

Министерство науки и высшего образования Российской Федерации

Томский государственный университет  
систем управления и радиоэлектроники

Кологривов В. А.  
Ефремова А. А.

**ИМИТАЦИОННАЯ МОДЕЛЬ ПЕРЕДАЧИ ДАННЫХ НА ОСНОВЕ ТЕХНОЛОГИИ  
LFM\_BPSK**

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

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., Ефремова А. А.**

К 61 Имитационная модель передачи данных на основе технологии LFM\_BPSK: методические указания по лабораторной работе для студентов радиотехнических специальностей / Кологривов В. А., Ефремова А. А. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2022 – 25 с.

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

Лабораторная работа «Имитационная модель передачи данных на основе технологии **LFM\_BPSK**» посвящена модельному исследованию **LFM\_BPSK**-модема с использованием пакета функционального моделирования **Simulink** системы для инженерных и научных расчетов **MatLab**.

Работа “Имитационная модель передачи данных на основе технологии **LFM\_BPSK**” относится к циклу лабораторных работ по разделу “Модуляция/Демодуляция”, входящему в дисциплины радиотехнических специальностей.

В описании сформулирована цель лабораторной работы, приведены краткие теоретические сведения о **LFM** и **PSK**-модуляции, **LFM\_MPSK** технологии, 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 № 3 от 27.10.22

УДК 621.396.61→621.376.3

ББК 32.811.7

© Кологривов В. А., Ефремова А. А., 2022

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

## ОГЛАВЛЕНИЕ

<b>1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ</b> .....	4
1.1 Теоретические сведения о <i>LFM</i> -модуляции .....	4
1.2 Теоретические сведения о <i>BPSK</i> -модуляции .....	5
1.3 Кратко о технологии <i>LFM_MPSK</i> .....	6
<b>2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ LFM_BPSK-МОДЕМА</b> .....	8
<b>3 ПРИНЦИП РАБОТЫ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ LFM_BPSK-МОДЕМА</b> .....	12
<b>4 КРАТКОЕ ОПИСАНИЕ ПАКЕТА SIMULINK И ИСПОЛЬЗУЕМЫХ БЛОКОВ</b> .....	18
4.1 Запуск и работа с пакетом <i>Simulink</i> .....	18
4.2 Описание используемых блоков библиотеки <i>Simulink</i> .....	19
<b>5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ</b> .....	23
<b>6 КОНТРОЛЬНЫЕ ВОПРОСЫ</b> .....	24
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b> .....	25

## 1 ЦЕЛЬ РАБОТЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Цель работы: разработка модели и сравнительное исследование с точки зрения помехоустойчивости и спектральной эффективности модема, реализованного на основе технологии *LFM\_BPSK* с использованием пакета функционального моделирования *Simulink*.

В теоретической части кратко представим материал о *LFM* модуляции, *BPSK* манипуляции и технологии *LFM\_BPSK* [1-4].

### 1.1 Теоретические сведения о *LFM*-модуляции

Линейная частотная модуляция (ЛЧМ) (*LFM*) – это вид частотной модуляции, в котором частота несущего сигнала изменяется по линейному закону.

На рисунке 1.1 представлен закон изменения частоты *LFM* радиоимпульса.

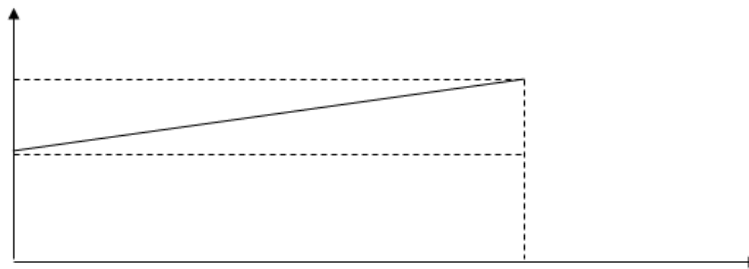


Рисунок 1.1 – Закон изменения частоты *LFM* радиоимпульса

*LFM*-сигналы применяются в радиолокации, как способ формирования и обработки радиоимпульсов, их применение позволяет повысить точность измерений.

Представим описание *LFM* в математическом виде:

Изменение частоты  $f(t)$  внутри импульсов с *LFM* происходит по линейному закону.

$$f(t) = f_0 + bt, \quad -\frac{T_c}{2} \leq t \leq \frac{T_c}{2},$$

где  $f_0 = \frac{F_{max} + F_{min}}{2}$  – центральное значение несущей частоты;

$b = \frac{F_{max} - F_{min}}{T_c}$  – крутизна изменения частоты *LFM* -сигнала;

$T_c$  – длительность сигнала;

$F_{max}$ ,  $F_{min}$  – максимальное и минимальное значение частоты радиосигнала.

Фаза сигнала с линейной частотной модуляцией:

$$\varphi(t) = 2\pi \int_0^t f(t) dt = 2\pi \left( f_0 t + \frac{b}{2} t^2 \right).$$

Следовательно, *LFM* сигнал можно представить в виде:

$$S_{LFM}(t) = S_0 \cos\{\varphi_0 + \varphi(t)\} = S_0 \cos\left\{\varphi_0 + 2\pi\left(f_0 t + \frac{b}{2} t^2\right)\right\},$$

где  $S_0$  – амплитуда сигнала;

$\varphi_0$  – начальная фаза.

## 1.2 Теоретические сведения о *BPSK*-модуляции

Двоичная фазовая модуляция (*BPSK*) является одной из простейших разновидностей цифровой модуляции, где значения фазы несущего колебания принимают значение  $0^\circ$  и  $180^\circ$  градусов.

Модулированный сигнал можно представить в виде:

$$m(t) = A_0 \cos(\omega_0 t + \varphi(t)), \quad (1.1)$$

где  $\varphi(t)$  – мгновенная фаза;

$A_0$  – амплитуда сигнала.

Фазовую модуляцию можно осуществить двумя способами, т.е. начальное значение фазы  $\varphi(t)$  может принимать значения  $0^\circ$  и  $180^\circ$ . В этом случае для реализации фазовой модуляции, как и в частотном модуляторе, можно использовать генератор управляемый напряжением (ГУН), который по сигналу управления может менять фазу колебания. Другой более простой способ используемый при построении используемых моделей связан с перемножением биполярной информационной битовой последовательности с колебаниями генератора несущей частоты.

При изменении фазы на  $180^\circ$  фазовая модуляция рассматривается как амплитудная модуляция с активной паузой, где амплитуда сигнала  $A_0$  принимает значения  $\pm 1$ .

Так как значения амплитуды сигнала  $\pm 1$  можно рассматривать как особый вариант логического нуля и единицы, то с использованием бинарной фазовой модуляции *BPSK* можно передавать обычный двоичный сигнал. Символьная скорость этого вида модуляции составляет  $1$  бит на символ.

На рисунке 1.2 представлена полярная диаграмма сигнала двоичной фазовой модуляции (*BPSK*).

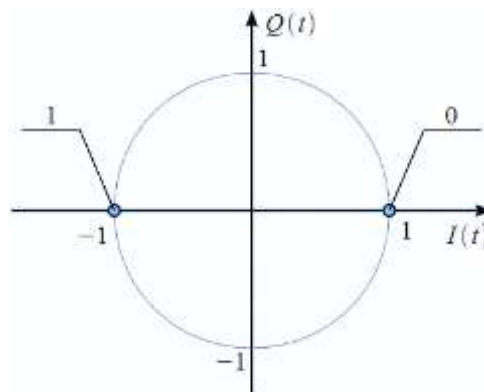


Рисунок 1.2 – Полярная диаграмма сигнала двоичной фазовой модуляции (*BPSK*)

Биполярная версия информационного битового потока соответствует состояниям  $\pm 1$  и межсигнальному расстоянию  $2$ . Это максимальное межсигнальное расстояние соответствует наибольшей помехоустойчивости *BPSK* или энергетической эффективности.

Спектральная эффективность *BPSK* определяется длительностью бита или скоростью передачи битового потока. У *BPSK* модуляции управляющий модулирующий символ и есть бит  $\tau_s = \tau_b$ , то есть *BPSK* модуляция имеет самый широкий спектр.

### 1.3 Кратко о технологии *LFM\_MPSK*

*LFM\_MPSK* технология представляет собой объединение широкополосных сигналов с традиционными методами модуляции узкополосных сигналов. Эта технология сродни технологии прямого расширения спектра, только для расширения спектра информационной последовательности используются *LFM* импульсы. Другими словами в *LFM\_MPSK* технологии вместо монохромных опорных гармонических генераторов используются повторяющиеся *LFM* косинусоидальные либо синусоидальные импульсы. Длительность повторяющихся *LFM* импульсов совпадает с длительностью битов либо модулирующих символов. Соответственно спектр таких сигналов будет определяться параметрами *LFM* импульсов. Для приёма *LFM\_MPSK* сигналов используется коррелятор.

Остановимся на формировании повторяющихся *LFM* импульсов на модельном уровне. Простейшая модель формирования включает два генератора повторяющейся последовательности *Repeating Sequence* с двумя параметрами – *Time values* и *Output values*. *Time values* задаётся массивом значений определяющих точки временного интервала. *Output values* также задаётся массивом значений определяющих форму импульса в заданном временном интервале. Интервал времени, задаваемый в обоих генераторах, совпадает с длительностью модулирующего символа. *Output values* первого генератора в форме линейной пилы (возрастающей либо убывающей) задаёт девиацию частоты, например, от  $\omega_{min}$  до  $\omega_{max}$  с необходимым числом промежуточных точек. *Output values* второго генератора просто повторяет *Time values*. *Output values* первого генератора можно представить как  $\omega_i \cdot t_i$ . *Output values* второго генератора можно представить как  $t_i$ . Тогда их произведение представляется как  $\omega_i \cdot t_i^2$ . Теперь, если подать, сформированные таким образом импульсы в качестве аргумента блока *sin* или *cos*, то на выходе получим повторяющиеся синусоидальные либо косинусоидальные *LFM* импульсы.

Наиболее простая модуляция *LFM\_BPSK* представляет собой манипуляцию знака *LFM* импульса. Биполярная информационная последовательность имеет два состояния  $\pm 1$ , что соответствует межсигнальному расстоянию равного 2. Для реализации *LFM\_BPSK* достаточно перемножить псевдослучайную биполярную информационную последовательность и периодически повторяющиеся *LFM* импульсы. Ширина спектра *LFM\_BPSK* сигнала определяется заданной девиацией частоты *LFM* импульса и скоростью перестройки частоты. Для приёма *LFM\_BPSK* сигнала достаточно использовать коррелятор. В плечах коррелятора в пределах бита запоминаются отсчеты синхронизированного опорного *LFM* импульса и принятого зашумлённого *LFM\_BPSK* модулированного сигнала. Затем накопленные отсчеты скалярно перемножаются (перемножаются и суммируются), масштабируются ограничителем до  $\pm 1$  и тем самым восстанавливаются принятые биты.

Квадратурная *LFM\_QPSK* модуляция. В фазовом кодере псевдослучайный информационный битовый поток преобразуется в пары (дибиты). Дибиты могут принимать 4-е состояния (00 01 10 11) и каждому из них в соответствии с кодом Грея приписываются фазовые состояния, например,  $\varphi_k = \pm \pi/4$ , равномерное распределение которых на единичной окружности даёт межсигнальное расстояние  $\sqrt{2}$ . Вычисляя значения  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ , получаем амплитуды квадратурных модулирующих импульсов (символов) равные  $\pm 1/\sqrt{2}$  длительностью  $\tau_s = 2 \cdot \tau_b$ . Квадратурные модулирующие импульсы поступают на первые входы умножителей квадратурного модулятора, а на вторые входы умножителей с формирователей поступают ортогональные косинусоидальные и синусоидальные *LFM* импульсы. Квадратурный модулятор заканчивается сумматором.

Прием *LFM\_QPSK* модулированных сигналов осуществляется двумя корреляторами для разделения квадратурных потоков. На одном плече модулятора накапливаются отсчеты соответствующего синхронизированного *LFM* импульса, а на другом плече коррелятора

запоминаются отсчеты принятого зашумлённого *LFM\_QPSK* модулированного сигнала. По окончании модулирующего символа отсчеты скалярно перемножаются (перемножаются и суммируются) и подаются на вход подсистемы фазового декодера. Фазовый декодер, выполненный программно, по принятым квадратурным проекциям восстанавливает квадранты комплексной плоскости, соответствующие фазовые состояния и дибиты в параллельной (векторной форме), а преобразователь из параллельного представления в последовательное – выдаёт последовательно биты принятого дибита.

**2-х канальная передача на основе *LFM\_QPSK*.** Так как фазовый кодер квадратурный *LFM\_QPSK* модулятора образует дибиты из информационной последовательности, то можно организовать формирование дибитов из двух независимых информационных потоков. При этом дибиты имеют длительность битов  $\tau_s = \tau_b$ . Это означает на основе *LFM\_QPSK* модуляции можно организовать передачу двух независимых информационных каналов в виде квадратурных потоков длительностью в бит.

В приёмной части с помощью корреляторов квадратурные потоки разделяются и подаются на фазовый декодер. Фазовый декодер, по принятым квадратурным проекциям восстанавливает фазовые состояния и дибиты в параллельной (векторной форме). Первая компонента вектора дибита соответствует восстановленным битам первого информационного источника. Вторая компонента вектора дибита соответствует восстановленным битам второго информационного источника.

**Квадратурная *LFM\_16PSK* модуляция.** В фазовом кодере псевдослучайный информационный битовый поток преобразуется в квадробиты. Квадробиты могут принимать 16 состояний и каждому из них в соответствии с кодом Грея приписываются фазовые состояния, равномерное распределение которых на единичной окружности даёт межсигнальное расстояние  $0.3902$ . Вычисляя значения  $\cos(\varphi_k)$  и  $\sin(\varphi_k)$ , получаем амплитуды квадратурных модулирующих импульсов (символов) длительностью  $\tau_s = 4 \cdot \tau_b$ . Квадратурные модулирующие импульсы поступают на первые входы умножителей квадратурного модулятора, а на вторые входы умножителей с формирователями поступают ортогональные косинусоидальные и синусоидальные *LFM* импульсы. Квадратурный модулятор заканчивается сумматором.

Прием *LFM\_16PSK* модулированных сигналов осуществляется двумя корреляторами для разделения квадратурных потоков. На одном плече модулятора накапливаются отсчеты соответствующего синхронизированного *LFM* импульса, а на другом плече коррелятора запоминаются отсчеты принятого зашумлённого *LFM\_16PSK* модулированного сигнала. По окончании модулирующего символа отсчеты скалярно перемножаются (перемножаются и суммируются) и подаются на вход подсистемы фазового декодера. Фазовый декодер, выполненный программно, по принятым квадратурным проекциям восстанавливает области комплексной плоскости, соответствующие фазовые состояния и квадробиты в параллельной (векторной форме), а преобразователь из параллельного представления в последовательное – выдаёт последовательно биты принятого квадробита.

**4-х канальная передача на основе *LFM\_16PSK*.** Так как фазовый кодер квадратурный *LFM\_16PSK* модулятора образует квадробиты из информационной последовательности, то можно организовать формирование квадробитов из четырёх независимых информационных потоков. При этом квадробиты имеют длительность битов  $\tau_s = \tau_b$ . Это означает на основе *LFM\_16PSK* модуляции можно организовать передачу четырёх независимых информационных каналов в виде квадратурных потоков длительностью в бит.

В приёмной части с помощью корреляторов квадратурные потоки разделяются и подаются на фазовый декодер. Фазовый декодер, по принятым квадратурным проекциям восстанавливает фазовые состояния и квадробиты в параллельной (векторной форме). Номер компоненты вектора квадробита соответствует восстановленным битам соответствующего информационного источника.

## 2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ LFM\_BPSK-МОДЕМА

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*.

Формирователь *LFM*-импульса выполнен на основе первого генератора *Repeating Sequence* с параметрами *Time values* от 0 до 1 с шагом 1/4 – как модельного времени и *Output values* от  $6\pi$  до  $30\pi$  с шагом  $6\pi$  – как девиации частоты. Также на основе второго генератора *Repeating Sequence* с параметрами *Time values* от 0 до 1 с шагом 1/4 и *Output values* от 0 до 1 с шагом 1/4 – как дополнительного модельного времени. После блока *Product* имеем аргумент *LFM* сигнала, который подаётся на блок *sin*, с выхода которого получаем последовательность *LFM*-импульсов. Длительность *LFM*-импульса совпадает с длительностью бита.

*LFM\_BPSK* модулятор выполнен на основе умножителя *Product*, на первый вход которого поступают сформированные *LFM* импульсы, а на второй – биполярная информационная последовательность. На выходе модулятора получаем последовательность фазоманипулированных *LFM*-импульсов.

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

Корреляционный прием и обработка сигнала реализованы на основе простейшего коррелятора при сдвиге  $\tau = 0$ . Плечи корреляционного приёмника состоят из экстраполяторов нулевого порядка *Zero-Order Hold* и блока *Buffer*, которые накапливают в течение бита отсчеты опорного колебания и принятого в шумах *LFM\_BPSK* модулированного радиосигнала. Накопленные отсчеты скалярно перемножаются как вектора и суммируются блоками *Product* и *Sum*. Блок *Frame Status Conversion* преобразовывает фреймовый тип данных в тип *double*. В данном случае реализуется накопление 64 отсчетов в течение бита. Блок *Zero-Order Hold* снимает отсчеты коррелятора, а блок *sign(x)* нормализует их к  $\pm 1$ . Таким образом, происходит восстановление передаваемого битового потока.

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

Подсистема *Subsystem Measuring Power* и блок *Scope* вычисляют и отображают изменение уровня мощности сигнала или смеси сигнал+шум в процессе измерения помехоустойчивости. Точка подключения измерителя мощности обусловлена блоком принятия решений, который соответствует блоку *Zero-Order Hold*.

Блоки *Spectrum Scope* и *Averaging Power Spectral Density* отображают спектры сигнальных и шумовых потоков.

Такая упрощенная реализация корреляционной обработки позволяет найти отсчет автокорреляционной функции опорного колебания и зашумленного радиосигнала при смещении  $\tau = 0$  и в зависимости от знака корреляции принять решение о приеме 1 или -1.

На рисунке 2.1 представлена функциональная модель *LFM\_BPSK* модема с корреляционным приёмником



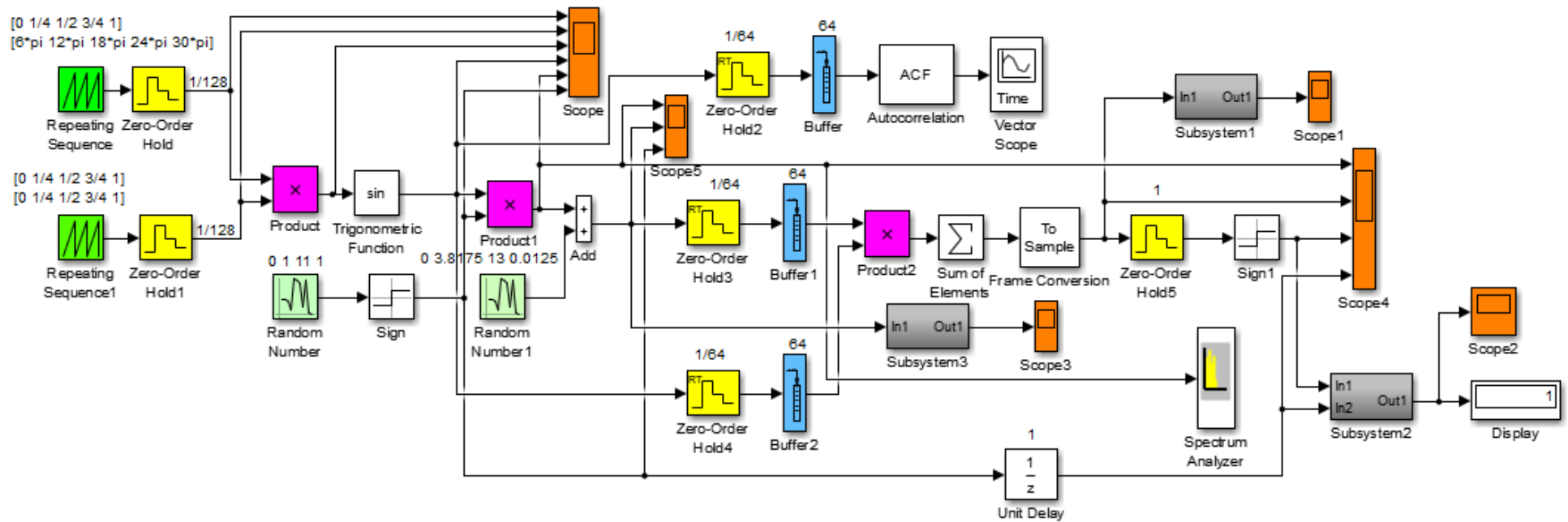


Рисунок 2.1 – Модель *LFM\_BPSK* модема с корреляционным приемником

На рисунке 2.2 представлена **подсистема измерения мощности** сигнала и/или шума.

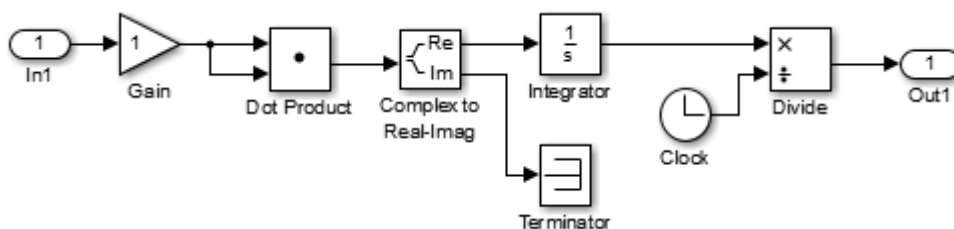


Рисунок 2.2 – Подсистема измерения мощности

Данная подсистема позволяет измерить мощность регулярных, случайных, вещественных и комплексных процессов. При помощи блока **Dot Product** процесс умножается на сопряженный. Далее с помощью блока **Complex to Real-Imag** выделяется вещественная часть. Блок **Integrator** вычисляет энергию процесса, а деление энергии (блок **Divide**) на время (блок **Clock**) вычисляет мощность как скорость поступления энергии.

На рисунке 2.3 представлена **подсистема подсчета ошибок**.

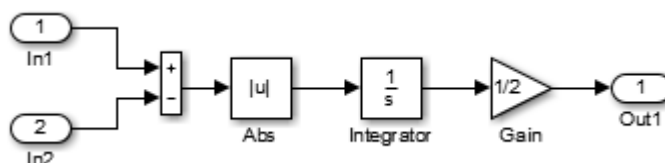


Рисунок 2.3 – Подсистема подсчета ошибок

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, который проходит через всю схему, на второй вход подается сигнал с генератора входной информационной последовательности с необходимой задержкой. К выходу подсистемы присоединен дисплей, на котором будет отображаться наличие ошибок и их количество.

**Модель коррелятора.** Корреляционный приемник реализован следующим образом (см. рис. 2.4):

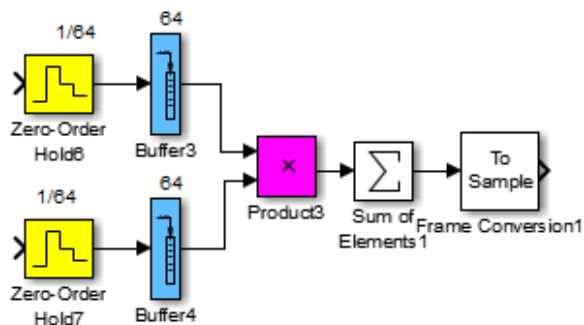


Рисунок 2.4 – Реализация корреляционного приемника

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 сигнала и колебаний опорного формирователя *LFM* импульсов. 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.

**Измерение отношения сигнал/шум (SNR).** Для того чтобы определить мощность сигнала без шума (*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 генератора псевдослучайной гауссовской последовательности модели канала распространения.

Отношение сигнал/шум (*SNR*) рассчитывается по следующей формуле:

$$SNR = \frac{S}{(SN-S)}. \quad (2.1)$$

Результаты измерений фиксируются в дБ.

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

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

### 3 ПРИНЦИП РАБОТЫ ФУНКЦИОНАЛЬНОЙ SIM-МОДЕЛИ LFM\_BPSK-МОДЕМА

Опишем принцип работы *LFM-BPSK* модема по функциональной модели, представленной на рис. 2.1.

На основе генераторов *Repeating Sequence* выполнены формирователи *LFM*-импульсов, изменяющиеся, например, по синусоидальному закону. Формирователь содержит по два генератора: первый задаёт девиацию частоты во время *LFM* импульса, а второй – дополнительное модельное время. Блоки *Product* реализуют аргумент *LFM* импульса и подают его на блок *sin*, с выходов которых получаем повторяющуюся последовательность *LFM*-импульсов длительностью  $\tau_b$ .

Псевдослучайный информационный биполярный битовый поток формируется генератором *Random Number*.

Модулятор *LFM\_BPSK* на основе умножителя (блок *Product*) перемножает повторяющуюся последовательность *LFM*-импульсов и биполярный информационный битовый поток.

Простейший имитатор канала распространения радиосигнала на основе блока *Sum* добавляет широкополосные шумы и позволяет менять интенсивность шумов в процессе измерения помехоустойчивости.

Корреляционный прием и обработка сигнала реализованы на основе простейшего коррелятора при сдвиге  $\tau = 0$ . Плечи корреляционного приёмника накапливают в течение бита отсчеты опорного колебания (опорного *LFM* импульса) и принятого в шумах *LFM\_BPSK* модулированного радиосигнала. Накопленные отсчеты, как вектора скалярно перемножаются и суммируются блоками *Product* и *Sum*. Блок *Frame Status Conversion* призван преобразовать фреймовый тип данных в тип *double*. В данном случае реализуется накопление 64 отсчетов в течение бита.

Принятые сигналы канала приёма с выхода корреляторов через блок *Zero Order Hold* подаются на блок *Sign*, для масштабирования амплитуд отсчетов к значениям  $\pm 1$ . В результате с выхода корреляционного приёмника получаем принятый информационный битовый поток.

Для иллюстрации работы модема на рис. 3.1 приведены фрагменты осциллограмм формирователя синусоидальных *LFM* импульсов, импульсов на выходе плеча квадратурного модулятора и информационный битовый поток.

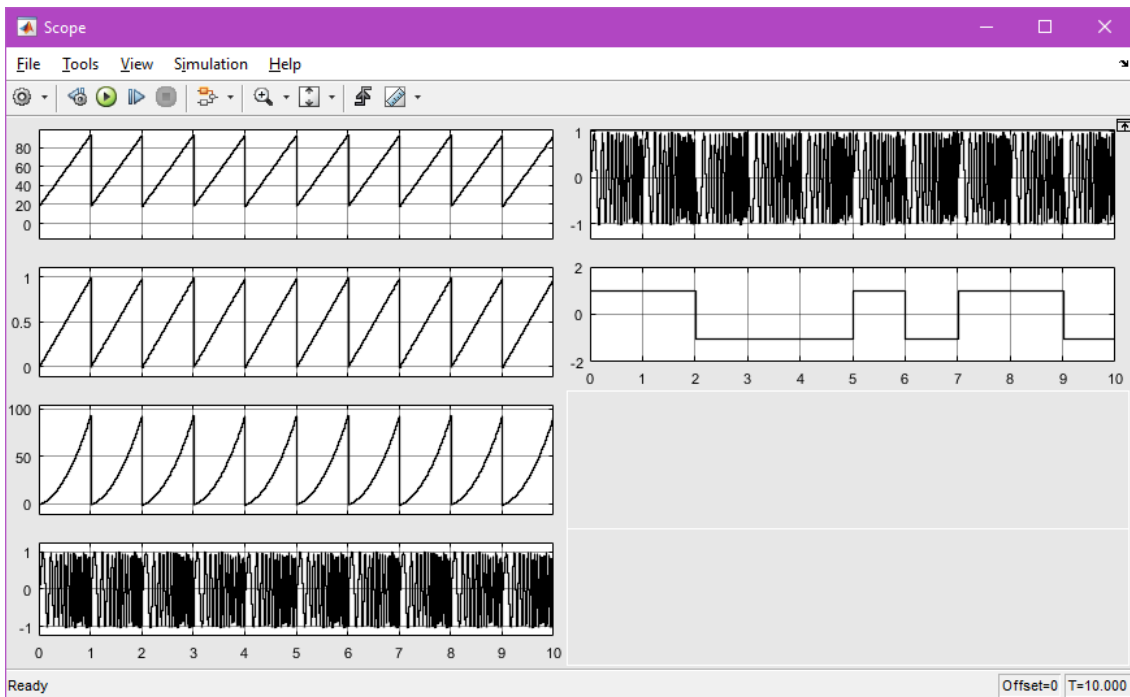


Рисунок 3.1 - Фрагменты осциллограмм формирователя синусоидальных *LFM* импульсов, импульсов на выходе плеча квадратурного модулятора и информационный битовый поток (1 - импульсы девиации частоты; 2 - импульсы модельного времени; 3 - импульсы аргумента *LFM* импульса; 4 – последовательность *LFM* импульсов; 5 - модулированные *LFM* импульсы; 6 - импульсы информационного битового потока)

На рисунке 3.2 приведены фрагменты осциллограмм радиоимпульсов каналов передачи до и после канала распространения и информационная битовая последовательность.

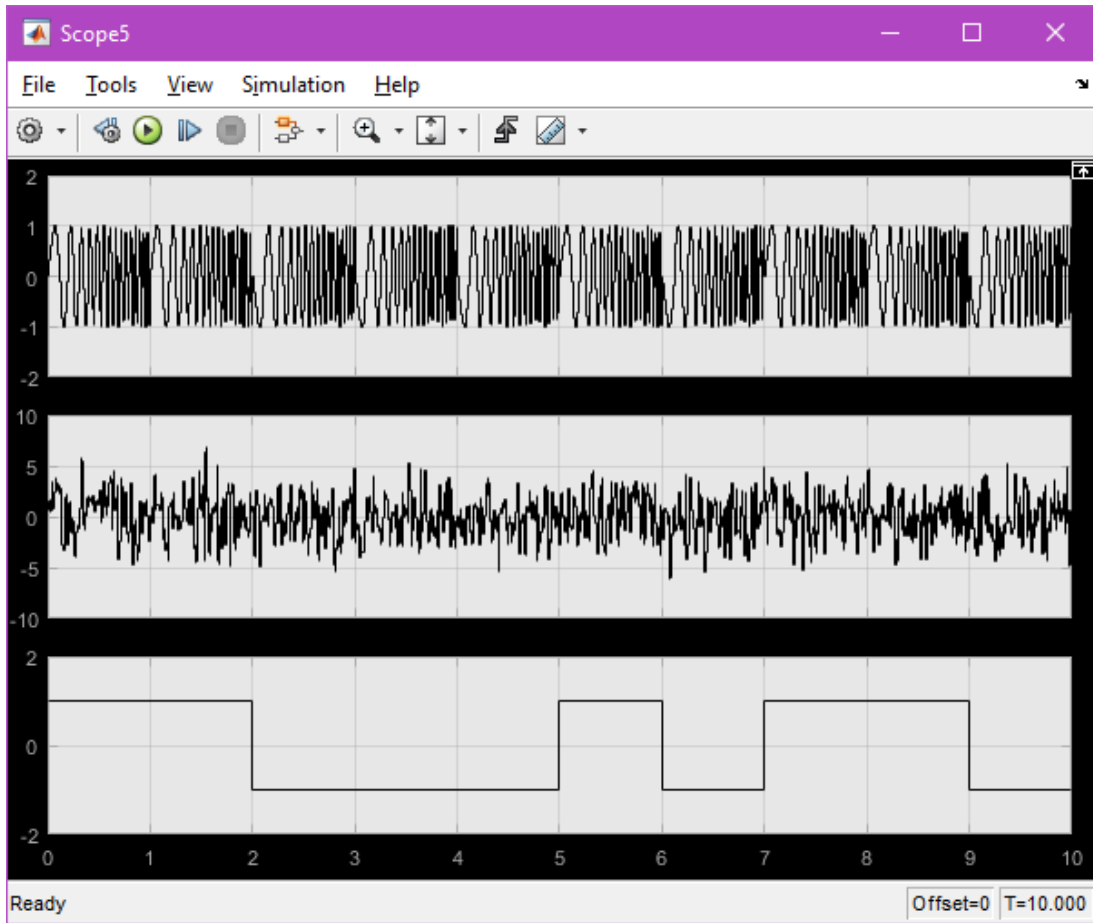


Рисунок 3.2 - Фрагменты осциллограмм радиоимпульсов каналов передачи до и после канала распространения и информационная битовая последовательность

На рисунке 3.3 приведены фрагменты осциллограмм последовательности модулированных *LFM\_BPSK* импульсов, принятых коррелятором сигналов, принятые и передаваемые биты.

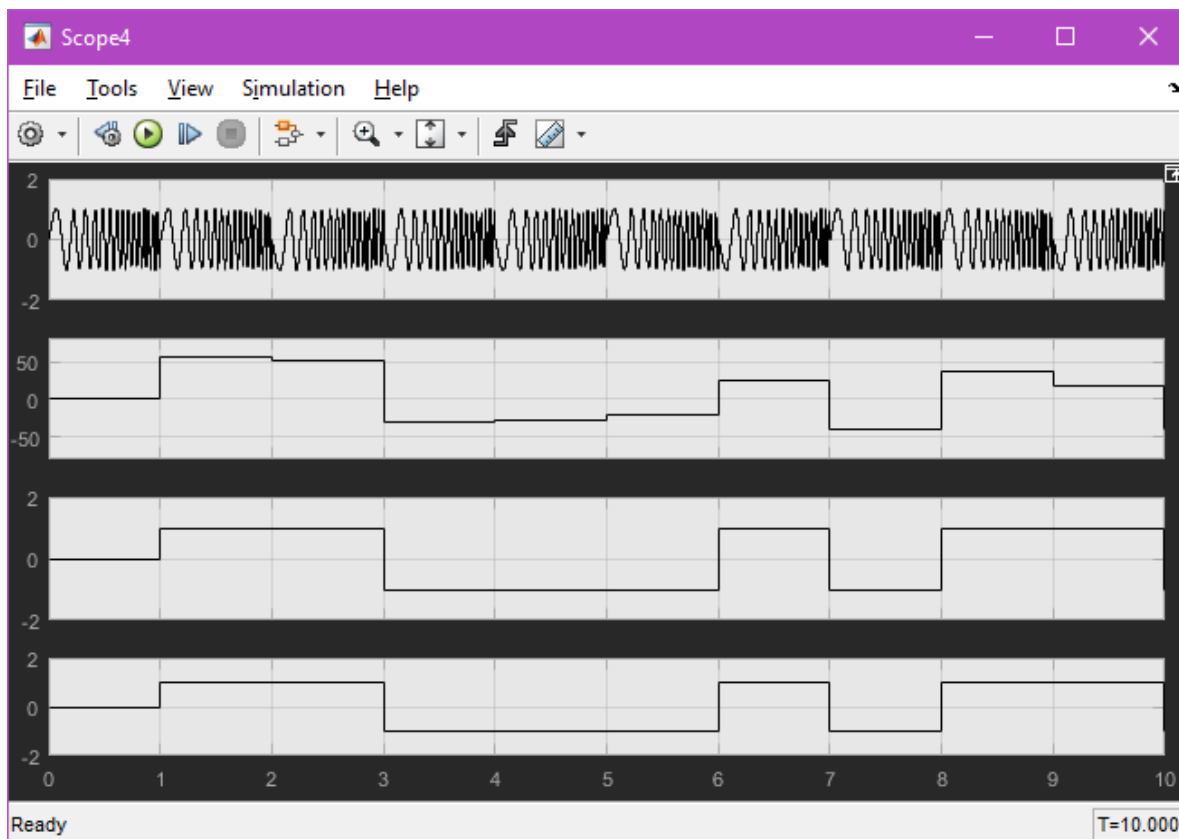


Рисунок 3.3 - Фрагменты осциллограмм последовательности модулированных *LFM\_BPSK* импульсов, принятых коррелятором сигналов, принятые и передаваемые биты

На рисунке 3.4 приведена автокорреляционная функция последовательности *LFM* импульсов.

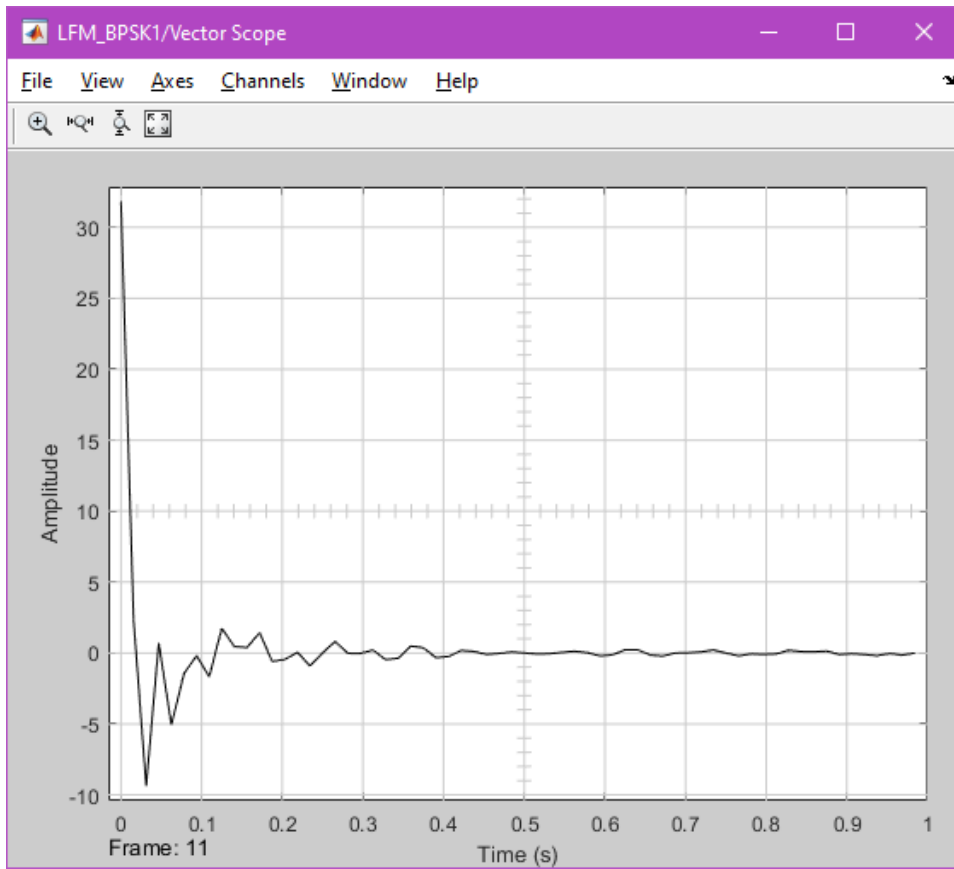


Рисунок 3.4 – Автокорреляционная функция последовательности *LFM* импульсов



На рисунке 3.5 приведен спектр *LFM-BPSK* модулированного сигнала.

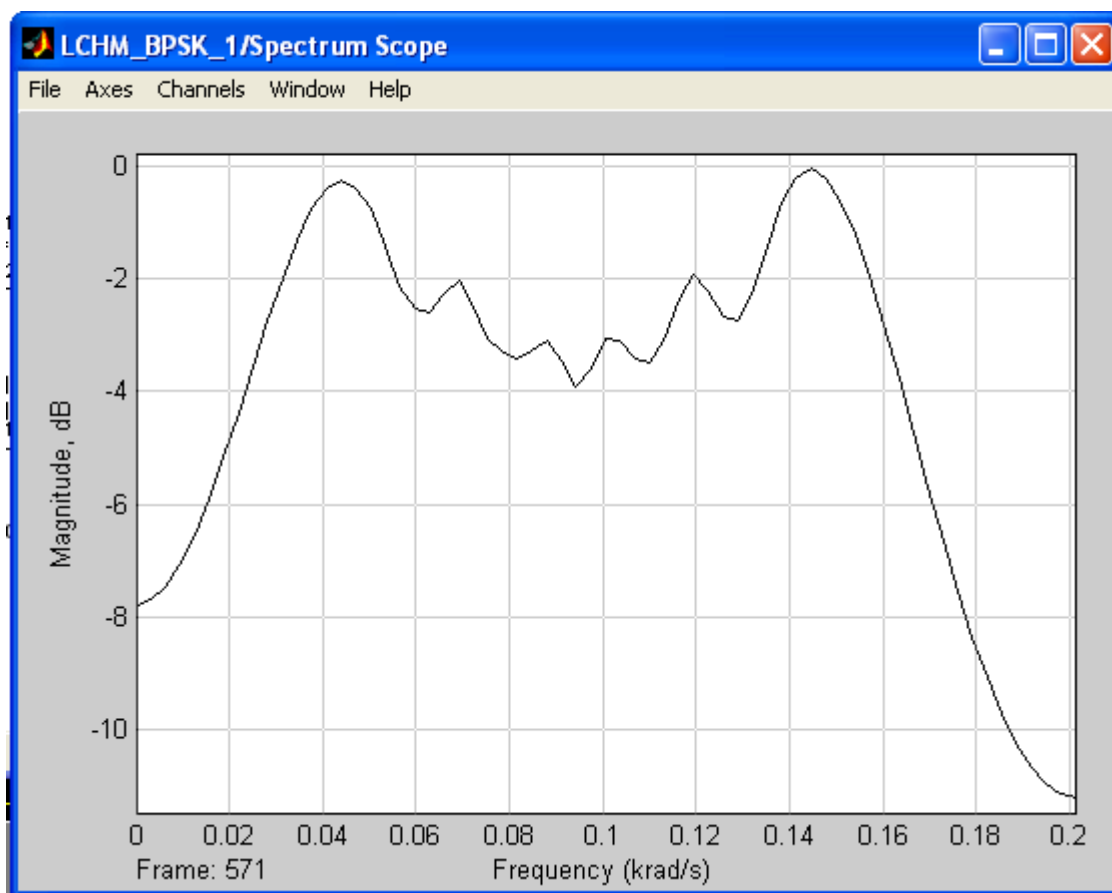


Рисунок 3.5 – Спектр *LFM-BPSK* модулированного сигнала


Приведённые осциллограммы модельного исследования *LFM-BPSK* модема 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 [5–8].

### 4.1 Запуск и работа с пакетом *Simulink*

Для того чтобы запустить пакет *Simulink*, необходимо предварительно выполнить запуск системы *MatLab*. После открытия командного окна системы *MatLab* необходимо запустить систему *Simulink*. Запустить систему *Simulink* можно тремя способами:

- 1) нажать кнопку  (*Simulink*) на панели инструментов системы *MatLab*;
- 2) в строке командного окна *MatLab* напечатать *Simulink* и нажать клавишу *Enter*;
- 3) в меню *File* выполнить опцию *Open* 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).

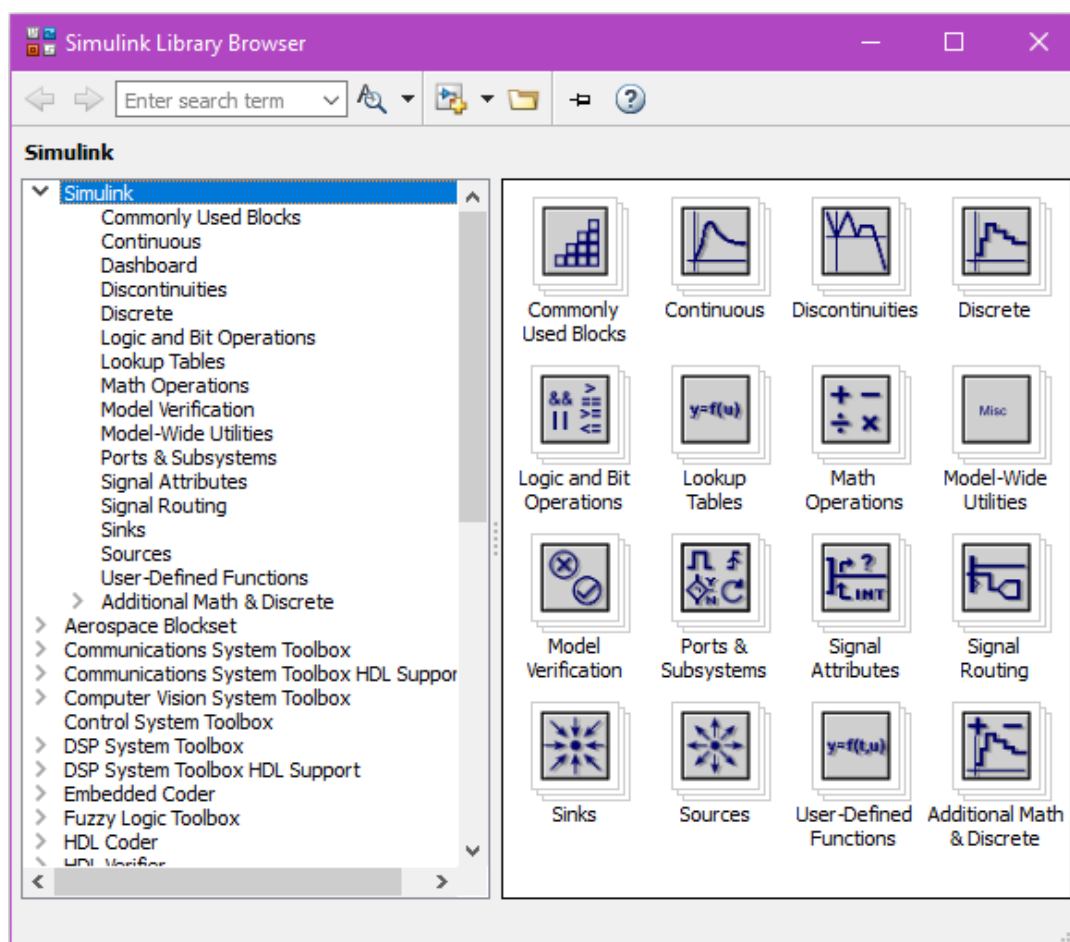


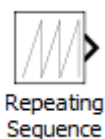
Рисунок 4.1 – Библиотека блоков *Simulink Library Browser*

На рисунке 4.1 в левой части окна выведена библиотека системы *Simulink*, а в правой части окна показаны ее разделы. Основная библиотека системы содержит следующие разделы:

- 1) *Continuous* – блоки аналоговых элементов;
- 2) *Discontinuous* – блоки нелинейных элементов;
- 3) *Discrete* – блоки дискретных элементов;
- 4) *Look-Up Tables* – блоки таблиц;
- 5) *Math Operations* – блоки элементов, определяющие математические операции;
- 6) *Model Verification* – блоки проверки свойств сигнала;
- 7) *Model-Wide Utilities* – раздел дополнительных утилит;
- 8) *Port & Subsystems* – порты и подсистемы;
- 9) *Signal Attributes* – блоки маршрутизации сигналов;
- 10) *Signal Routing* – блоки маршрутизации сигналов;
- 11) *Sinks* – блоки приема и отображения сигналов;
- 12) *Sources* – блоки источников сигнала;
- 13) *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 *LFM\_BPSK*-модема:



Repeating Sequence

**Repeating Sequence** – выводит периодический скалярный сигнал, имеющий форму волны, что вы задаете использование параметры **Output values** и **Time values**. Параметр **Time values** задает вектор выходных времен. Параметр **Output values** задает вектор из амплитуд сигнала в соответствующие выходные времена. Вместе, эти два параметра задают выборку выходной формы волны в точках, измеренных с начала интервала, на котором форма волны повторяется (период сигнала).

По умолчанию обоими параметрами является **[0 2]**. Эти настройки по умолчанию задают пилообразную форму волны, которая повторяет каждый **2** секунды от запуска симуляции и имеют максимальную амплитуду **2**.



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.



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:

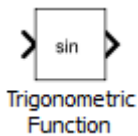
$$\begin{aligned} & -1, \text{ где } x < 0; \\ \mathit{sign} & = 0, \text{ где } x = 0; \\ & 1, \text{ где } x > 0. \end{aligned}$$

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



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.



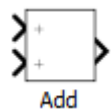
Trigonometric Function

**Trigonometric Function** – тригонометрическая функция. Назначение: выдает 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.



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.



Buffer

**Buffer** – Буферная входная последовательность к меньшему или большему формату кадра. Всегда выполняет основанную на системе координат обработку. Блок перераспределяет данные в каждом столбце входа, чтобы произвести выход с различным форматом кадра. Буферизация сигнала к большему формату кадра дает к выходу с более медленной частотой кадров, чем вход.



Autocorrelation

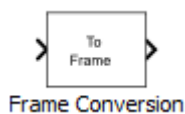
**Autocorrelation** – в этом блоке вычисляется автокорреляционная функция сегмента сигнала. Число отсчетов корреляционной функции определяется заданной величиной порядка предсказания, то есть числом предыдущих отсчетов сигнала, по которым предсказывается значение последующего отсчета.



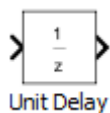
Sum

**Sum** – блок выполняет сложение или вычитание на его входных параметрах. Этот блок может добавить или вычесть скаляр, вектор или матричные входные параметры. Это может также свернуть элементы сигнала и

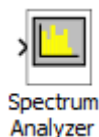
выполнить суммирование. Блок *Sum* сначала преобразует тип входных данных в свой тип данных аккумулятора, затем выполняет заданные операции. Блок преобразует результат в свой тип выходных данных с помощью заданного округления и режимов переполнения.



**Frame Conversion** – данный блок передает вход до выхода и устанавливает выходной режим выборки на значение параметра *Sampling mode of output signal*, который может быть любой *Frame-based* или *Sample-based*. Задайте режим выборки выходного сигнала *Sample-based*.



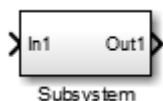
**Unit delay** – блок единичной дискретной задержки. Назначение: выполняет задержку дискретного сигнала на заданный шаг модельного времени. Параметры блока: *Initial conditions* – начальное значение выходного сигнала; *Sample time* – шаг модельного времени.



**Spectrum Analyzer** – блок, упомянутый здесь как осциллограф, отображает спектры частоты сигналов. Панель *Spectrum Settings* появляется в правой стороне окна *Spectrum Analyzer*. Эти настройки управляют, как спектр вычисляется. Чтобы показать Настройки Спектра, в меню *Spectrum Analyzer*, выбирают *View>Spectrum Settings*. Диалоговое окно *Configuration Properties* управляет визуальными аспектами Спектр Анализатора. Чтобы открыть *Configuration Properties*, в меню *Spectrum Analyzer*, выбирают *View>Configuration Properties*. Средства управления появляется в виде диалогового окна *Style*, Спектр Анализатора. Чтобы открыть Свойства стиля, в меню *Spectrum Analyzer*, выбирают *View>Style*. Диалоговое окно *Axes Scaling* управляет пределами осей Спектра Анализатора. Чтобы открыть свойства *Axes Scaling*, в меню *Spectrum Analyzer*, выбирают *Tools>Axes Scaling>Axes Scaling Properties*.



**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 *ЛКМ* 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*...).



**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.



**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.

## 5 ЭКСПЕРИМЕНТАЛЬНОЕ ЗАДАНИЕ

### Исходные данные.

Для исследования *LFM\_BPSK* модема в генераторе *Random Number* задаются параметры *0 1 11 1*, с помощью параметра *Sample Time* задается длительность бита  $\tau_b = 1$ . В формирователе *LFM*-импульсов для блока *Repeating Sequence* заданы параметры модельного времени [*0 1/4 1/2 3/4 1*] (*Time values*) и параметры девиации частоты [*6\*pi 12\*pi 18\*pi 24\*pi 30\*pi*] (*Output values*). В блоке *Zero Order Hold* задаем параметр *Sample Time* = *1/128*. В плечах корреляционного приёмника в блоке *Zero Order Hold* параметр *Sample Time* = *1/64*, а в блоке *Buffer* параметр *Output buffer size* = *64*. В шумовой псевдослучайной последовательности с гауссовским распределением *Random Number* заданы параметры *0 3.8175 13 0.0125*. Все основные параметры блоков, в качестве подсказок указаны на рис. 2.1 над блоками.

### Экспериментальное задание.

1. Собрать Sim-модель *LFM\_BPSK*-модема в соответствии с рисунком 2.1. В блоке *Repeating Sequence* задать параметры девиации частоты от *6\*pi* до *30\*pi*.
2. Вычислить отношение сигнал/шум (*SNR*) при *1, 3, 5* и *8* ошибках, и оценить помехоустойчивость модема.
3. Сравнить результаты и сделать выводы.
4. Составить отчёт по проделанной работе.
5. Ответить на контрольные вопросы по лабораторной работе.

## 6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите принцип *LFM*-модуляции.
2. Опишите принцип двоичной фазовой манипуляции (*BPSK*).
3. Какими преимуществами обладает *LFM*-модуляция?
4. Как формируется последовательность *LFM* импульсов?
5. Как устроен модулятор *LFM\_BPSK*?
6. Опишите принцип работы корреляционного приемника.
7. Где применяется линейная частотная модуляция?
8. Опишите методику измерения помехоустойчивости модема.
9. Какова связь автокорреляционной функции со спектральной плотностью?
10. Для чего может быть использована автокорреляционная функция *ACF*?



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лукашин И.В. Исследование применения ЛЧМ-сигналов для передачи данных по радиоканалу. Электронный адрес: // <http://omoled.ru/publications/view/414> (дата обращения 7.11.22).
2. Антипенский Р. Моделирование источников сложных сигналов // Журнал «Современная электроника», 2007, №9.– С. 47-49.
3. Леонидов В.В. Конспект лекций «Модуляция и демодуляция цифровых сигналов». Учебно-методический комплект по дисциплине «Цифровая обработка». МГТУ имени Н.Э. Баумана. Электронный адрес: <https://leonidov.su/wp-content/uploads/2020/04/Modulation-and-Demodulation-of-Digital-Signals-Lecture-V.V.-Leonidov.pdf> (дата обращения 7.11.22).
4. Речкина Д.А. Модем на основе LFM-BPSK сигналов. ВКР бакалавра. - Томск: ТУСУР, 2022.- 55 с.
5. Гультяев, А.К. MatLab 5.3. Имитационное моделирование в среде Windows: Практическое пособие / А.К. Гультяев – СПб.: КОРОНА принт, 2001.– 400 с.
6. Черных, И.В. Simulink: среда создания инженерных приложений. / Под общ. ред. В.Г. Потемкина – М.: ДИАЛОГ-МИФИ, 2003.– 496 с.
7. Дьяконов, В.П. MatLab 6.5 SP1/7 + Simulink 5/6. Основы применения. Сер. Библиотека профессионала / В. П. Дьяконов - М.: СОЛОН-Пресс, 2005.– 800 с.
8. 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.