

Министерство науки и высшего образования Российской Федерации

Томский государственный университет  
систем управления и радиоэлектроники

Кологривов В. А.  
Игумнова Н.А.

**ИССЛЕДОВАНИЕ 4FSK МОДУЛЯЦИИ ПРИ НЕОРТОГОНАЛЬНОМ РАЗНЕСЕНИИ  
НЕСУЩИХ**

Методические указания по лабораторной работе в среде функционального моделирования  
Simulink системы MatLab для студентов направления «Инфокоммуникационные технологии  
и системы связи»

Томск 2022

УДК 621.396.61+621.376.3

ББК 32.811.7

К 61

**Рецензент:**

**Мещеряков А.А.**, доцент кафедры радиотехнических систем ТУСУР, канд. техн. наук

**Кологривов В. А., Игумнова Н. А.**

К 61 Исследование 4FSK модуляции при неортогональном разнесении несущих: методические указания по лабораторной работе для студентов направления «Инфокоммуникационные технологии и системы связи» / Кологривов В. А., Игумнова Н. А. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2022 – 20 с.

Настоящие методические указания по лабораторной работе составлены с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО).

Лабораторная работа «Исследование 4FSK модуляции при неортогональном разнесении» посвящена модельному исследованию 4FSK -модема с использованием пакета функционального моделирования Simulink системы для инженерных и научных расчетов MatLab.

Работа “Исследование 4FSK модуляции при неортогональном разнесении” относится к циклу лабораторных работ по разделу “Модуляция/Демодуляция”, входящему в дисциплины по направлению “Инфокоммуникационные технологии и системы связи”.

В описании сформулирована цель лабораторной работы, приведены краткие теоретические сведения о FSK-модуляции, краткая характеристика пакета Simulink системы MatLab, описание виртуального лабораторного макета и используемых блоков библиотеки Simulink, а также требования к модельному исследованию и контрольные вопросы, ответы на которые необходимы для успешной защиты лабораторной работы.

Одобрено на заседании каф. РТС протокол № 4 от 27.01.22

УДК 621.396.61→621.376.3

ББК 32.811.7

© Кологривов В. А., Игумнова Н. А., 2022

© Томск. гос. ун-т систем упр. и радиоэлектроники, 2022

## ОГЛАВЛЕНИЕ

<b>1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О FSK-МОДУЛЯЦИИ</b> .....	4
1.1 Теоретические сведения о FSK-модуляции .....	4
<b>2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ MSK-МОДЕМА</b> .....	7
<b>3 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK И ИСПОЛЬЗУЕМЫХ БЛОКОВ</b> .....	13
3.1 Запуск и работа с пакетом Simulink.....	13
3.2 Описание используемых блоков библиотеки Simulink .....	15
<b>4 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ</b> .....	18
<b>5 КОНТРОЛЬНЫЕ ВОПРОСЫ</b> .....	19
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b> .....	20

## 1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О FSK-МОДУЛЯЦИИ

Цель работы: изучение влияния способа разнесения несущих частот на помехоустойчивость 4FSK модемов с использованием пакета функционального моделирования *Simulink*.

### 1.1 Теоретические сведения о FSK-модуляции

При частотной манипуляции частота генерируемого сигнала зависит от текущего информационного символа. В простейшем случае символу «0» соответствует частота  $f_1$ , а символу «1» частота  $f_2$ . Для увеличения скорости передачи информации увеличивают количество используемых частот и тогда каждой частоте соответствует уже не один бит, а несколько. В 4FSK модуляции используется 4 частоты, и каждой частоте соответствует один из дибитов: 00, 01, 10, 11 [1].

В стандартном случае для модуляции выбирают ортогональные частоты. Для того, чтобы получить неортогональный разнос используют следующие способы:

1. Уменьшение длительности модуляционного символа (менее периода  $T$ ) при сохранении неизменного частотного разноса между поднесущими ( $1/T$ );
2. Уменьшение частотного разноса между поднесущими при сохранении неизменной длительности модуляционного символа [2].

В данной лабораторной работе используется второй способ получения неортогонального разнесения несущих.

В данном случае реализуем модуляцию с помощью генератора управляемого напряжением (ГУН), который заменяет четыре независимых генератора. Прием ведется с помощью корреляционного приёмника, для его работы необходимы 4-е опорных генератора колебаний несущих на приёмной стороне.

При корреляционном приеме определяется значение функции корреляции между принимаемым и опорным сигналом. Функция корреляции является функцией параметра смещения  $\tau$  и обычно функция автокорреляции симметрична относительно  $\tau$  и может содержать боковые пики. Функция корреляции определяется соотношением:

где  $S$  – полезный сигнал;  
 $N$  – АБГШ.

При полном совпадении сигналов будет получен положительный максимум значения функции корреляции. Если сигналы находились в противофазе, то максимум станет отрицательным. При отсутствии схожести между сигналами значение функции корреляции будет равно нулю. При частичном совпадении и при наличии помех значение функции корреляции находится в диапазоне от нуля до значения ее максимума.

Альтернативным способом приема является синхронное детектирование (рисунок 1.1).

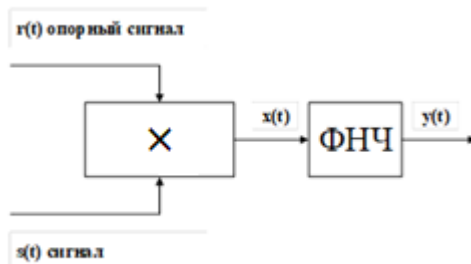


Рисунок 1.1 – Синхронный детектор

На входы умножителя поступают сигнал опорного генератора несущей и принимаемый сигнал. Сигнал на выходе умножителя будет иметь вид:

где  $\omega_c - \omega_s$  – разностная частота;  
 $\omega_c + \omega_s$  – суммарная частота.

Для обнаружения используется составляющая на разностной частоте, ее выделяет ФНЧ после умножения сигналов. Отклик на выходе фильтра будет лишь тогда, когда результирующая частота близка к частоте опорного сигнала. Если частота сигнала равна опорной частоте, то разностная частота равна нулю и в результате умножения будет получено изменение постоянной составляющей сигнала [3].

При синхронном приеме, поэтому формула примет следующий вид:

После ФНЧ фильтрующего высокочастотные составляющие остается:

Таким образом, с точностью до постоянной составляющей выделяется принятый символ.

Функциональная схема 4FSK-модема представлена на рисунке 1.2.

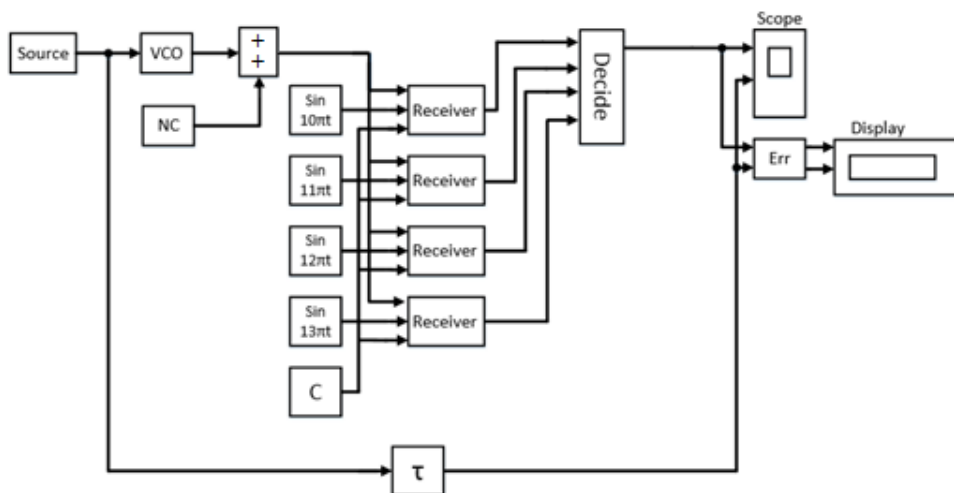


Рисунок 1.2 – Функциональная схема 4FSK-модема с использованием генератора управляемого напряжением (VCO)

Функциональная модель содержит: источник информационного потока данных (Source), модулятор (VCO), модель канала распространения (Sum+NC), приемник-демодулятор (Receiver), опирающийся на значение порога (C), подсистему принятия решений (Decide), детектор ошибок (Err). На осциллограф и детектор ошибок с необходимой задержкой ( $\tau$ ) подается исходная информационная последовательность.

## 2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ MSK-МОДЕМА

Источником сигнала является блок *Random Number*. Данный блок генерирует последовательность псевдослучайных значений. Далее этот поток преобразуется в однополярный поток случайной последовательности (1 и 0). Блок *Sign* нормирует амплитуду сигнала к  $\pm 1$  или 0 в зависимости от поступающего сигнала, последующее сравнение с нулем превращает биполярную последовательность в однополярную. Преобразование происходит с помощью блоков *Sign*, *Constant* и *Relational operator*. Далее с помощью блоков *Unit Delay* и *Mux* образуются дибиты, необходимые для дальнейшей модуляции. Однополярная случайная последовательность поступает в генератор, управляемый напряжением, реализованный с помощью блока *Matlab function*. Функция, прописанная в данном блоке, выглядит следующим образом (рисунок 2.1):

```
function freq=dibit_frequency_ort(x);  
% Преобразование кода в частоту  
% x- бит  
% freq- частота (круговая) рад/сек  
  
if x==[0; 0]; freq=10*pi; end;  
if x==[0; 1]; freq=12*pi; end;  
if x==[1; 0]; freq=14*pi; end;  
if x==[1; 1]; freq=16*pi; end;
```

Рисунок 2.1 –Функция dibit-frequency

В управляемом генераторе установлены несущие частоты, и в зависимости от поступающих символов проходит переключение этих частот (модуляция). При ортогональный частотный разнос равен  $2\pi$ , а неортогональный разнос частот несущих равен  $\pi$ . Далее с помощью блока *Sin* формируются гармонические колебания несущих частот.

После источника и модулятора следует модель канала распространения. Она состоит из блоков *Sum* и *Random Number*. Блок *Random Number* в данном случае выполняет роль источника шумов канала. Подбором шага *Sample time* псевдослучайной последовательности с нормальным распределением обеспечивается широкополосность шума канала. С помощью параметра *Variance* (дисперсия) подбирается мощность шумов канала распространения.

Прием сигналов с помощью корреляторов основан на сравнении поступающего сигнала с опорным сигналом либо с набором опорных сигналов. О совпадении судят по значению корреляционной функции в течении периода следования сигнала.

Т.к. используются дискретные сигналы, то значение функции корреляции вычисляется через сумму произведений отсчетов принятого сигнала и колебаний опорного генератора. В схему приемника входят генераторы опорных сигналов синхронизированные с передатчиком т.к. они необходимы для вычисления корреляции между принятым и опорным сигналом. После приема осуществляется восстановление исходной последовательности.

После приема последовательность битов проходит через блок *Matlab function*, в котором прописана следующая функция (рисунок 2.2):

```

function db=frequency_dibit(x);
% Преобразование частоты
% (номера частотного генератора) в дибит
% x- кодовый двоичный вектор
% db- дибит выходной последовательности

switch num2str(x)
case '1000'
    db=[0 0];
case '0100'
    db=[0 1];
case '0010'
    db=[1 0];
case '0001'
    db=[1 1];
otherwise
    db=[0 0];
end;

```

Рисунок 2.2 – Функция frequency–dibit

При приеме сигнала и совпадении несущей частоты с частотой опорного генератора канала обработки на выходе коррелятора будет наблюдаться заметное изменение значения корреляции. И блок сравнения с порогом выдаст значение 1. В то время как в других каналах обработки эти изменения будут носить шумоподобный характер и блок сравнения с порогом будет выдавать нули. Таким образом, выходы каналов обработки формируют вектор, положение 1 в котором указывает на номер несущей частоты. Функция frequency–dibit по значению вектора вырабатывает дибит, соответствующий данной опорной частоте.

На рисунке 2.3 представлен 4FSK-модем, в котором модуляция выполняется с помощью генератора, управляемого напряжением (ГУН), а демодуляция методом корреляционного приема.



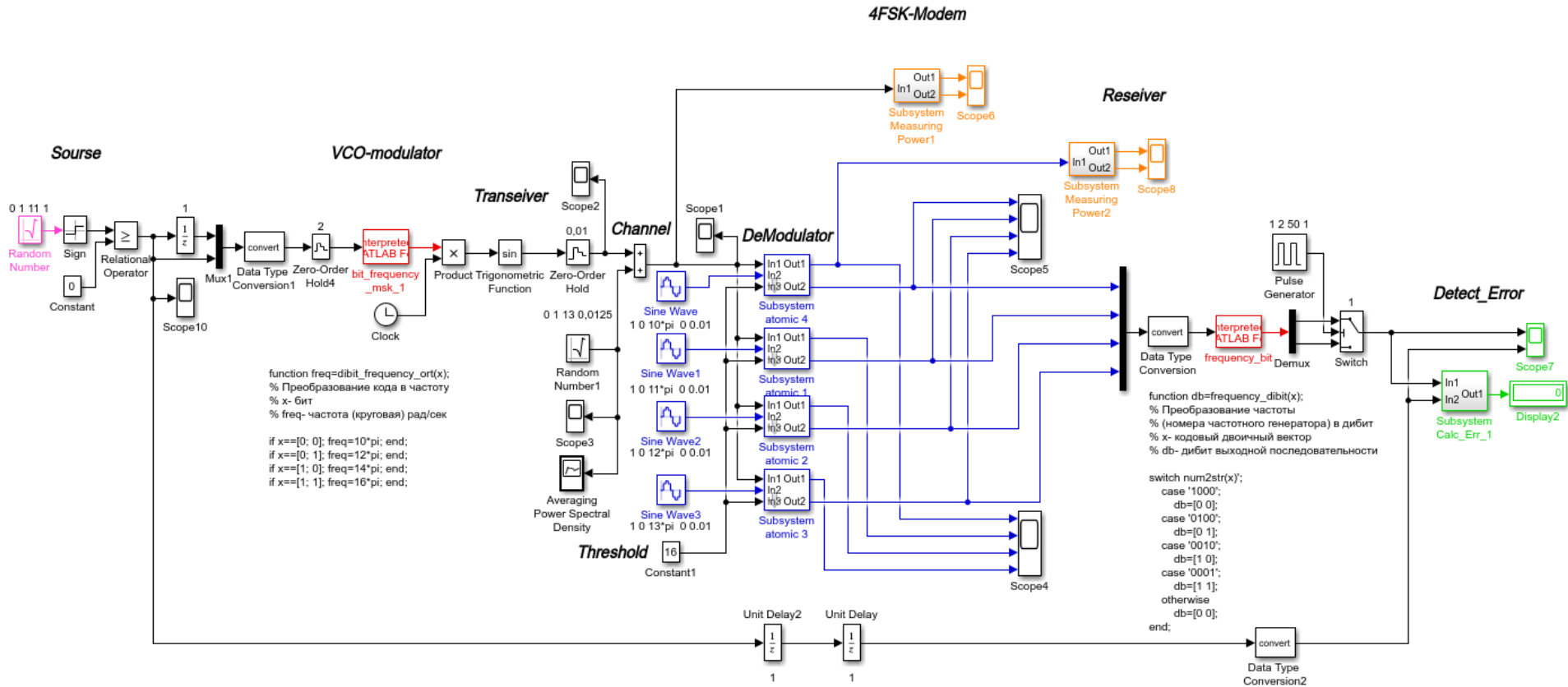


Рисунок 2.3 – Модем 4FSK на основе ГУН и корреляционного приема

**Модель ГУНа.** Рассмотрим подробнее реализацию генератора управляемого напряжением (рисунок 2.4):

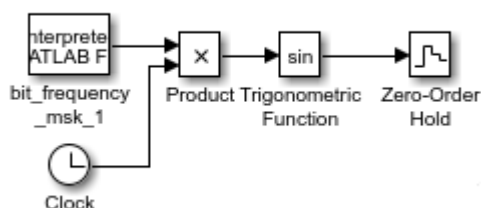


Рисунок 2.4 – Реализация генератора управляемого напряжением

Для работы модели необходимо получить сигнал  $\omega$ . Блок *Matlab Function* преобразует поступающий поток дибитов в соответствующие значения круговых частот несущих колебаний  $\omega$ . Значение частоты зависит от поступившего символа-дибита. Далее полученный поток перемножается с текущим временем блока *Clock* и поступает на вход блока *Trigonometric Function*. После перемножения поток принимает вид  $\omega t$  а на выходе блока появляется гармоническое колебание  $\sin(\omega t)$ . Далее сигнал проходит через блок *Zero-Order Hold*, данный блок задает шаг прорисовки функции.

**Модель коррелятора.** Корреляционный приемник реализован следующим образом (см. рисунок 2.5):

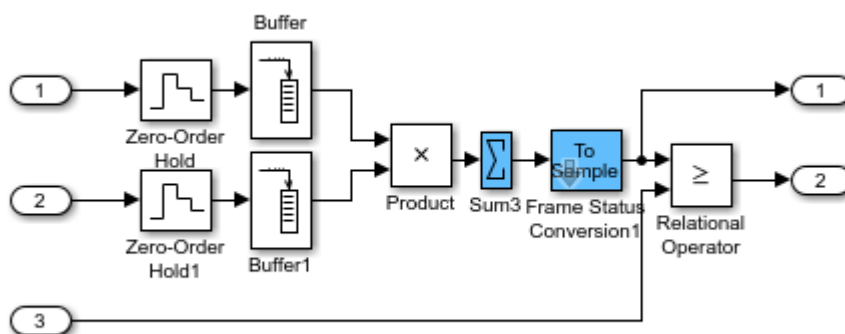


Рисунок 2.5 – Реализация корреляционного приемника

При реализации функциональной модели корреляционного приема решили ограничиться значением функции корреляции при  $\tau = 0$ .

Т.к. используются дискретные (цифровые) сигналы, то значение функции корреляции вычисляется через сумму произведений отсчетов принятого сигнала и колебаний опорного генератора соответствующей несущей. Модель коррелятора содержит два плеча накопления отсчетов. В каждом плече с помощью блоков *Zero-Order Hold* сигнал дискретизируется на отсчеты (в данном случае на 64) и накапливается в блоках *Buffer*. Далее накопленные отсчеты попарно перемножаются в блоке *Product*. Затем вычисляется сумма этих произведений блоком *Sum*, т.е. реализуется скалярное произведение векторов накопленных

отсчетов. Далее происходит сравнение полученного значения с пороговым для принятия решения о том, есть ли полезный сигнал в данном канале.

**Подсистема измерения мощности** сигнала и/или шума выглядит следующим образом (см. рисунок 2.6).

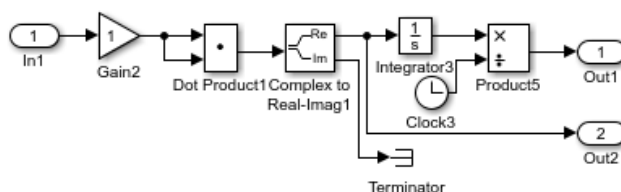


Рисунок 2.6 – Подсистема измерения мощности

Подсистема позволяет измерить мощность как регулярных, так и случайных процессов, как вещественных, так и комплексных. С помощью блока **Dot Product** процесс умножается на сопряженный. Блоком **Complex to Real-Imag** выделяется вещественная часть. Блок **Integrator** вычисляет энергию процесса, а деление блоком **Product** энергии на время (блок **Clock**) вычисляет мощность как скорость поступления энергии.

**Подсистема подсчета ошибок** представлена на рисунке 2.7.

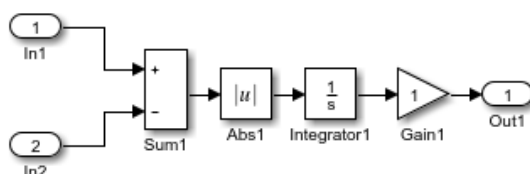


Рисунок 2.7 – Подсистема подсчета ошибок

В детекторе ошибок вычисляется интеграл от модуля разности входных последовательностей, т.е. площадь разностного процесса. Затем, в зависимости от длительности битов и одно- или биполярности подбирается множитель **Gain**, переводящий интеграл разности в эквивалентное количество битов (ошибок).

На первый вход подается сигнал, прошедший через всю схему, на второй вход подан сигнал с генератора входной информационной последовательности с необходимой задержкой. К выходу подсистемы присоединен дисплей, на котором в случае наличия ошибок, будет указано их количество.

**Измерение SNR.** Отношение сигнал/шум (SNR) измеряется по следующей методике: для того чтобы определить мощность сигнала без шумовой составляющей необходимо отсоединить генератор шума от модели канала распространения. Для того чтобы определить мощность смеси полезного сигнала с шумом генератор шумов должен быть подключен к каналу распространения. Изменение отношения сигнал/шум достигается вариацией параметра дисперсии генератора псевдослучайной гауссовской последовательности модели канала распространения.

SNR рассчитывается по формуле, представленной ниже:

(2.1)

Результаты измерений фиксируется в дБ.

При большом числе испытаний, частота появлений битовых ошибок, т.е. отношение числа ошибок к общему числу битов, стремится к вероятности битовых ошибок.


Таким образом, измерение SNR при соответствующем числе ошибок позволяет определить точки водопадоподобной характеристики, позволяющей оценить помехоустойчивость модема.

### 3 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK И ИСПОЛЬЗУЕМЫХ БЛОКОВ

Пакет *Simulink* разработан компанией *Mathworks* 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 [4 – 7].

#### 3.1 Запуск и работа с пакетом Simulink

Для запуска пакета *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 3.1).

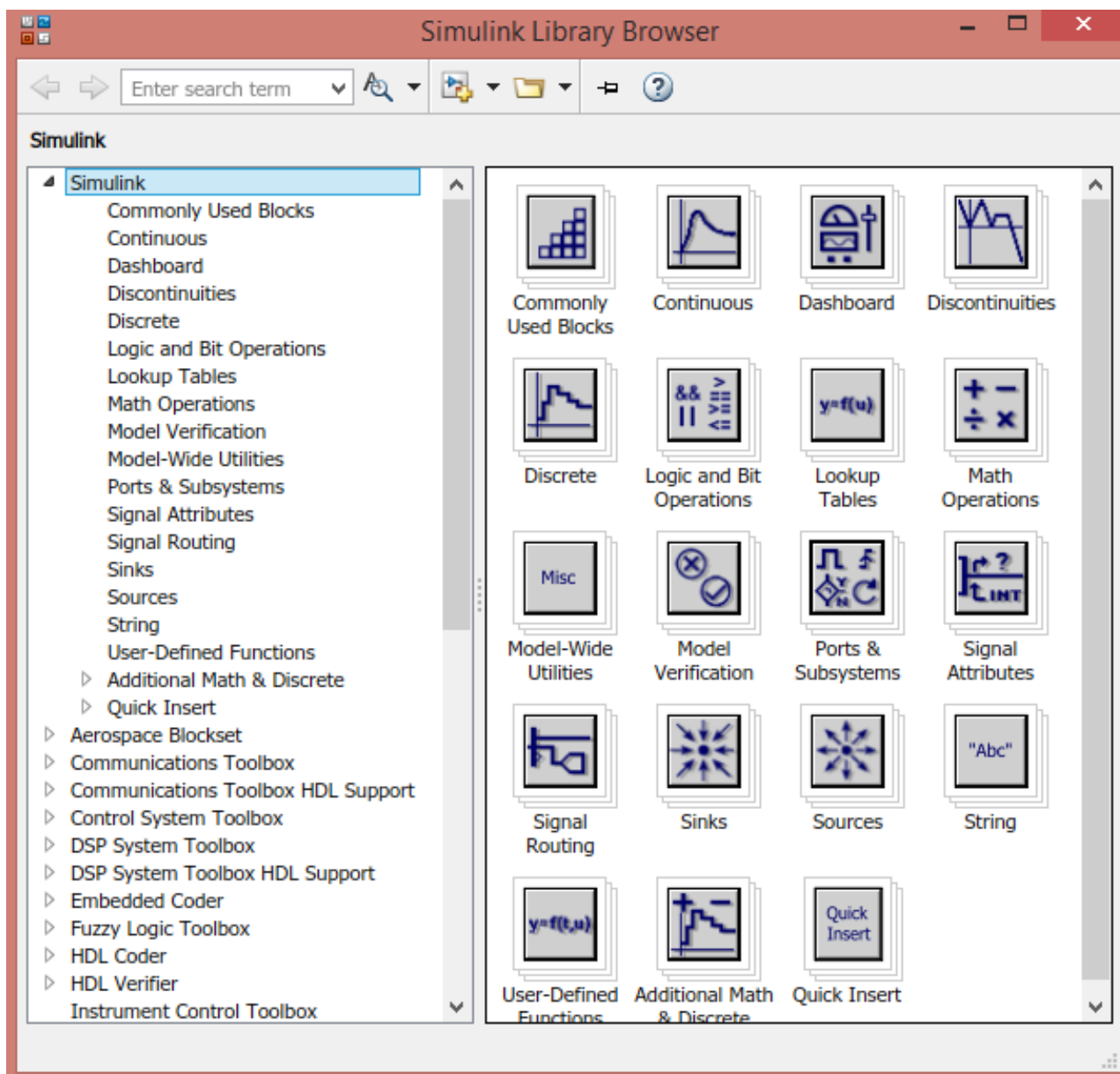


Рисунок 3.1 – Библиотека блоков *Simulink Library Browser*

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

- *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 маршрутизации сигналов;
- *Signal Routing* – блоки маршрутизации сигналов;
- *Sinks* – блоки приема и отображения сигналов;
- *Sources* – блоки источников сигнала;
- *User-Defined Function* – функции, определяемые пользователем.

### 3.2 Описание используемых блоков библиотеки Simulink

Ниже описаны основные блоки базовых разделов библиотеки *Simulink*, используемые в функциональной схеме MSK-модема:



Random Number

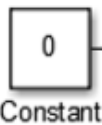
*Random Number* – блок источника случайного дискретного сигнала с нормальным распределением. Назначение: формирование случайного сигнала с нормальным распределением уровня сигнала. Параметры блока: *Mean* - среднее значение сигнала, *Variance* - дисперсия (среднеквадратическое отклонение), *Initial seed*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



Sign

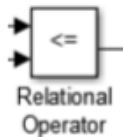
*Sign* – блок определения знака сигнала. Назначение: определяет знак входного сигнала, при этом, если  $x$  – входной сигнал, то сигнал на выходе определяется выражением:

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



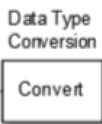
Constant

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



Relational Operator

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



Data Type Conversion  
Convert

*Data Type Conversion* – блок преобразования типа сигнала. Назначение: блок преобразует тип входного сигнала. Параметры: *Data type* – тип данных выходного сигнала. Может принимать значения (выбираются из списка): *auto*, *double*, *single*, *int8*, *int16*, *int32*, *uint8*, *uint16*, *uint32* и *boolean*. *Saturate on integer overflow* (флажок) – подавляет переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.



Product

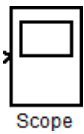
*Product* – блок умножения и деления. Назначение: вычисление произведения текущих значений сигналов. Параметры блока: *Number of inputs* – количество входов, может задаваться как число или как список знаков. В списке знаков можно использовать знаки: \* – умножить и / – разделить. *Multiplication* – способ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



Sine Wave

*Sine Wave* – блок источника синусоидального сигнала. Назначение:

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



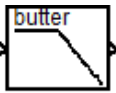
Scope

*Scope* – блок осциллографа. Назначение: построение графиков исследуемых сигналов как функций времени. Открытие окна осциллографа производится двойным щелчком ЛКМ на пиктограмме блока. В случае векторного сигнала каждая компонента вектора отображается отдельным цветом. Настройка окна осциллографа выполняется с помощью панелей инструментов, позволяющих: осуществить печать содержимого окна осциллографа; установить параметры, в частности, *Number of axes* – число входов осциллографа, *Time range* – отображаемый временной интервал и другие; изменить масштабы графиков; установить и сохранить настройки; перевести в плавающий режим и так далее.



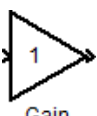
Add

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



Analog Filter Design

*Analog Filter Design* – блок аналогового фильтра заданного метода проектирования и типа из подраздела *Filter Design*; подраздела *Filtering*, раздела *DSP Blockset*. Назначение: аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. Параметры блока: *Design method* – метод проектирования, выбирается из списка: *Butterworth* – фильтр Баттерворта; *Chebyshev I* – фильтр Чебышева 1-го рода; *Chebyshev II* – фильтр Чебышева 2-го рода; *Elliptic* – фильтр эллиптический; *Bessel* – фильтр Бесселя. *Filter type* – тип фильтра, выбирается из списка: *Lowpass* – нижних частот; *Highpass* – верхних частот; *Bandpass* – полосно-пропускающий; *Bandstop* – полосно-заграждающий. Далее для каждого метода проектирования и типа фильтра выдается свой список параметров. Так для фильтра Баттерворта типа нижних частот параметрами являются: *Filter order* – порядок фильтра; *Passband edge frequency (rads/sec)* – нижняя граничная частота (радиан в секунду). Для других методик проектирования и типов фильтров определяемые параметры очевидны.



Gain

*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.

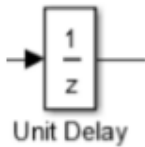


Zero-Order Hold

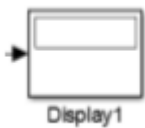
*Zero-Order Hold* – экстраполятор нулевого порядка. Назначение:



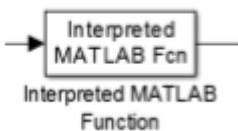
экстраполяция входного сигнала на интервале дискретизации. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания интервала дискретизации. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге дискретизации. Параметры блока: *Sample time* – такт дискретности. Блок экстраполятора нулевого порядка может использоваться также для согласования работы дискретных блоков, имеющих разные такты дискретности.



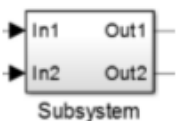
*Unit delay* – блок единичной дискретной задержки. Назначение: выполняет задержку дискретного сигнала на заданный шаг модельного времени. Параметры блока: *Initial conditions* – начальное значение выходного сигнала; *Sample time* – шаг модельного времени.



*Display* – блок цифрового дисплея. Назначение: отображает значение сигнала в виде числа. Параметры: *Format* – формат отображения данных. Параметр *Format* может принимать следующие значения: *short* – 5 значащих десятичных цифр, *long* – 15 значащих десятичных цифр, *short\_e* – 5 значащих десятичных цифр и 3 символа степени десяти, *long\_e* – 15 значащих десятичных цифр и 3 символа степени десяти, *bank* – "денежный" формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа; *Decimation* – кратность отображения входного сигнала, при *Decimation* = 1 отображается каждое значение входного сигнала, при *Decimation* = 2 отображается каждое второе значение, при *Decimation* = 3 – каждое третье значение и т.д; *Sample time* – шаг модельного времени. Определяет дискретность отображения данных; *Floating display* (флажок) – перевод блока в "свободный" режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения выполняется щелчком ЛВМ на соответствующей линии связи. В этом режиме для параметра расчета *Signal storage reuse* должно быть установлено значение off (вкладка *Advanced* в окне диалога *Simulation parameters...*).



*MatLab Fcn* – блок задания функции. Назначение: задает выражение в стиле языка программирования *MatLab*. Параметры: *MatLab function* – Выражение на языке *MatLab*. *Output dimensions* – размерность выходного сигнала. Значение параметра минус 1 предписывает блоку определять размерность автоматически. *Output signal type* – тип выходного сигнала. Выбирается из списка: *real* – действительный сигнал, *complex* – комплексный сигнал, *auto* – автоматическое определение типа сигнала; *Collapse 2-D results to 1-D* – преобразование двумерного выходного сигнала к одномерному.



*Subsystem* – виртуальная и монолитная подсистемы. Доступ к окну параметров подсистемы осуществляется через меню *Edit* командой *Block Parameters*. Параметры: *Show port labels* – показать метки портов, *Treat as atomic unit* (флажок) – считать подсистему монолитной. Таким образом, блоки виртуальной и монолитной подсистем – это один и тот же блок, отличающийся значением данного параметра. *Access* – доступность подсистемы для изменений. Выбирается из списка: *ReadWrite* – пользователь может открывать и изменять подсистему, *ReadOnly* – пользователь может открывать подсистему только для просмотра, *NoReadOrWrite* – пользователь не может открывать и изменять подсистему; *Name of error callback function* – имя функции используемой для обработки ошибок возникающих в данной подсистеме.

## 4 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

### Исходные данные.

Для ортогонального разнесения несущих при длительности бита информационного потока  $T_b$ , в функции **dibit-frequency** несущие прописать равными  $10\pi$ ,  $12\pi$ ,  $14\pi$ ,  $16\pi$ ; для обеспечения широкополосности шума в генераторе шума параметр **sample time** задать равным 0,0125; в корреляционном приёмнике для блока *Zero-Order Hold* параметр **sample time** равен  $1/64$ , в блоке *Buffer* параметр **Output buffer size** – 64; значение константы, являющейся порогом для корреляторов составляет 16; для корректной работы блока *Averaging Power Spectral Density* следующие параметры: **Length of buffer** – 128, **Number of points for fft** – 512, **Plot after how many points** – 64, **Sample time** – 0,025.

Для неортогонального разнесения несущих при длительности бита информационного потока  $T_b$ , в функции **dibit-frequency** несущие прописать равными  $10\pi$ ,  $11\pi$ ,  $12\pi$ ,  $13\pi$ ; для обеспечения широкополосности шума в генераторе шума параметр **sample time** задать равным 0,0125; в корреляционном приёмнике для блока *Zero-Order Hold* параметр **sample time** равен  $1/64$ , в блоке *Buffer* параметр **Output buffer size** – 64; значение константы, являющейся порогом для корреляторов составляет 16; для корректной работы блока *Averaging Power Spectral Density* следующие параметры: **Length of buffer** – 128, **Number of points for fft** – 512, **Plot after how many points** – 64, **Sample time** – 0,025.

### Экспериментальное задание.

1. Собрать Sim-модель 4FSK модема и написать Matlab-функции, показанные на рисунках 2.1 и 2.2. Параметры модели задать для ортогонального разнесения несущих ( $10\pi$ ,  $12\pi$ ,  $14\pi$ ,  $16\pi$ ).

2. Определить при каком уровне шума появляется вероятность появления битовой ошибки равная 0,001. По полученным данным рассчитать SNR (в dB). Длину информационного потока установить порядка 1000 битов. Данные исследования занести в отчет.

3. Изменить разнос частот на неортогональный набор ( $10\pi$ ,  $11\pi$ ,  $12\pi$ ,  $13\pi$ ).

4. Повторить пункт 2 для неортогонального разнесения несущих.

5. Сравнить результаты и сделать выводы.

7. Составить отчёт по проделанной работе.

## 5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Принцип частотной манипуляции.
2. Реализация частотной манипуляции с помощью независимых генераторов.
3. Реализация частотной манипуляции с помощью ГУНа.
4. Ортогональный и неортогональный разносы несущих при частотной манипуляции.
5. Понятие модуляционного символа.
6. Возможные способы приема частотно-модулированного сигнала.
7. Опишите принцип приема на основе прямого преобразования (синхронного детектирования).
8. Опишите принцип корреляционного приема.
9. Опишите методику измерения отношения сигнал/шум.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Галкин, В.А. Цифровая мобильная радиосвязь. Учебное пособие для вузов / В.А. Галкин – М.: Горячая линия–Телеком, 2007.– 432 с.
2. Завьялов, С.В. Повышение спектральной эффективности многочастотных неортогональных сигналов. Диссертация на соискание ученой степени кандидата технических наук. – СПб.: ФГАОУ ВО «Санкт-Петербургский государственный политехнический университет», 2015.– 161 с.
3. Степанов, А.В. Синхронный детектор [Электронный ресурс] Практикум кафедры физики колебаний – Москва: Московский государственный университет имени М.В. Ломоносова, 1997. Режим доступа: <https://lektsii.net/4-96803.html> (дата обращения 25.01.2022).
4. Гультияев, А.К. MatLab 5.3. Имитационное моделирование в среде Windows: Практическое пособие / А.К. Гультияев – СПб.: КОРОНА принт, 2001.– 400 с.
5. Черных, И.В. Simulink: среда создания инженерных приложений. / Под общ. ред. В.Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2003.– 496 с.
6. Дьяконов, В.П. MatLab 6.5 SP1/7 + Simulink 5/6. Основы применения. Сер. Библиотека профессионала / В. П. Дьяконов - М.: СОЛОН-Пресс, 2005.– 800 с.
7. Дьяконов, В.П. MatLab 6.5 SP1/7 + Simulink 5/6 в математике и моделировании. Сер. Библиотека профессионала / В. П. Дьяконов - М.: СОЛОН-Пресс, 2005.– 576 с.