

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего профессионального образования**  
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ**  
**УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**Кафедра радиотехнических систем (РТС)**

Утверждаю:  
Зав. каф. РТС, проф., д.т.н.  
\_\_\_\_\_ Мелихов С.В.  
\_\_\_\_\_ 2016 г.



**Кологривов В.А., Хазиахметова Р.З.**

**ИССЛЕДОВАНИЕ ОСНОВНЫХ ХАРАКТЕРИСТИК СИСТЕМ ФАПЧ**

**Учебно-методическое пособие по лабораторной работе**  
**для студентов направления**  
**«Инфокоммуникационные технологии и системы связи»**  
**по дисциплине**  
**«Сети и системы мобильной связи»**

Разработчики:  
доц. каф. РТС  
\_\_\_\_\_ Кологривов В.А.  
\_\_\_\_\_ Хазиахметова Р.З.

**2016**

**Кологривов В. А., Хазиахметова Р.З.**

**«Исследование основных характеристик систем ФАПЧ»:** Учебно-методическое пособие по лабораторной работе для студентов направления «Инфокоммуникационные технологии и системы связи» по дисциплине «Сети и системы мобильной связи». – Томск: ТУСУР. Научно-образовательный портал, 2016.– 32 с.

Учебно-методическое пособие содержит описание функциональной модели для исследования функциональных схем фазовой автоподстройки частоты (ФАПЧ) цифровых телекоммуникационных систем, выполненной в среде функционального моделирования *Simulink*, системы для инженерных и научных расчетов *MatLab*.

В пособии приведены краткие теоретические сведения о фазовой автоподстройке частоты, краткая характеристика пакета *Simulink* системы *MatLab*, описание виртуального лабораторного макета и используемых блоков библиотеки *Simulink*.

### Аннотация

Лабораторная работа «Исследование основных характеристик **ФАПЧ**» посвящена экспериментальному исследованию фазовой автоподстройки частоты с использованием пакета функционального моделирования *Simulink*, системы для инженерных и научных расчетов *MatLab*.

Работа «Исследование основных характеристик **ФАПЧ**» относится к циклу лабораторных работ, входящему в дисциплины по направлению «Инфокоммуникационные технологии и системы связи».

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

Лабораторная работа рассчитана на два лабораторных задания по 4 часа каждая.

**Содержание**

<b>1. Цель работы. Краткие сведения из теории</b>	<b>5</b>
<b>2. Краткое описание пакета Simulink</b>	<b>11</b>
<b>2.1. Общая характеристика пакета Simulink</b>	<b>11</b>
<b>2.2. Запуск и работа с пакетом Simulink</b>	<b>11</b>
<b>3. Описание лабораторного макета</b>	<b>14</b>
<b>4. Описание используемых блоков библиотеки Simulink</b>	<b>19</b>
<b>5. Экспериментальное задание</b>	<b>30</b>
<b>6. Контрольные вопросы</b>	<b>31</b>
<b>Список использованных источников</b>	<b>32</b>

## 1 Цель работы. Краткие сведения из теории

**Цель работы:** Изучить режимы работы и параметры схем фазовой автоподстройки частоты (**ФАПЧ**) цифровых систем передач данных. Разработка виртуального лабораторного макета.

### Теоретическая часть

Системы **ФАПЧ** (**PLL - Phase Locked Loop**) относятся к классу следящих систем и предназначены для слежения за фазой сигнала (точнее, за частотой и фазой). Принципиальное отличие систем **ФАПЧ** от систем **АПЧ**, в которых информация о фазе сигнала теряется. Благодаря этой особенности системы **ФАПЧ** применяются гораздо шире, чем системы **АПЧ**. Они используются в качестве следящих фильтров доплеровских систем измерения скоростей, для стабилизации промежуточной частоты приёмников, синхронного детектирования сигналов, в качестве демодуляторов ЧМ- и ФМ-сигналов, в синтезаторах частот, в следящих измерителях координат (дальности, угла).

Принцип действия системы **ФАПЧ** рассмотрим на примере простейшей системы, описываемой функциональной схемой, приведенной на рисунке 1.1.

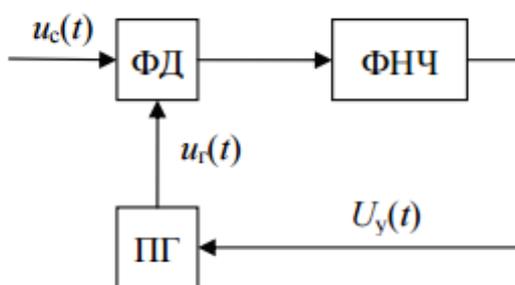


Рисунок 1.1 – Функциональная схема системы **ФАПЧ**

На входы фазового дискриминатора (**ФД**) поступает сигнал  $u_c(t)$  (обычно промежуточной частоты) и опорное колебание  $u_r(t)$ ,

вырабатываемое подстраиваемым генератором. Если частоты сигнала и генератора совпадают (начальная частотная расстройка  $\Delta f = f_c - f_{г0} = 0$ , то в стационарном режиме напряжение на выходе **ФД** равно нулю, а фазовый сдвиг  $\varphi_{ст} = \varphi_c - \varphi_{г} = -\pi/2$ .

Обычно начальная расстройка  $\Delta f_0 \neq 0$  (вследствие нестабильности частот сигнала и подстраиваемого генератора, доплеровского сдвига частоты сигнала). В этом случае на выходе дискриминатора возникают биения с частотой  $\Delta f$ . Если бы обратная связь была разомкнута, то на выходе **ФД** наблюдалось бы гармоническое колебание с частотой, равной частотной расстройке. Благодаря обратной связи управляющее напряжение  $U_y(t)$ , воздействуя на подстраиваемый генератор (управляющий элемент – варикап), осуществляет частотную модуляцию колебания  $u_{г}(t)$ . В зависимости от знака управляющего напряжения частота биений то возрастает, то убывает (рисунок 1.2). В результате конечной длительности положительные и отрицательные полуволны напряжений биений оказываются разными, вследствие чего на выходе дискриминатора образуется постоянная составляющая напряжения, которая стремится уменьшить частоту биений до нуля.

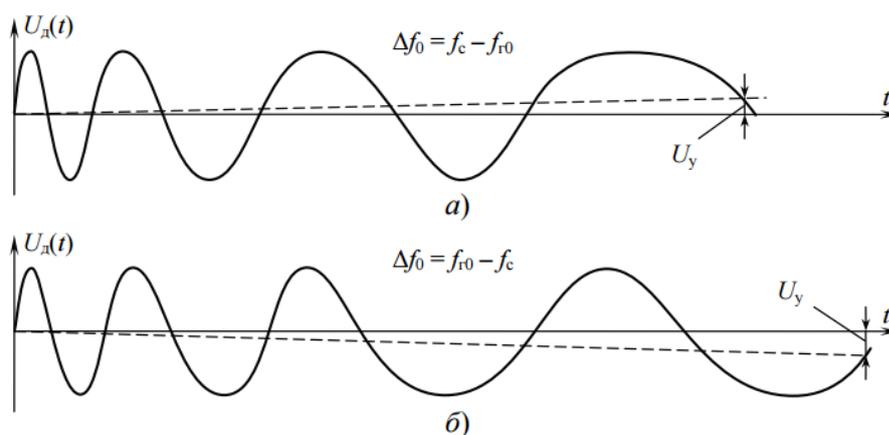


Рисунок 1.2 – Зависимость управляющего напряжения и частоты биения

Фильтр нижних частот пропускает постоянное напряжение, обусловленное частотной модуляцией сигнала. Знак управляющего напряжения определяется знаком частотной расстройки: напряжение положительное при  $f_c > f_{г0}$  (рисунок 1.2, а) и отрицательное при  $f_c < f_{г0}$  (рисунок 1.2, б). В зависимости от знака  $U_y$  средняя частота генератора либо возрастает, либо убывает относительно начального значения  $f_{г0}$  (соответствует  $U_y = 0$ , стремясь к значению, равному  $f_c$  (рисунок 1.3).

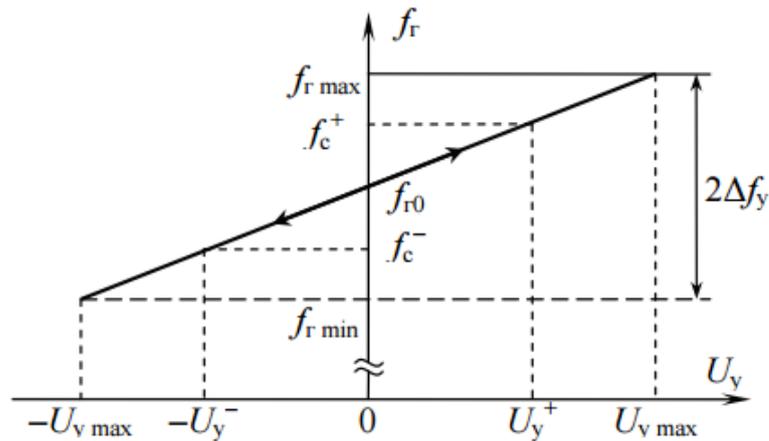


Рисунок 1.3 - Зависимость  $U_y$  от средней частоты генератора

Если начальная расстройка не превышает значения  $\Delta f_3$ , определяющего полосу захвата, то по истечении времени  $t_3$ , называемого временем захвата, в системе устанавливается стационарный режим (режим слежения или удержания), при котором частотная расстройка  $\Delta f = 0$ , ( $f_c = f_{г0}$ ), а фазовый сдвиг сигнала относительно колебания генератора  $\varphi_{ст} = -\pi/2 + \varphi$ , где  $\varphi$  – фазовая ошибка. Если начальная расстройка  $\Delta f_0 > \Delta f_3$ , то управляющее напряжение становится недостаточным для её компенсации и установление стационарного режима невозможно. Этот режим (режим биений) характеризуется тем, что средняя частота генератора отличается от частоты сигнала, а разность фаз непрерывно нарастает.

Полоса захвата системы **ФАПЧ** определяется диапазоном перестройки генератора, формой дискриминационной характеристики, а также структурой (видом передаточной функции) фильтра нижних частот [1].

Все цифровые системы связи требуют наличия нескольких уровней синхронизации на стороне приемника. Когерентные схемы демодуляции сигнала подразумевают наличие на стороне приемника опорного колебания, с которым производится сравнение принимаемого сигнала. Основой многих систем синхронизации является контур фазовой автоподстройки частоты (рисунке 1.4). В результате работы петли **ФАПЧ** сигнал местного гетеродина подстраивается по фазе и частоте с принимаемым сигналом, что и позволяет осуществить когерентное детектирование. Таким образом, для работы петли **ФАПЧ** необходимо наличие в сигнале устойчивого синусоидального колебания.

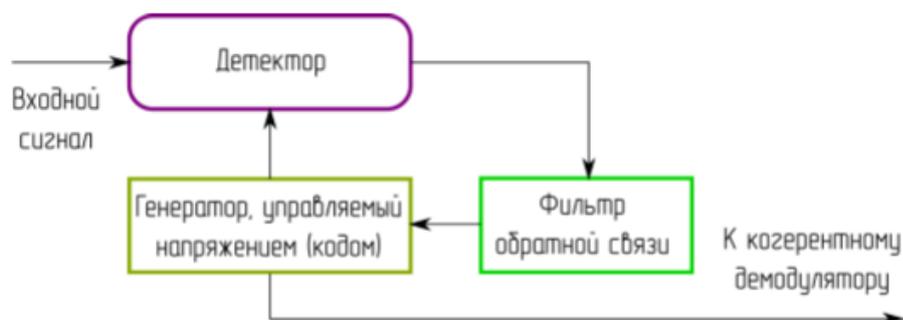


Рисунок 1.4 – Структурная схема петли фазовой автоподстройки частот

Большинство современных систем связи работают с подавлением несущего колебания, так как оно практически не несет полезной информации. В таком случае классический контур **ФАПЧ** не может восстановить несущее колебание на приемном конце линии связи. Например, при равной вероятности появления сигналов логического нуля и логической единицы на входе модулятора **BPSK**, усредненная энергия несущего колебания будет равна нулю. Кроме того для синхронизации несущего колебания необходимо устранить модуляцию. Сделать это можно путем

возведения входного сигнала в квадрат по схеме, предложенной отечественным ученым *А.А. Пистолькорсом* (рисунок 1.5).

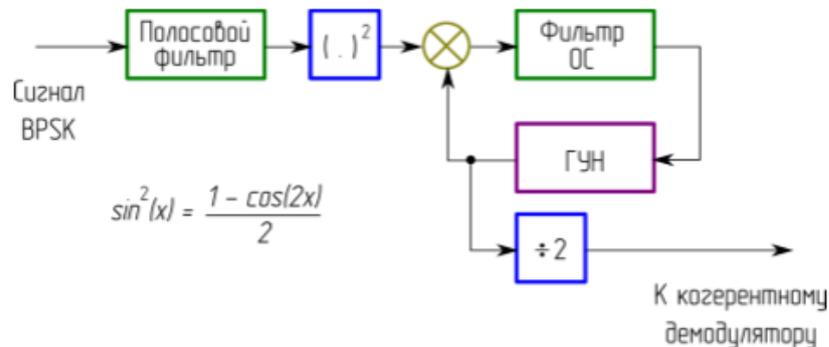


Рисунок 1.5 – Схема восстановления несущего колебания с возведением сигнала в квадрат (схема *Пистолькорса*)

Возведение в квадрат сигнала с **BPSK** снимает с него модуляцию, что позволяет выполнить синхронизацию на удвоенной частоте несущего колебания при помощи классической петли **ФАПЧ**. Для синхронизации сигналов **QPSK** необходимо возведение сигнала в четвертую степень. Другой разновидностью схем восстановления несущего колебания является синфазно-квадратурная петля, предложенная *Джоном Костасом* (рисунок 1.6).

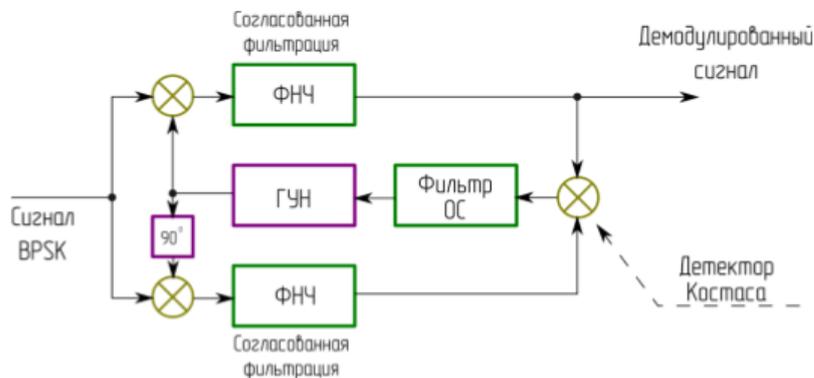


Рисунок 1.6 – Синфазно-квадратурная схема восстановления несущего колебания (петля *Костаса*)

Главным отличием петли Костаса от предыдущей схемы является отсутствие операции возведения сигнала в квадрат, реализация которой в цифровом виде требует повышения разрядности шины данных. Необходимо отметить, что с точки зрения теоретических характеристик эти схемы равнозначны [2].

## 2 Краткое описание пакета *Simulink*

### 2.1 Общая характеристика пакета *Simulink*

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

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

Для запуска системы *Simulink* необходимо выполнить запуск системы *MatLab*. После открытия командного окна системы *MatLab* нужно запустить систему *Simulink*. Существует три способа запуска системы *Simulink*:

- 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).

При применении двух первых способов открывается окно обозревателя библиотеки блоков (*Simulink Library Browser*). Если нам не требуется добавление новых блоков, а нужно лишь открыть уже готовую модель и провести моделирование, то следует воспользоваться третьим способом.

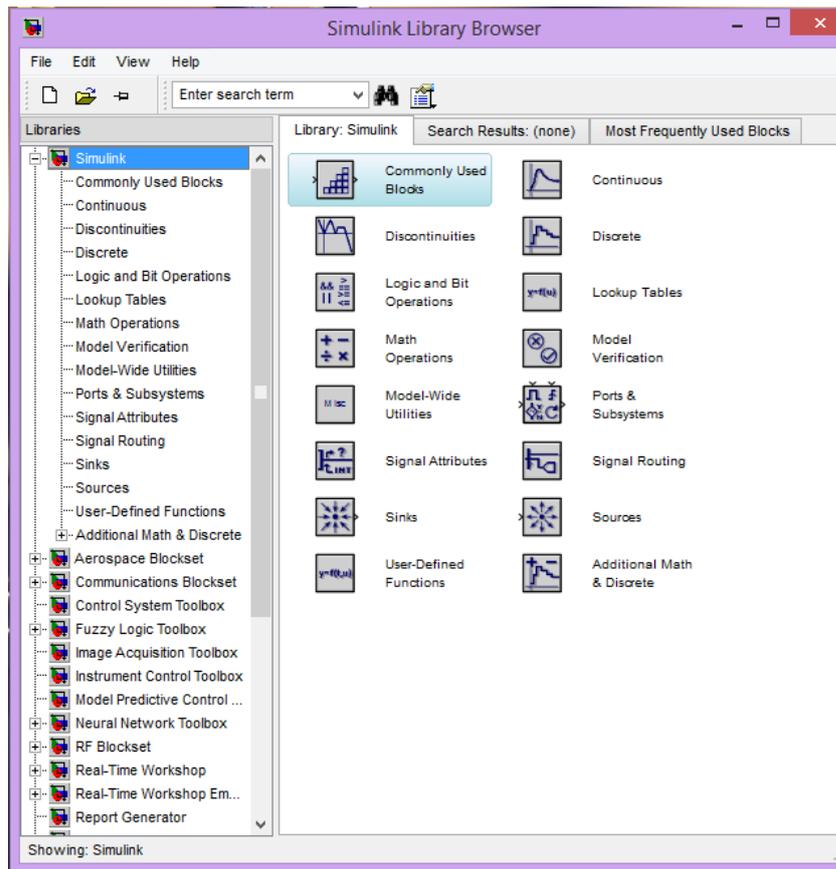


Рисунок 2.1. – Библиотека блоков *Simulink Library Browser*

На рисунке 2.1 выведена библиотека системы *Simulink* и показаны ее разделы. Основная библиотека системы содержит следующие разделы:

- **Commonly Used Blocks** – наиболее часто используемые блоки;
- **Continuous** – блоки аналоговых элементов;
- **Discontinuous** – блоки нелинейных элементов;
- **Discrete** – блоки дискретных элементов;
- **Logic and Bit Operations** – Логические и битовые операции;
- **Lookup Tables** – блоки таблиц;
- **Math Operations** – блоки элементов, определяющие математические операции;
- **Model Verification** – блоки проверки свойств сигнала;
- **Model-Wide Utilities** – раздел дополнительных утилит;
- **Port&Subsystems** – порты и подсистемы;

- **Signal Attributes** – блоки задания свойств сигналов;
- **Signal Routing** – блоки маршрутизации сигналов;
- **Sinks** – блоки приема и отображения сигналов;
- **Sources** – блоки источников сигнала;
- **User-Defined Function** – функции, определяемые пользователем;
- **Additional Math & Discrete** – дополнительная и дискретная математика.

Список разделов библиотеки представлен в виде дерева, и правила работы с ним являются общими для списков такого вида: пиктограмма свернутого узла дерева содержит символ «+», а пиктограмма развернутого – символ «-».

Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши (ЛКМ). При выборе соответствующего раздела библиотеки его содержимое отображается в правой части окна.

При работе элементы разделов библиотек "**перетаскивают**" в рабочую область удержанием *ЛКМ* на соответствующих изображениях. Для соединения элементов достаточно указать курсором мыши на начало соединения и затем при нажатии левой кнопки мыши протянуть соединение в его конец.

При двойном щелчке *ЛКМ* на выделенном блоке всплывает меню, в котором задаются параметры блоков.

Работа *Simulink* происходит на фоне открытого окна системы *MatLab*, закрытие которого приведёт к выходу из *Simulink*.

### 3 Описание лабораторного макета

Вариант реализации функциональной модели модема на основе **BPSK** модуляции, с использованием восстановления несущей по схеме **Костаса** представлена на рисунке 3.1.

Функциональная модель состоит из передатчика и приемника (демодулятора, схемы восстановления несущей по схеме **Костаса** и схемы регенерации формы принятого сигнала). Помимо этого модель включает блоки качания частоты несущей с заданной девиацией.

**Передатчик.** С помощью блоков **Random Number** и **Sign** формируем псевдослучайную информационную последовательность. С помощью блоков **Sine Wave**, **Constant**, **Sum**, **Product1** и **trigon.function (cos)** формируем гармоническую несущую, с заданной частотой и амплитудой девиации. Блок **Product** реализует **BPSK** модулятор.

**Приемник.** Синфазно-квадратурная схема восстановления несущего колебания (петля **Костаса**) состоит из двух квадратурных каналов замкнутых в «кольцо **Костаса**» и образованного с помощью перемножителя, фильтра нижних частот, усилителя, генератора управляемого напряжением с квадратурным выходом.

Функциональная модель модема представлена на рисунке 3.1.

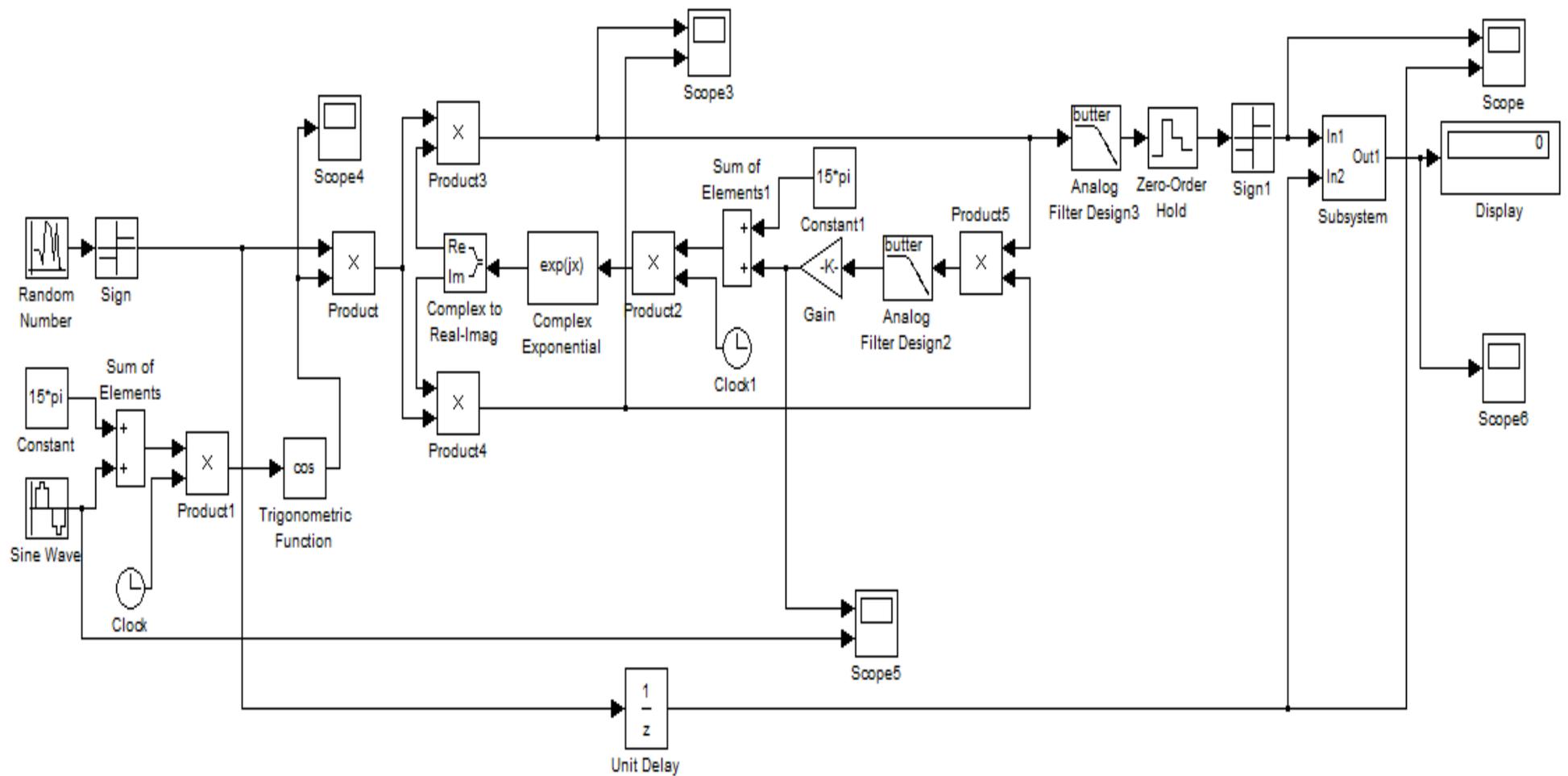


Рисунок 3.1 – Функциональная модель **BPSK** модема с фазовой автоподстройкой частоты по схеме **Костаса**

**Схема Костаса.** Схема фазовой автоподстройки частоты, совмещает в себе демодулятор и петлю обратной связи. С выходов квадратурных перемножителей организуется обратная связь с помощью **ФНЧ** и **ГУН** с квадратурным выходом – блок **Complex To Real-Imag**. В приемнике принятый радиосигнал подаётся на квадратурный преобразователь - два блока **Product**. На вторые входы перемножителей квадратурного преобразователя подаются квадратурные составляющие петли обратной связи. С целью выделения низкой разности частот в кольце обратной связи сигналы фильтруются узкополосным **ФНЧ** и подаются на **ГУН** с квадратурным выходом. Напряжение с выхода узкополосного **ФНЧ** является управляющим для **ГУНа** (опорный генератор, усилитель). **ГУН** формирует гармонический сигнал в аналитическом (комплексном) виде, который разделяется на квадратурные составляющие в качестве опорного сигнала.

Рассмотрим принцип работы схемы *Костаса* для **BPSK** сигнала (рисунок 3.2). Опорные колебания вырабатываются в **ГУН** (генератор управляемый напряжением). Частота **ГУН** подстраивается под частоту принимаемой несущей с помощью петли **ФАПЧ**, роль **ФД** (фазовый детектор) выполняют перемножители и **ФНЧ** (фильтр нижних частот). Разность сигналов после перемножителя в петле обратной связи используется для управления **ГУН**. Таким образом, **ФАПЧ** обеспечивает точную установку частоты несущей.

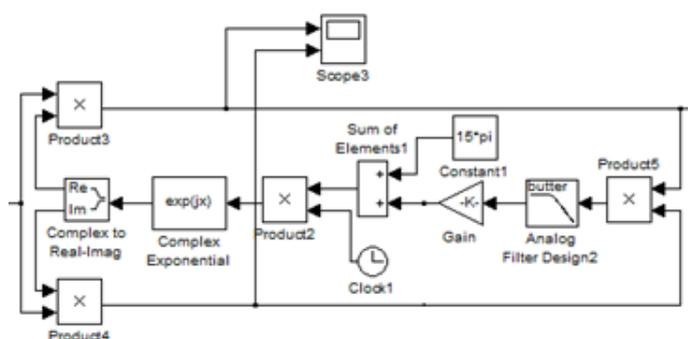


Рисунок 3.2 – Синфазно-квадратурная схема восстановления несущего колебания (петля Костаса)

**Демодулятор** состоит из перемножителя, фильтра нижних частот. Фильтр нижних частот убирает вторую гармонику несущего колебания возникающую в перемножителе (преобразователе).

**Регенератор формы** битовых импульсов включает блок взятия отсчетов – экстраполятор нулевого порядка **Zero\_Order Hold** и блок функции знака **Sign**.

Вторым вариантом реализации фазовой автоподстройки частоты является функциональная модель модема на основе **BPSK** модуляции, с использованием восстановления несущей по схеме **Пистолькорса**. Модель представлена на рисунке 3.3. С помощью блоков **Random Number** и **Sign** формируем псевдослучайную информационную последовательность. Блоки **Sine Wave**, **Constant**, **Sum**, **Product1** и **trigon.function (cos)** формируют гармоническую несущую, с заданной девиацией частоты. Блок **Product** реализует **BPSK** сигнал. Блок **MathFunction** выполняет возведение сигнала в квадрат.

В схеме восстановления несущей используется возведение в квадрат принятого сигнала, с целью снятия с него модуляции. В результате синхронизация выполняется на удвоенной частоте несущего колебания при помощи классической петли **ФАПЧ**. В кольце обратной связи, с помощью **ФНЧ**, выделяется закон девиации несущей частоты. Так как подстройка происходит на второй гармонике, девиация частоты оказывается удвоенной, поэтому для того чтобы с выхода **ГУН** можно было получить несущую необходимо использовать делитель частоты. Так как, делитель частоты в **Simulink** выполнить проблематично было решено использовать закон изменения несущей. Закон изменения несущей, как управляющий сигнал, подаем на новый **ГУН**, с выхода которого получаем восстановленную несущую.

Остальная часть функциональной модели приемника приемного тракта традиционна. Функциональная модель модема представлена на рисунке 3.3.

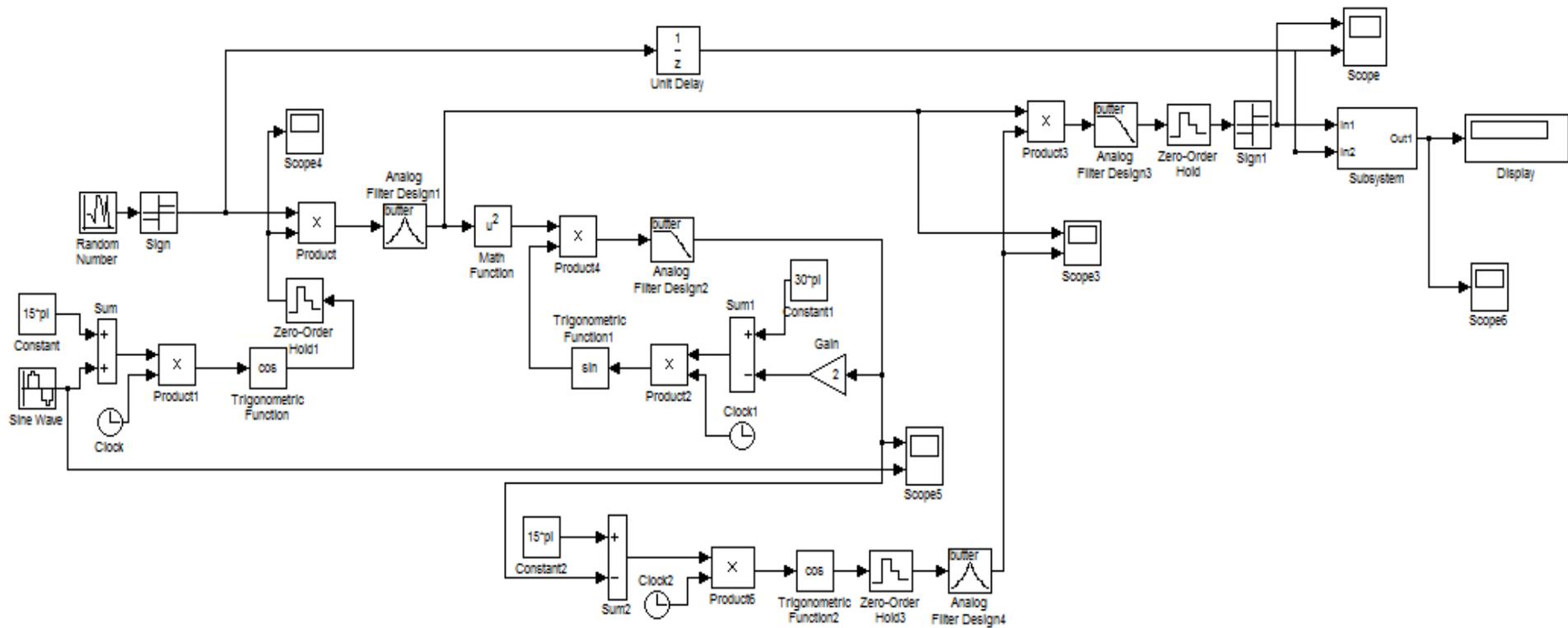


Рисунок 3.3 – Функциональная модель **BPSK** модема с фазовой автоподстройкой частоты по схеме **Пистолькорса**

## 4 Описание используемых блоков библиотеки Simulink

Ниже описаны основные блоки базовых разделов библиотеки *Simulink* [3], используемые в функциональных моделях модема на основе **BPSK** модуляции, с использованием восстановления несущей по схеме **Костаса** и по схеме **Пистолькорса**.



**Random Number** – источник случайного сигнала с нормальным распределением. *Назначение:* формирование случайного сигнала с равномерным распределением уровня сигнала. *Параметры блока:* **Mean** – среднее значение сигнала; **Variance** – дисперсия; **Initial seed** – начальное значение генератора случайного сигнала; **Sample time** – такт дискретности.



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

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

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



Scope

**Scope**— блок осциллографа. *Назначение:* построение графиков исследуемых сигналов как функций времени. Открытие окна осциллографа производится двойным щелчком **ЛКМ** на пиктограмме блока. 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.



Sine Wave

**Sine Wave**— блок источника синусоидального сигнала. *Назначение:* формирование синусоидального сигнала с заданной частотой, амплитудой, фазой и смещением. Для формирования выходного сигнала блоком могут использоваться два алгоритма. Вид алгоритма определяется параметром **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.

*Параметры блока в режиме **Time-based**:* **Amplitude** – амплитуда; **Bias** – постоянная составляющего сигнала (смещение); **Frequency (rads/sec)** – частота (рад/с); **Phase (rads)** – начальная фаза (рад); **Sample time** – такт

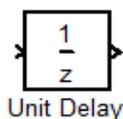
дискретности. Параметр может принимать следующие значения: **0** (по умолчанию) – используется при моделировании непрерывных систем; **> 0** (положительное значение) – задается при моделировании дискретных систем; **-1** (минус один) – такт дискретности устанавливается таким же, как и в предшествующем блоке. Флажок **Interpreted vector parameters as 1 – D** – интерпретировать вектор как массив скаляров. Для очень больших значений времени точность вычисления значений сигнала падает.

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

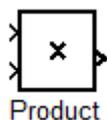
$$N = \text{Samples per second} = 1/(f \cdot T);$$

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

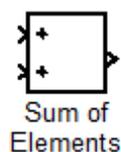
**Number of offset samples** – начальная фаза сигнала, задается количеством тактов дискретности  $l = \varphi \cdot p / (2 \cdot \pi)$ . **Sample time** – такт дискретности. Флажок **Interpret vector parameters as 1 - D** – интерпретировать вектор как одномерный. В данном режиме ошибка округления не накапливается, поскольку **Simulink** начинает отсчет номера текущего шага с нуля для каждого периода.



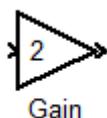
**Unit delay** – блок единичной дискретной задержки. *Назначение:* выполняет задержку дискретного сигнала на заданный такт дискретности. *Параметры блока:* **Initial conditions** – начальное значение выходного сигнала; **Sample time** – такт дискретности (при задании значения параметра равного **-1** такт дискретности наследуется от предшествующего блока).



**Product**– блок умножения и деления. *Назначение:* вычисление произведения текущих значений сигналов. *Параметры блока:* **Number of inputs** – количество входов, может задаваться как число или как список знаков. В списке знаков можно использовать знаки: \* - умножить и / - разделить. **Multiplication** – способ выполнения операции, может принимать значения из списка: **Element-wise** – поэлементный; **Matrix** – матричный. Флажок **Show additional parameters** – показать дополнительные параметры. При выставленном флажке отображается окно списка **Output data type mode**, в нашем случае флажок не используется.



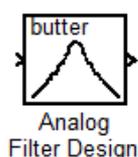
**Sum**– блок сумматора. *Назначение:* вычисление алгебраической суммы текущих значений входных сигналов. *Параметры блока:* **Icon shape** – форма блока, выбирается из списка: **round** – круг; **rectangular** – прямоугольник. **List of sign** – список знаков из набора: + - плюс; - - минус, | - разделитель. 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.



**Gain**– блок усилителя. *Назначение:* выполняет умножение входного сигнала на постоянный коэффициент; *Параметры блока:* **Gain**-коэффициент усиления. **Multiplication** – способ выполнения операции,

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

Блоки **Gain** и **Matrix Gain** по сути есть один и тот же блок, но с разными начальными установками параметра **Multiplication**.



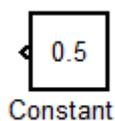
**Analog Filter Design** – блок аналогового фильтра заданного метода проектирования и типа из подраздела **Filter Design**. *Назначение:* аналоговая фильтрация низкочастотных составляющих спектра входного сигнала. *Параметры блока:* **Design method** – метод проектирования, выбирается из списка: **Butterworth**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

других методик проектирования и типов фильтров определяемые параметры очевидны.



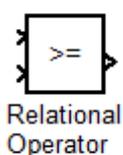
**Zero-Order Hold**– экстраполятор нулевого порядка. *Назначение:*

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



**Constant**– блок источника постоянного сигнала. *Назначение:* задает

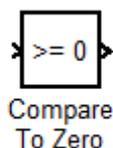
постоянный по уровню сигнал. *Параметры блока:* **Constant value**– постоянная величина. **Interpret vector parameters as 1-D** – интерпретировать вектор как массив скаляров. **Show additional parameters** – показать дополнительные параметры. При выставленном флажке появится окно списка Output data type mode. **Output data type mode** – выбор типы выходных данных. **Output data type** – тип выходных данных. **Output Scaling Mode** – способ масштабирования выходного сигнала. **Output scaling value** – величина масштаба.



**Relational Operator**– блок выполнения операций отношения.

*Назначение:* блок сравнивает текущие значения входных сигналов.

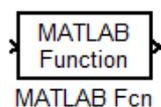
*Параметры блока:* Relational Operator- тип операции отношения, выбирается из списка. **Show additional parameters** – показать дополнительные параметры. **Require all inputs to have same data type**- все входы должны иметь одинаковый тип данных. **Output data type mode** – выбор типы выходных данных. **Output data type** – тип выходных данных. **Enable zero crossing detection**-фиксировать прохождение сигнала через нулевой уровень.



**Compare To Zero**– блок сравнения с нулем. *Назначение:* сравнить с нулем заданный сигнал. *Параметры блока:* **Operation** – операции сравнения. Выбираются из списка. **Output data type mode** – выбор типы выходных данных.



**Demux**– демультиплексор. *Назначение:* разделяет входной векторный сигнал на отдельные составляющие. *Параметры блока:* **Number of Outputs**-количество выходов. **Display option**- способ отображения, выбирается из списка: **bar**-вертикальный узкий прямоугольник черного цвета; **none**-прямоугольник с белым фоном без отображения меток входных сигналов. **Bus Selection Mode**-режим деления векторных сигналов по шине.

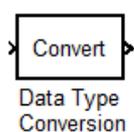


**MatLab Function**– блок задания М-функции. *Назначение:* задает выражение в стиле языка программирования MatLab. *Параметры блока:* **MatLab Function**- выражение на языке MatLab. **Output dimensions** – размерность выходного сигнала. **Output signal type**– тип выходного сигнала. Выбирается из списка: **real**-действительный сигнал; **complex**- комплексный

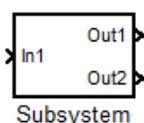
сигнал; **auto-** автоматическое определение типа сигнала. **Collapse 2-D results to 1-D-**преобразование двумерного выходного сигнала в одномерный.



**Display**– цифровой дисплей. *Назначение:* отображает значение сигнала в виде числа. *Параметры блока:* **Format** –формат отображения данных, может принимать следующие значения: **short** – 5 цифр, включая десятичную точку; **long**- 15 цифр с фиксированной точкой ; **short\_e** -5 цифр с плавающей точкой; **long\_e** -15 цифр с плавающей точкой; **bank**– банковский формат. **Decimation** – прореживание. **Sample time** – такт дискретности. **Floating display**– изменяющийся режим.



**Data Type Conversion**– преобразователь типа сигнала. *Назначение:* преобразует тип входного сигнала. *Параметры блока:* **Data Type-** тип данных выходного сигнала. **Saturate on integer overflow-** подавлять переполнение целого.

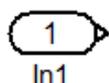


**Subsystem**– создание подсистем. *Назначение:* Подсистема-это фрагмент Simulink-модели, оформленный в виде отдельного блока. Использование подсистем при составлении модели преследует следующие цели:

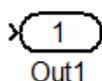
- Уменьшает количество одновременно отображаемых на экране блоков, что облегчает восприятие модели
- Позволяет создавать и отлаживать отдельные фрагменты модели, что повышает технологичность создания модели
- Позволяет создавать собственные библиотеки

- Позволяет синхронизировать параллельно работающие подсистема
- Позволяет включить в модель собственные справочные средства
- Позволяет связывать подсистему с М-файлом, обеспечивая запуск этого файла при открытии подсистемы.

*Параметры блока:* **Show port labels** – показать метки портов. **Treat as atomic unit** – считать подсистему неделимой. **Read/Write Permissions** – разрешить чтение и запись. Допустимы три опции: **ReadWrite**-чтение и запись; **ReadOnly** -только чтение; **NoReadOrWrite** ни чтение, ни записи. **Name of error callback function** – имя функции ответного вызова.

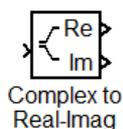


**Import**– входной порт. *Назначение:* Создает входной порт для подсистемы или выполняет считывание сигнала с рабочей области MATLAB в модель. *Параметры блока:* **Port number**-номер порта. **Port dimensions**-размерность входного сигнала. **Sample time** – такт дискретности. **Show additional parameters**-показать дополнительные параметры. **Data type** – выбор типа выходных данных. **Output data type**-тип выходных данных. **Output Scaling Mode** –способ масштабирования выходного сигнала. **Output scaling value** – величина масштаба. **Sampling mode**-режим.

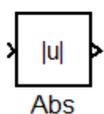


**Outport**– выходной порт. *Назначение:* Создает входной порт для подсистемы или для модели верхнего уровня иерархии. *Параметры блока:* **Port number**-номер порта. **Output when disabled**-вид сигнала на выходе подсистемы, в случае если подсистема выключена. Используется для подсистем, управляемых внешним сигналом. Может принимать следующие значения: **held**- выходной сигнал подсистемы равен последнему рассчитанному значению; **reset**- выходной сигнал подсистемы равен

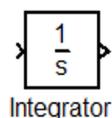
значению, задаваемому параметром `initial output`. **Initial output**-начальное значение.



**Complex to Real-Imag**– блок вычисления действительной и (или) мнимой части комплексного числа. *Назначение:* вычисляет действительную и (или) мнимую часть комплексного числа. *Параметры блока:* **Output**-выходной сигнал. Тип сигнала выбирается из списка: **Real**-действительная часть; **Image** – мнимая часть; **Real& Image**- действительная и мнимая часть.



**Abs**– блок вычисления модуля. *Назначение:* Выполняет вычисление абсолютного значения величины сигнала. *Параметры блока:* **Saturate on integer overflow**-подавлять переполнение целого. **Enable zero crossing detection**-фиксировать прохождение сигнала через нулевой уровень.



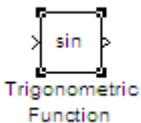
**Integrator**– интегратор. *Назначение:* Интегрирование входного сигнала. *Параметры блока:* **External reset**- внешний сброс. Выбирается из списка: **none** - нет (сброс не выполняется); **rising** - нарастающий сигнал (передний фронт сигнала); **falling**-спадающий сигнал (задний фронт сигнала); **either**- нарастающий либо спадающий сигнал; **level**-ненулевой сигнал (сброс выполняется, если сигнал на управляющем входе становится не равным нулю). **Initial condition source**- источник начального значения выходного сигнала. Выбирается из списка: **internal**-внутренний; **external**-внешний. **Initial condition**-начальное условие. **Limit output**-ограничение выходного сигнала. **Upper saturation limit**- верхний предел выходного сигнала. **Lower saturation limit**-нижний предел выходного сигнала. **Show saturation port**-показать на пиктограмме порт насыщения. Выходной сигнал данного порта

может принимать следующие значения: нуль, если интегратор не находится на ограничении; +1, если выходной сигнал интегратора достиг верхнего предела; -1, если выходной сигнал интегратора достиг нижнего предела. **Show state port** - отобразить/скрыть порт состояния блока. **Absolute tolerance**-абсолютная погрешность. **Enable zero crossing detection**-определять прохождение сигнала через нулевой уровень.



Clock

**Clock**– источник времени. *Назначение:* Формирует сигнал, величина которого на каждом шаге равна текущему времени моделирования. *Параметры блока:* **Display time**-отображение значения времени на пиктограмме блока. **Decimation** – прореживание.



Trigonometric Function

**Trigonometric Function**- тригонометрическая функция. *Назначение:* вычисляет 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.

### 5 Экспериментальное задание №1 (на 4 часа)

1. Собрать *Sim*-модель для исследования функциональной схемы фазовой автоподстройки частоты, на основе восстановления несущей частоты по схеме **Костаса**, в соответствии с рисунком 3.1.

2. Выставить параметры блоков *Sim*-модели для схемы **Костаса**. Так, для блока источника синусоидального сигнала (*Sine Wave*), выставить, например:  $Amplitude = 1/25$ ;  $Frequency = \pi/10$ ;  $Sample\ time = 1/500$ . Для *Analog Filter Design*, который находится в петле обратной связи, выставить  $Filter\ order = 2$ ,  $Passband\ edge\ frequency = \pi/6$ . Для блока усилителя (*Gain*), выставить  $Gain = -0.4$ .

3. Для оценки влияния петли фазовой автоподстройки частоты исключить петлю обратной связи и зафиксировать количество ошибок, обусловленное блоком качания несущей передатчика.

4. Добавить в канал распространения, выполненного в виде сумматора шумы на основе блока *Random Number*, выставить, например:  $Variance = 0,05$  и  $0,1$ ,  $Sample\ time = 0,0125$ . Зафиксировать количество ошибок.

5. Пронаблюдать, зафиксировать и пояснить основные осциллограммы, иллюстрирующие работу.

6. Понять принцип работы функциональной схемы фазовой автоподстройки частоты, на основе восстановления несущей частоты по схеме **Костаса**.

7. Сделать вывод о проделанной работе.

8. Оформить отчет.

### 6 Экспериментальное задание №2 (на 4 часа)

1. Собрать *Sim*-модель для исследования функциональной схемы фазовой автоподстройки частоты, на основе восстановления несущей частоты по схеме **Пистолькорса**, в соответствии с рисунком 3.3.

2. Выставить параметры блоков *Sim*-модели для схемы **Пистолькорса**. Так, для блока источника синусоидального сигнала (*Sine Wave*), выставить,

например:  $Amplitude = 1/25$ ;  $Frequency = \pi/15$ ;  $Sample\ time = 1/500$ . Для *Analog Filter Design2* выставить  $Filter\ order = 2$ ,  $Passband\ edge\ frequency = \pi/10$ .

3. Для оценки влияния петли фазовой автоподстройки частоты исключить петлю обратной связи и зафиксировать количество ошибок, обусловленное блоком качания несущей передатчика.

4. Добавить в канал распространения, выполненного в виде сумматора шумы на основе блока *Random Number*, выставить, например:  $Variance = 0,1$  и  $0,5$ ,  $Sample\ time = 0,0125$ . Зафиксировать количество ошибок.

5. Пронаблюдать, зафиксировать и пояснить основные осциллограммы, иллюстрирующие работу.

6. Понять принцип работы функциональной схемы фазовой автоподстройки частоты, на основе восстановления несущей частоты по схеме **Пистолькорса**.

7. Сделать вывод о проделанной работе.

8. Оформить отчет.

## 7 Контрольные вопросы

1. Что такое системы фазовой автоподстройки частоты (**ФАПЧ, PLL - Phase Locked Loop**)?
2. В качестве чего применяются системы **ФАПЧ**?
3. Объясните принцип работы системы **ФАПЧ**.
4. Объясните принцип работы системы **ФАПЧ** на основе схемы **Костаса**?
5. Объясните принцип работы системы **ФАПЧ** на основе схемы **Пистолькорса**?
6. Объясните принцип работы подсистемы детектора ошибок.

**Список использованных источников**

1. Радиоавтоматика. Учеб. пособие / В.Н. Бондаренко, В.Н. Тяпкин, Д.Д. Дмитриев [и др.]; ред. В.Н. Бондаренко. – Красноярск: Сиб. федер. ун-т . – 2013.– 172 с.
2. Системы связи. Подвижные системы связи. Лекции: учебно-методическое пособие [Электронный ресурс] / сост. Н. М. Боев. – Электрон. дан. – Красноярск: Сиб. федер. ун-т, 2013.- 60 с.
3. Черных, И.В. **Simulink**: среда создания инженерных приложений. / Под общ. ред. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2003.- 496 с.
4. Дьяконов, В.П. **MatLab 6.5 SP1/7 + Simulink 5/6** в математике и моделировании. Сер. Библиотека профессионала / В.П. Дьяконов. - М.: СОЛОН-Пресс, 2005.- 576 с.