительные параметры. **Require All Inputs To Have Same Data Type -**вс е входы должны иметь одинаковый тип данных. **Output Data Type Mode** – выбор типа выходных данных. **Output Data Type** – тип выходных данных. **Enable Zero Crossing Detection-**фиксировать прохождение сигнала через нулевой уровень.

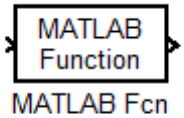


Compare To Zero – блок сравнения с нулем. *Назначение:* сравнить с нулем заданный сигнал. *Параметры блока:* **Operation** – операции сравнения. Выбираются из списка. **Output Data Type Mode** – выбор типа выходных данных.

Demux – демультиплексор. *Назначение:* разделяет входной векторный сигнал на отдельные составляющие. *Параметры блока:* **Number Of Outputs**-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. **Bus Selection Mode -** режим разделения векторных сигналов по шине.



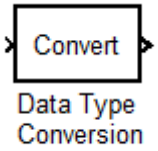
Averaging Power Spectral Density – анализатор усредненной спектральной плотности мощности. *Назначение:* отобразить частотную зависимость спектральной плотности мощности. *Параметры блока:* **Length Of Buffer** длина буфера **– Number Of Points For Fft**-количество точек. **Plot After How Many Points –** количество точек по которому выводится график.**Sample Time**-такт дискретности.



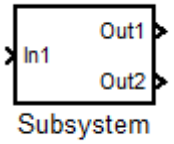
MATLAB Function – блок задания **M** – функции. *Назначение:* задает выражение в стиле языка программирования **MATLAB**. *Параметры блока:* **MATLAB Function** – выражение на языке **MATLAB. Output Dimensions** – размерность выходного сигнала.**Output Signal Type** –

тип выходного сигнала. Выбирается из списка: **real** - действительный сигнал; **complex-** комплексный сигнал; **auto-** автоматическое определение типа сигнала. **Collapse 2-Dresultsto 1-D**-преобразование двумерного выходного сигнала в одномерный.

Display – цифровой дисплей. *Назначение:* отображает значение сигнала в виде числа. *Параметры блока:* **Format** – формат отображения данных, может принимать следующие значения: **short –** 5 цифр, включая десятичную точку; **long-** 15 цифр с фиксированной точкой; **short _e-** 5цифр с плавающей точкой; **long_e-**15цифр с плавающей точкой; **bank**– банковский формат. **Decimation** – прореживание. **Sample Time** – такт дискретности. **Floating Display**– изменяющийся режим.



Data Type Conversion – преобразователь типа сигнала. *Назначение:* преобразует тип входного сигнала. *Параметры блока:* **Data Type -** тип данных выходного сигнала. **Saturate On Integer Overflow -** подавлять переполнение целого.

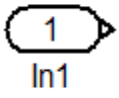


Subsystem – создание подсистем. *Назначение:* Подсистема - это фрагмент **Simulink** - модели, оформленный в виде отдельного блока. Использование подсистем при составлении модели преследует следующие цели:

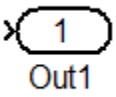
- Уменьшает количество одновременно отображаемых на экране блоков, что облегчает восприятие модели
- Позволяет создавать и отлаживать отдельные фрагменты модели, что повышает технологичность создания модели
- Позволяет создавать собственные библиотеки

- Позволяет синхронизировать параллельно работающие подсистема
- Позволяет включить в модель собственные справочные средства
- Позволяет связывать подсистему с М-файлом, обеспечивая запуск этого файла при открытии подсистемы

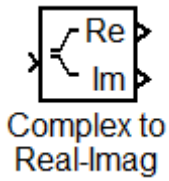
Параметры блока: **Show Portlabels** – показать метки портов. **Treat As Atomic Unit** – считать подсистему неделимой. **Read/Write Permissions** – разрешить чтение и запись. Допустимы три опции: **Read Write -** чтение и запись; **Read Only** - только чтение; **No Read Or Write** чтение записи. **Name of error call back function** – имя функции ответного вызова.



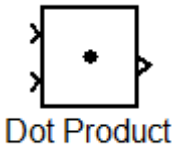
Import – входной порт. *Назначение:* Создает входной порт для подсистемы или выполняет считывание сигнала с рабочей области MATLAB в модель. *Параметры блока:* **Portnumber -** номер порта. **Portdimensions -** размерность входного сигнала. **Sampletime** – такт дискретности. **Show Additional Parameters -** показать дополнительные параметры. **Data Type** – выбор типа выходных данных.**Output Data Type-** тип выходных данных. **Output Scaling Mode –** способ масштабирования выходного сигнала. **Output Scaling Value** – величина масштаба. **Sampling mode -** режим.



Outport – выходной порт. *Назначение:* Создает входной порт для подсистемы или для модели верхнего уровня иерархии. *Параметры блока :***Portnumber -** номер порта. **Output When Disabled -** вид сигнала на выходе подсистемы, в случае если подсистема выключена. Используется для подсистем, управляемых внешним сигналом. Может принимать следующие значения: **held -** выходной сигнал подсистемы равен последнему рассчитанному значению; **reset-** выходной сигнал подсистемы равен значению, задаваемому параметром initialoutput. **Initial Output -** начальное значение.



Complex to Real –**Imag –** блок вычисления действительной и (или) мнимой части комплексного числа. *Назначение:* вычисляет действительную и (или) мнимую часть комплексного числа. *Параметры блока:* **Output -** выходной сигнал. Тип сигнала выбирается из списка: **Real** - действительная часть; **Image** – мнимая часть; **Real&Imag –** действительная и мнимая часть.



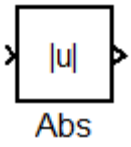
Dot Product – блок скалярного произведения. *Назначение:* Выполняет вычисление скалярного произведения двух векторов. *Параметры блока:* нет. Блок выполняет вычисление выходного сигнала в соответствии с выражением:

$$y = sum\ (conj\ (u1)).*u2),$$

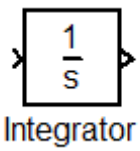
где $u1$ и $u2$ – выходные векторы;

conj – операция вычисления комплексно-сопряженного числа;

sum – операция вычисления суммы.



Abs – блок вычисления модуля. *Назначение:* Выполняет вычисление абсолютного значения величины сигнала. *Параметры блока:* **Saturate On Integer Overflow -** подавлять переполнение целого. **Enable Zero Crossing Detection -** фиксировать прохождение сигнала через нулевой уровень.



Integrator – интегратор. *Назначение:* Интегрирование входного сигнала. *Параметры блока:* **External Reset -** внешний сброс.

Выбирается из списка: **none** - нет (сброс не выполняется); **rising** – нарастающий сигнал (передний фронт сигнала); **falling –** спадающий сигнал (задний фронт сигнала);**either**- нарастающий либо спадающий сигнал; **level** - ненулевой сигнал (сброс выполняется, если сигнал на управляющем входе становится не равным нулю). **Initial Condition Source –** источник начального значения выходного сигнала.

Выбирается из списка: **internal** - внутренний; **external**-внешний. **Initial Condition –** начальное условие. **Limit output –**ограничение выходного сигнала. **Upper Saturation Limit**-верхний предел выходного сигнала. **Lower Saturation Limit –** нижний предел выходного сигнала. **Show Saturation Port** -показать на пиктограмме порт насыщения. Выходной сигнал данного порта может принимать следующие значения: нуль, если интегратор не находится на ограничении; +1, если выходной сигнал интегратора достиг верхнего предела; -1, если выходной сигнал интегратора достиг нижнего предела. **Show State Port** - отобразить/скрыть порт состояния блока. **Absolute Tolerance –** абсолютная погрешность. **Enable Zero Crossing Detection** - определять прохождение сигнала через нулевой уровень.

Terminator – концевой приемник. *Назначение:* Блок применяется как заглушка для сигнала, поступающего с выхода другого блока. В том случае, когда выход блока оказывается не подключенным к входу другого блока **Simulink** выдает предупреждение в командном окне системы MATLAB. Для исключения таких ситуаций следует использовать блок **Terminator**. *Параметры блока:* Нет

Clock – источник времени. *Назначение:* Формирует сигнал, величина которого на каждом шаге равна текущему времени моделирования. *Параметры блока:* **Displaytime –** отображение значения времени на пиктограмме блока. **Decimation** – прореживание.

5. Экспериментальное задание

Прежде всего, отметим, что моделирование в среде **Simulink** ведется во временной области с использованием относительных масштабов по времени и частоте. Длительность битов принята равной $\boldsymbol{\tau}$=**1**, длина исследуемой импульсной последовательности в зависимости от ситуации составляет $\boldsymbol{L}$=**1000** $\boldsymbol{bit}$. Несущая частота выбрана порядка **15**·$\boldsymbol{\pi}$. В процессе исследования отслеживалось соотношение сигнал/шум (**SNR**).

Задание 1:

1. Собрать функциональную модель для исследования цифровой системы реализованных на основе **BPSK-**модема без кодера, в соответствии с рисунком 3.3. Найти значение **SNR** при числе ошибок $n_{err} = 1$ и 10^3 шагов.

2. Собрать функциональную модель для исследования цифровой системы реализованную на основе **BPSK-**модема с кодером в соответствии с рисунком 3.5. Найти значение **SNR** при числе ошибок $n_{err} = 1$ и 10^3 шагов.

3. Собрать функциональную модель для исследования цифровой системы реализованную на основе **8-PSK-**модема без кодера, в соответствии с рисунком 3.12. Найти значение **SNR** при числе ошибок $n_{err} = 1$ и 10^3 шагов.

4. Собрать функциональную модель для исследования цифровой системы реализованную на основе **8-PSK-**модема с кодером, в соответствии с рисунком 3.13. Найти значение **SNR** при числе ошибок $n_{err} = 1$ и 10^3 шагов.

5. Сопоставить полученные значения **SNR** и скорости передач модемов с точки зрения удовлетворения компромиссных требований. Написать отчет.

6. Контрольные вопросы

1. Назовите основные виды фазовой манипуляции?
2. В чем заключается смысл совместного применения модуляции и кодирования?
3. Структура **8-PSK** модулятора?
4. Принцип реализации **8-PSK** демодулятора?
5. Основная идея помехоустойчивого кодирования?
6. Принцип блочного алгебраического кодирования?
7. Принцип синдромного декодирования?
8. Нарисовать упрощенную структурную схему **BPSK**-модема и объяснить принцип её работы?
9. Нарисовать упрощенную структурную схему **8-PSK**-модема и объяснить принцип её работы?
10. Какая модуляция является более помехоустойчивой **BPSK** или **QPSK**?
11. Какая модуляция **BPSK** или **8-PSK** обеспечивает большую скорость передачи?
12. Какая модуляция **BPSK** или **8-PSK** является более узкополосной?

Список использованных источников

1. Основы цифровых технологий. Часть II. Методы модуляции. Помехоустойчивость. Авторы: Песков С.Н., зам. директора по науке компании “Контур-М”, к.т.н; Барг А.И., руководитель управления кабельного ТВ компании “Контур-М”; Балков М.В., зам. начальника проектного отдела компании “Контур-М”. http://www.konturm.ru/download/stat/2005/290805.pdf

2. Обработка сигналов и изображений. А.Б. Сергиенко. Communications Toolbox - обзор. Функции модуляции/демодуляции. http://matlab.exponenta.ru/communication/index.php

3. Манипуляция https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F

4. Модуляция и кодирование. ВГУ, ФКН, ИС, Коваль А.С., http://www.cs.vsu.ru/~kas/doc/infonets/infonets04.pdf

5. Гультяев, А.К. **MatLab 5.3**. Имитационное моделирование в среде Windows: Практическое пособие / А.К. Гультяев. – СПб.: КОРОНА принт, 2001.- 400 с.

6. Черных, И.В. **Simulink**: среда создания инженерных приложений. / Под общ.ред. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2003.- 496 с.

7. Дьяконов, В.П. **MatLab 6.5 SP1/7 + Simulink 5/6** в математике и моделировании. Сер. Библиотека профессионала / В.П. Дьяконов. - М.: СОЛОН-Пресс, 2005.- 576 с.