

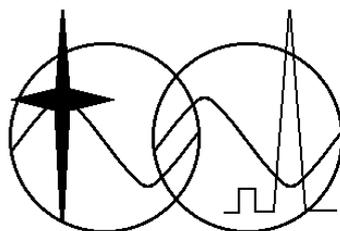
**Министерство образования и науки Российской Федерации**

**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 МОДЕМА ВОС-МОДУЛЯЦИИ**

**Описание лабораторной работы для студентов специальности  
“Системы мобильной связи”  
по дисциплине  
«Теоретические основы систем мобильной связи»**

Разработчики:  
доц. каф. СРС  
\_\_\_\_\_ Кологривов В.А.  
\_\_\_\_\_ Сумекенова А.К.

**2015**

**СОДЕРЖАНИЕ**

<b>1. Цель работы. Краткие сведения из теории</b>	<b>3</b>
<b>2. Краткое описание пакета Simulink</b>	<b>7</b>
<b>2.1. Общая характеристика пакета Simulink</b>	<b>7</b>
<b>2.2. Запуск и работа с пакетом Simulink</b>	<b>7</b>
<b>3. Описание лабораторного макета</b>	<b>9</b>
<b>4. Описание используемых блоков библиотеки Simulink</b>	<b>17</b>
<b>5. Экспериментальное задание</b>	<b>25</b>
<b>6. Контрольные вопросы</b>	<b>25</b>
<b>Список использованных источников</b>	<b>26</b>

## 1. ЦЕЛЬ РАБОТЫ. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

**Цель работы:** Изучить перспективные способы модуляции современных спутниковых навигационных систем, простейшие модели телекоммуникационных устройств и влияния меандровой модуляции на спектральную картину спутниковых радионавигационных сигналов (СРНС).

### Теоретическая часть

Вариант классической реализации функциональной модели модема на основе квадратурной модуляции с использованием ВОС-модуляции и демодуляции фазы несущей частоты приведен на рисунке 1.1.

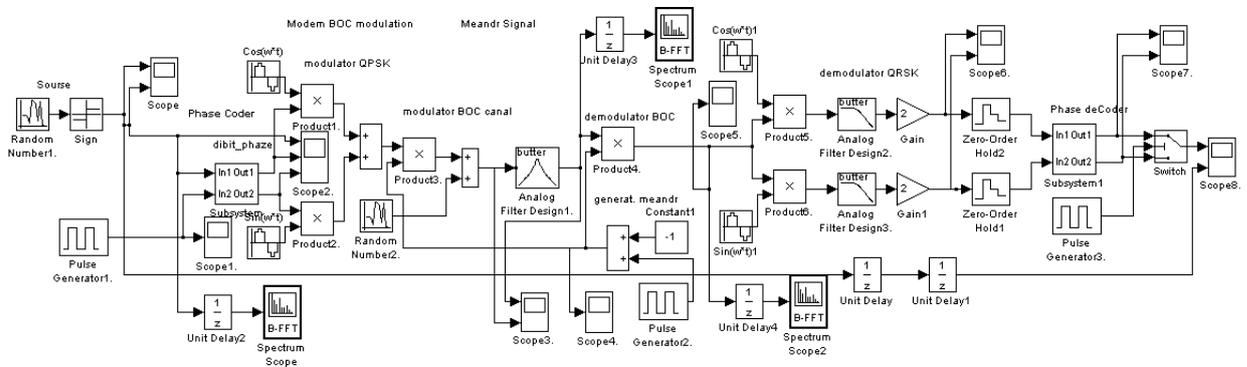


Рисунок 1.1 – Функциональная модель модема на основе квадратурной модуляции с использованием ВОС-модуляции

На практике чаще используют простейший вид фазовой модуляции **BPSK** (Binary phase shift keying), как наиболее помехоустойчивую. В нашем случае использована квадратурная фазовая манипуляция **QPSK** (Quadrature phase shift keying) с фазовым кодером.

Введение дополнительной меандровой или **ВОС**-модуляции с целью расщепления спектра радиосигнала реализуется обычно перемножением биполярной псевдослучайной последовательности (**ПСП**) дальномерного кода и биполярной меандровой последовательности (**БМП**) с периодом кратно укладывающемся в элемент **ПСП** (специфика **ВОС** – сигналов).

Модуляция **ПСП** дальномерного кода биполярной меандровой последовательностью (**ВОС** – модуляция)  $d(t)$  может быть записана в виде:

$$d(t-t_0) = g(t-t_0) \cdot r(t-t_0), \quad (1)$$

где  $g(t)$  – собственно ПСП дальномерного кода;  $r(t)$  – меандровая последовательность, отражающая специфику нового класса СРНС.

В нашем случае из соображений простоты и универсальности ВОС-модулятор реализован умножением фазоманипулированного колебания несущей частоты на биполярную меандровую последовательность. ВОС-демодуляция радионавигационного сигнала на приемной стороне реализуется повторным умножением принимаемого модулированного дальномерного сигнала на аналогичную биполярную меандровую последовательность.

Фазовая квадратурная модуляция и демодуляция с использованием фазового кодера предполагает программно-аппаратурную реализацию. Входная последовательность сигнала (дальномерного кода) разбивается на дибиты. Блок **MatLab Fcn** назначает дибитам фазовые состояния  $\varphi_k$ . Далее блоки **cos** и **sin** тригонометрических функций вычисляют  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ , которые в качестве модулирующих импульсов поступают на вторые входы умножителей квадратурного модулятора. На первые входы умножителей поступают опорные колебания несущей частоты  $\cos(\omega \cdot t)$  и  $\sin(\omega \cdot t)$ . Суммирование сигналов с выходов перемножителей завершает квадратурную фазовую модуляцию

$$\cos(\omega \cdot t) \cdot \cos(\varphi_k) + \sin(\omega \cdot t) \cdot \sin(\varphi_k) = \cos(\omega \cdot t - \varphi_k).$$

Демодуляция на приемной стороне выполняется путем разветвления и подачи принятого модулированного сигнала на вторые входы умножителей квадратурного демодулятора. На первые входы умножителей подаются опорные колебания несущей частоты  $\cos(\omega \cdot t)$  и  $\sin(\omega \cdot t)$  синхронизированные с колебаниями опорных генераторов передающей стороны. В результате на выходах умножителей получаем квадратурные составляющие

$$\cos(\omega \cdot t - \varphi_k) \cdot \cos(\omega \cdot t) = \frac{1}{2} \cdot \langle \cos(\varphi_k) + \cos(2 \cdot \omega \cdot t - \varphi_k) \rangle;$$

$$\cos(\omega \cdot t - \varphi_k) \cdot \sin(\omega \cdot t) = \frac{1}{2} \cdot \langle \sin(\varphi_k) + \sin(2 \cdot \omega \cdot t - \varphi_k) \rangle.$$

Пропуская эти составляющие через **ФНЧ** и усиливая вдвое, получаем восстановленные квадратурные модулирующие импульсы пропорциональные  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ . Далее еще один блок **MatLab Fcn** квадратурным проекциям  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$  ставит в соответствие дибиты  $(d_k, d_{k+1})$ .

Рассмотрим пример для определения частот, на которые расщепляется спектр при **ВОС**-модуляции.

### Пример 1.

Пусть имеем несущую частоту  $\omega = 20 \cdot \pi$  и период меандрового колебания  $T_m = 1/2$ , тогда частота  $f = 1/T_m = 2$ , а круговая частота  $\omega = 4 \cdot \pi$ . Отсюда находим две несущие частоты расщепленного спектра:

$$\omega_1 = 20 \cdot \pi - 4 \cdot \pi = 16 \cdot \pi,$$

$$\omega_2 = 20 \cdot \pi + 4 \cdot \pi = 24 \cdot \pi.$$

Для фильтрации необходимой полосы сигнала в канале используем полосовой фильтр (блок **Analog Filter Design 1**). Для настройки фильтра рассмотрим пример расчета верхней и нижней граничных частот.

### Пример 2.

Несущая частота  $\omega = 20 \cdot \pi$ , период меандрового колебания  $T_m = 1/2$ , длительность бита меандра  $\tau = 1$ .

Граничные частоты фильтра для приема по двум составляющим, несущих частот равных  $16 \cdot \pi$  и  $24 \cdot \pi$  легко определить исходя из спектра сигнала **ВОС**-модуляции, который приведен на рисунке 1.2.

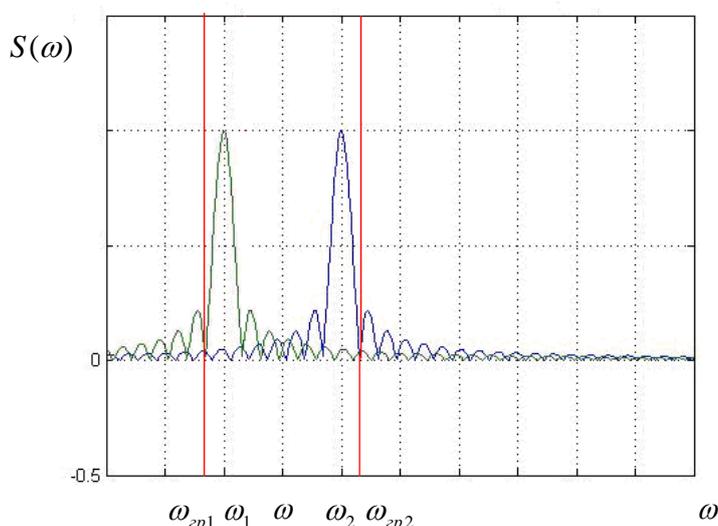


Рисунок 1.2 – Спектр сигнала **ВОС**-модуляции

$$\omega_{cp1} = \omega_1 - \frac{2 \cdot \pi}{\tau} = 16 \cdot \pi - 2 \cdot \pi = 14 \cdot \pi,$$

тогда используя формулу относительной расстройки при средней частоте диапазона  $\omega = 20 \cdot \pi$ , получим

$$\frac{\omega_{\omega 2}}{\omega} = \frac{\omega}{\omega_{\omega 1}}; \quad \omega_{\omega 2} = \frac{\omega^2}{\omega'_{\omega 1}} = \frac{(20 \cdot \pi)^2}{14 \cdot \pi} \approx 28,57 \cdot \pi,$$

где  $\frac{2 \cdot \pi}{\tau}$  - полоса частот главного лепестка прямоугольного сигнала. Спектр прямоугольного сигнала изображен на рисунке 1.3.

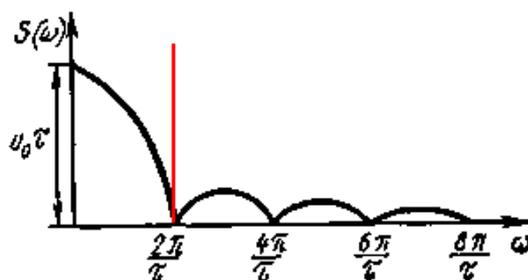


Рисунок 1.3 – Спектр прямоугольного модулирующего сигнала

В системах спутниковой навигации возникает проблема занятости диапазона частот, поэтому рассмотрим случаи, когда можно принимать по одной боковой составляющей спектральной плотности. С этой целью осуществляем перестройку канального фильтра (блок **Analog Filter Design 1**). На **примере 3** рассчитаем границы фильтра для приема по левой составляющей, несущей частоты равной  $16 \cdot \pi$ .

### Пример 3.

Несущая частота сигнала  $\omega = 20 \cdot \pi$ , период меандрового колебания  $T_m = 1/2$ , длительность бита меандра  $\tau = 1$ . Полоса пропускания фильтра, обеспечивающая прохождение главного лепестка спектральной плотности меандра равна  $\Delta\omega = 2 \cdot \pi$ . Расчет границ по левой составляющей расщепленного спектра с несущей  $\omega_1 = 16 \cdot \pi$ .

Нижнюю граничную частоту  $\omega_{\omega 1}$  найдем по формуле:

$$\omega'_{гр1} = \omega_1 - \Delta\omega = 16 \cdot \pi - 2 \cdot \pi = 14 \cdot \pi.$$

Верхнюю граничную частоту  $\omega''_{гр1}$  найдем из формулы относительной расстройки:

$$\frac{\omega_1}{\omega'_{гр1}} = \frac{\omega''_{гр1}}{\omega_1}; \quad \omega''_{гр1} = \frac{\omega_1^2}{\omega'_{гр1}} = \frac{(16 \cdot \pi)^2}{14 \cdot \pi} = 18,29 \cdot \pi.$$

## 2. КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK

### 2.1 Общая характеристика пакета Simulink

Пакет **Simulink** разрабатывается компанией **Mathworks** ([www.mathworks.com](http://www.mathworks.com)) и распространяется в составе математического пакета **MatLab**. Пакет основан на графическом интерфейсе и является типичным средством визуально-ориентированного программирования. Он обладает обширной библиотекой готовых блоков с модифицируемыми параметрами для построения моделей рассматриваемых систем и наглядными средствами визуализации результатов моделирования [2, 3, 4].

### 2.2 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**. Это можно сделать одним из трех способов:

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

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

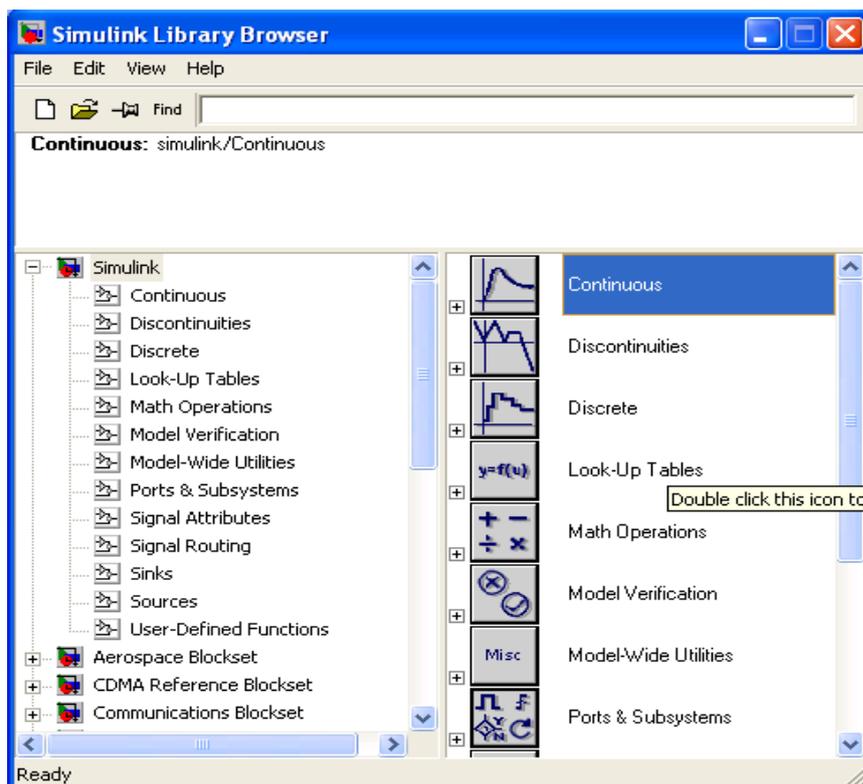


Рисунок 2.1. – Библиотека блоков **Simulink Library Browser**

На рисунке 2.1 выведена библиотека системы **Simulink** (в левой части окна) и показаны ее разделы (в правой части окна). Основная библиотека системы содержит следующие разделы:

- **Continuous** – блоки аналоговых элементов;
- **Discontinuous** – блоки нелинейных элементов;
- **Discrete** – блоки дискретных элементов;
- **Look-Up Tables** – блоки таблиц;
- **Math Operations** – блоки элементов, определяющие математические операции;
- **Model Verification** – блоки проверки свойств сигнала;
- **Model-Wide Utilities** – раздел дополнительных утилит;
- **Port&Subsystems** – порты и подсистемы;
- **Signal Attributes** – блоки задания свойств сигналов;
- **Signal Routing** – блоки маршрутизации сигналов;
- **Sinks** – блоки приема и отображения сигналов;
- **Sources** – блоки источников сигнала;
- **User-Defined Function** – функции, определяемые пользователем.

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 приведёт к выходу из **Simulink**.

### 3. ОПИСАНИЕ ЛАБОРАТОРНОГО МАКЕТА

Приведем краткое описание работы модема на основе квадратурной модуляции с использованием **ВОС**-модуляции по функциональной модели представленной на рисунке 3.1.

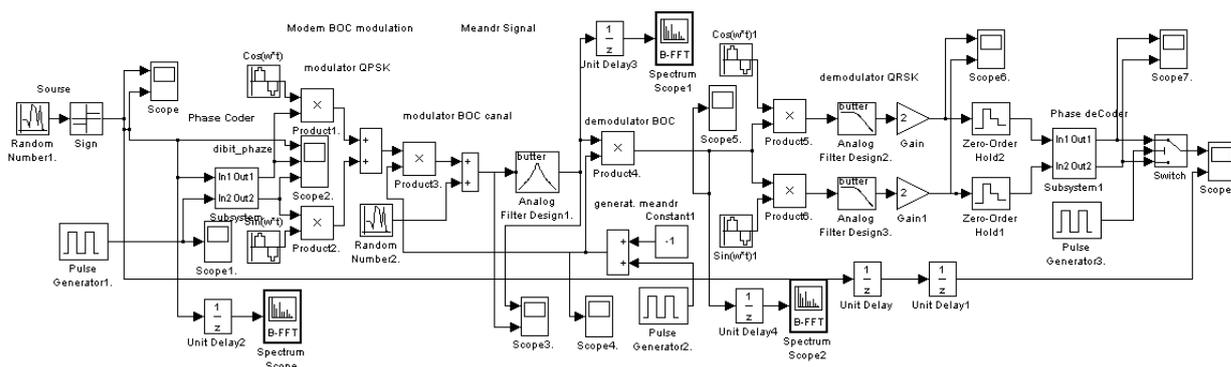


Рисунок 3.1 – Функциональная модель модема на основе квадратурной модуляции с использованием **ВОС**-модуляции

Входной поток данных реализуется блоком источника случайного сигнала с равномерным распределением (**Uniform Random Number**), на выходе которого включен блок определения знака сигнала (**Sign**). Изменяя

параметры блока источника **Initial seed** и **Sample time**, задаем вид случайной последовательности и длительности импульсов.

С выхода схемы формирования входной импульсной последовательности сигнал поступает на фазовый кодер. Он может быть представлен в виде функциональной схемы (см. рисунок 3.2).

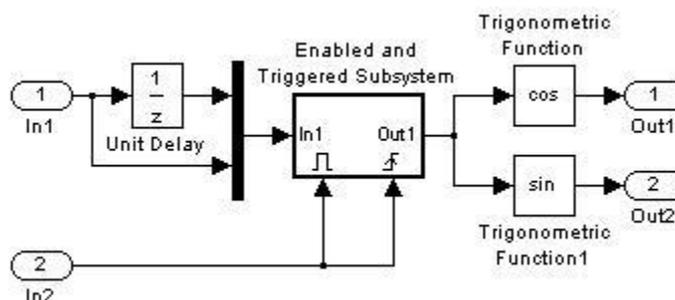


Рисунок 3.2 – Функциональная схема составного блока фазового кодера

Здесь на вход **In1** поступает входная импульсная последовательность, а на вход **In2** прямоугольные импульсы с опорного тактового генератора (**Pulse Generator**) с периодом равным дибиту и скважностью **2**.

Преобразования битовой последовательности импульсов в векторное представление дибитов, реализовано путем подачи текущего и задержанного на один такт предыдущего импульсов на блок мультиплексора (**Mux**).

В подсистеме **Enabled and Triggered Subsystem** преобразование из дибита в фазу производится функцией **dibit\_cod** включенного в подсистему блока **Matlab Fcn** по коду Грея в виде:  $(-1,-1) \rightarrow \pi/4$ ,  $(-1,1) \rightarrow 3\pi/4$ ,  $(1,1) \rightarrow -3\pi/4$ ,  $(1,-1) \rightarrow -\pi/4$ .

С выхода подсистемы полученные фазы через блоки (**Trigonometric Function**) в виде импульсов  $\cos(\varphi)$  и  $\sin(\varphi)$  поступают на выход фазового кодера.

Далее этап модуляции реализован в виде квадратурной фазовой манипуляции. Текущие значения квадратурных составляющих несущей частоты  $\cos(\omega t)$  и  $\sin(\omega t)$  умножаются на значения, полученные в фазовом кодере  $\cos(\varphi)$  и  $\sin(\varphi)$ , соответственно.

Заключительная часть модуляции содержит блок умножения (**Product3**), определяющая специфику ВОС-сигналов. ВОС-модулированный сигнал образуется путем перемножения собственно QPSK модулированного сигнала и меандрового колебания.

Меандровое колебание реализовано в виде суммы сигналов с генератора прямоугольных импульсов (**Pulse Generator**) с амплитудой равной **2** и источника постоянного уровня (**Constant**) с амплитудой равной **-1**. Блок умножения (**Product3**) завершает передающую часть функциональной схемы модема. С блока умножения сигнал поступает на простейший имитатор канальных помех выполненного в виде блока сумматора (**Sum**). На второй вход которого подается случайная псевдослучайная импульсная последовательность (блок **Random Number 2**), которая используется в качестве имитатора шумов канала передачи.

Сигнал с ВОС-модулятора вместе с шумами поступает на вход полосового канального фильтра (блок **Analog Filter Design**) предназначенного для отфильтрации сопутствующих помех канала и сигналов соседних каналов.

Демодулятор реализован двумя умножителями (**Product3** и **Product4**), на первые входы которых подается принятый сигнал, а на вторые входы умножителей поступают сигналы с опорных генераторов несущей частоты, изменяющиеся по законам  $\cos(\omega t)$  и  $\sin(\omega t)$  и синхронизированных с генераторами передающей части. С выходов умножителей сигналы поступают на ФНЧ (блоки **Analog Filter Design** и **Analog Filter Design1**), которые отфильтровывают высокочастотные составляющие сигналов умножителей, и далее усиливаются блоками **Gain** и **Gain1**. Низкочастотные составляющие спектра сигналов умножителей, пропорциональные принятым модулирующим квадратурным импульсам подаются для предварительного сглаживания на необязательные блоки экстраполяторов нулевого порядка (**Zero-Order Hold** и **Zero-Order Hold1**). Далее сглаженные модулирующие импульсы поступают на схему принятий решений.

Схема принятия решений реализована на основе двухканального блока подсистемы **Enabled Subsystem** управляемого опорным генератором прямоугольных импульсов (**Pulse Generator1**). Блоками принятия решений каждого канала являются встроенные блоки **Sign1** и **Sign2**, реализующие нулевой порог срабатывания. С выхода схемы принятия решений,

регенерированные импульсы (дибиты) поступают на вход схемы восстановления исходной последовательности импульсов (битов).

Схема восстановления исходной последовательности импульсов (битов) по принятым квадратурным модулирующим импульсам (дибитам) реализована на основе двухпортового переключателя (**Multiport Switch**) и управляемого опорным генератором прямоугольных импульсов (**Pulse Generator2**) с периодом равным дибиту и скважностью 2. Поочередная выборка отсчетов из квадратурных двухбитовых последовательностей восстанавливает принятую последовательность битов цифрового сигнала.

Для визуализации работы модема на основе квадратурной модуляции с фазовым кодером и с использованием ВОС-модуляции приведены осциллограммы блоков осциллографов (**Scope**), подключенных к узловым точкам функциональной модели, и, отображающие основные этапы преобразования сигнала в модуляторе и демодуляторе.

На рисунке 3.3 приведены осциллограммы входной импульсной последовательности при  $\tau = 1$ , сформированной источником случайного сигнала с равномерным распределением и блоком определения знака сигнала, а также модулирующие квадратурные последовательности после фазового кодера.

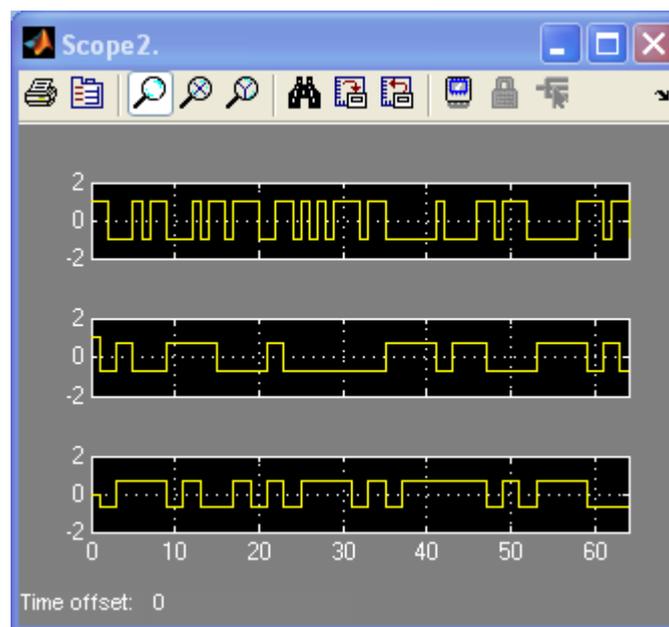


Рисунок 3.3 – Осциллограммы входной импульсной последовательности, модулирующие квадратурные последовательности после фазового кодера

На рисунке 3.4. приведена осциллограмма принятого фазоманипулированного сигнала на несущей частоте и этого же сигнала после прохождения полосового фильтра.

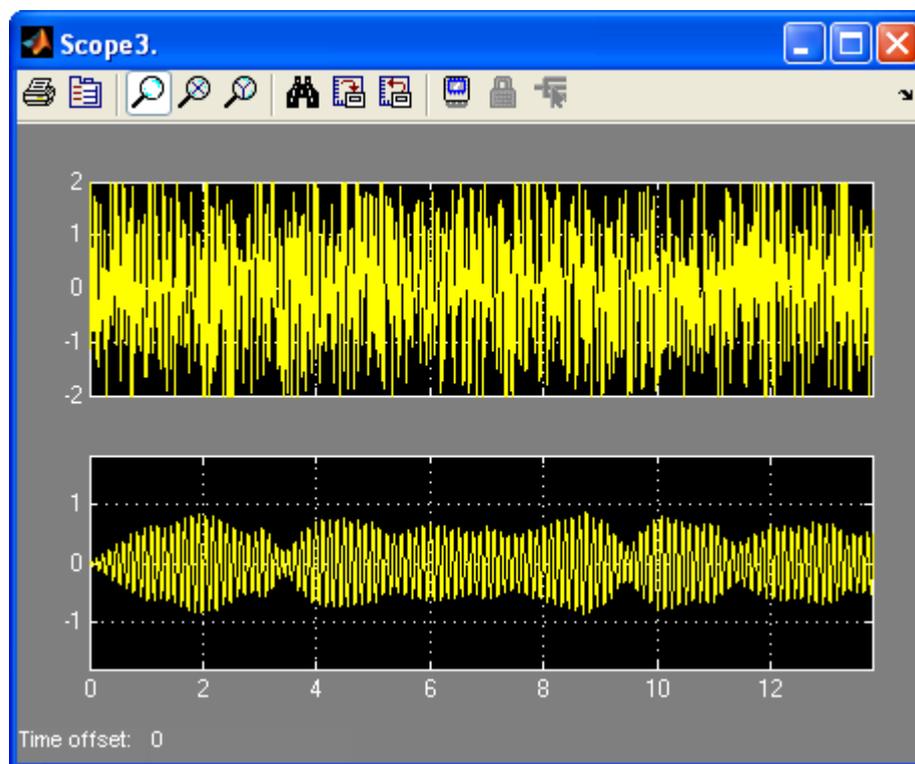


Рисунок 3.4 – Осциллограммы принятого фазоманипулированного сигнала на несущей частоте до и после прохождения ПФ

На рисунке 3.5 приведена осциллограмма меандровой последовательности прямоугольных импульсов длительностью равной  $(1/2) \cdot \tau$  и скважностью 2, формируемая опорным тактовым генератором с подставкой **Constant**.

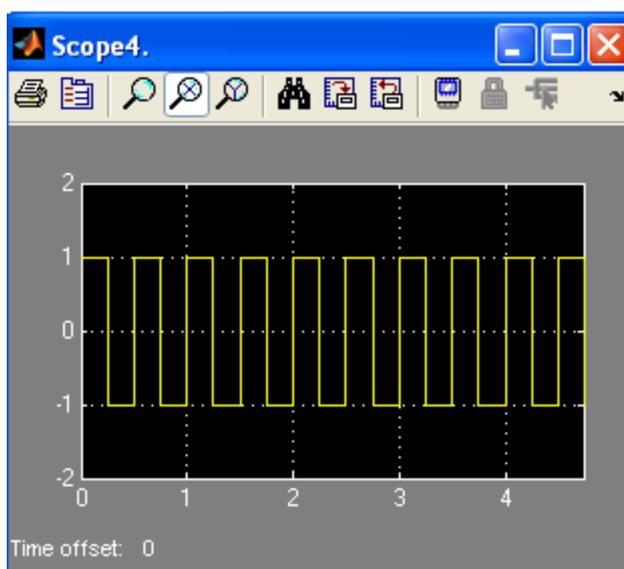


Рисунок 3.5 – Осциллограмма меандровой последовательность

На рисунке 3.6 приведены осциллограммы сигналов на выходах **ФНЧ** демодулятора, соответствующих принятым модулирующим импульсам квадратурных каналов модулятора.

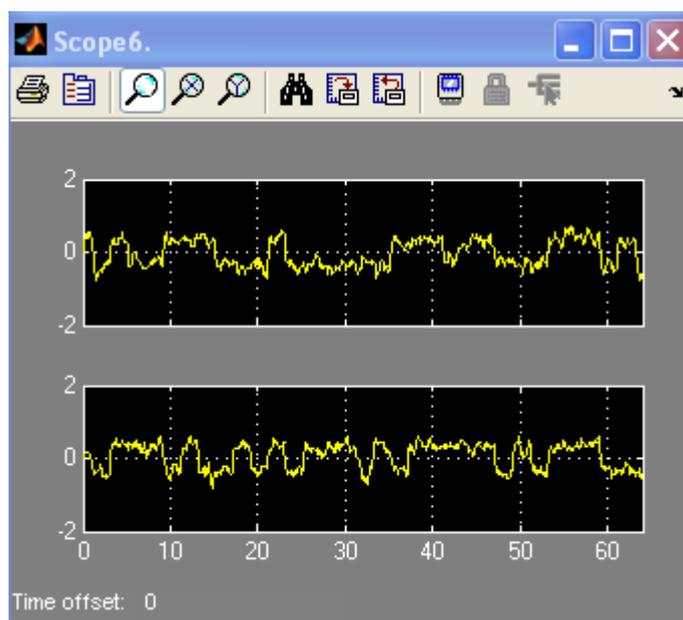


Рисунок 3.6 – Осциллограммы сигналов на выходах **ФНЧ** демодулятора, соответствующих принятым модулирующим импульсам квадратурных каналов модулятора

На рисунке 3.7 приведены осциллограммы принятых и восстановленных схемой принятия решений модулирующих импульсов квадратурных каналов.

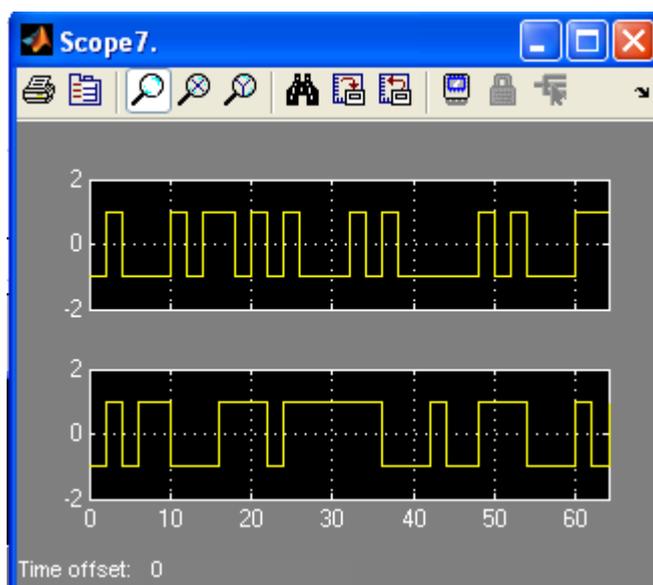


Рисунок 3.7 – Осциллограммы принятых и восстановленных схемой принятия решений модулирующих импульсов квадратурных каналов

На рисунке 3.8 приведены осциллограммы переданного и принятого сигнала на выходе схемы восстановления исходной последовательности импульсов по квадратурным последовательностям схемы принятия решений демодулятора.

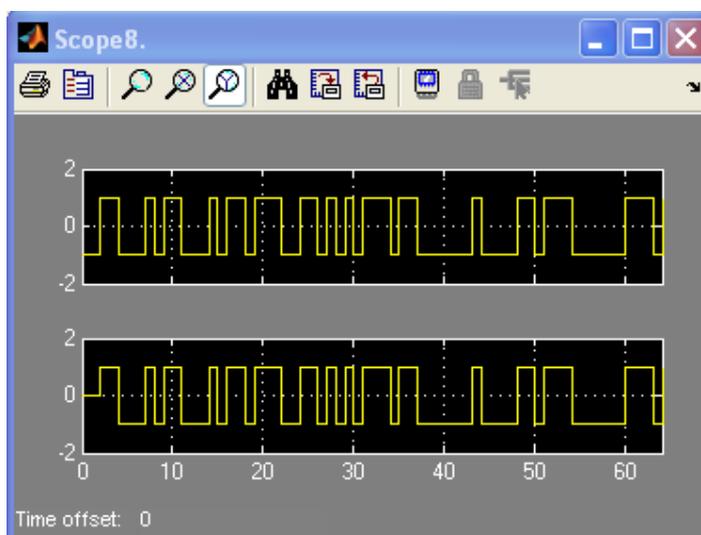


Рисунок 3.8 – Осциллограммы переданного и принятого сигнала на выходе схемы регенерации формы импульсов

На рисунках 3.9 - 3.10 приведены спектральные плотности мощности меандровой псевдослучайной последовательности, принятой по двум и одной боковой составляющей, соответственно, и помех тракта передачи, отображаемая блоком **Spectrum Scope**.

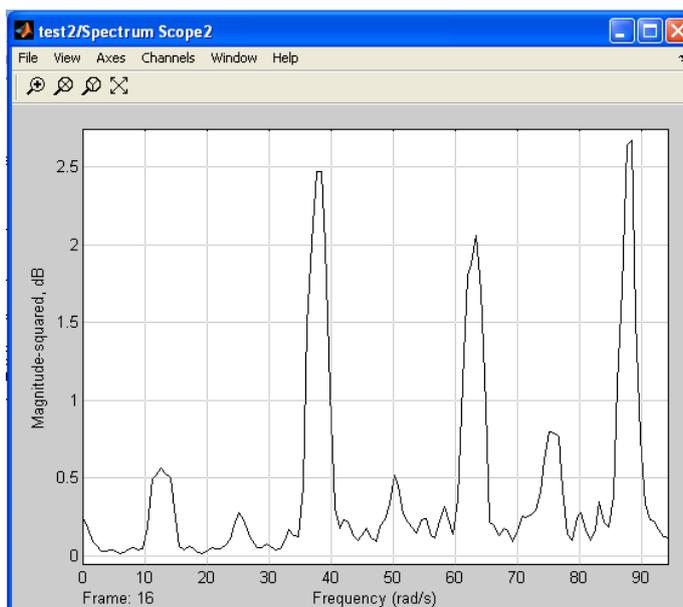


Рисунок 3.9 – Спектральная плотность мощности МПСЦ, принятой по двум боковым составляющим

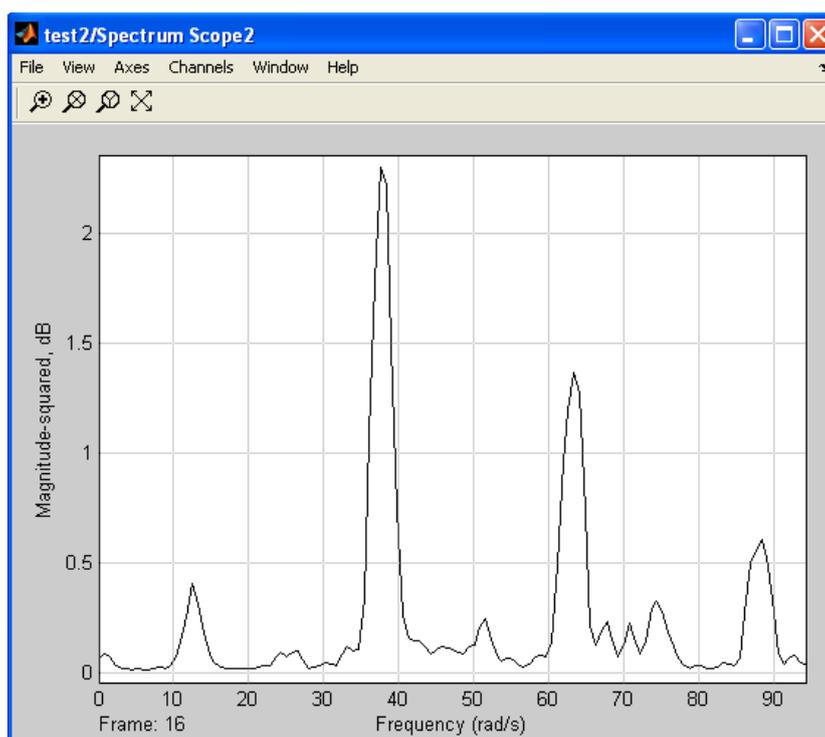


Рисунок 3.10 – Спектральная плотность мощности МПСЦ, принятая по одной боковой составляющей

Рисунки, где приведены энергетические спектры, демонстрирует их свойства: **ВОС**-модулированные сигналы позволяют полнее использовать выделенный спектр частот и допускают сосуществование различных сигналов на одной несущей частоте.

## 4. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ БЛОКОВ БИБЛИОТЕКИ SIMULINK

Ниже описаны основные блоки базовых разделов библиотеки **Simulink** [2, 3, 4], используемые в функциональной схеме классического варианта модема квадратурной фазовой манипуляции, с естественным принципом модуляции и демодуляции фазовых состояний несущей частоты.



**Uniform Random Number** – источник случайного дискретного сигнала с нормальным распределением. *Назначение:* формирование случайного сигнала с равномерным распределением уровня. *Параметры блока:* **Minimum** – минимальный уровень сигнала; **Maximum** – максимальный уровень сигнала; **Initial seed** – 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

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

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

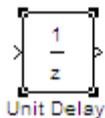


**Switch** – блок переключателя. *Назначение:* переключение входных сигналов по сигналу управления. *Параметры блока:* **Criteria for passing first input** – условие прохождения сигнала с первого входа, значение выбирается из списка: **u2>=Threshold** – сигнал управления больше или равен пороговому значению; **u2>Threshold** – сигнал управления больше порогового значения; **u2~=Threshold** – сигнал управления не равен пороговому значению. **Threshold** – порог; флажок **Show additional parameters** – показать дополнительные параметры. При выставленном

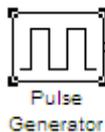
флажке отображаются дополнительные окна списков, в нашем случае флажок не используется.



**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.



**Unit delay** – блок 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).



**Pulse generator** – блок источника импульсного сигнала. *Назначение:* формирование сигнала в форме прямоугольных импульсов. *Параметры блока:* **Pulse Type** – способ формирования сигнала, может принимать два значения: **Time-based** – по текущему времени; **Sample-based** – по величине такта дискретности и количеству шагов моделирования. Вид окна параметров зависит от выбранного способа формирования сигнала. **Amplitude** – амплитуда; **Period** – период, задается в секундах при способе **Time-based** или количеством тактов при способе **Sample-based**; **Pulse width** – ширина импульса, задается в процентах от периода при способе **Time-based** или количеством тактов при способе **Sample-based**; **Phase delay** – фазовая задержка, задается в секундах при способе **Time-based** или количеством тактов при способе **Sample-based**; **Sample time** – такт

дискретности; флажок **Interpret vector parameters as 1 - D** – интерпретировать вектор как массив скаляров.



**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 (рад/с); **Phase (rads)** – начальная фаза (рад); **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

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

где  $\Delta t$  - постоянная величина, равная значению **Sample time**. В данном режиме ошибка округления для больших значений времени меньше, чем в предыдущем режиме.

Выходной сигнал источника в режиме **Time-based (по количеству тактов на период для дискретных систем)** определяется выражением

$$y = A \cdot \sin(2 \cdot \pi \cdot k \cdot f \cdot T + \varphi) + b = A \cdot \sin(2 \cdot \pi \cdot k + l) / N + b ,$$

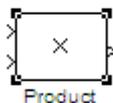
где  $A$  - амплитуда сигнала;  $f$  - частота сигнала в Гц;  $T$  - такт дискретности;  $N$  - количество тактов в секунду;  $k$  - номер текущего шага,  $k = 0, \dots, N - 1$ ;  $\varphi$  - начальная фаза сигнала;  $l$  - начальная фаза, заданная количеством тактов;  $b$  - постоянная составляющая (смещение) сигнала.

*Параметры блока в режиме **Sample-based: Amplitude*** – амплитуда; **Bias** – постоянная составляющего сигнала (смещение); **Samples per period** – количество тактов на один период синусоидального сигнала:

$$N = \text{Samples per second} = 1/(f \cdot T);$$

$$p = \text{Samples per period} = 2 \cdot \pi \cdot N .$$

**Number of offset samples** – начальная фаза сигнала, задается количеством тактов дискретности  $l = \varphi \cdot p / (2 \cdot \pi)$ . **Sample time** – такт дискретности. Флажок **Interpret vector parameters 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.



**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:* **Icon shape** – форма блока, выбирается из списка: **round** – круг; **rectangular** – прямоугольник. **List of sign** – список знаков из набора: + - плюс; - - минус, | - делитель. 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.

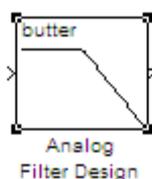


**Random Number** – источник случайного сигнала с нормальным распределением уровня. *Назначение:* формирование сигнала с равномерным распределением уровня. Параметры блока: **Mean** – среднее значение; **Variance** – дисперсия; Initial seed – Начальное значение генератора случайного сигнала; **Sample time** – такт дискретности; флажок **Interpreted vector parameters as 1 – D** – интерпретировать вектор как массив скаляров. В нашем случае блок используется для моделирования шумов канала передачи данных.



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

Блоки **Gain** и **Matrix Gain** по сути есть один и тот же блок, но с разными начальными установками параметра **Multiplication**. Для векторного входного сигнала **u** и параметра **Gain** в виде вектора-строки **K**, при способе выполнения операции **Element-wise K\*u** получаем вектор покомпонентного умножения векторов **K\*u**, а при способе выполнения операции **Matrix K\*u** получаем скалярное произведение векторов **K\*u**. Параметр блока **Gain** может быть положительным или отрицательным, как больше, так и меньше единицы. Коэффициент усиления можно задавать в виде скаляра, матрицы, вектора или вычисляемого выражения. Если входной сигнал действительного типа, а коэффициент усиления комплексный, то выходной сигнал будет комплексным. В случае отличия типа входного сигнала от типа коэффициента усиления **Simulink** пытается выполнить приведение типа усиления к типу входного сигнала и выдает сообщение об ошибке, если такое приведение невозможно.

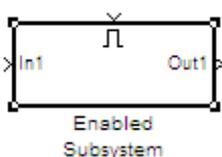


**Analog Filter Design** – блок аналогового фильтра заданного метода проектирования и типа из подраздела **Filter Design**; подраздела **Filtering**, раздела **DSP Blockset** библиотеки **Simulink** [4].  
*Назначение:* аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. *Параметры 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.



**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.



**Enabled Subsystem** – блок **E**-подсистемы. Исходный шаблон блока содержит в своем составе блок **Enable** – блок управления; входной и выходной порты и линию связи между ними. После того как блок подсистемы скопирован из библиотеки в окно модели, он становится доступным для редактирования. Блок может содержать в своем составе другие блоки и подсистемы. Число входных и выходных портов определяется конфигурацией подсистемы и устанавливается пользователем. *Назначение:* **E**-подсистема активизируется при наличии положительного сигнала на управляющем входе. Если входной сигнал векторный, то подсистема активизируется, если хотя бы один элемент принимает положительное значение. Величина выходного сигнала в том случае, если система заблокирована, определяется настройками выходных портов подсистемы (блоки **Output**). Сама **E**-подсистема является неделимой подсистемой. Её свойства определяются параметрами блока **Enabled Subsystem** и совпадают со свойствами блока **Subsystem**. *Параметры блока:* **Show port labels** – показать метки портов; **Treat as atomic unit** – считать подсистему неделимой; **Read/Write Permissions** – разрешить чтение и запись, допустимы 3 опции: **ReadWrite** – чтение и запись; **Read Only** – только чтение; **NoReadOrWrite** – ни чтения ни записи. **Name of error callback function** – имя функции ответного вызова.

Блок **Enable** – управляющий вход. *Параметры блока:* **States when enabling** – состояние при запуске, задает состояние подсистемы при каждом запуске и выбирается из списка: **held** – использовать предыдущее активное

состояние; **reset** – использовать начальное (исходное состояние). **Show output port** – показать выходной порт, при установленном флажке на пиктограмме блока **Enable** появляется дополнительный выходной порт, сигнал с которого может быть использован для управления блоками внутри подсистемы. Флажок **Enable zero crossing detection** – предписывает фиксировать прохождение сигнала через нулевой уровень.

Пользователь может автоматически создать блок **Subsystem**. Для этого необходимо выделить с помощью мыши нужный фрагмент модели и выполнить команду **Create Subsystem** из меню **Edit** в окне модели. Выделенный фрагмент будет помещен в подсистему, а входы и выходы будут снабжены соответствующими портами. В дальнейшем, если это необходимо, можно объявить этот блок неделимым, изменив параметры, или сделать условно управляемым, добавив управляющие входы. Отменить группировку блоков в подсистему можно командой **Undo**.

## 5. ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

1. Собрать **Sim**-модель модема на основе **QPSK** модуляции с использованием **ВОС**-модуляции и демодуляции фазы несущей частоты, в соответствии с рисунком 1.1.

2. Выставить параметры блоков **Sim**-модели, обращая внимание на граничные частоты полосового фильтра.

3. Пронаблюдать и зафиксировать основные осциллограммы, иллюстрирующие работу модема с использованием **ВОС**-модуляции.

4. Получив основные осциллограммы, проанализировать значение двух несущих частот расщепленного спектра.

5. Изменить параметры блока **Analog Filter Design** для приема по одной боковой составляющей и пронаблюдать работу модема.

## 6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое **ВОС**-модуляция и перечислить основные ее преимущества?

2. Объясните каким образом реализован **ВОС**-модулятор.

3. Объясните работу фазовой квадратурной модуляция и демодуляция с использованием фазового кодера.

4. Чем обусловлено использование ортогональных несущих  $\cos(\omega \cdot t)$  и  $\sin(\omega \cdot t)$  в фазовом квадратурном модуляторе и демодуляторе?

5. Объясните принцип работы формирователя квадратурных модулирующих импульсов фазового манипулятора.

6. Объясните принцип работы схемы восстановления исходной последовательности импульсов по принятым квадратурным модулирующим импульсам.

7. Какую функцию в схеме выполняет блок **Enabled Subsystem** со встроенными блоками **Sign** и как можно модифицировать реализацию данной функции?

8. Какую функцию в схеме выполняют блоки **Analog Filter Design**?

9. Изобразите функциональную схему квадратурного фазового модулятора исследуемого **PSK** модема с **ВОС**-модуляцией.

10. Изобразите функциональную схему квадратурного фазового демодулятора исследуемого **PSK** модема с **ВОС**-модуляцией.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение, 2-е изд., исправл.: Пер. с англ. – М.: Изд. дом “Вильямс”, 2003.- 1104 с.
2. Гульятяев А.К. **MatLab 5.3**. Имитационное моделирование в среде **Windows**: Практическое пособие. – СПб.: КОРОНА принт, 2001.- 400 с.
3. Черных И.В. **Simulink**: среда создания инженерных приложений. / Под общ. ред. В.Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2003.- 496 с.
4. Дьяконов В.П. **MatLab 6.5 SP1/7 + Simulink 5/6** в математике и моделировании. Сер. Библиотека профессионала. - М.: СОЛОН-Пресс, 2005.- 576 с.
5. Кологривов В.А. Исследование **QPSK** модема (реализация с фазовым кодером): Описание лабораторной работы для студентов – ТУСУР, 2012. – 35 с.