

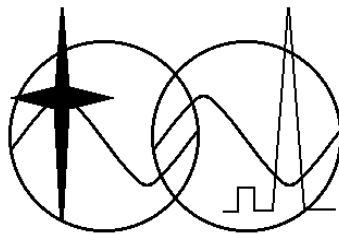
**Министерство образования и науки Российской Федерации**

**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.  
\_\_\_\_\_ 2014 г.



**ИССЛЕДОВАНИЕ ТЕХНОЛОГИИ ММО, ПОСТРОЕННОЙ ПО  
СХЕМЕ АЛАМОУТИ**

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

Разработчики:  
доц. каф. СРС  
\_\_\_\_\_ Кологривов В.А.;  
Ст. гр. 1В1  
\_\_\_\_\_ Писаренко Н.С.

## АННОТАЦИЯ

Лабораторная работа “**Исследование технологии ММО, построенной по схеме Аламоути**” посвящена экспериментальному исследованию технологии **ММО**, выявлению ее явных достоинств в условиях многолучевости канала распространения радиосигнала с использованием пакета функционального моделирования **Simulink**.

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

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

**СОДЕРЖАНИЕ**

<b>1. Цель работы. Краткие сведения из теории</b>	<b>4</b>
<b>2. Краткое описание пакета Simulink</b>	<b>10</b>
<b>2.1. Общая характеристика пакета Simulink</b>	<b>10</b>
<b>2.2. Запуск и работа с пакетом Simulink</b>	<b>10</b>
<b>3. Описание структуры и принципов функционирования Simulink-модели технологии ММО по схеме Аламути</b>	<b>12</b>
<b>4. Описание используемых блоков библиотеки Simulink</b>	<b>21</b>
<b>5. Экспериментальное задание</b>	<b>28</b>
<b>6. Контрольные вопросы</b>	<b>28</b>
<b>Список использованных источников</b>	<b>29</b>

## 1. ЦЕЛЬ РАБОТЫ. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

**Цель работы:** Изучить структуру и принцип работы технологий **MISO**, **SIMO**, **MIMO**, построенных по схеме **Аламоути**, выявить их достоинства по сравнению с технологией **SISO**, реализовать модели в пакете функционального моделирования **Simulink**.

### Теоретическая часть

Вариант реализации функциональной модели **MIMO** приведен на рисунке 1.5 (смотреть в конце пункта 1).

Прежде чем говорить о практической реализации модели, приведем общие сведения о ней [1].

В каналах систем мобильной связи действует комплекс помех и искажений. Сильные замирания сигнала в канале затрудняют оценку переданных сообщений и приводят к ухудшению качества передачи информации. Традиционным методом повышения помехоустойчивости таких систем является разнесение.

Среди различных методов разнесения на передаче перспективным является *пространственно-временное кодирование (ПВК)*, реализация которого предполагает не только передачу информационных сигналов через несколько антенн, но и соответствующее их кодирование.

Принцип схемы **ПВК** заключается в том, что из битов исходной информационной последовательности в соответствии с правилами, определяемыми данным кодом, формируются сигналы. Эти сигналы излучаются антеннами передатчика одновременно и в течение заданного интервала времени. Причем из одних и тех же последовательностей битов формируется несколько различных сигналов, т. е. вводится избыточность, которая может быть уменьшена при использовании кодов с исправлением ошибок.

Идея применения в беспроводной связи технологии с несколькими антеннами была сформулирована давно, но только за последние 20 лет исследователи сделали основную часть научных открытий в этой области. В середине 90-х и была изобретена технология **MIMO** – технология передачи данных  $n$  антеннами и их приема  $m$  антеннами. Суть метода состоит в том, что для передачи сигнала используется не одна, а сразу несколько антенн, удаленные друг от друга.

Устоявшиеся системы беспроводной связи используют технологию **SISO** (Single Input Single Output) – система с одной передающей антенной и одной принимающей антенной.

Наиболее распространенной технологией для повышения эффективности радиосвязи при многолучевом распространении сигналов, является разнесенный прием.

Для приема сигнала используется несколько антенн, расположенных на расстоянии друг от друга, обеспечивающим статистическую независимость канала.

Пример такого решения - технология **SIMO** (Single Input Multiple Output) – система с одной передающей антенной и несколькими приемными антеннами или система с разнесенным приемом. Применяют и обратный подход, используют систему с несколькими передающими антеннами и одной принимающей антенной - **MISO** (Multiple Input Single Output) – или система с разнесенной передачей.

Для существенного повышения пропускной способности по сравнению с вышеперечисленными системами применяется технология **MIMO** (Multiple Input Multiple Output) – система с несколькими передающими антеннами и несколькими приемными антеннами.

Преимущество технологии **MIMO** — возможность пространственного уплотнения сигналов (Spatial Division Multiplexing, **SDM**).

Для противодействия замираниям используется разнесение сигналов по времени, частоте и в пространстве. Разнесение по времени или частоте может быть реализовано в классической системе **SISO**, однако комбинированное разнесение в пространстве является отличительной чертой именно технологии **MIMO**.

Блок-схема системы **MIMO** изображена на рисунке 1.1.



Рисунок 1.1 - Блок-схема системы **MIMO**

Систему с **MIMO**, состоящую из передатчика, имеющего  $N_t$  антенн, и приемника, имеющего  $N_r$  антенн, можно изобразить в виде, представленном на рисунке 1.2.

Рассмотрим кодирование и декодирование сигнала в **MIMO** на примере схемы Аламоути.

В схеме **Аламоути** символы разделяются на пары — четный и нечетный. Для их передачи требуется два интервала времени  $\tau_{ii}$ . В первом интервале передающая антенна 1 будет излучать сигнал символа  $s_1$ , в то время как антенна 2 — сигнал символа  $s_2$ . В следующем временном интервале антенна 1 передает сигнал, сформированный из  $-s_2^*$ , а антенна 2 — из  $s_1^*$ . Причем операции сопряжения и отрицания применяются ко всем комплексным отсчетам символа.

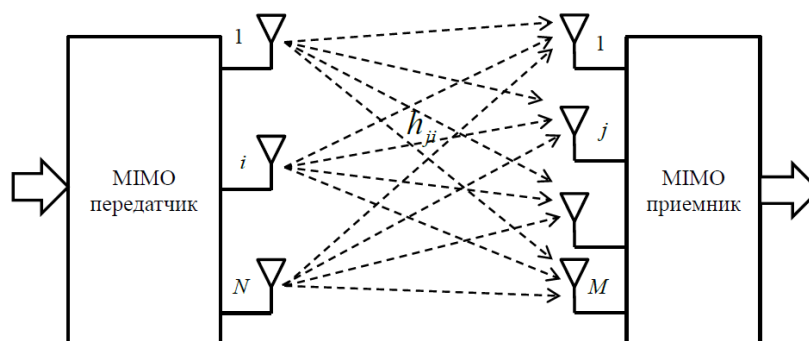


Рисунок 1.2 – Схематическое изображение приемо-передающей системы с **MIMO**

В дальнейшем для упрощения записи выражений будем рассматривать пару комплексных отсчетов, соответствующих одной и той же поднесущей:  $s_1$  — из символа  $s$  и  $s_2$  — из  $s$ . Тогда кодовая матрица схемы **Аламоути** имеет вид:

$$\mathbf{S} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix}.$$

Физический смысл таких манипуляций заключается в том, что если сигналы, излучаемые антеннами передатчика, в первом временном интервале были синфазны, то во втором они будут в противофазе и наоборот. Если же фазы сигналов в первом интервале ортогональны, то они останутся такими и во втором интервале, с той лишь разницей, что опережение фазы излучения поочередно будет возникать то на первой, то на второй антенне. Кроме того, ортогональными будут и сигналы, последовательно излучаемые каждой антенной.

Для приема кодированного по **Аламоути** двухсимвольного сигнала достаточно одной приемной антенны и пары временных отсчетов, в каждом из которых на приемную антенну поступают сигналы

$$\begin{cases} r_1 = s_1 h_1 + s_2 h_2 + n_1 \\ r_2 = -s_2^* h_1 + s_1^* h_2 + n_2 \end{cases},$$

где  $h_1$  и  $h_2$  - коэффициенты передачи **MIMO**-канала первой и второй антеннами соответственно (для рассматриваемой поднесущей), образующие вектор передаточных характеристик:  $\vec{h} = [h_1, h_2]$ ,  $n_1$  и  $n_2$  - компоненты, учитывающие влияние **АБГШ** (аддитивного белого гауссового шума), которые также можно представить в виде вектора:  $\vec{n} = [n_1, n_2]$ , тогда взаимосвязь переданных символов коэффициентов передачи каналов, шумов каналов и принятых символов можно представить в виде

$$\vec{r} = \vec{h}\mathbf{S} + \vec{n},$$

где  $\vec{r} = [r_1, r_2]$ . Векторы  $h$ ,  $n$ ,  $r$  — являются строками, что означает прием на одну антенну и описание процесса кодирования в течение двух интервалов времени.

Декодирование сигналов происходит в два этапа. На первом шаге компоненты  $n_1$  и  $n_2$  принимаются равными нулю, производится преобразование Фурье, и линейный приемник вычисляет оценочные значения переданных отсчетов:

$$\begin{cases} \tilde{s}_1 = \frac{r_1 h_1^* + r_2^* h_2}{|h_1|^2 + |h_2|^2} \\ \tilde{s}_2 = \frac{r_1 h_2^* - r_2^* h_1}{|h_1|^2 + |h_2|^2} \end{cases}.$$

Обозначим множество комплексных отсчетов, соответствующих точкам сигнального созвездия как  $Z$ . Второй этап декодирования заключается в том, что по стратегии наибольшего правдоподобия из множества  $Z$  выбирается тот отсчет, который бы минимизировал расстояние между ним и его оценочным значением, полученным на первом этапе. Математическая запись этого этапа для схемы **Аламути** выглядит следующим образом:

$$\begin{cases} \hat{s}_1 = \arg \min_{s \in Z} d^2(s, \tilde{s}_1) \\ \hat{s}_2 = \arg \min_{s \in Z} d^2(s, \tilde{s}_2) \end{cases}.$$

Оценочные значения по максимуму правдоподобия  $\hat{s}_1$ ,  $\hat{s}_2$  поступают на вход демодулятора.

На практике реализация **MIMO** происходит по следующей схеме:

- из дибита формируем символ **Аламути** (coder\_alamouti);
- производим комплексную модуляцию, для этого представляем наш символ в комплексной форме;

- переносим на несущую частоту получившуюся последовательность, применяя фазовую манипуляция;
- производим декодирование по Аламоути (демодулируем радиосигнал и восстанавливаем исходную последовательность, преобразуя комплексные числа в исходный сигнал).

```

1 function y=coder_alamouti(x);
2 % function y=coder_alamouti(x);
3 % функция кодирования биполярного дибита
4 % по Аламоути
5 if x(1)==1; s1=1+i*1; else s1=-1-i*1; end;
6 if x(2)==1; s2=1+i*1; else s2=-1-i*1; end;
7 % y=[s1 s2 -s2' s1'];
8 y=[s1 -s2' s2 s1'];

```

Рисунок 1.3 – Функция кодирования дибита по Аламоути

```

1 function y=decoder_alamouti(x);
2
3 switch x(1);
4 case (1+i*1)
5     y(1)=1;
6 case (1-i*1)
7     y(1)=1;
8 case (-1-i*1)
9     y(1)=-1;
10 case (-1+i*1)
11     y(1)=-1;
12 otherwise
13     y(1)=0;
14 end;
15 switch x(2);
16 case (1+i*1)
17     y(2)=1;
18 case (-1+i*1)
19     y(2)=1;
20 case (-1-i*1)
21     y(2)=-1;
22 case (1-i*1)
23     y(2)=-1;
24 otherwise
25     y(2)=0;
26 end;
27
28

```

Рисунок 1.4 – Функция декодирования символов Аламоути



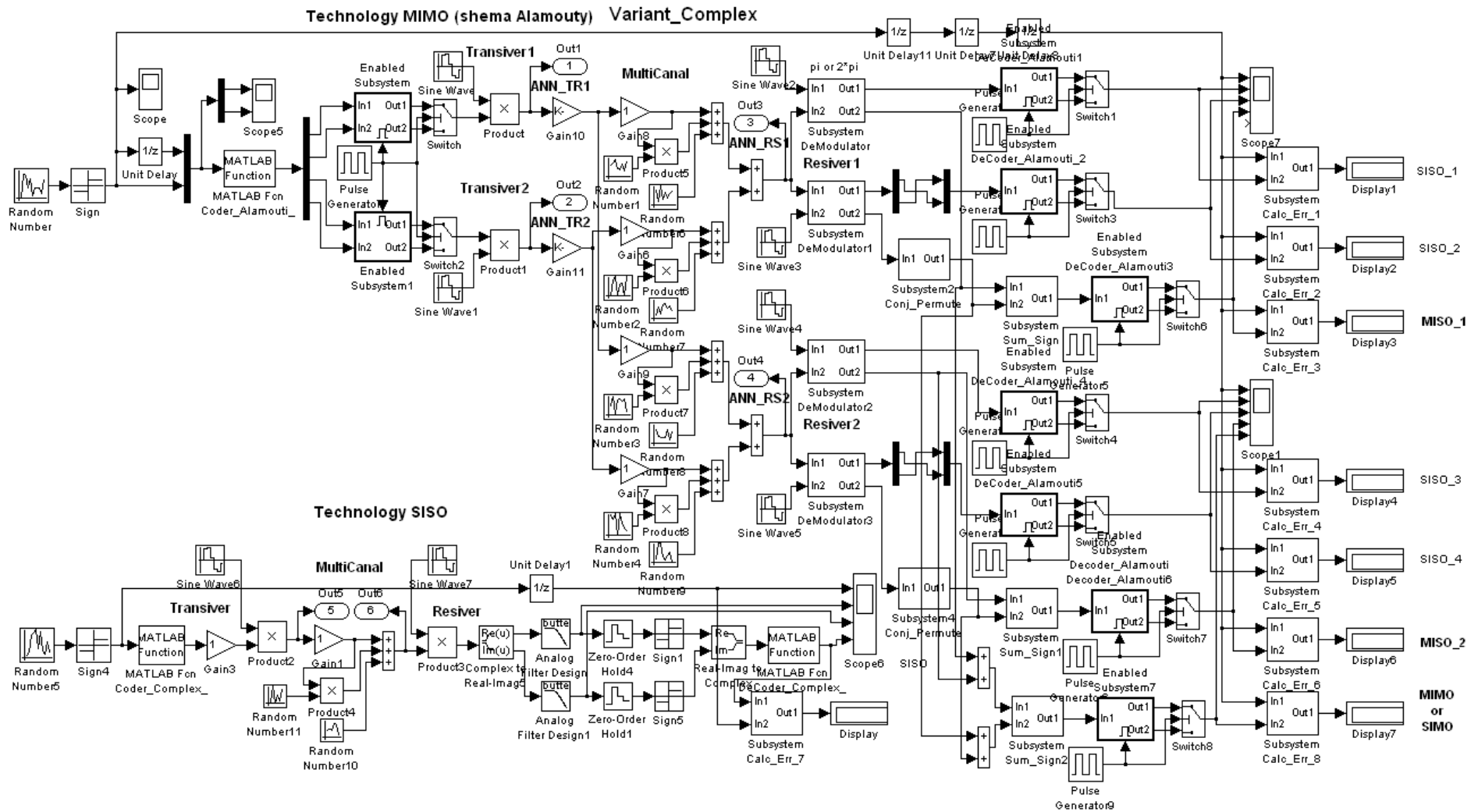


Рисунок 1.5 - Функциональные модели MISO, SIMO, MIMO и SISO


## 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**).

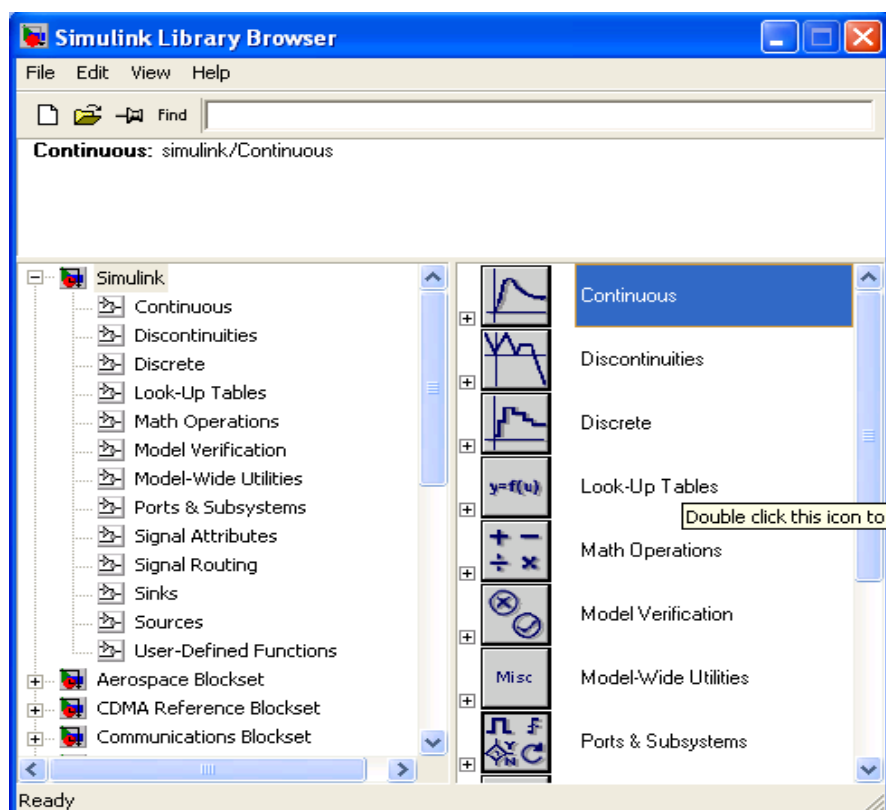


Рисунок 2.1. – Библиотека блоков **Simulink Library Browser**

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 СТРУКТУРЫ И ПРИНЦИПОВ ФУНКЦИОНИРОВАНИЯ SIMULINK-МОДЕЛИ ТЕХНОЛОГИИ ММО ПО СХЕМЕ АЛАМОУТИ

Технология **ММО** относится к многоантенным технологиям пространственного кодирования с целью увеличения скорости либо помехоустойчивости передачи в условиях многолучевого распространения радиосигналов.

С целью упрощения модели и с учетом ограничений накладываемых функциональной средой моделирования **Simulink** предлагается простая функциональная модель технологии **ММО** по схеме **Аламоути** с квазистатическим представлением многолучевых каналов распространения радиосигналов при двух передающих и двух приемных антеннах. Другими словами, в отличие от чисто математического подхода к технологии **ММО** по схеме **Аламоути**, здесь рассматривается возможная аппаратная реализация технологий средствами среды функционального моделирования **Simulink**.

Источник биполярного цифрового сигнала моделируется блоком датчика случайных чисел **Random Number** с шагом  $\tau=1$  и блоком **Sign**.

Далее с использованием разветвления, элемента задержки и блока **mux** формируются вектора-дибиты  $(d_1, d_2)$ ,  $(d_2, d_3)$ ,  $(d_3, d_4)$  и т.д. С помощью блока **MatLab Fcn** использующего функцию **Coder\_Alamouti** и блока **mux** производится комплексное кодирование дибит и формирование комплексного символа **Аламоути**  $[s_1; -s_2^*; s_2; s_1^*]$  длительностью  $\tau=1$ . Предварительное комплексное кодирование биполярного дибита осуществляется следующим образом: если  $d_1=1$ ; то  $s_1=1+i*1$  иначе  $s_1=-1-i*1$ ; если  $d_2=1$ ; то  $s_2=1+i*1$  иначе  $s_2=-1-i*1$ . На рисунке 3.1 приведена таблица комплексного кодирования биполярной битовой последовательности.

Первые два элемента (вектор-дизл\_1) символа **Аламоути**  $(s_1; -s_2^*)$  предназначены для передачи первой антенной  $A_1$ , соответственно в первый и

второй интервалы времени ( $t_1, t_2$ ). Третий и четвертый элементы (вектор-диэл\_2) символа **Аламоути** ( $s_2; s_1^*$ ) предназначены для передачи второй антенной A2, соответственно в первый и второй интервалы времени ( $t_1, t_2$ ). На рисунке 3.2 приведен символ **Аламоути** в координатах – номер передающей антенны, интервал передачи. Здесь  $k$ , а  $k$ , где  $k$  – номер текущего бита.


Рисунок 3.1 – Схема кодирования биполярной битовой последовательности

Рисунок 3.2 – Символ **Аламоути** в координатах (номер передающей антенны, интервал передачи)

В соответствии с таблицей комплексного кодирования битовой последовательности, представленной на рисунке 3.1, реализуется функция **coder\_alamouti**.

```
function y=coder_alamouti(x);
% function y=coder_alamouti(x);
% Функция комплексного кодирования биполярного дибита
% и формирования символа Аламоути

if x(1)==1; s1=1+i*1; else s1=-1-i*1; end;
if x(2)==1; s2=1+i*1; else s2=-1-i*1; end;
y=[s1; -s2'; s2; s1'];
```

Для исключения промежуточных дибит типа ( $d_2, d_3$ ), ( $d_3, d_4$ ) и т.д. и расширения по длительности пар элементов (векторов-диэлов) символа **Аламоути** до  $\tau=2$  используются блоки **Enabled Subsystem**, управляемые генератором прямоугольных импульсов **Pulse Generator** с периодом  $T=2$  и скважностью  $q=2$ . Для развертки пар элементов символа **Аламоути** (векторов-диэлов) во времени с целью последовательной передачи антеннами A1 и A2 используются двухпортовые переключатели **Switch**, управляемые, упомянутым выше, генератором прямоугольных импульсов **Pulse Generator**.

Для формирования радиосигналов излучаемых антеннами A1 и A2 используется **BPSK** модуляция, осуществляемая путем перемножения блоком **Product** элементов символа **Аламоути** с колебаниями несущей

частоты  $\omega=20\pi$  генерируемых блоками **Sine Wave**. С целью возможности разделения в приемнике сигналов принятых с двух передающих антенн, фазы колебаний опорных генераторов несущих колебаний передатчиков разнесены по фазе на  $\pi/2$ . Таким образом, использована квадратурная технология передачи информации на одной несущей частоте. В результате модуляции антеннами излучаются комплексные фазо-манипулированные радиосигналы на одной несущей частоте  $\omega=20\pi$ .

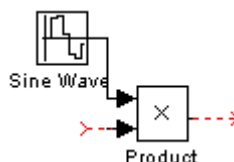


Рисунок 3.3 – **Simulink**-модель блока **BPSK** модулятора

Блоки **Gain**, установленные в модели после антенн использованы для возможности изменения уровней излучаемых радиосигналов.

Многолучевые каналы распространения радиосигналов между двумя передающими и двумя приемными антеннами также реализуются блоками **Gain**, которые позволяют изменять детерминированные условия передачи сигналов (фединги каналов) между передающими и приемными антеннами. Флуктуации коэффициентов (статистические составляющие) передач сигналов между антеннами моделируются перемножением (блок **Product**) передаваемой антенной сигнала и псевдослучайной последовательности генерируемой блоком **Random Number** и последующим сложением (блок **Sum**) с детерминированной частью. Для моделирования разноса приемных антенн с целью достижения статистической независимости каналов передач между антеннами используется параметр **Initial Seed** блоков **Random Number**, кроме того интенсивность флуктуаций задается параметром дисперсии **Variance**, а параметр **Sample time** позволяет изменять характерное время флуктуаций. Кроме флуктуаций коэффициентов передач каналов в модель каналов распространения через блоки **Sum** включены источники шумов (помех) в виде дополнительных блоков генераторов **Random Number**.

На приемной стороне сигналы с выходов каждой из приемных антенн разветвляются и подаются на блоки демодуляторов **Subsystem DeModulator**, на вторые входы которых подаются колебания с синхронизированных с точностью до фазы опорных генераторов **Sine Wave**. Фазы опорных генераторов разветвленных каналов, как и в модуляторах, разнесены на  $\pi/2$ .

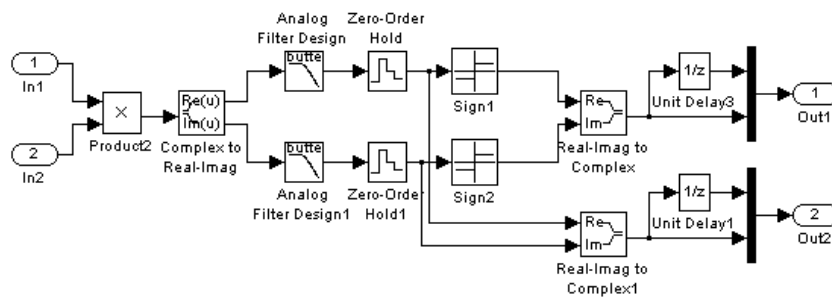


Рисунок 3.4 – Simulink-модель блока **Subsystem DeModulator**

Так как демодулированные сигналы комплексны, то после перемножителя **Product** они разделяются на вещественные квадратурные составляющие с помощью блока **Complex to Real-Imag**. В каждом квадратурном канале демодулятора для фильтрации второй гармоники несущей используются блоки **Analog Filter Design**, порядок фильтра обычно выбирается небольшим ( $n=2$ ), а полоса пропускания по первым нулям спектральной плотности составляет . Синхронизация принимаемых отсчетов комплексных элементов символов осуществляется блоками экстраполяторов нулевого порядка **Zero-Order Hold**, которые удерживают начальный уровень сигнала на протяжении интервала времени указанного параметром **Sample time**. Далее предусмотрены две возможности формирования выходного сигнала блока демодулятора **DeModulator**. Для подачи на основной выход вещественные квадратурные компоненты блоком **Sign** трансформируются в значения (нормируются), объединяются в комплексную компоненту блоком **Real-Imag to Complex**, далее с использованием разветвления, элемента задержки и блока **mux** формируются принятые векторы-диэлы ( $s1; -s2^*$ ) или ( $s2; s1^*$ ) в зависимости от принимаемого сигнала излученного первой A1 или второй A2 передающими антеннами. В данном случае принятые векторы-диэлы могут быть переданы непосредственно на вход блока декодера **DeCoder\_Alamouti**.

Для подачи на дополнительный (вспомогательный) выход вещественные квадратурные компоненты, минуя операцию **Sign**, объединяются в комплексную компоненту блоком **Real-Imag to Complex**, далее с использованием разветвления, элемента задержки и блока **mux** формируются принятые векторы-диэлы ( $s1; -s2^*$ ) или ( $s2; s1^*$ ) в зависимости от принимаемого сигнала излученного первой A1 или второй A2 передающими антеннами. Исключение трансформации принятых диэлов блоком **Sign** позволяет складывать сигналы нескольких статистически независимых каналов передачи с целью снижения вероятности ошибочного приема.

В каналах приема векторов-диэлов ( $s2; s1^*$ ) передаваемых второй передающей антенной A2 на основном выходе блока демодулятора **Subsystem DeModulator** перед блоком декодера **Subsystem**

**DeCoder\_Alamouti** осуществляется перестановка элементов с помощью перекрещенного соединения блоков **demux** и **mux**.

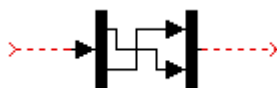


Рисунок 3.5 – **Simulink**-модель перекрещенного соединения блоков **demux** и **mux**

В каналах приема векторов-диэлов ( $s_2$ ;  $s_1^*$ ) передаваемых второй передающей антенной  $A_2$  на дополнительном выходе блока демодулятора **Subsystem DeModulator** перед блоками **Subsystem Sum\_Sign** и декодера **Subsystem DeCoder\_Alamouti** осуществляется перестановка элементов специальным блоком **Subsystem Conj\_Permute**. Блок **Subsystem Conj\_Permute** выполняет: операцию комплексного сопряжения элементов, смены знака второго элемента и собственно перестановку с помощью перекрещенного соединения блоков **demux** и **mux**. В результате такой перестановки вектор-диэл вида ( $s_2$ ;  $s_1^*$ ) преобразуется в вектор-диэл вида ( $s_1$ ;  $-s_2^*$ ), что позволяет производить сложение векторов-диэлов статистически независимых каналов с целью повышения помехоустойчивости суммарного канала.

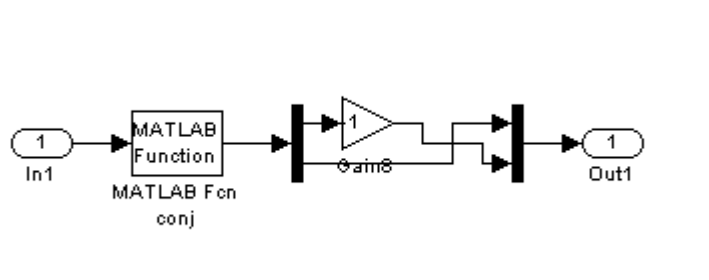


Рисунок 3.6 – **Simulink**-модель блока **Subsystem Conj\_Permute**

Полученные с разных антенн вектора-диэлы поступают на вход блока **Subsystem Sum\_Sign** с целью суммирования и нормировки. Блок **Sum** суммирует, предварительно приведенные к одному виду, комплексные сигналы пришедшие на входы. Блок **Complex to Real-Imag** разделяет суммарный комплексный сигнал на вещественные квадратурные составляющие. Блоки **Sign** трансформируют их значения в  $\pm 1$  и с помощью блока **Real-Imag to Complex** преобразуются нормированные комплексные сигналы.



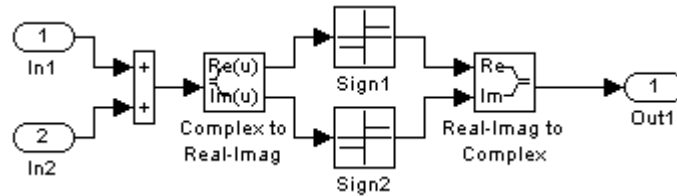


Рисунок 3.7 – Simulink-модель блока **Subsystem Sum\_Sign**

Декодер Аламоути блок **Subsystem DeCoder\_Alamouti** представляет собой блок **Enabled Subsystem** содержащий блок **MatLab Fcn** с блоком **demux** на выходе и использующий функцию **DeCoder\_Alamouti\_**. Блок **Enabled Subsystem** использован для расширения длительности декодируемых биполярных импульсов вдвое с целью последующего преобразования параллельного (векторного) представления в последовательное с помощью блока двухпортового переключателя **Switch**. Блоки **Enabled Subsystem** и **Switch** управляются генератором прямоугольных импульсов **Pulse Generator**.

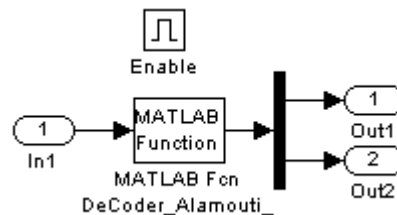


Рисунок 3.8 – Simulink-модель декодера Аламоути


Рисунок 3.9 – Схема декодирования символа Аламоути и учетом перестановки элементов переданных второй антенной

```
function y=decoder_alamouti_(x);
% function y=decoder_alamouti_(x);
% Функция декодирования символа Аламоути
% с учетом перестановки элементов переданных второй антенной

switch x(1);
    case (1+i*1)
        y(1)=1;
    case (1-i*1)
```

```

y(1)=1;
case (-1-i*1)
y(1)=-1;
case (-1+i*1)
y(1)=-1;
otherwise
y(1)=0;
end;
switch x(2)
case (1+i*1)
y(2)=1;
case (-1+i*1)
y(2)=1;
case (-1-i*1)
y(2)=-1;
case (1-i*1)
y(2)=-1;
otherwise
y(2)=0;
end;

```

На выходе каждого канала приема и обработки установлен блок детектора битовых ошибок **Subsystem Calc\_Err\_** содержащий два входа (опорного и принимаемого битовых потоков), вычитатель на базе блока **Sum**, блока взятия модуля **Abs** и блок аналогового интегратора **Integrator**. Для визуализации числа битовых ошибок использован блок **Display**.

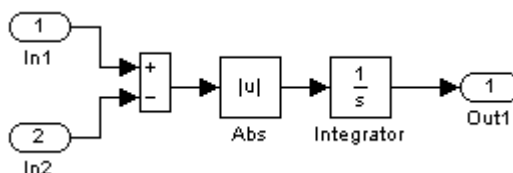


Рисунок 3.10 – **Simulink**-модель детектора ошибок

**Резюме:** Итак, **Simulink**-модель **MIMO** технологии по схеме **Аламоути** состоит из:

генератора информационной биполярной последовательности импульсов;

комплексного кодера дибит и формирователя символа **Аламоути**;

**BPSK** квадратурных модуляторов каналов первой и второй передающих антенн;

многолучевого канала распространения сигнала между двумя передающими и двумя приемными, содержащего стационарную и флуктуирующую составляющую передачи, а также шумы каналов распространения;

**BPSK** демодуляторов квадратурных составляющих сигналов принимаемых каждой приемной антенной от двух передающих;

блоков сложения демодулированных сигналов принятых приемными антеннами для реализации технологий **MISO, SIMO, MIMO**;

декодеров принятых комплексных и сформированных сигналов (символов **Аламоути**);

детекторов ошибок принимаемых сигналов по каждому из каналов.

Для исследования помехоустойчивости технологий **MISO** и **MIMO** запускается модель представленная на рисунке 1.5, а для исследования технологии **SIMO** необходимо отключить каналы на входах приемником одной из передающих антенн и соответствующий вход нижнего блока **Subsystem Sum\_Sign**. В результате отключений сигнала одной из передающих антенн детектор канала **MIMO** покажет число ошибок по технологии **SIMO**.

### Структура и принцип функционирования **Simulink**-модели технологии **SISO**

Для обеспечения сравнительного анализа технологий **MISO, SIMO, MIMO** и **SISO** опишем функциональную **Simulink**-модели технологии **SISO**.

Источник биполярного цифрового сигнала моделируется блоком датчика случайных чисел **Random Number** с шагом  $\tau=1$  и блоком **Sign**.

С помощью блока **MatLab Fcn** использующего функцию **Coder\_Complex\_** и блока **mux** производится комплексное кодирование биполярной последовательности. Алгоритм кодирования представлен листингом **MatLab**-функции

```
function y=coder_complex_(x);
% function y=coder_complex_(x);
% Функция комплексного кодирования
```

```
if x==1; y=1+i*1; end;
if x==-1; y=-1-i*1; end;
```

Для формирования радиосигнала излучаемого антенной используется **BPSK** модуляция, осуществляемая путем перемножения блоком **Product** комплексных элементов с колебаниями несущей частоты  $w=20*\pi$  генерируемых блоками **Sine Wave**. В результате модуляции антенной излучается комплексный фазо-манипулированный радиосигнал на несущей частоте  $w=20*\pi$ .

Блок **Gain**, установленный в модели перед модулятором использован для возможности изменения уровня излучаемого радиосигнала.

Многолучевый канал распространения радиосигнала между передающей и приемной антеннами также реализуется блоком **Gain**, который позволяет изменять детерминированные условия передачи сигналов (фединг

канал) между передающей и приемной антеннами. Флуктуации коэффициентов (статистические составляющие) передач сигналов между антеннами моделируются перемножением (блок **Product**) передаваемой антенной сигнала и псевдослучайной последовательности генерируемой блоком **Random Number** и последующим сложением (блок **Sum**) с детерминированной частью. Для моделирования разноса приемных антенн с целью достижения статистической независимости каналов передач между антеннами используется параметр **Initial Seed** блоков **Random Number**, кроме того интенсивность флуктуаций задается параметром дисперсии **Variance**, а параметр **Sample time** позволяет изменять характерное время флуктуаций. Кроме флуктуаций коэффициентов передач каналов в модель каналов распространения через блоки **Sum** включены источники шумов (помех) в виде дополнительных блоков генераторов **Random Number**.

На приемной стороне сигнал с выход приемной антенны подаются на демодулятор блок **Product**, на второй вход которого подаются колебания с синхронизированного с точностью до фазы опорного генератора **Sine Wave**.

Так как демодулированные сигналы комплексны, то после перемножителя **Product** они разделяются на вещественные квадратурные составляющие с помощью блока **Complex to Real-Imag**. В каждом квадратурном канале демодулятора для фильтрации второй гармоники несущей используются блоки **Analog Filter Design**, порядок фильтра обычно выбирается небольшим ( $n=2$ ), а полоса пропускания по первым нулям спектральной плотности составляет .

Синхронизация принимаемых отсчетов комплексных элементов символов осуществляется блоками экстраполяторов нулевого порядка **Zero-Order Hold**, которые удерживают начальный уровень сигнала на протяжении интервала времени указанного параметром **Sample time**. Для подачи на основной выход вещественные квадратурные компоненты блоком **Sign** трансформируются в значения (нормируются), объединяются в комплексную компоненту блоком **Real-Imag to Complex** и далее передаются непосредственно на вход блока декодера **DeCoder\_Complex\_**.

Декодер представляет собой блок **MatLab Fcn**, использующий функцию **DeCoder\_Complex\_**. Алгоритм декодирования представлен листингом **MatLab**-функции

```
function y=decoder_complex_(x);
% function y=decoder_complex_(x);
% Функция комплексного декодирования

switch x;
    case (1+i*1)
        y=1;
    case (-1-i*1)
        y=-1;
```

```

otherwise
    y(1)=0;
end;

```

На выходе каждого канала приема и обработки установлен блок детектора битовых ошибок **Subsystem Calc\_Err\_** содержащий два входа (опорного и принимаемого битовых потоков), вычитатель на базе блока **Sum**, блока взятия модуля **Abs** и блок аналогового интегратора **Integrator**. Для визуализации числа битовых ошибок использован блок **Display**.

#### 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

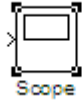
$$sign(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

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

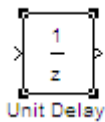


**Switch** – блок переключателя. *Назначение:* переключение входных сигналов по сигналу управления. *Параметры блока:* **Criteria for passing first input** – условие прохождения сигнала с первого входа, значение выбирается из списка: **u2>=Threshold** – сигнал управления больше или равен пороговому значению;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

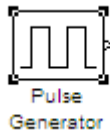
порогового значения;  $u2 \sim \text{Threshold}$  – сигнал управления не равен пороговому значению. **Threshold** – порог; флажок **Show additional parameters** – показать дополнительные параметры. При выставленном флажке отображаются дополнительные окна списков, в нашем случае флажок не используется.



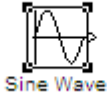
**Scope** – блок осциллографа. *Назначение:* построение графиков исследуемых сигналов как функций времени. Открытие окна осциллографа производится двойным щелчком ЛКМ на пиктограмме блока. В случае векторного сигнала каждая компонента вектора отображается отдельным цветом. Настройка окна осциллографа 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

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

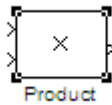
где  $A$  - амплитуда сигнала;  $f$  - частота сигнала в Гц;  $T$  - такт дискретности;  $N$  - количество тактов в секунду;  $k$  - номер текущего шага,  $k = 0, \dots, N - 1$ ;  $\varphi$  - начальная фаза сигнала;  $l$  - начальная фаза, заданная количеством тактов;  $b$  - постоянная составляющая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

Параметры блока в режиме **Sample-based: Amplitude** – амплитуда; **Bias** – постоянная составляющая 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.

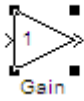


**Random Number** – источник случайного сигнала с

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

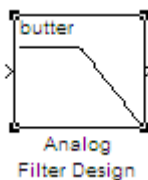


нашем случае блок используется для моделирования шумов канала передачи данных.



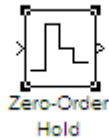
**Gain** – блок усилителя. *Назначение:* блок **Gain** умножает входной сигнал на постоянный коэффициент; *Параметры блока:* **Multiplication** – способ выполнения операции, значение параметра выбирается из списка: **Element-wise  $K*u$**  – поэлементный; **Matrix  $K*u$**  – матричный, коэффициент усиления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



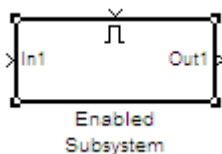
**Analog Filter Design** – блок аналогового фильтра заданного метода проектирования и типа из подраздела **Filter Design**; подраздела **Filtering**, раздела **DSP Blockset** библиотеки **Simulink** [4]. *Назначение:* аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. *Параметры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

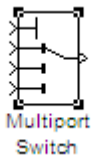
запуске и выбирается из списка: **held** – использовать предыдущее активное состояние; **reset** – использовать начальное (исходное состояние). **Show output port** – показать выходной порт, при установленном флажке на пиктограмме блока **Enable** появляется дополнительный выходной порт, сигнал с которого 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**.



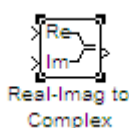
**Repeating Sequence** – источник периодического сигнала.

*Назначение:* формирование заданного пользователем периодического сигнала. *Параметры блока:* **Time values** – вектор значений времени; **Output values** – вектор значений сигнала. Блок выполняет линейную интерполяцию выходного сигнала для моментов времени не совпадающих со значениями, заданными вектором **Time values**.



**Multiport Switch** – блок многовходового переключателя.

*Назначение:* выполняет переключение входных сигналов на выход по сигналу управления, задающему номер активного входного порта. *Параметры блока:* **Number of inputs** – количество входов; флажок **Show additional parameters** – показать дополнительные параметры, в нашем случае не используется. Блок **Multiport Switch** пропускает на выход сигнал с того входного порта, номер которого равен текущему значению управляющего сигнала. Если управляющий сигнал не является сигналом целого типа, то блок **Multiport Switch** производит округление значения в соответствии со способом, выбранным в графе дополнительного параметра **Round integer calculations toward**.



**Real-Imag to Complex** – блок вычисления комплексного

числа по его действительной и мнимой частям. *Назначение:* вычисление комплексного значения по его проекциям на действительную и мнимую оси.

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

## 5. ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

1. Изучить функциональные модели для исследования технологий **SISO**, **MISO**, **SIMO** и **MIMO** работающих по схеме **Аламоути** в соответствии с рисунком 1.5.

2. Изучить подсистемы моделей в соответствии с рисунками 3.4, 3.6, 3.7 и 3.8. Разобраться с параметрами блоков модели.

3. Проанализировать используемые в модели **MatLab**-функции для исследуемой модели `coder_alamouti_`, `coder_complex_`, `decoder_alamouti_`, `decoder_complex_` в соответствии с приведенными листингами.

4. Провести исследование технологий при шумах канала передачи ; `sample time=0.05`; при дисперсиях флуктуаций передач каналов т.е. ; при двух характерных временах замираний каналов передачи `sample time=2.5, 5`; время счета `Stop time=1000` или `10000`.

4. В пакете **MatLab** изобразить графики зависимости величины битовой ошибки от отношения сигнал/шум, используя **MatLab**-функцию **semilogy** для всех исследуемых технологий.

5. В отчет включить рисунки исследуемых схем, необходимые графики и пояснения.

## 6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое технология **MIMO**? В чем ее принципиальное отличие от **SISO**?

2. Каковы преимущества технологии **MIMO** в отличие от других технологий?

3. Какие проблемы могут возникнуть при передаче сигнала по многолучевому каналу? Как с ними бороться?

4. Какие вы знаете методы разнесения? Основная идея пространственно временного кодирования?

5. Изобразите блок-схему **MIMO**. Кратко расскажите о функциях каждого отдельного блока.

6. Какова основная идея кодирования и декодирования по **Аламоути**? В чем преимущества схемы **Аламоути**?

7. Каков физический смысл кодовой матрицы схемы **Аламоути**?

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Банкет В.Л., Незгазинская Н.В., Токарь М.С. Методы пространственно-временного кодирования для систем радиосвязи. ж. Цифровые технологии, №6, 2009, С. 5-16
2. Крейнделин В.Б., Резнев А.А. Пространственно-временное кодирование в сетях связи четвертого поколения. Материалы Международной научно-технической конференции 3-7 декабря 2012 г. Москва INTERMATIC-2012, часть 5, С. 112-115.
3. Рашич А.В. Сети беспроводного доступа WiMAX: учеб. пособие / Рашич А.В. – СПб.: Изд-во Политехн. ун-та, 2011.– 179 с.
4. Слюсар В.И. Системы **ММО**: принципы построения и обработка сигналов. ж. Электроника: наука, технология, бизнес, №8, 2006, С. 52-58.