

Министерство науки и высшего образования Российской Федерации

Томский государственный университет  
систем управления и радиоэлектроники

Кологривов В. А.  
Куценко М.А.

**ИССЛЕДОВАНИЕ ПОДШУМОВОЙ ТЕХНОЛОГИИ ПЕРЕДАЧИ НА ОСНОВЕ QPSK  
МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ**

Методические указания по лабораторной работе в среде функционального моделирования  
Simulink системы MatLab для студентов радиотехнических специальностей

Томск 2023

УДК 621.391.6  
ББК 32.884.1  
К 61

**Рецензент:**

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

**Кологривов В. А., Куценко М.А.**

К 61 Исследование подшумовой технологии передачи на основе **QPSK** модуляции и комплекснозначных последовательностей: методические указания по лабораторной работе для студентов радиотехнических специальностей / Кологривов В. А., Куценко М.А. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2023 – 25 с.

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

Лабораторная работа «Исследование подшумовой технологии передачи на основе **QPSK** модуляции и комплекснозначных последовательностей» с использованием пакета функционального моделирования *Simulink* системы для инженерных и научных расчетов *MatLab*.

Работа «Исследование подшумовой технологии передачи на основе **QPSK** модуляции и комплекснозначных последовательностей» относится к циклу лабораторных работ по разделу «Скрытые системы передачи», входящему в дисциплины радиотехнических специальностей.

В описании сформулирована цель лабораторной работы, приведены краткие теоретические сведения о подшумовой технологии передачи на основе **QPSK** модуляции и комплекснозначных последовательностей, 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 № 7 от 22.02.23

УДК 621.391.6  
ББК 32.884.1

© Кологривов В. А., Куценко М.А., 2023  
© Томск. гос. ун-т систем упр. и радиоэлектроники, 2023

## ОГЛАВЛЕНИЕ

<b>1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О ПОДШУМОВОЙ ТЕХНОЛОГИИ ПЕРЕДАЧИ НА ОСНОВЕ QPSK МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ .....</b>	<b>4</b>
1.1 Краткие теоретические сведения .....	4
1.2 Математическое описание корреляционных свойств комплекснозначной последовательности...	5
<b>2 ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ ПОДШУМОВОЙ ПЕРЕДАЧИ НА ОСНОВЕ QPSK МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ .....</b>	<b>6</b>
<b>3 ПРИНЦИП РАБОТЫ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ ПОДШУМОВОЙ ПЕРЕДАЧИ НА ОСНОВЕ QPSK МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ .....</b>	<b>12</b>
<b>4 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK И ИСПОЛЬЗУЕМЫХ БЛОКОВ .....</b>	<b>16</b>
4.1 Запуск и работа с пакетом <i>Simulink</i> .....	16
4.2 Описание используемых блоков библиотеки <i>Simulink</i> .....	18
<b>5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ.....</b>	<b>23</b>
<b>6 КОНТРОЛЬНЫЕ ВОПРОСЫ.....</b>	<b>24</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>25</b>

# 1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О ПОДШУМОВОЙ ТЕХНОЛОГИИ ПЕРЕДАЧИ НА ОСНОВЕ QPSK МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

**Цель работы:** исследование подшумовой технологии передачи на основе использования *QPSK* модуляции и комплекснозначных матриц и векторов – помехоустойчивости, энергетической и спектральной эффективности, с использованием пакета функционального моделирования *Simulink*. Моделирование производится в относительных масштабах времени и частоты.

## 1.1 Краткие теоретические сведения

Согласно теоретическим представлениям использование подшумовой передачи широкополосных сигналов на основе комплекснозначных последовательностей с компактной автокорреляционной функцией позволяет добиться скрытности [1-3]. За основу взяты комплекснозначные матрицы  $G1$  и  $G3$  [1] размером  $3 \times 4$  с компактной  $ACF$ , которые для удобства реализации построчно трансформированы в вектор.

При модуляции несущего гармонического колебания текущей информационной последовательностью реализуется достаточно узкополосный спектр, который в условиях многолучевого распространения сильно поражается интерференцией лучей распространения пришедших в точку приёма с разной задержкой. В связи с этим пришли к выводу о необходимости расширении спектра.

Для расширения спектра необходимо увеличить базу информационного сигнала (произведение полосы сигнала  $B$  на длительность  $T$ ). Для обычных информационных сигналов  $B \cdot T \sim 1$  и для увеличения базы используют, например технологию расширения спектра повторяющейся *pin*- или чип-последовательностью коротких импульсов имеющей псевдослучайный статистический закон распределения. В процессе расширения спектра *pin*-последовательность, имеющая скажем гауссовский характер распределения и длительностью с информационный символ или бит, перемножается с информационной последовательностью.

В результате ширина спектра  $B$  будет определяться длительностью импульсов чип-последовательности вписываемой в символ или бит. При этом база сигнала и ширина спектра увеличится в число чипов вписываемых в символ или бит. Теперь интерференционные искажения, имеющие узкополосный характер, будут поражать лишь часть спектра сигнала и их последствия могут быть преодолены помехоустойчивым кодированием.

Теперь, если бы удалось построить набор статистически независимых псевдослучайных последовательностей и использовать их в соседних каналах передачи, то это позволило бы строить многоканальные системы передачи. В этом и состоит суть известной технологии *CDMA* (кодового разделения каналов).

Существует несколько подходов к синтезу псевдослучайных последовательностей. Один из подходов основан на использовании так называемых регистров сдвига с обратными связями (ОС) (*LFSR*). Псевдослучайные последовательности реализуемые структурами *LFSR* называют  $M$ -последовательностями. В свою очередь структура регистров сдвига с ОС может быть описана полиномами и если структура регистра сдвига соответствует так называемому примитивному полиному, то регистр генерирует неповторяющуюся псевдослучайную последовательность максимальной длины определяемой выражением  $L = (2^n - 1)$ , где  $n$ - порядок примитивного полинома. Чем выше порядок полинома, тем больше среди них примитивных. Число примитивных полиномов  $n$ -го порядка можно определить, используя функцию Эйлера.

Проблема построения псевдослучайных последовательностей для построения многоканальных систем с кодовым разделением каналов предполагает одновременное достижение двух условий – автокорреляционные характеристики каждой из последовательностей должны быть компактны, т.е. иметь явно выраженный максимум без боковых лепестков, взаимокорреляционные характеристики любой пары последовательностей не должны иметь значительных лепестков. Перечисленные условия делают проблему синтеза набора псевдослучайных последовательностей нетривиальной.

Авторы работ [1-3] доказывают, что переход от вещественных псевдослучайных последовательностей к комплексным способствует нахождению последовательностей с более компактной АСФ.

## 1.2 Математическое описание корреляционных свойств комплекснозначной последовательности

Вектор комплекснозначной последовательности полученный построчным преобразованием из матрицы  $GI$  [1] имеет вид:

$$v = [j, (1+j)/\sqrt{2}, (1+j)/\sqrt{2}, -j, (1-j)/\sqrt{2}, -(1+j)/\sqrt{2}, -1, -j, -1, -(1+j)/\sqrt{2}, 1, 0].$$

Установив правило построения комплекснозначной последовательности из комплексных значений по модулю равных  $1$ , построили комплекснозначную последовательность длиной  $16$ .

$$v = [1, -1, j, -j, (1+j)/\sqrt{2}, (1-j)/\sqrt{2}, (-1+j)/\sqrt{2}, -(1+j)/\sqrt{2}, (1+j \cdot \sqrt{2})/\sqrt{3}, (1-j \cdot \sqrt{2})/\sqrt{3}, (-1+j \cdot \sqrt{2})/\sqrt{3}, -(1+j \cdot \sqrt{2})/\sqrt{3}, (\sqrt{2}+j)/\sqrt{3}, (\sqrt{2}-j)/\sqrt{3}, (-\sqrt{2}+j)/\sqrt{3}, -(\sqrt{2}+j)/\sqrt{3}].$$

С целью оценки автокорреляционных свойств последовательности вычисляем скалярное произведение вектора с транспонированным вектором и получим  $v * v^t = 16$ . Полученное значение чисто вещественно и практически равно длине последовательности, что свидетельствует о хороших автокорреляционных свойствах последовательности.

Именно эту комплекснозначную чиповую последовательность и будем вписывать в каждый бит информационной последовательности в качестве расширяющей pin-последовательности.

## 2 ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ ПОДШУМОВОЙ ПЕРЕДАЧИ НА ОСНОВЕ QPSK МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

На рис. 2.1 приведена функциональная модель системы подшумовой передачи широкополосных сигналов на основе **QPSK** модуляции и комплекснозначных последовательностей.

**Описание модели.** Источник биполярной информационной последовательности реализован на основе генератора псевдослучайной последовательности с гауссовским распределением **Random Number** и блока двухстороннего ограничителя на основе функции **sign(x)**. Длина бита информационной последовательности задаётся параметром **Sample Time=1** генератора **Random Number**. Так как на входе фазового кодера должна быть одноволярная информационная последовательность, то с помощью блока оператора сравнения **Relational Operator** организуем это преобразование. Блок **Data Type Conversion** необходим для преобразования типа из **boolean** в тип **double**.

Комплекснозначная последовательность реализована с применением генератора повторяющейся последовательности **Repeating Sequence** в качестве модельного времени длиной **16**, а сама комплекснозначная последовательность подавалась на выход генератора с помощью функции **ccc\_seq**, представленной скрин файлом на рис. 2.2.

Функциональная модель подсистемы фазового кодера **QPSK** модулятора приведена на рис. 2.3. На основе блоков **Unit Delay** и **Mux** реализовано преобразование информационного битового потока в дибиты. Блок **Zero Order Hold** исключает участие текущего бита в разных дибитах. Программная реализация сопоставления разным дибитам фазовых состояний  $\varphi_k$  осуществляется **MatLab** функцией **dibit\_phase** (см. рис. 2.4.). Блоки **cos** и **sin** вычисляют значения  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ .

Модулятор биполярной информационной последовательности расширяющей комплекснозначной последовательностью выполнен на основе блоков умножителей **Product** и блока **Sum**. На первые входы умножителей поступают вычисленные значения  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ , а на вторые входы умножителей подаются соответственно действительные и мнимые составляющие комплекснозначной цифровой последовательности.

Квадратурный модулятор несущего колебания расширенной информационной последовательности также выполнен на основе блоков **Product** и блока **Sum**. На первые входы умножителей подаем разветвленный сигнал с выхода модулятора информационной последовательности расширяющей комплексной последовательностью, а на вторые входы умножителей с блоков **Sine Wave** подаются несущие колебания  $\cos(\omega_0 t)$  и  $\sin(\omega_0 t)$  с частотой  $\omega_0 = 75 * \pi$  радиан.

Модель канала распространения реализована на основе блока **Sum**, на второй вход которого поступает шумовая псевдослучайная последовательность, выполненная на основе генератора **Random Number** с гауссовским распределением. Параметр генератора **Sample Time** позволяет реализовать необходимую широкополосность шумов канала распространения, а параметр **Variance** регулировать мощность шумов в процессе измерения помехоустойчивости (зависимости вероятности битовой ошибки от соотношения сигнал/шум **SNR**).

Квадратурный демодулятор по аналогии с модулятором реализован с использованием блоков **Product** и блоков **ФНЧ (Analog Filter Design)** с граничными частотами  $\omega_g = 16 * \pi$  радиан/сек. Для компенсации коэффициента **1/2** тригонометрических преобразований на выходах **ФНЧ** использованы блоки **Gain = sqrt(2)**.

## Test\_PodShum\_Pered\_Of\_Matr\_ACF\_3\_5 QPSK

SNRin=S/N=0.1483(0.481-0.1483)=0.4457 (-3.5091 dB) Nc=3.2 dt=1/16 n\_buff=32 n=1000 bez RF dF=16\*pi n\_err=1 ACF\_3\_5 QPSK

SNRout=S/N=57.75/(66-57.75)=7.000 (8.4510 dB) Nc=3.2 dt=1/16 n\_buff=32 n=1000 bez RF dF=16\*pi n\_err=1 ACF\_3\_5 QPSK

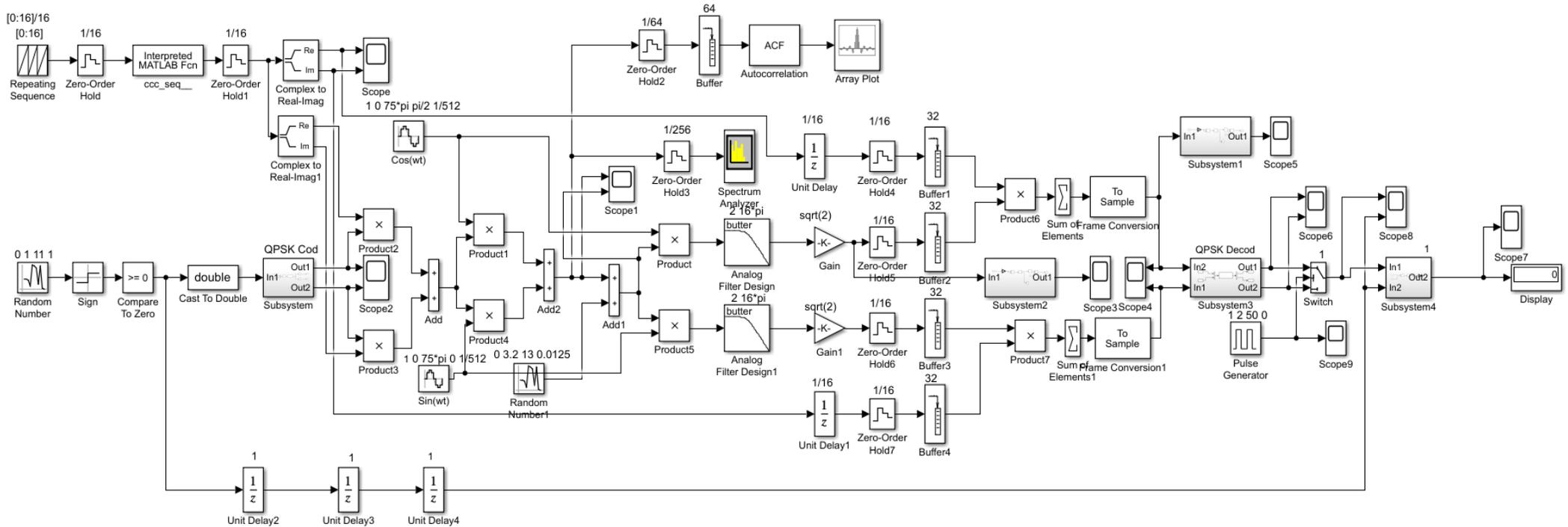


Рисунок 2.1 – Функциональная модель системы подшумовой передачи широкополосных сигналов на основе *QPSK* модуляции и комплекснозначных последовательности

```

function c=ccc_seq__(x)
% function c=ccc_seq__(x);
% Функция генерации заданной дискретной
% (комплексной) последовательности видеоимпульсов
% x- входное значение текущего модельного времени
% c- текущее значение выходной комплексной импульсной последовательности
% здесь вектор v получен из матр Q1(3*4) из статьи построчным развертыванием в
% вектор v

v=[1          -1          j          -j          (1+j)/sqrt(2)  (1-j)/sqrt(2) ...
  (-1+j)/sqrt(2) (-1+j)/sqrt(2) (1+j*sqrt(2))/sqrt(3) (1-j*sqrt(2))/sqrt(3) (-1+j*sqrt(2))/sqrt(3) -
  (1+j*sqrt(2))/sqrt(3) ...
  (sqrt(2)+j)/sqrt(3) (sqrt(2)-j)/sqrt(3) (-sqrt(2)+j)/sqrt(3) -(-sqrt(2)+j)/sqrt(3)];
m=round(1+x); % вычисление текущей координаты вектора
c=v(m);      % текущее значение выходной последовательности

```

Рисунок 1.2 – Скрин-файл функции генерации псевдослучайной комплекснозначной последовательности

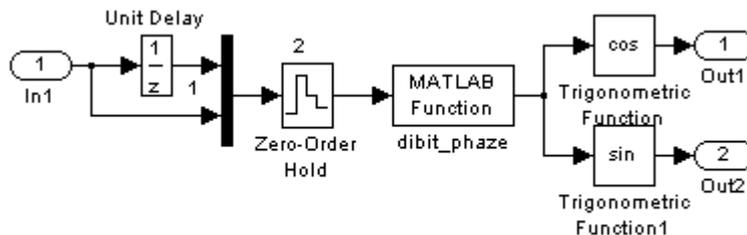


Рисунок 2.3 – Функциональная модель фазового кодера *QPSK* модулятора

```

function fi=dibit_phaze(x);

```

```

if x==[0; 0]; fi=pi/4; end;
if x==[0; 1]; fi=3*pi/4; end;
if x==[1; 1]; fi=-3*pi/4; end;
if x==[1; 0]; fi=-pi/4; end;

```

Рисунок 2.4 – Скрин файл *MatLab* функция *dibit\_phaze*

Корреляционные приёмники квадратурных составляющих выполнены на блоках экстраполяторов нулевого порядка *Zero Order Hold* и блоков *Buffer*, которые накапливают в течение дибита 32 отсчета повторяющейся опорной комплекснозначной последовательности и отсчеты принятых в шумах квадратурных составляющих информационных дибитов модулированных комплекснозначной последовательностью. Накопленные отсчеты, как вектора скалярно перемножаются и суммируются блоками *Product* и *Sum*, и через блоки *Frame Status* подаются на входы подсистемы фазового декодера (см. рис. 2.5).

Блок *Mux* на входе формирует вектор принятых квадратурных составляющих.

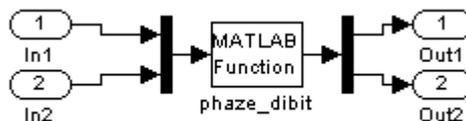


Рисунок 2.5 – Функциональная модель подсистемы фазового декодера

Программная реализация фазового декодера - сопоставления разным наборам принятых квадратурных проекций соответствующего дибита осуществляется *MatLab* функцией *dibit\_phase* (см. рис. 2.6.). Выходной блок DeMux передает декодированный дибит на схему преобразования параллельного представления в последовательный поток.

```
function y=phaze_dibit(x);

if ((x(1)==0) & (x(2)==0)); y=[0 0]; end;
if ((x(1)>=0) & (x(2)>=0)); y=[0 0]; end;
if ((x(1)<0) & (x(2)>=0)); y=[0 1]; end;
if ((x(1)<0) & (x(2)<0)); y=[1 1]; end;
if ((x(1)>=0) & (x(2)<0)); y=[1 0]; end;
```

Рисунок 2.6 – Скрин файл *MatLab* функция *phase\_dibit*

Схема преобразования параллельного представления принятого дибита в последовательный поток реализована на блоках *Switch* и *Pulse Generator*. Используя тот факт, что передаваемый дибит имеет длительность вдвое превышающую длительность бита, удастся с помощью блока *Pulse Generator* переключать входы блока *Switch* и тем самым преобразовывать параллельный поток дибитов в последовательный поток битов.

Подсистема *Subsystem Calc Err*, блоки *Display* и *Scope* призваны отображать число ошибок передачи и фиксации момента их возникновения.

Подсистема *Subsystem Measuring Power* и блок *Scope* призваны вычислять и отображать изменение уровня мощности сигнала или смеси сигнал+шум в процессе измерения помехоустойчивости. Точкой подключения измерителя мощности выбран выход корреляционного приемника квадратурной составляющей *QPSK* модулированного сигнала. Блок *Spectrum Scope* призван отображать спектры сигнальных и шумовых потоков.

**Корреляционный приём.** 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  $\tau = 0$  и может содержать боковые пики. Функция корреляции непрерывного сигнала определяется соотношением:

$$R(\tau) = \int x(\tau) * y(t - \tau) d\tau,$$

где  $y(t) = x(t) + n(t)$ ;  
 $x(t)$  – полезный сигнал;  
 $n(t)$  – АБГШ.

При полном совпадении сигналов будет получен положительный максимум значения функции корреляции. Если сигналы находились в противофазе, то максимум станет отрицательным. При отсутствии похожести 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. Для дискретного и цифрового сигнала функция корреляции определяется через сумму произведения функций отсчетов

$$R(m) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{\infty} x(n) \cdot y(n - m).$$

**Детектор ошибок.** Модель подсистемы детектора ошибок *Subsystem Calc Err* приведена на рис. 2.7.

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 поступает сигнал с генератора входной информационной последовательности с необходимой задержкой. К выходу подсистемы присоединены дисплей и осциллограф, на дисплее в случае наличия ошибок, будет указано их количество, а на осциллографе можно фиксировать время их возникновения.

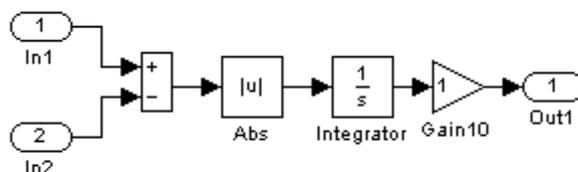


Рисунок 2.7 - Модель подсистемы детектора ошибок

**Измеритель мощности.** Модель подсистемы измерителя мощности *Subsystem Measuring Power* приведена на рис. 2.8.

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.

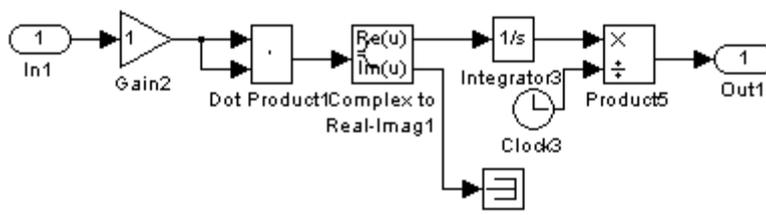


Рисунок 2.8 - Модель подсистемы измерителя мощности

**Модель коррелятора.** Корреляционный приемник реализован следующим образом

(см. рис. 2.9):

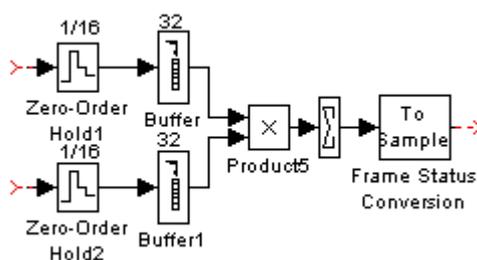


Рисунок 2.9 – Реализация корреляционного приемника

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

Т.к. используются дискретные (цифровые) сигналы, то значение функции корреляции вычисляется через сумму произведений отсчетов принятого в шумах сигнала и отсчетов опорного генератора комплекснозначной последовательности с компактной *ACF*. Модель коррелятора содержит два плеча накопления отсчетов. В каждом плече с помощью блоков *Zero-Order Hold* сигнал дискретизируется на отсчеты в течении бита (в данном случае на *16*) и накапливается в блоках *Buffer* в течении принимаемого дибита (т.е. *32* отсчета). Далее накопленные отсчеты попарно перемножаются в блоке *Product*. Затем вычисляется сумма этих произведений блоком *Sum*, т.е. реализуется скалярное произведение векторов накопленных отсчетов.

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

*SNR* рассчитывается по формуле, представленной ниже:

$$SNR = S / (SN - S)$$

Результаты измерений фиксируется в *дБ*.

При большом числе испытаний, частота появления битовых ошибок, т.е. отношение числа ошибок к общему числу битов, стремится к вероятности битовых ошибок *P<sub>b</sub>*.

**Рекомендации** для проведения исследования помехоустойчивости. При отключенном генераторе шумов канала распространения на выходе преобразователя измеряется уровень мощности сигнала *S*. Подключив генератор шумов канала распространения и меняя параметр *Variance* добиваемся скажем *1*-ой ошибки на *1000* битов, получаем одну точку водопадоподобной кривой. Далее, постепенно увеличивая мощность шумов с помощью параметра *Variance* добиваемся по очереди *3*-х, *5*-ти и *8*-ми ошибок, получаем *4*-е точки водопадоподобной кривой, которую строим в полулогарифмическом масштабе с помощью *MatLab*- функции *semilogy*.

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

### 3 ПРИНЦИП РАБОТЫ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ ПОДШУМОВОЙ ПЕРЕДАЧИ НА ОСНОВЕ QPSK МОДУЛЯЦИИ И КОМПЛЕКСНОЗНАЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Опишем принцип работы функциональной *Sim*-модели модема подшумовой передачи на основе *QPSK* модуляции и комплекснозначных последовательностей представленной на рис. 2.1. Источник на основе генератора *Random Number*, блока *Sign* и блока сравнения *Relational Operator* формирует псевдослучайный информационный однополярный битовый поток.

Комплекснозначная последовательность формируется генератором повторяющейся последовательности *Repeating Sequence* в качестве модельного времени длиной *16*, а сама комплекснозначная последовательность подается на выход генератора с помощью *MatLab*-функции.

Подсистема фазового кодера *QPSK* модулятора осуществляет преобразование информационного битового потока в дибиты, сопоставляет разным дибитам фазовые состояния  $\varphi_k$  и вычисляет значения  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ .

Модулятор однополярной информационной последовательности расширяющей комплекснозначной последовательностью выполнен на основе блоков умножителей *Product* и блока *Sum*. На умножители, с одной стороны, поступают вычисленные значения  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ , а с другой стороны соответственно действительные и мнимые составляющие комплекснозначной чиповой последовательности.

Квадратурный модулятор несущего колебания расширенной информационной последовательности также выполнен на основе блоков *Product* и блока *Sum*. На входы умножителей, с одной стороны, подаем разветвленный сигнал с выхода модулятора информационной последовательности расширяющей комплексной последовательностью, а с другой стороны с блоков *Sine Wave* подаются несущие колебания  $\cos(\omega_0 t)$  и  $\sin(\omega_0 t)$  с частотой  $\omega_0 = 75 * \pi$  радиан.

Модель канала распространения реализована на основе блока *Sum*, на второй вход которого поступает шумовая псевдослучайная последовательность реализуемая генератором *Random Number* с гауссовским распределением.

Квадратурный демодулятор по аналогии с модулятором реализован с использованием блоков *Product* и блоков *ФНЧ (Analog Filter Design)* с граничными частотами  $\omega_g = 16 * \pi$  радиан/сек и блоков *Gain = sqrt(2)*.

Корреляционный приёмники квадратурных составляющих на блоках экстраполяторов нулевого порядка *Zero Order Hold* и блоков *Buffer* накапливают в течение дибита *32* отсчета повторяющейся опорной комплекснозначной последовательности и отсчеты принятых в шумах квадратурных составляющих информационных дибитов модулированных комплекснозначной последовательностью. Накопленные отсчеты скалярно перемножаются и суммируются блоками *Product* и *Sum*, и через блоки *Frame Status* подаются на входы подсистемы фазового декодера.

Подсистема фазового декодера сопоставляет разным наборам принятых квадратурных проекций соответствующего дибита.

Схема преобразования параллельного представления принятого дибита в последовательный поток реализована на блоках *Switch* и *Pulse Generator*.

Подсистема *Subsystem Calc Err*, блоки *Display* и *Scope* призваны отображать число ошибок передачи и фиксации момента их возникновения.

Подсистема *Subsystem Measuring Power* и блок *Scope* призваны вычислять и отображать изменение уровня мощности сигнала или смеси сигнал+шум в процессе измерения помехоустойчивости. Точка подключения измерителя мощности выбран выход корреляционного приемника квадратурной составляющей *QPSK* модулированного сигнала.

Блок *Spectrum Scope* призван отображать спектры сигнальных и шумовых потоков.

Измерение отношения сигнал/шум *SNR* может быть осуществлено как на входе так и на выходе коррелятора, что позволяет оценить влияние коррелятора на *SNR* в зависимости от конкретных условий.

Для иллюстрации работы модема под шумовой передачей на основе *QPSK* модуляции и комплекснозначных последовательностей на рис. 3.1 приведены фрагменты осциллограмм повторяющейся комплекснозначной расширяющей последовательности.

На рисунке 3.2 приведены фрагменты осциллограмм несущего колебания модулированного информационной последовательностью расширенной комплекснозначной последовательностью.

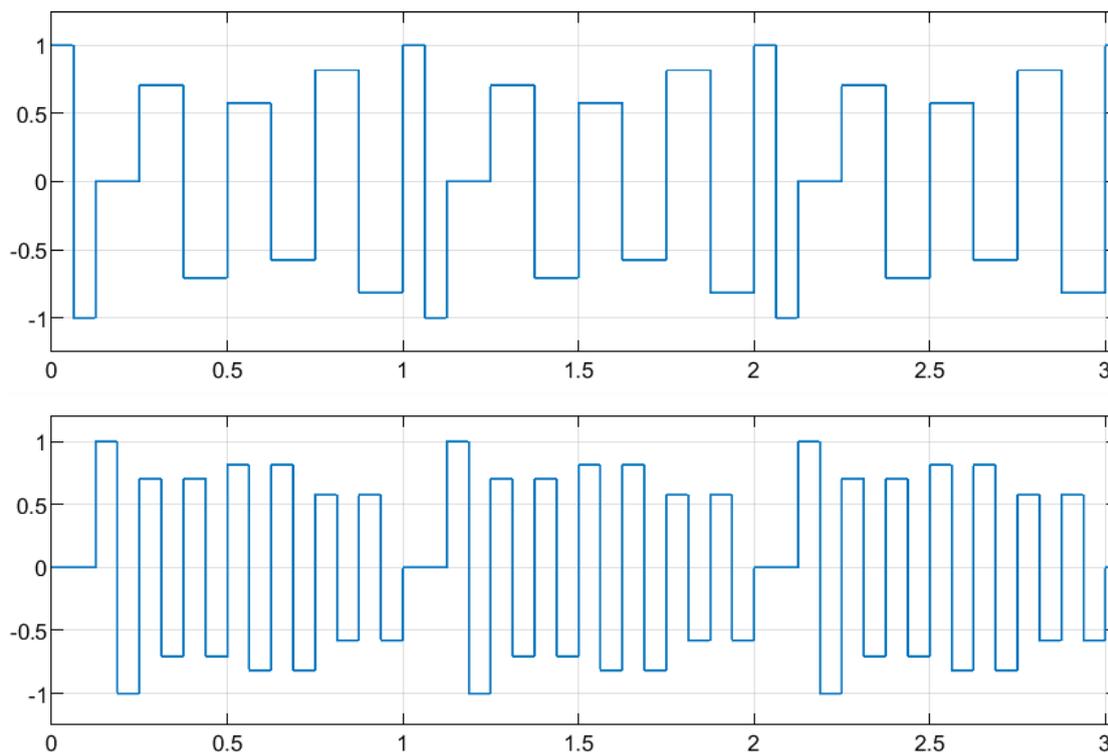


Рисунок 3.1 - Фрагменты осциллограмм повторяющейся комплекснозначной расширяющей последовательности (действительная и мнимая части)

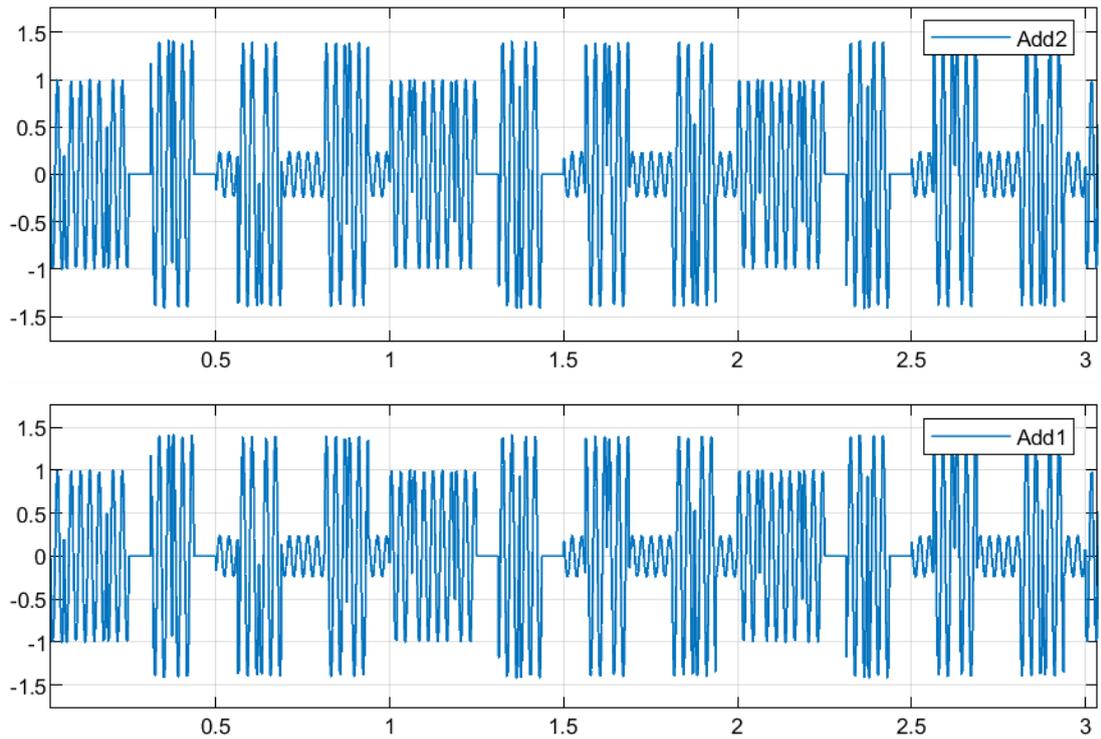


Рисунок 3.2 - Фрагмент осциллограммы несущего колебания модулированного информационной последовательностью расширенной комплекснозначной последовательностью

На рисунке 3.3 приведена спектрограмма сигнала модема подшумовой передачи на основе *QPSK* модуляции и комплекснозначных последовательностей в отсутствие шумов канала распространения.

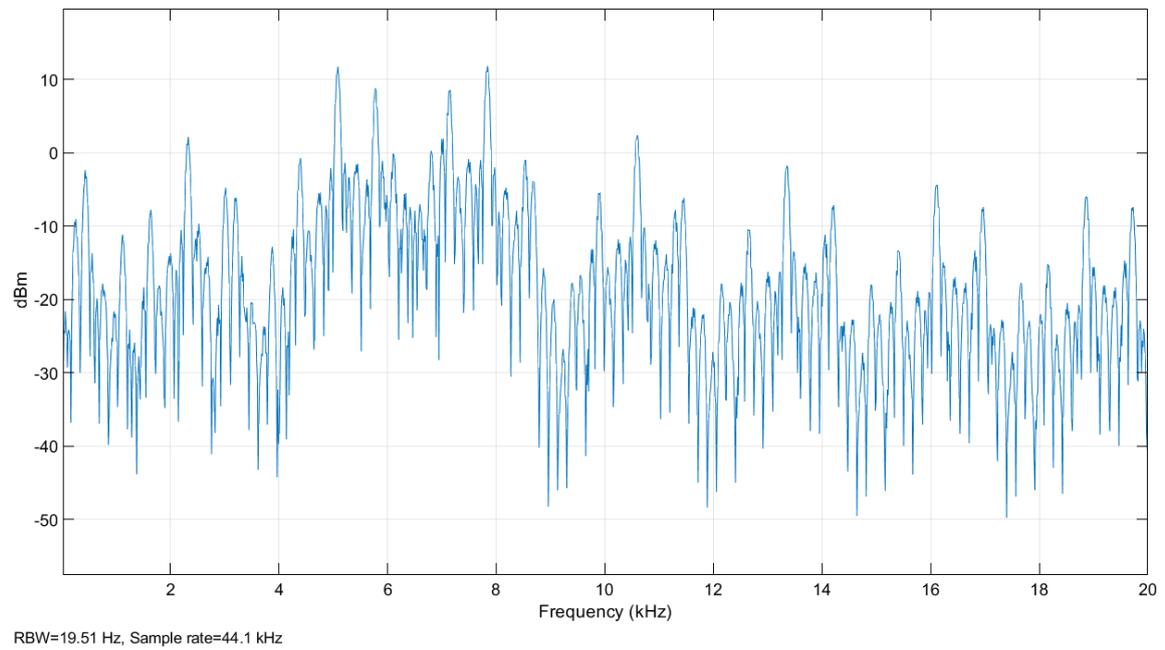


Рисунок 3.3 - Спектрограмма сигнала модема подшумовой передачи на основе *QPSK* модуляции и комплекснозначных последовательностей в отсутствие шумов канала распространения

На рисунке 3.4 приведены фрагменты осциллограмм на выходах корреляторов в отсутствие шумов канала распространения.

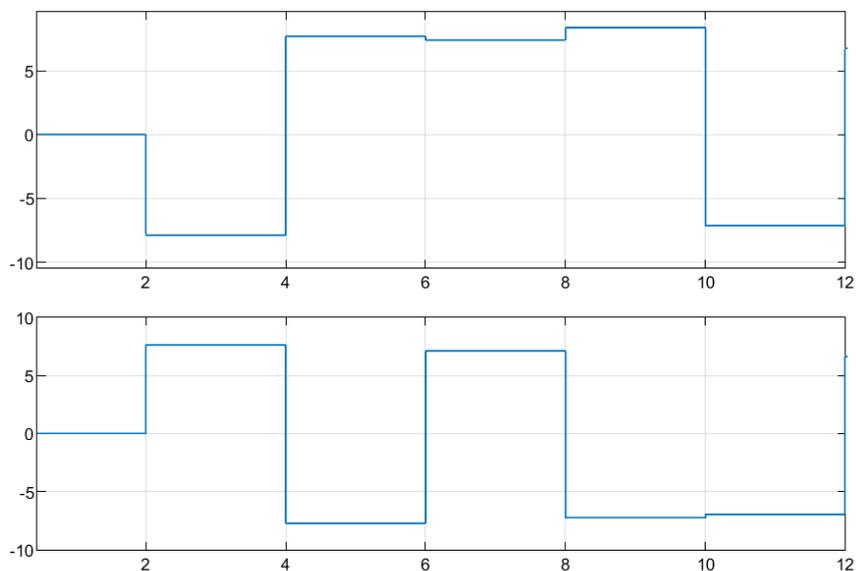


Рисунок 3.4 - Фрагменты осциллограмм на выходах корреляторов в отсутствие шумов канала распространения

На рисунке 3.5 приведены фрагменты осциллограмм на выходе модема и передаваемые биты в отсутствие шумов канала распространения.

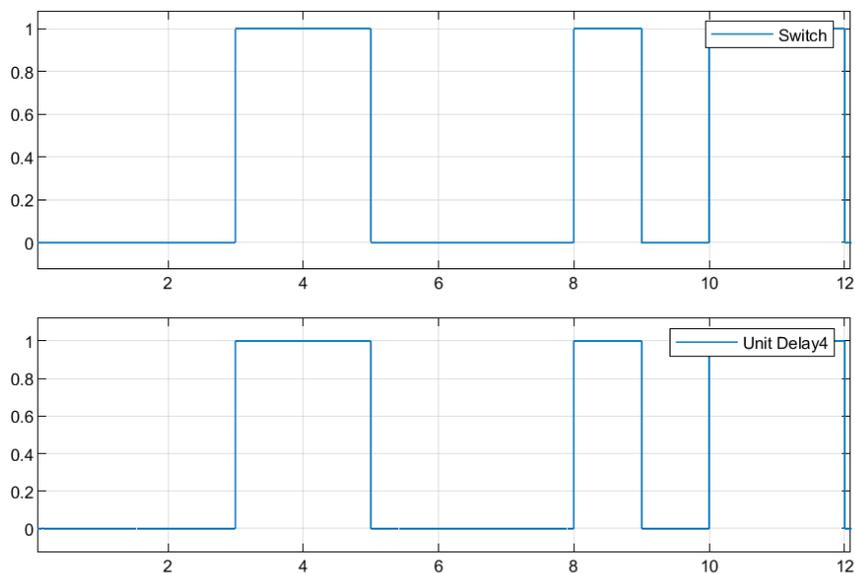


Рисунок 3.5 - Фрагменты осциллограмм на выходе модели и передаваемые биты в отсутствие шумов канала распространения

Приведённые осциллограммы модельного исследования **Sim**-модели подшумового модема на основе **QPSK** модуляции и комплекснозначной последовательности призваны ориентировать студента в процессе выполнения лабораторной работы.

## 4 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].

### 4.1 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 4.1).

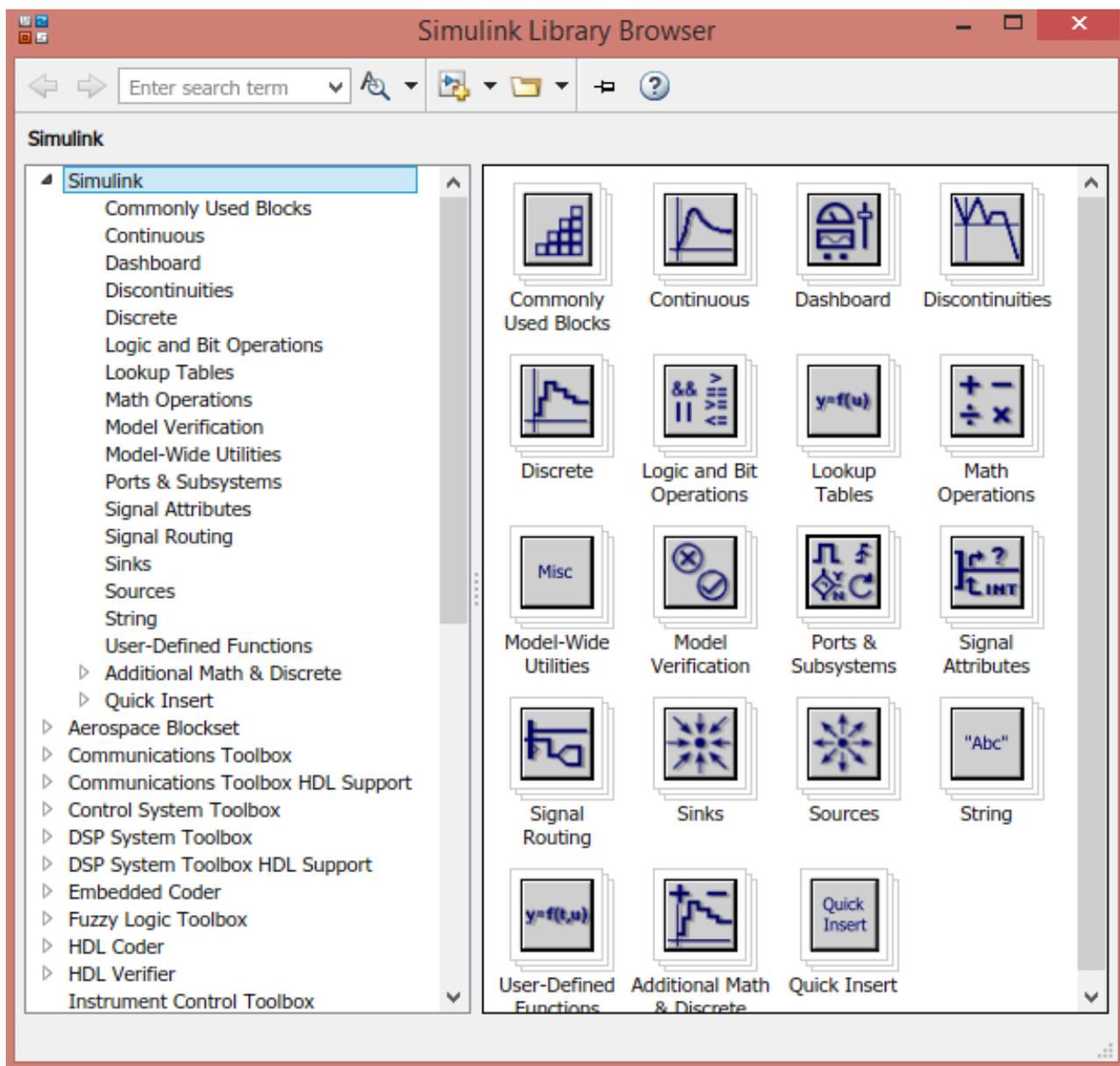


Рисунок 4.1 – Библиотека блоков *Simulink Library Browser*

На рисунке 4.1 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.

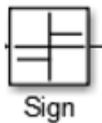
## 4.2 Описание используемых блоков библиотеки *Simulink*

Ниже описаны основные блоки базовых разделов библиотеки *Simulink*, используемые в функциональной схеме **OQPSK**-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.

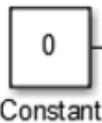


Sign

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

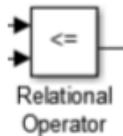
$$\text{sign} = \begin{cases} -1, & \text{где } x < 0; \\ 0, & \text{где } x = 0; \\ 1, & \text{где } x > 0. \end{cases}$$

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



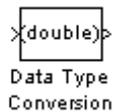
Constant

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



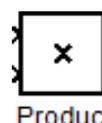
Relational Operator

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



Data Type Conversion

**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

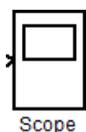
**Product** – блок умножения и деления. Назначение: вычисление произведения текущих значений сигналов. Параметры блока: **Number of inputs** – количество входов, может задаваться как число или как список знаков. В списке знаков можно использовать знаки: \* – умножить и / – разделить. **Multiplication**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



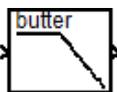
Scope

**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.



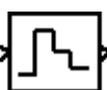
Analog Filter Design

**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

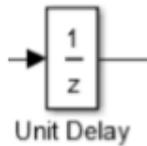
**Gain** – блок усилителя. Назначение: блок **Gain** умножает входной сигнал на постоянный коэффициент; Параметры блока: **Multiplication** – способ выполнения операции, значение параметра выбирается из списка: **Element-wise  $K*u$**  – поэлементный; **Matrix  $K*u$**  – матричный, коэффициент усиления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



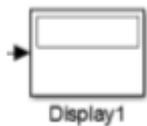
Zero-Order Hold

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

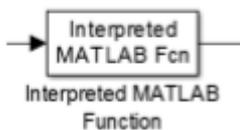
экстраполяция входного сигнала на интервале дискретизации. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания интервала дискретизации. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге дискретизации. 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.

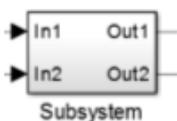


**Display** – блок цифрового дисплея. Назначение: отображает значение сигнала в виде числа. Параметры: **Format** – формат отображения данных. Параметр **Format** может принимать следующие значения: **short** – 5 значащих десятичных цифр, **long** – 15 значащих десятичных цифр, **short\_e** – 5 значащих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



**Переключатель.** Назначение: Выполняет переключение входных сигналов по сигналу управления.

Параметры блока: **Criteria for passing first input:** Значение параметра выбирается из списка:

**U2 >= Threshold** – входной сигнал больше или равен пороговому значению;

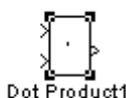
$U2 > Threshold$  – входной сигнал больше порогового значения;  
 $U2 \sim = Threshold$  – входной сигнал не равен пороговому значению.  
 $Threshold$  – порог.



**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.



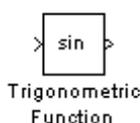
**Источник времени.** *Назначение:* Формирует сигнал, величина которого на каждом шаге равна текущему времени моделирования. *Параметры блока:* **Display time:** Включение отображения времени на пиктограмме; **Decimation:** Шаг обновления времени.



**Блок скалярного произведения.** *Назначение:* Выполняет вычисление скалярного произведения двух векторов. *Параметров блока нет.* Блок выполняет вычисление выходного сигнала в соответствии с выражением

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

где  $u1$  и  $u2$  – входные векторы;  $\text{conj}$  – операция комплексного сопряжения;  $\text{sum}$  – операция суммирования.



**Trigonometric Function-** тригонометрическая функция. *Назначение:* вычляет выбранную тригонометрическую функцию. *Параметры блока:* **Function-** вид вычисляемой функции. Вид функции выбирается из списка: **sin, cos, tan, asin, acos, atan, atan2, sinh, cosh, tanh.** **Output signal type-** тип выходного сигнала. Тип выходного сигнала выбирается из списка: **auto-** автоматическое определение типа, **real-** действительный сигнал, **complex-** комплексный сигнал. При векторном или матричном входном сигнале блок выполняет поэлементное вычисление заданной функции.



**Блок вычисления комплексного числа** по его действительной и мнимой части. *Параметры блока:*

**Input:** Значение параметра выбирается из списка:

**Real** – действительная часть;

**Image** – мнимая часть;

**Real&Imag** – действительная и мнимая часть.



**Блок вычисления действительной и/или мнимой части** комплексного числа. *Параметры блока:*

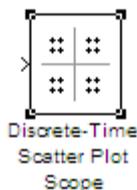
**Output:** Тип сигнала выбирается из списка:

**Real** – действительная часть;

**Image** – мнимая часть;

**Real&Imag** – действительная и мнимая часть.

Входной сигнал блока может быть скалярным, векторным или матричным.



**Discrete-Time Scatter Plot Scope** – блок отображения диаграммы рассеяния фазовых состояний сигнала из подраздела **Comm Sinks** раздела **Communication Blockset** библиотеки **Simulink** [6]. *Назначение:* отображение диаграммы рассеяния фазовых состояний за счет влияния полосы пропускания и помех тракта передачи. *Параметры блока:* Флажок **Show Plotting Properties** – показать графические установки; **Samples per Symbol** – шаг периода символа; **Offset (samples)** –

смещение шагов; **Points displayed** – число отсчетов сигнала, начиная с которого отображается диаграмма; **New points per display** – число отсчетов при обновлении отображения; флажки **Show Rendering Properties**, **Show Axes Properties**, **Show Figure Properties** – показать свойства отображения, осей и фигуры в нашем случае не используются.



**Discrete-Time Signal Trajectory Scope** – блок отображения диаграммы переходов фазовых состояний сигнала из подраздела **Comm Sinks** раздела **Communication Blockset** библиотеки **Simulink** [6]. *Назначение:* отображение диаграммы рассеяния фазовых состояний за счет влияния полосы пропускания и помех тракта передачи. *Параметры блока:* Флажок **Show Plotting Properties** – показать графические установки; **Samples per Symbol** – шаг периода символа; **Offset (samples)** –

смещение шагов; **Points displayed** – число отсчетов сигнала, начиная с которого отображается диаграмма; **New points per display** – число отсчетов при обновлении отображения; флажки **Show Rendering Properties**, **Show Axes Properties**, **Show Figure Properties** – показать свойства отображения, осей и фигуры в нашем случае не используются.

В нашем случае диаграммы блоков **Discrete-Time Scatter Plot Scope** и **Discrete-Time Signal Trajectory Scope** позволяют оценить влияние метода модуляции, полосы пропускания и шумов канала передачи на фазовую диаграмму состояний принятого сигнала.



**Array Plot** – блок отображения векторов или массивов. *Описание:* строит векторы или массивы данных. График массива представляет собой векторный график, на котором данные равномерно распределены по оси x. Доступные измерения: **Data Cursors** – измерить значения сигнала с помощью вертикальных и горизонтальных курсоров; **Signal Statistics** - отобразить максимум, минимум, разность между пиками, среднее значение, медиану и среднеквадратичное значение выбранного сигнала; **Peak Finder** – найти максимумы, показывая значения по оси x ,

в которых они происходят.

## 5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

### Исходные данные.

Для исследования подшумовой технологии передачи на основе использования *QPSK* модуляции и комплекснозначных последовательностей с компактной *ACF* устанавливаем в генераторе *Random Number* с помощью параметра *Sample Time* длительность бита информационного потока  $\tau_b = 1$ . Расширяющая комплекснозначная последовательность реализована с применением генератора повторяющейся последовательности *Repeating Sequence* в качестве модельного времени длиной *16*, а сама комплекснозначная последовательность подаётся на выход генератора с помощью функции *ccc-seq\_\_* с шагом *1/16*. Для обеспечения широкополосности шума в генераторах шума канала распространения параметры *Sample Time* задать равным *0.0125*; частоту опорных гармонических генераторов несущей выставить равными  $\omega_0 = 75 \cdot \pi$  радиан/сек. Шаг взятия отсчетов экстраполяторами нулевого порядка *Zero Order Hold* параметр *Sample Time* установить равным *1/16*, а параметр длины блоков *Buffer* взять равным *32*, что позволит накапливать за дибит *32* отсчета комплекснозначной последовательности. Все основные параметры блоков функциональной модели, в качестве подсказок указаны на рис. 2.1 над блоками.

### Экспериментальное задание.

1. Собрать *Sim*-модель модема подшумовой технологии передачи на основе *QPSK* модуляции и повторяющейся расширяющей комплекснозначной последовательности с компактной *ACF*, показанную на рисунке 2.1. Параметры моделей задать, как показано на рис. 2.1. Отладку моделей вести при отключенных шумах канала распространения.

2. Определить при каком уровне шума появляется вероятность появления битовой ошибки равная *0.001* для исследуемой модели. Измерения *SNR* произвести как на входе так и на выходе коррелятора. По полученным данным рассчитать *SNR* (в *dB*). Длину информационного потока установить порядка *1000* битов. Данные исследования занести в отчет.

3. Сравнить результаты измерения помехоустойчивостей (*SNR*) по входу и выходу коррелятора и сделать выводы о скрытности системы передачи.

4. Составить отчет по проделанной работе (цели, задачи, функциональная схема, принцип работы, наиболее важные осциллограммы и спектрограммы, выводы).

При необходимости экспериментальное задание может быть модифицировано преподавателем.

## 6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Скрытые системы передачи, применение, разновидности.
2. Подшумовая система передачи на основе **QPSK** модуляции и расширяющей комплекснозначной последовательности.
3. Принцип работы корреляционного приёмника.
4. Принцип прямого расширения спектра.
5. Принцип **QPSK** модуляции.
6. Влияние коррелятора на **SNR** принимаемого широкополосного сигнала.
7. Требования, предъявляемые к расширяющей последовательности.
8. От чего зависит вероятность появления битовых ошибок?
9. Особенности формирования комплекснозначной расширяющей последовательности.
10. Что означает компактная автокорреляционная функция **ACF**?
11. Чем определяется полоса пропускания псевдослучайного битового потока?

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гришенцев А.Ю., Елсуков А.И., Коробейников А.Г., Сидоркина И.Г. Разработка и модельная реализация приемо-передающего устройства скрытого подшумового обмена широкополосными радиосообщениями / Вестник чувашского университета, №3, 2017.- С. 195-206.
2. Гришенцев А.Ю., Коробейников А.Г. Алгоритм поиска, некоторые свойства и применение матриц с комплексными значениями элементов для стеганографии и синтеза широкополосных сигналов / Журнал радиоэлектроники, №5, 2016.- 20 с.
3. Гришенцев А.Ю. Способ синтеза и применение шумоподобных широкополосных сигналов в задачах организации защищенных каналов связи // Радиотехника № 9, 2017.- С. 91–101.
4. Гультяев А.К. MatLab 5.3. Имитационное моделирование в среде Windows: Практическое пособие / А.К. Гультяев – 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.