

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»
Кафедра «Информатики и вычислительной техники»

И.А. Стефанова

**МОДЕЛИРОВАНИЕ СИСТЕМ
ТЕЛЕКОММУНИКАЦИЙ
В СИСТЕМЕ
MATLAB + SIMULINK**

Учебное пособие

Самара
2018

УДК: 004.42: 519.85

С

Рекомендовано к изданию методическим советом ПГУТИ,
в качестве учебного пособия протокол № 41, от 23.04.2018 г.

Рецензенты:

проф., зав. кафедры «МСИБ» ПГУТИ,
д.т.н., проф. Карташевский В. Г.,

к.т.н., проф. каф. «МЕХАТРОНИКА В АВТОМАТИЗИРОВАННЫХ ПРОИЗВОДСТВАХ» САМГУПС,
Засов В. А.

Стефанова, И. А.

С Моделирование систем телекоммуникаций в системе *MATLAB+Simulink*: учебное пособие / И. А. Стефанова. – Самара: ПГУТИ, 2018. – 112 с.

Учебное пособие содержит краткие теоретические сведения и 4 лабораторные работы, позволяющих студентам, магистрантам освоить процесс создания сложных имитационных моделей с помощью пакета *Simulink*. В пособии приводятся модели телекоммуникационных подсистем и систем, описание их работы, параметры настройки блоков моделирования и варианты заданий, позволяющие разнообразить выполнение моделей при их реализации. Кроме того, показано как с помощью интеграции простых подсистем строятся более сложные системы с использованием математической системы *MATLAB* и пакета *Simulink*.

Учебное пособие разработано по направлениям подготовки: бакалавра «11.03.02 *Инфокоммуникационные технологии и системы связи (ИКТ)*», бакалавра «11.03.01 *Радиотехника*» для использования на практических занятиях по дисциплине «Информатика». Кроме того, может быть полезно для студентов, магистрантов и аспирантов, специализирующихся в области моделирования инфокоммуникационных систем и может быть полезно при изучении устройств и систем телекоммуникаций.

ISBN

©, Стефанова И.А., 2018

Введение

В соответствии с новыми общеобразовательными стандартами высшего образования в программу бакалавриата телекоммуникационных направлений входят вопросы, связанные с освоением инновационных технологий, средств, способов и методов, направленных на создание условий для обработки, хранения и обмена информацией на расстоянии с использованием различных сетевых структур. А также освоение технических и аппаратных средств, способов и методов обработки, хранения и обмена информации в различных системах передачи данных.

В связи с этим разработаны новые рабочие программы по дисциплине «Информатика». Важное место в них занимают лабораторные работы, способствующие развитию практического применения теоретических знаний. Ранее разработанные лабораторные работы, охватывали лишь дидактические единицы, посвященные программированию и моделированию в современной системе *MATLAB*, что недостаточно в условиях новых требований к образованию бакалавра.

Система *MATLAB* с пакетом расширения *Simulink* могут помочь организации учебных курсов по изучению систем связи. Данные курсы могут содержать следующие разделы: это базовая информация о цифровых системах, цифровой обработке сигналов, изучение методов модуляции, кодов коррекции ошибок и их влияние на производительность системы, искажения, которые претерпевает сигнал при распространении по каналам связи, фильтрация сигналов, методы синхронизации. И это основные системы. Сюда могут входить и более продвинутые системы.

С помощью системы *MATLAB* + *Simulink* можно сместить внимание студентов с математических манипуляций в сторону постановки задачи, поиска решений и визуализации, и разработки моделей, основанных на фундаментальных принципах. Это позволит студентам создавать модели, исполнять их и конструировать на их базе более сложные системы.

Сейчас уже невозможно представить себе современную науку без широкого применения математического [1] и компьютерного моделирования [2, 7], суть которого состоит в замене исходного объекта его образом – моделью и дальнейшем изучении модели с помощью реализуемых на компьютерах вычислительно-логических алгоритмов. Эксперименты с моделями объектов позволяют, опираясь на мощь современных вычислительных методов и технических средств информатики, подробно и глубоко изучать объекты в достаточной полноте, недоступной чисто теоретическим подходам.

Большую помощь в этом могут оказать математическая система *MATLAB* с пакетом расширения *Simulink* [3, 4, 8]. Этот тандем дает возможность ставить задачу, основываясь на основных принципах и производить вычисление без ошибок, использовать общие методы и подходы из различных дисциплин, чтобы развивать алгоритмические навыки и принимать участие в междисциплинарных разработках. Так же делать визуализацию и непрерывно оценивать постановку задачи и предположений. Развивать концептуальное понимание того, с чем работает исследователь, моделирующей систему.

Учебное пособие содержит краткие теоретические сведения о моделировании систем и цикл из четырех лабораторных работ, направленных на создание сложных систем, а также на освоение принципов декомпозиции, моделирования и интеграции простых моделей в сложные системы.

Пособие содержит задания и методические указания по их выполнению, описание блоков моделирования и настройки их параметров, а также тестовые задания для проверки знаний, полученных в результате выполнения практических работ.

Данное учебное пособие может использоваться на практических занятиях по дисциплинам «Информатика» при подготовке бакалавров по телекоммуникационным направлениям дневной и заочной форм обучения. Кроме того, может использоваться в дисциплинах, изучающих устройства и моделирование систем связи. Пособие может быть так же полезно аспирантами и магистрам в их научной работе.

Автор выражает благодарность студентам ПГУТИ: Книжнину М. В., Павлову Е. С., Павлову К. В. за проявленный интерес к работе и участие в ней.

1. Компьютерное моделирование сложных систем

Компьютерное моделирование – это замещение исследуемого объекта (оригинала) его моделью, реализуемое с помощью компьютерной техники. Моделирование обычно выполняется с целью познания свойств оригинала путем исследования его модели, а не самого объекта. Разумеется, моделирование оправдано в том случае, когда оно проще создания самого оригинала или, когда последний по каким-то причинам лучше вообще не создавать.

Моделирование предполагает 2 основных этапа:

- 1) разработка самой модели;
- 2) исследование модели и получение выводов.

При этом на каждом из этапов решаются разные задачи и используются отличающиеся по сути методы и средства. На практике как правило применяют различные методы моделирования.

Для компьютерного моделирования важно наличие определенного программного обеспечения. При этом программное обеспечение, средствами которого может осуществляться компьютерное моделирование, может быть, как достаточно универсальным (например, обычные текстовые и графические процессоры), так и весьма специализированными, предназначенными лишь для определенного вида моделирования (пакеты прикладных программ).

Обычно в компьютерном моделировании различные виды моделирования дополняют друг друга. Так, если математическая формула очень сложна, что не дает явного представления об описываемых ею процессах, то на помощь приходят графические и имитационные модели. Компьютерная визуализация может быть намного дешевле реального создания натуральных моделей.

Если система сложна, а требуется проследить за каждым ее элементом, то на помощь могут прийти компьютерные имитационные модели. На компьютере можно воспроизвести последовательность временных событий, а потом обработать большой объем информации. Однако следует четко понимать, что компьютер является хорошим инструментом для создания и исследования моделей, но он их не придумывает. Абстрактный анализ окружающего мира с целью воссоздания его в модели выполняет сам человек.

Одной из важных проблем в области разработки и создания современных сложных технических систем является исследование динамики их функционирования на различных этапах проектирования, испытания и эксплуатации.

Сложными системами называются системы, состоящие из большого числа взаимосвязанных и взаимодействующих между собой элементов. При исследовании сложных систем возникают задачи исследования как отдельных видов оборудования и аппаратуры, входящих в систему, так и системы в целом.

При построении сложных систем надо понимать, что абсолютно точных моделей не бывает. Обеспечение требований точности и адекватности модели приводит к увеличению объема и сложности моделирующей программы.

Сложные системы, как правило, состоят из большого числа элементов, выполняющих различные функции. Получение модели достаточно простого вида для таких систем чаще всего невозможно. Поэтому в этом случае модель системы строят по модульному принципу, т.е. систему представляют состоящей из элементов, блоков и подсистем. Каждая система может быть разделена на подсистемы, а подсистемы – на блоки, а блоки в свою очередь – на элементы. Таким образом, сложная модель функционально подразделяется на подмодели.

Основная цель разбиения системы – это получение элементов, математическое описание которых может быть выполнено. Такой подход достаточно эффективен и может быть легко осуществлен. При этом можно строить и совершенствовать модель итерационным методом, добавляя к основной схеме блок за блоком. Построение модели из блоков дает возможность экспериментировать при ее реализации в процессе машинной имитации.

Построение модели сложной системы начинают с анализа отдельных «элементарных» процессов, протекающих в объекте моделирования, и выделения отдельных блоков, соответствующих какому-либо элементу или процессу системы. Для этого функции модели разделяют на логические подфункции с более высоким уровнем детализации. Если какой-либо первоначально выбранный блок оказывается чрезмерно сложным, его расчленивают (с сохранением связей) на конечное число более мелких

блоков нижнего уровня, образующих подсистему, соответствующую разделенному блоку. Этот процесс деления на подсистемы и блоки продолжается до необходимого уровня детализации описания системы, т.е. до получения таких блоков, которые в условиях данной задачи будут признаны достаточно простыми и удобными для непосредственного математического описания. Блоки, не подлежащие дальнейшему расчленению на части, являются элементами сложной системы.

Таким образом, в общем случае сложная система является многоуровневой, состоящей из взаимосвязанных элементов, объединяемых в блоки и подсистемы различных уровней. Используя современные языки программирования, можно получить модель, максимально приближенную к изучаемой системе.

Чтобы построить сложную модель, надо для себя задать и решить два вопроса:

1. Что мы хотим узнать? – тогда мы поймем модель какого типа нам нужна.
2. Что нам не важно? – чем мы можем пренебречь при построении модели, тогда мы определим уровень абстракции модели.

Как же моделировать сложный объект на практике? Его надо разделить на части. В свою очередь каждую часть можно разделить еще на другие части и так далее, пока не дойдем до компонент. Этот процесс называется *декомпозицией*. Теперь каждый объект становится уже не таким сложным и его можно смоделировать отдельно. Затем каждый объект можно усложнить и снова отладить модель, затем смоделировать еще более сложную композицию и опять отладить и так шаг за шагом можно получить более сложную модель. Этот процесс называется *интеграцией*. Проводя многократную интеграцию более простых компонент можно получить систему довольно большой степени сложности.

Фактически для создания сложной системы только нужно ответить на два вопроса и выполнить три действия, приведенные на рис. 1.



Рис. 1. Процесс создания сложной модели

При составлении модели, состоящей из отдельных блоков, в зависимости от назначения математической модели возможно применить один из двух подходов:

Структурный подход – это моделирование внутреннего механизма блока. В этом случае модель должна отражать механизм взаимодействия узлов, элементов и деталей рассматриваемого блока; должны моделироваться как внутренняя структура блока, так и функционирование его элементов. Этот подход должен применяться тогда, когда задачей моделирования является, например, проверка структуры блока, правильности взаимодействия его частей и общей логики работы модели. Критерием правильности структуры блока является выполнение блоком функции или функций, заданной в ходе моделирования.

Функциональный подход – это моделирование функции блока. В этом случае блок рассматривается как «черный ящик», его внутренний механизм может не моделироваться; задается лишь передаточная функция блока в целом. Этот подход применим к тем блокам, внутреннее содержание которых не описывается данной моделью. Такие блоки рассматриваются как неделимые элементы моделируемой системы.

Выбор того или иного подхода к моделированию функциональных блоков зависит от поставленной задачи. В ряде случаев моделирующий алгоритм бывает настолько сложным для реализации с помощью имеющихся в наличии вычислительных средств, что требуется измерить формулировку исходной задачи моделирования для упрощения математического описания. Это упрощение часто достигается за счет снижения точности математической модели путем сокращения полноты математического

описания при исключении из модели части параметров или взаимодействий моделируемого объекта.

При декомпозиции блоков сложных телекоммуникационных систем удобно разделить их на типовые элементы, в которых протекают сходные между собой электронные процессы. Для выделения типовых элементов (процессов) и определения их природы используют несколько основных критериев:

1. общность математического описания (модели) процессов, т. е. идентичность материальных и энергетических связей. Такая общность модели учитывает электрофизические особенности процессов;
2. общность аппаратурно-технологического оформления процессов, отражающая их целевое назначение и условия реализации.

Взаимодействие элементов в процессе функционирования сложной системы рассматривается как результат совокупности воздействий каждого элемента на другие элементы. Воздействие, представленное некоторым набором характеристик, называют сигналом. Каждый элемент системы в общем случае может принимать входные сигналы и выдавать выходные. Сигналы передаются по каналам связи между элементами сложной системы.

Совокупность элементов некоторой системы может рассматриваться как ее подсистема. Обычно подсистемы являются некоторыми самостоятельно функционирующими частями системы. Например, в многоканальной аппаратуре связи можно выделить подсистемы, соответствующие отдельным источникам сообщения или абонентским линиям. Правильное выделение подсистем сложной системы способствует упрощению расчетов и построению модели исследуемого объекта при его моделировании и более наглядной интерпретации результатов моделирования. Модель подсистемы составляется в виде структуры из моделей блоков и элементов и целиком входит в полную модель системы. Поскольку подсистема – это самая крупная, функционирующая отдельно от общих связей, структурная единица, важным этапом работы является ее декомпозиция, основанная на выявлении и оценке различных воздействующих факторов. После исследования каждой части подсистемы в отдельности производят объединение полученных сведений в единое целое.

2. Имитационное моделирование

Имитационное моделирование используется для анализа и синтеза сложных систем в тех случаях, когда аналитическая модель слишком сложна или ее не удается разработать. Имитационная модель [5] отображает процессы в системе при наличии внешних воздействий на систему. При этом связь между параметрами внутренних, внешних элементов и выходными параметрами задана в ней неявно в виде алгоритма моделирования.

При имитационном моделировании реализующий модель алгоритм воспроизводит процесс функционирования системы во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательностью их протекания во времени, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы. Время в такой модели – один из ключевых моментов, позволяющих исследовать систему в процессе функционирования.

Объектами имитационного моделирования могут быть локальные и глобальные вычислительные сети, телефонные и телеграфные сети, системы энергоснабжения и т.п.

Анализ работы подобных систем основан на изучении процесса прохождения сигналов по цепи.

Таким образом, работа с имитационной системой представляет собой эксперимент, осуществляемый на компьютере. Преимуществами имитационного моделирования являются:

- Большая близость к реальной системе, чем у математических моделей. Промежуточные результаты имитационного моделирования, в отличие от математического, имеют четкий физический смысл. Это облегчает обнаружение ошибок в программе – в особенности при работе в интерактивном режиме;

- Блочный принцип дает возможность верифицировать каждый блок до его включения в общую систему;

- Использование зависимостей более сложного характера, не описываемых простыми математическими соотношениями. Имитационные модели позволяют достаточно просто учитывать такие

факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и т.п., которые часто создают трудности при аналитических исследованиях.

Перечисленные достоинства определяют и недостатки:

- построить имитационную модель дольше, труднее и дороже, чем математическую;
- для работы с имитационной системой необходимо наличие подходящей по классу ЭВМ;
- взаимодействие пользователя и модели (интерфейс) должно быть не слишком сложным, удобным и хорошо известным;
- построение имитационной модели требует более глубокого изучения реального процесса.

Метод имитационного моделирования основан на использовании имитационных моделей, реализуемых на компьютере, для исследования процесса функционирования сложных систем. Для реализации метода необходимо разработать специальный алгоритм моделирования. В соответствии с этим алгоритмом в компьютере вырабатывается информация, описывающая элементарные процессы исследуемой системы с учетом взаимосвязей и взаимных влияний. При этом моделирующий алгоритм строится в соответствии с логической структурой системы с сохранением последовательности протекающих в ней процессов и отображением основных состояний системы.

Основными этапами метода имитационного моделирования являются:

1. моделирование входных и внешних воздействий;
2. воспроизведение работы моделируемой системы (моделирующий алгоритм);
3. интерпретация и обработка результатов моделирования.

Перечисленные этапы метода имитационного моделирования многократно повторяются для различных наборов входных и внешних воздействий, образуя внутренний цикл моделирования. Во внешнем цикле организуется просмотр заданных вариантов моделируемой системы. Процедура выбора оптимального варианта управляет просмотром вариантов, внося соответствующие коррективы в имитационную модель и в модели входных и внешних воздействий.

Метод имитационного моделирования позволяет решать, как простые задачи, так и более сложные. Исследуемая система может одновременно содержать элементы непрерывного и дискретного действия, быть подверженной влиянию многочисленных случайных факторов сложной природы, описываться весьма громоздкими соотношениями и т.п. Этот метод не требует создания специальной аппаратуры для каждой новой задачи и позволяет легко изменять значения параметров исследуемых систем и начальных условий. Эффективность метода имитационного моделирования тем более высока, чем на более ранних этапах проектирования системы он начинает использоваться.

Следует, однако, помнить, что метод имитационного моделирования является численным методом и обладает существенным недостатком – его решение всегда носит частный характер. Решение соответствует фиксированным значениям параметров системы и начальных условий. Для анализа системы приходится многократно моделировать процесс ее функционирования, варьируя исходные данные модели. Таким образом, для реализации имитационных сложных моделей необходимо наличие ЭВМ высокой производительности.

Как показывает практика натуральные испытания сложных систем связаны с большими затратами времени и средств. Проведение испытаний предполагает наличие готового образца системы или ее физической модели, что затрудняет использование этого метода на этапе проектирования системы.

Исследование функционирования сложных систем с помощью компьютерного моделирования их работы помогает сократить время и средства на разработку. Затраты рабочего времени и материальных средств на реализацию метода имитационного моделирования оказываются незначительными по сравнению с затратами, связанными с натурным экспериментом. Результаты моделирования по своей ценности для практического решения задач близки к результатам натурального эксперимента.

Для компьютерного моделирования системы необходимо записывать моделирующий алгоритм на одном из входных языков. В качестве входных языков для решения задач моделирования могут быть с успехом использованы универсальные алгоритмические языки высокого уровня, *Cu*, *MATLAB* и др.

3. Основные сведения о системе MATLAB+ Simulink

Эффективное использование имитационного моделирования невозможно без применения компьютера и соответствующего программного обеспечения.

Система *MATLAB* является одной из самых крупных и мощных систем компьютерной математики [2] и предназначена для автоматизации математических и научно-технических расчетов с данными, представленными в матричной форме.

Система *MATLAB* поставляется совместно с главным расширением *Simulink*, обеспечивающим визуально-ориентированную подготовку имитационных моделей систем различного назначения и выполнение их моделирования.

Пакет *MATLAB* поставляется как комплекс *MATLAB + Simulink + Toolboxes + Blocksets*. Базируясь на принципах визуально-ориентированного программирования, *Simulink* позволяет выполнять моделирование сложных устройств с высокой степенью достоверности и наглядными средствами представления результатов.

Пакет *Toolboxes* – является расширением *MATLAB* и содержит наборы функций для определенной предметной области. Для телекоммуникационного направления представляют интерес пакеты: *Communications Toolbox*, *Signal Processing Toolbox*, *Control System Toolbox*, *Filter Design Toolbox*, *RF Toolbox*, *Statistics Toolbox*.

Пакет *Blocksets* – является расширением пакета *Simulink* и содержит наборы блоков, методов и функций для определенной предметной области. Для телекоммуникационных задач используются такие пакеты, как *Communications Blockset*, *Signal Processing Blockset*, *RF Blockset*, *Simscape*, *SimElectronics* и другие.

Главное расширение *MATLAB + Simulink* служит для имитационного моделирование моделей, состоящих из графических блоков с заданными свойствами (параметрами). Компоненты моделей, в свою очередь, являются графическими блоками и моделями, которые содержатся в ряде библиотек и с помощью мыши могут переноситься в основное окно и соединяться друг с другом необходимыми связями. В состав моделей могут включаться источники сигналов различного вида, виртуальные регистрирующие приборы, графические средства анимации.

Для начала работы с *Simulink* достаточно активизировать

кнопку *Simulink* в панели инструментов. При этом откроется окно браузера библиотек *Simulink Library Browser*, показанной на рис. 2 внутри окна системы MATLAB. Браузер включает в себя базовые блоки и блоки дополнения.

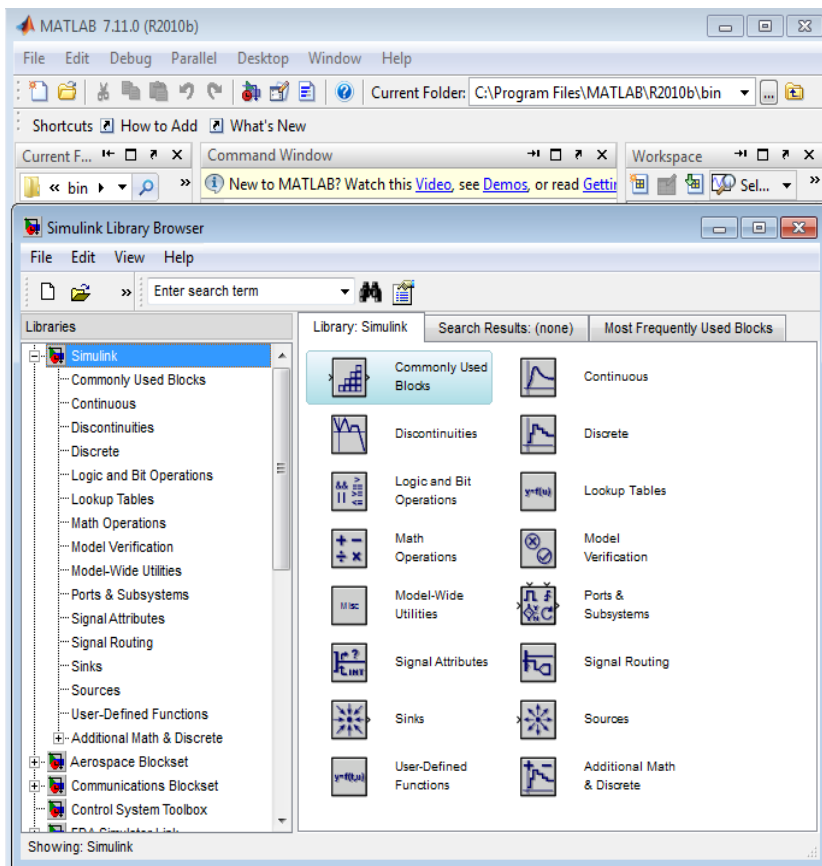


Рис. 2. Окно браузера библиотек Simulink в окне MATLAB

Интерфейс *Simulink* полностью соответствует стилю интерфейса приложений Windows. Меню системы содержит следующие пункты:

File, Edit, View – общепринятые для Windows-приложений команды и операции,

Simulation – управление процессом моделирования,

Format – операции форматирования модели,

Tools – управление видом анализа.

В окне браузера содержится иерархическое дерево компонентов библиотек *Simulink* рис.2 слева, справа – содержимое выбранной в левой панели папки с блоками. Для просмотра нужного раздела библиотеки достаточно выделить его мышью – в правой части окна *Simulink Browser Library* появится набор пиктограмм компонентов активного раздела библиотеки.

Все модели создаются в следующей последовательности:

1) запустить *MATLAB*;

2) нажать на кнопку «*Simulink*» в панели инструментов системы *MATLAB*, при этом появится окно браузера библиотек *Simulink Library Browser* (рис. 2);

3) создать модель командой *File => New => Model*. Это приведет к созданию пустого окна модели (*untitled*). С помощью мыши, расположить окно браузера и окно модели таким образом, чтобы они не перекрывали друг друга (см. рис. 3);

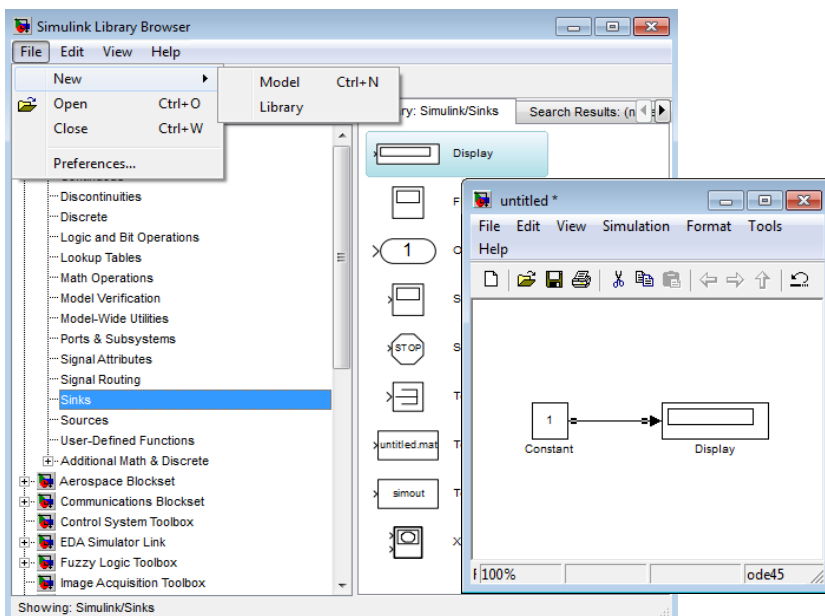


Рис. 3. Окно браузера и модели *untitled*

4) с помощью технологии *drag-and-drop* перетащить мышью из папок браузера в окно модели нужные блоки;

5) соединить блоки коннекторами (*connector* – соединительная линия). Для этого нужно указать курсором на «выход» блока, нажать левую клавишу мыши и, не отпуская ее, провести линию к входу другого блока. После чего клавишу мыши отпустить. При этом в модели отобразится коннектор со стрелкой. Удаляется линия с помощью ее выделения и нажатия клавиши *<Delete>*;

6) установить для каждого блока соответствующие свойства. Для этого нужно по блоку сделать двойной щелчок мыши, что приведет к появлению окна со свойствами блока (*Parameters*). В открывшемся окне установить нужные параметры блока;

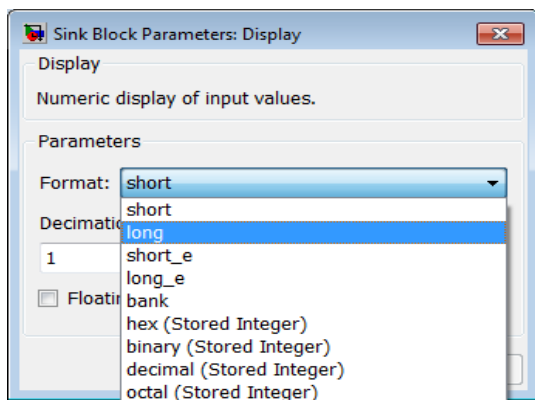


Рис. 4. Окно установки параметров блока *Display*

7) сохранить модель в виде файла командой *File => Save as* в своей папке. Имена папки и файла должно содержать **латинские** символы (допускаются цифры, но не на первом месте);

8) включить симулирование (моделирование) командой *Simulation => Start* (или кнопкой на панели инструментов). Проконтролировать результат моделирования.

Вместе с рабочим окном *Simulink* выводится окно с перечнем разделов основной библиотеки компонентов. Это окно открывает доступ к другим пакетам компонентов (*Blocksets & Toolboxes*) (рис. 3) и примерам их применения (*Demos*).

3.1. Применение подсистем

Одной из особенностей *Simulink* является возможность создания моделей из блоков, содержащих встроенные подмодели (субблоки). Внутри подсистемы первого уровня, в свою очередь, могут располагаться подсистемы второго уровня и т.д. Это напоминает ситуацию, когда сложная система набирается из отдельных систем – модулей, каждый из которых, в свою очередь, является системой или устройством.

Такой принцип конструирования сложных моделей имеет важные достоинства:

- появляется возможность разбивки решаемой задачи на ряд более мелких задач, решаемыми подсистемами;
- каждая подсистема может отлаживаться отдельно и использоваться в полной системе уже после отладки;
- существенно упрощается вид основной модели за счет маскирования второстепенных блоков;
- облегчается модификация полной модели за счет модификации ее более простых подсистем.

3.2. Создание подсистемы из части основной модели

Simulink дает возможность выделить в любой модели некоторый блок (или группу блоков) и тут же превратить его (их) в подсистему. Для создания подсистемы – компактного представления модели необходимо выделить нужные блоки в виде прямоугольника и использовать команду пункта меню **Edit => Create Subsystem** (Редактирование => Создание подсистемы);

После выполнения команды *Create Subsystem* на месте выделенных блоков появится блок подсистемы.

Пример создания подсистемы приведен на рис. 5. Допустим мы решили создать блок, который преобразует входной синусоидальный сигнал в три типа сигналов: квантованный сигнал, сигнал с задержкой во времени, сигнал с ограничением. Создаем модель, приведенную на рис. 5. На рисунке показано, какие блоки выделяются для создания подсистемы, это Quantizer, Relay, Saturation и мультиплексор Mux. После выделения блоков для создания подсистемы используется команда *Create Subsystem*. На месте выделенных блоков появится блок подсистемы Рис. 6.

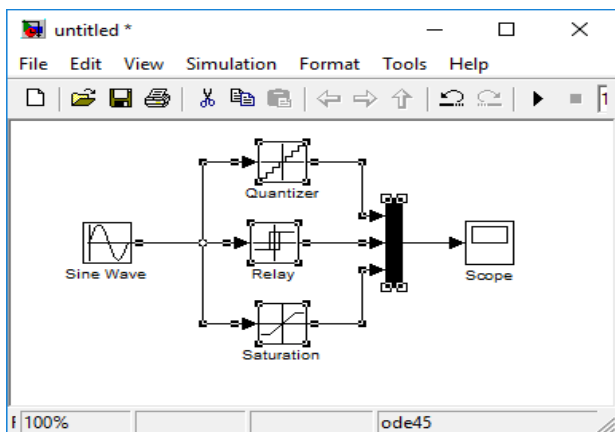


Рис. 5. Пример создания подсистемы

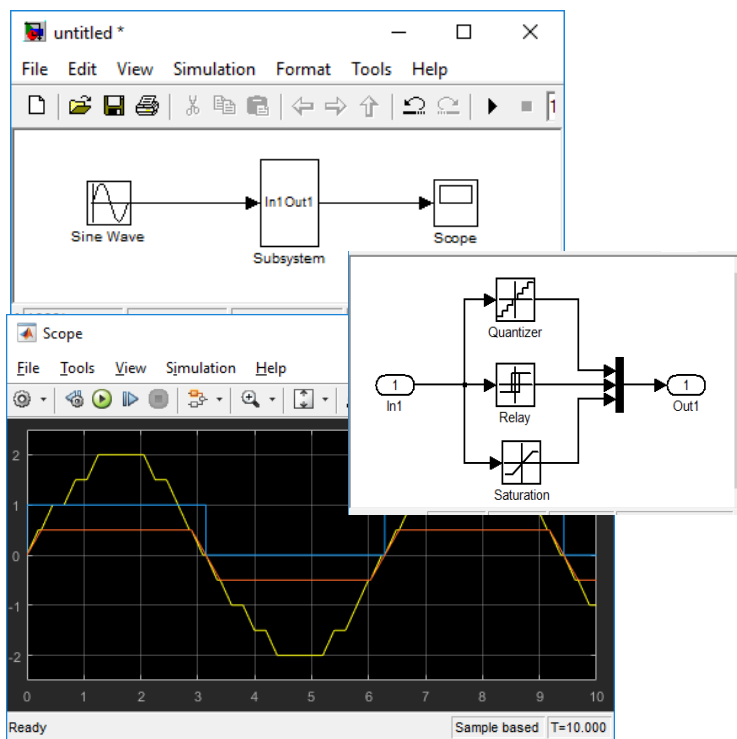


Рис. 6. Модель с подсистемой

Для вызова, просмотра и модификации подсистемы достаточно навести на нее курсор мыши и дважды щелкнуть левой кнопкой мыши.

Подсистема отличается от основной модели только наличием порта ввода **In** и порта вывода **Out**. Порты обозначаются овалом с номером внутри и подписями. Возможно несколько входов In1, In2,...и несколько выводов Out1, Out2,...С помощью этих портов подсистема включается в состав основной модели. Порты ввода служат для приема данных подсистемами, а порты вывода – для передачи выходных данных в основную модель.

На рис. 7 показана подсистема SM, где группа блоков, выполняет функции суммирования двоичных чисел и имеет три входа и два выхода. С помощью входных портов в подсистему подаются двоичные коды и перенос, с помощью выходных – результаты суммирования и переноса подаются в основную схему.

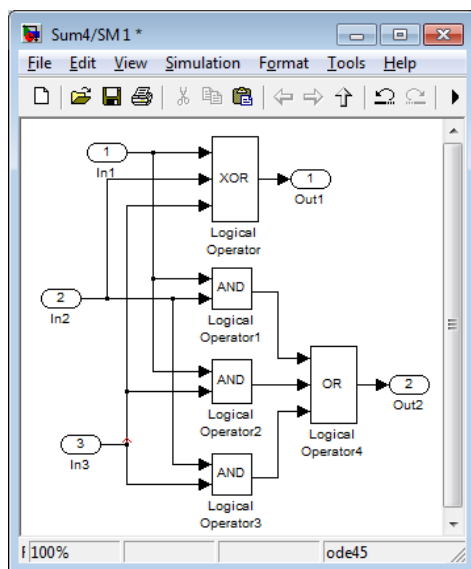


Рис. 7. Пример создания подсистемы

Подсистема имеет окно свойства. Его можно вызвать для просмотра и редактирования с помощью команды меню **Edit => Model Properties** (Редактирование => Свойства модели).

Порты ввода вывода имеют свои окна параметров.

3.3. Библиотека Ports & Subsystems

Для построения моделей с подсистемами *Simulink* имеет библиотеку Ports & Subsystems, окно которой представлено на рис. 8. Это одна из самых крупных библиотек.

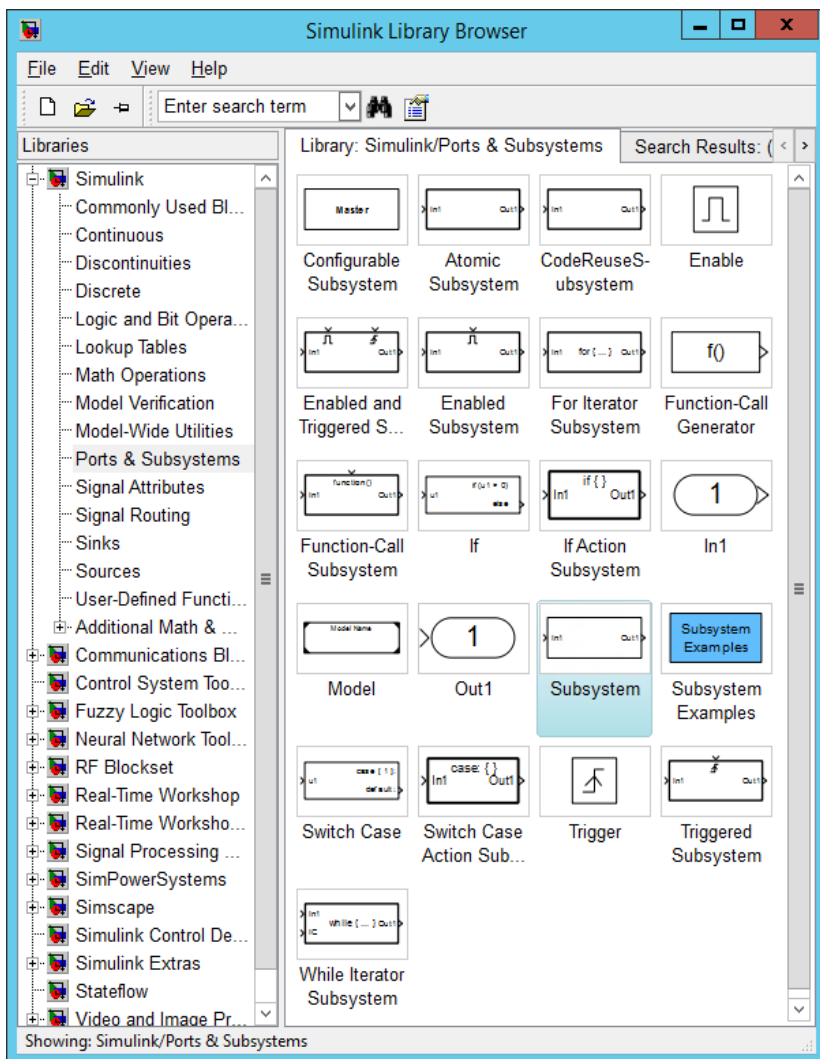


Рис. 8. Окно с блоками библиотеки портов и подсистем

Библиотека содержит:

- блоки входных и выходных портов для связи с моделью,
- блоки организации запуска *Trigger, Enable, Function-call Generator*,
- неделимая подсистема *Automatic Subsystem*. При моделировании блоки такой системы моделируются в первую очередь, а затем уже моделируются другие блоки.

Модели, содержащие блоки подсистем *Subsystem*, имеют произвольный порядок моделирования блоков.

3.4. Особенности применения подсистем

В *Simulink* выделены в отдельный класс так называемые *виртуальные подсистемы*. Они содержат макросы и готовятся на языках С или С++. Они поддерживают графическую иерархию в модулях. Ряд таких созданных подсистем (рис. 8) включен в раздел библиотеки *Simulink – Port & Subsystems* [4]. К ним относятся:

Configurable Subsystems – блок конфигурации подсистем;

If – блок задания условных операций;

Subsystem – блок ввода подсистемы в модель;

Subsystem Examples – открытие окна с примерами описания семантики подсистем. Дает детальное ознакомление с системами;

Switch Case – блок задания оператора переключений.

Эти виртуальные системы являются инструментальными средствами для конфигурации подсистем и задания управляющих структур.

Не виртуальные подсистемы (рис. 8) выделяются рамкой из линий двойной толщины. К ним относятся:

Automatic Subsystem – это элементарные подсистемы, которые выполняются как единое целое;

Enable Subsystems – это Е-подсистемы, выполняемые только при наличии на специальном управляющем порту разрешающего сигнала, уровень которого превышает 0;

Trigger Subsystems – это триггерные Т-подсистемы, управляемые по триггерному, управляющему порту;

Function-call Subsystems – подсистема запроса функции. Эти подсистемы используются как m-файлы, но готовятся как блоки *Simulink*;

Enabled and Trigger Subsystems – это ЕТ-подсистемы, имеющие свойства Е-подсистем и Т-подсистем;

Action Subsystems – это подсистемы, действия которых задается операцией и управляются инициатором подсистем, например, блоком *If* или *Switch Case*;

While-Subsystems – подсистемы, выполняющие итерационные действия под управлением цикла типа *while* (пока);

For-Subsystems – подсистемы, выполняющие итерационные действия под управлением цикла типа *for* с заданным числом повторений итераций.

3.5. Создание подсистемы на основе блока *Subsystem*

Путь создания подсистемы из части основной модели не всегда приемлем и не всегда удачен. На практике зачастую сначала создают подсистему, а затем уже общую модель системы.

Рассмотрим пример создания функционального генератора разно полярных прямоугольных, треугольных и трапецеидальных импульсов, которых нет в библиотеке *Simulink*. Для этого:

1. Откроем окно для будущей модели функционального генератора. И перетащим в него блок *Subsystem*.



2. Наведем курсор мыши на этот блок и дважды щёлкнем мышью. Откроется окно подсистемы (рис.9), содержащее лишь блоки In1 и Out1 соединенные коннектором.

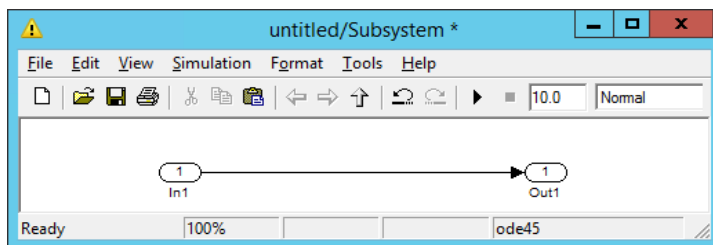


Рис. 9. Создание окна подсистемы.

3. Выделим блок In1 и коннектор и удалим их клавишей

<Delete>. Открыв библиотеку *Simulink* можно строить подсистему.

4. Для генерации меандра достаточно вычесть из стандартного однополярного прямоугольного сигнала (длительность импульса и паузы равны) с амплитудой 2 константу, равную 1. Для получения треугольного сигнала можно использовать интегратор. И для получения разно полярный сигнал с единичной амплитудой, надо вычесть из него константу 0,5 и усилить вдвое масштабирующим блоком *Gain*. На рис. 10 показано создание подсистемы, генерирующей прямоугольные и треугольные разно полярные импульсы. В подсистему надо добавить еще один блок выхода *Out2*.

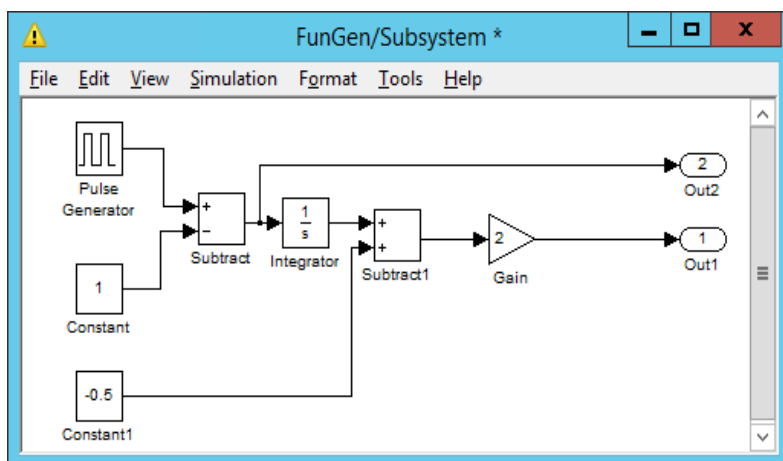


Рис. 10. Создание подсистемы

5. В окне основной модели находится один блок *Subsystem*, созданный выше. Теперь добавим в модель интегратор, усилитель и осциллограф с 3 входами. В результате модель приобретает вид, изображенный на рис. 11.

6. Запустить созданную модель. При необходимости настроить свойства блоков до желаемого результата. Осциллограммы на рис. 12 показывают, что построенная модель соответствует поставленной задаче. Модель формирует все три типа сигналов: треугольный, прямоугольный и трапецидальный.

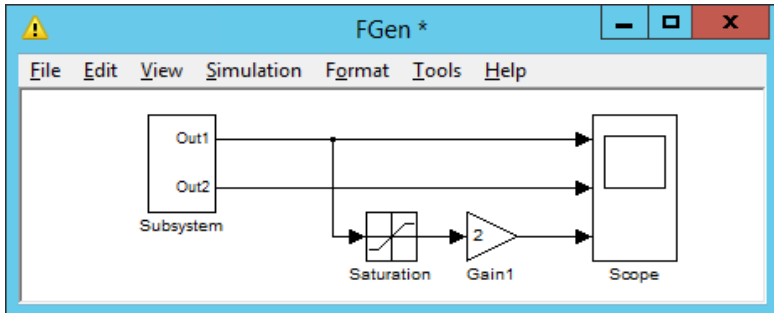


Рис. 11. Модель функционального генератора

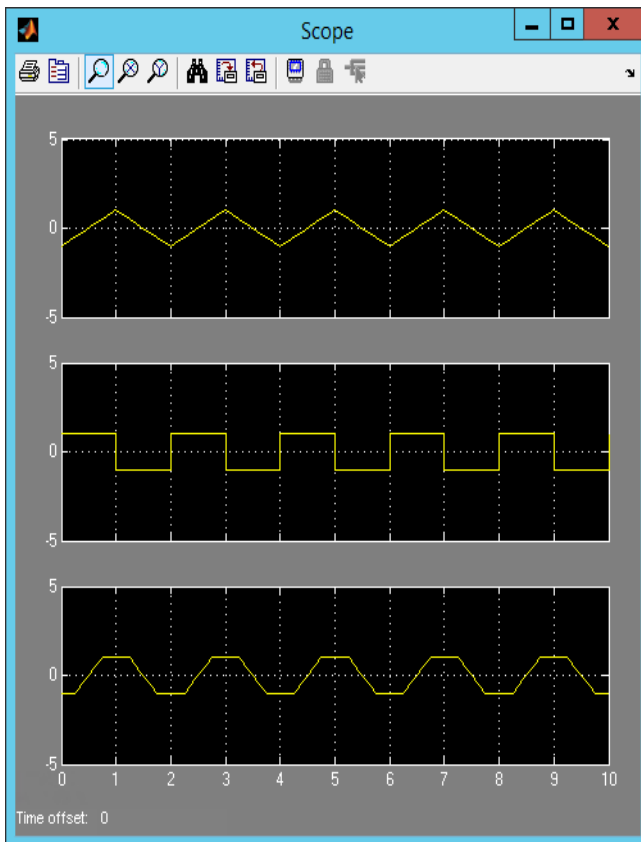


Рис. 12. Осциллограммы работы модели

3.6. Управляемые подсистемы

У управляемых подсистем поведение задается внешним воздействием. Эти системы обеспечивают упрощение создания сложных систем и дают решение задачи синхронизации параллельных процессов. В *Simulink* для создания управляемых подсистем служат блоки *Enable* и *Trigger*, которые размещаются только в подсистемах.

В зависимости от логики работы управляемых подсистем они делятся на три основных типа:

Е-подсистемы – подсистемы, управляемые блоком *Enable* (Разрешение). Этот блок воспринимает управляющий сигнал на входе *Enable* и разрешает или запрещает работу подсистемы;

Т- подсистемы – подсистемы, управляемые блоком *Trigger* (Переключение). Этот блок разрешает работу подсистемы только во время заданного события на управляющем входе *Trigger*;

ЕТ- подсистема – подсистемы, управляемые как блоком *Enable*, так и блоком *Trigger*. Она является комбинацией двух первых управляемых подсистем. Приоритетным является вход *Enable*. С блоками *Enable* и *Trigger* ЕТ- подсистемы можно работать независимо друг от друга (рис. 13).

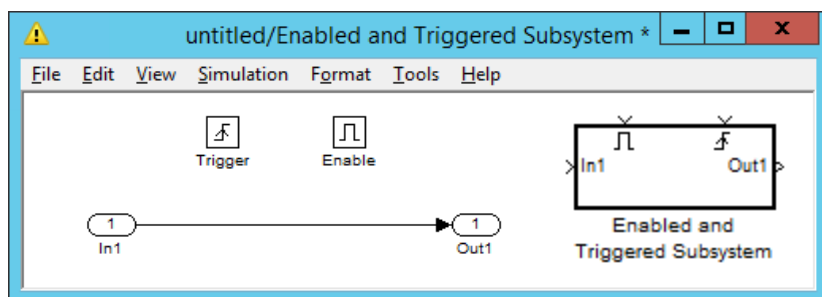


Рис. 13. ЕТ – подсистема

При работе с этими блоками необходимо сначала установить нужную подсистему в модель, затем настроить ее свойства, используя окно параметры подсистемы, собрать саму подсистему и запустить модель на исполнение и произвести испытание на ее работоспособность. Примеры работы с управляемыми подсистемами приведены в 5 главе.

3.7. Маскирование подсистем

Маскирование подсистем позволяет оформить подсистему как полноценный библиотечный блок, т.е. добавить в него окно параметров, пиктограмму, справочный материал и атрибуты [11].

Маскирование дает следующие преимущества:

- расширяет возможности пользователя по управлению параметрами модели;
- позволяет создавать дружелюбный интерфейс;
- повышает наглядность блок-диаграмм;
- расширяет возможности построения сложных моделей;
- защищает модель от несанкционированного доступа.

Маскирование подсистемы выполняется с помощью редактора маски *Mask Editor*. Для запуска редактора необходимо выделить маскируемую подсистему и выполнить команду **Edit => Mask Subsystem**. После запуска *Mask Editor* на экран будет выведено окно редактора (рис. 14), имеющее 4 вкладки:

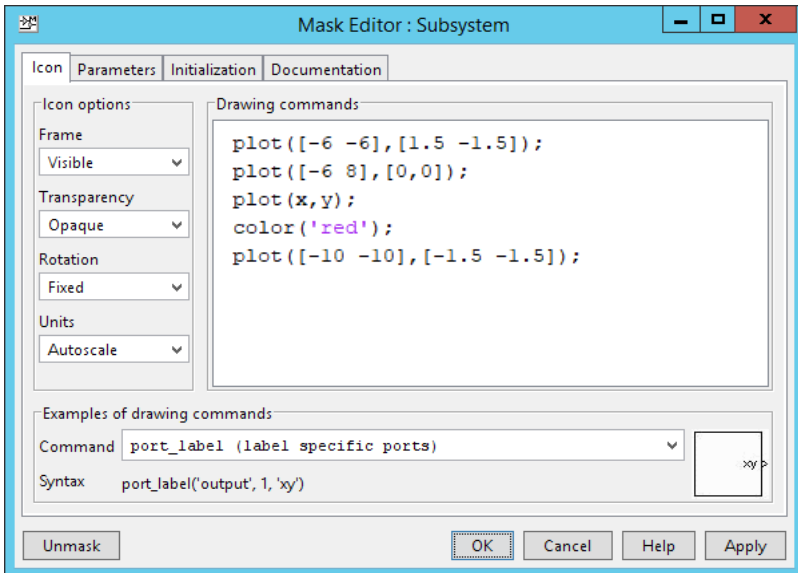


Рис. 14. Окно диалога *Mask Editor*, вкладка *Icon*

- *Icon* (Пиктограмма) обеспечивает создание пиктограммы подсистемы;

- *Parameters* (Параметры) дает возможность создать окно диалога для ввода параметров;
- *Initialization* (Инициализация) задает выражения для инициализации блока;
- *Documentation* (Документация) позволяет ввести описание блока и создает справку.

В нижней части окна имеется 5 кнопок управления *Editor*:

- *Unmask* – снять маску с подсистемы;
- *OK* – сохранить внесенные изменения и закрыть окно;
- *Cancel* – отменить внесенные изменения и закрыть окно;
- *Help* – открыть окно справки редактора маски;
- *Apply* – сохранить внесенные изменения и закрыть окно.

Повторный вызов окна осуществляется командой ***Edit => Edit Mask***. После маскирования подсистемы двойной щелчок по ее пиктограмме открывает окно задания параметров, а не его модели. Открыть саму подсистему для редактирования можно опцией ***Edit => Look under mask*** или контекстного меню.

Пиктограмма подсистемы создается с помощью вкладки *Icon*, представленная на рис. 14.

Вкладка содержит три области:

Drawing commands (операторы графики). Окно ввода графических операторов, куда вводятся команды рисования допустимые на языке *MATLAB*. На рис. 14 графические операторы строят оси, форму сигнала, точку красного цвета в нижнем углу.

Для вывода текста в подсистему достаточно использовать в окне *Drawing commands* команду `disp('text')`. Пример вывода текста в пиктограмму приведен на рис. 15.

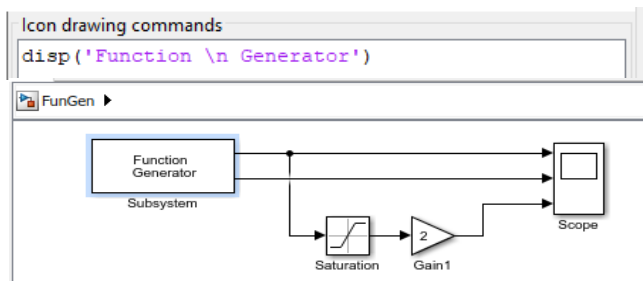


Рис. 15. Модель функционального генератора

Для отображения на пиктограмме некоторого изображения (рис. 2.6 и 2.7) из графического файла используют оператор: `image(imread('<имя файла>'))`.

Icon options (опции пиктограммы). В этой области расположены 4 окна раскрывающихся списков для задания свойств пиктограммы.

Frame – отображение рамки (видимая / невидимая);

Transparency – прозрачность (непрозрачна / прозрачна);

Rotation – вращение (фиксированная / вращение;

Units – способ рисования (автоматическое масштабирование, когда рисунок занимает максимально возможную площадь внутри пиктограммы / нормируемое относительно левого нижнего угла (0, 0) и правого верхнего угла (1, 1) в пикселях).

Examples of drawing commands (примеры операторов графики). В этой области имеется окно раскрывающегося списка *Command*, с помощью которого можно выбрать нужный оператор, и строка *Syntax*, где приводится синтаксис выбранного оператора. Здесь же расположено окно, в котором отображается построенная пиктограмма.

Окно задания параметров создается с помощью вкладки *Parameters* редактора маски, приведенное на рис. 16.

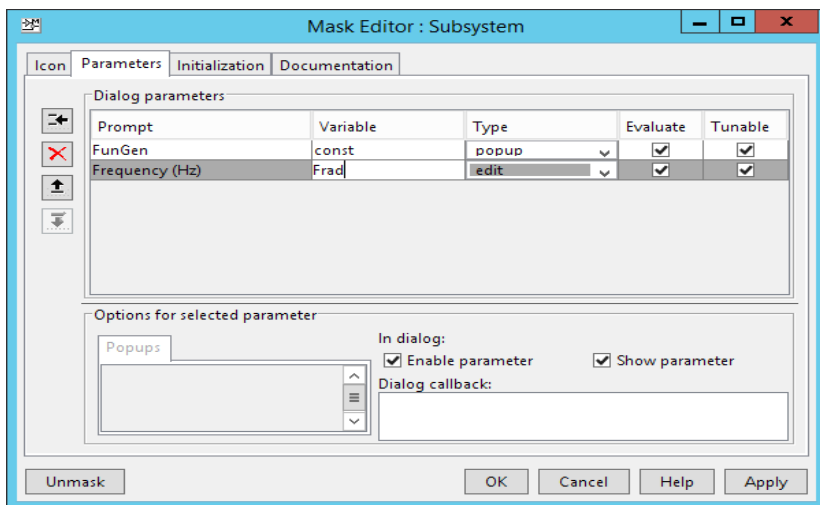


Рис. 16. Окно диалога *Mask Editor*, вкладка *Parameters*

В разделе *Dialog Parameters* необходимо внести:

- *Prompt* – текстовое описание параметров, обычно используется название подсистемы;
- *Variable* – имя переменной, с помощью которой задается данный параметр;
- *Type* – тип элемента интерфейса: *Edit* – редактируемое поле ввода, *Checkbox* – флажок, *PopUp* – раскрывающийся список;
- *Evaluate* – вычислитель. Установка флажка означает что в строке ввода параметра можно вводить вычисляемое выражение, при отсутствии флажка переменной присваивается текстовое выражение;
- *Tunable* – изменить. Установка флажка означает, что пользователь может изменить значение параметра в процессе моделирования.

В левой верхней части вкладки находятся кнопки *Add* (Добавить), *Delete* (Удалить), *Move Up* (передвинуть вверх), *Move down* (Передвинуть вниз), позволяющие добавить или удалить новый параметр окна диалога или изменить его положение в общем списке параметров.

Для создания поля ввода параметра с его описанием (рис. 17), необходимо выполнить следующие действия:

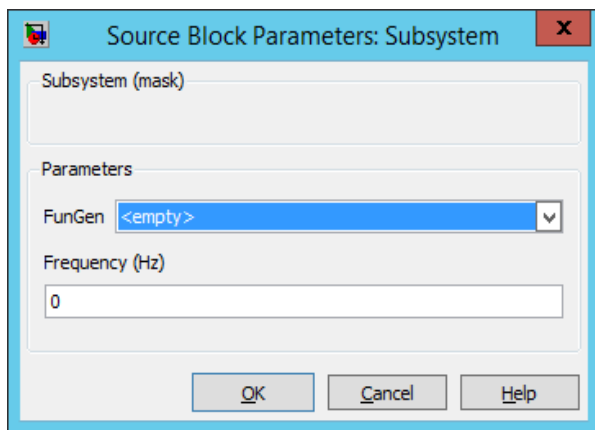


Рис. 17. Вид окна диалога, созданного с помощью редактора маски

1. Нажать кнопку *Add* (Добавить);
2. Внести описание параметра в поле *Prompt* (текстовое);
3. Задать имя переменной, связанной с параметром в поле *Variable* (Переменная);
4. Выбрать тип интерфейса из списка *Type*;
5. Установить свойства параметров *Evaluate*, *Tunable*;
6. Задать свойства для каждого параметра в панели *Option for selected parameter*.

Команды инициализации следует задавать на вкладке *Initialization* (рис. 18). Команды представляют собой обычные выражения на языке *MATLAB* и могут включать любые конструкции, допустимые в языке *MATLAB*.

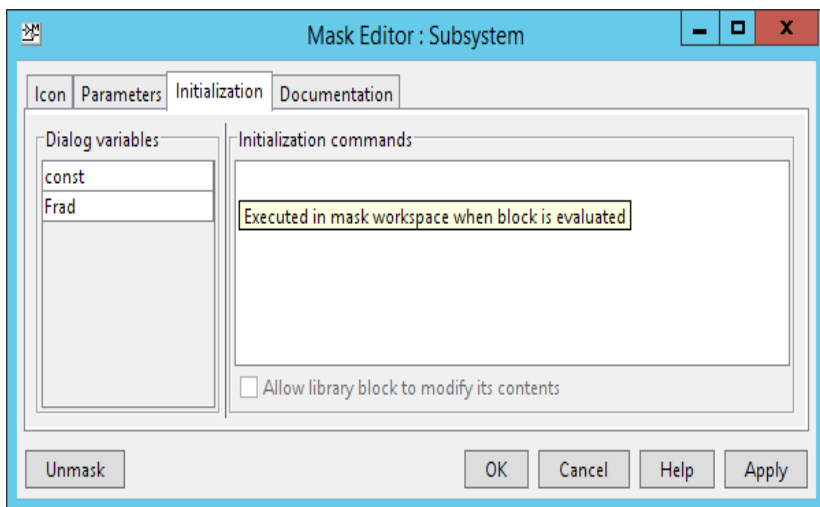


Рис. 18. Окно диалога *Mask Editor*, вкладка *Initialization*

На вкладке *Initialization* находятся две панели:

Dialog variable (переменные окна диалога) – список переменных, заданный в поле *Variable* на вкладке *Parameters*. Эти переменные можно использовать в командах инициализации.

Initialization commands (команды инициализации) – выражение на языке *MATLAB*, которые выполняются при инициализации модели.

3.8. Пакет расширения *Signal Processing*

Для специалистов телекоммуникационного направления большой интерес представляет еще один пакет расширения – *Signal Processing*, обеспечивающий обширные возможности создания программ обработки сигналов для современных научных и технических приложений. В пакете используется разнообразная техника фильтрации и новейшие алгоритмы спектрального анализа. Пакет содержит модули для разработки линейных систем, анализа временных рядов. Что делает его не заменимым при работе в таких областях как обработка аудиоданных, видеоинформации, телекоммуникации [3].

Основные свойства пакета:

- Моделирование сигналов и линейных систем,
- Проектирование, анализ и реализация цифровых и аналоговых фильтров,
- Быстрое преобразование Фурье, дискретное косинус и другие преобразования,
- Оценка спектров и статистическая обработка сигналов,
- Параметрическая обработка временных рядов,
- Генерация сигналов различной формы.

Пакет *Signal Processing* является идеальной оболочкой для анализа и обработки сигналов. В нем используются проверенные практикой алгоритмы, выбранные по критериям максимальной эффективности и надежности. Пакет содержит широкий спектр алгоритмов для представления сигналов и линейных моделей. Кроме того, он включает полный набор методов для создания цифровых фильтров с разнообразными характеристиками. Позволяет быстро разрабатывать фильтры с нужной полосой пропускания с графической интерпретацией проходящих процессов.

Графический интерфейс позволяет проектировать фильтры, задавая требования к ним в режиме переноса объектов мышью. Пакет включает функции для вычисления дискретного преобразования Фурье, дискретного косинус преобразования, часто применяемых для анализа, кодирования и фильтрации. Новый графический интерфейс позволяет просматривать и визуально оценивать характеристики сигналов, проектировать фильтры, производить спектральный анализ.

На рисунке 19 изображен вид окна библиотеки пакета *Signal Processing*.

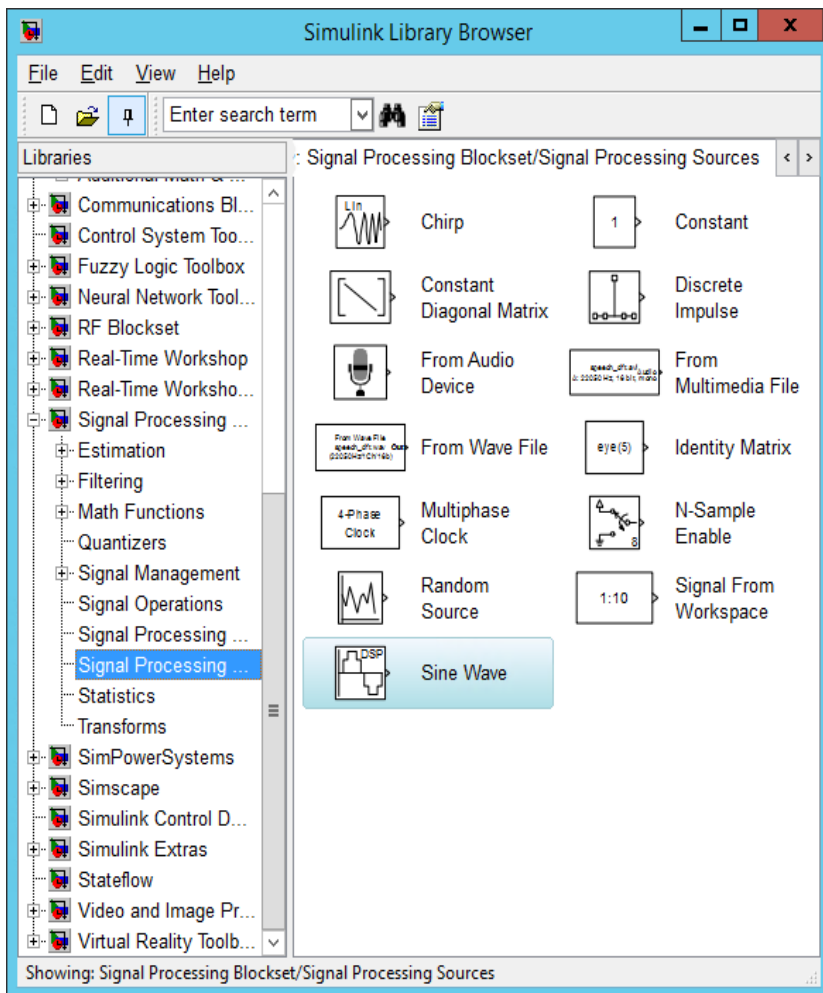


Рис. 19. Окно с блоками пакета *Signal Processing* в окне *MATLAB*

4. Технология моделирования в Simulink

После создания *Simulink*-модели приступают к процессу моделирования, который включает в себя следующие этапы:

- настройка модели;
- настройка обмена данными модели с рабочим пространством памяти *Workspace*;
- настройка диагностики моделирования;
- симулирование (моделирование) – запуск, пауза, остановка.

4.1. Настройка модели

Настройку модели можно провести с помощью окна *Configuration Parameters* (Параметры конфигурации), которое вызывается по команде меню *Simulation\ Configuration Parameters*.

Настройка модели заключается в задании параметров моделирования в разделе *Solver* (Решатель), выделенный в левом верхнем углу окна панели *Select*, представленного на рис. 20.

Параметры моделирования задаются на правой панели:

1. В группе *Simulation time* (Время моделирования) указывается интервал моделирования, задаваемый его границами *Start time* (начальное время) моделирования в секундах и *Stop time* (конечное время) моделирования (по умолчанию 10 с).

2. В группе *Solver Options* (Параметры решателя) выбираются параметры решателя.

- **Type** (Тип) – тип решателя – с фиксированным (*Fixed-step*) и переменным (*Variable-step*) шагом моделирования, выбираемые из списка *Type*. При моделировании аналоговых систем – выбирают решатель с переменным шагом моделирования *Variable-step*, а дискретных систем – решатель с фиксированным шагом моделирования *Fixed-step*
- **Max step size** – максимальный шаг моделирования в сек.;
- **Min step size** – минимальный шаг моделирования в сек.;
- **Initial step size** – начальный шаг моделирования в сек.;
- **Number of consecutive min steps** – допустимое количество уменьшений минимального шага моделирования;

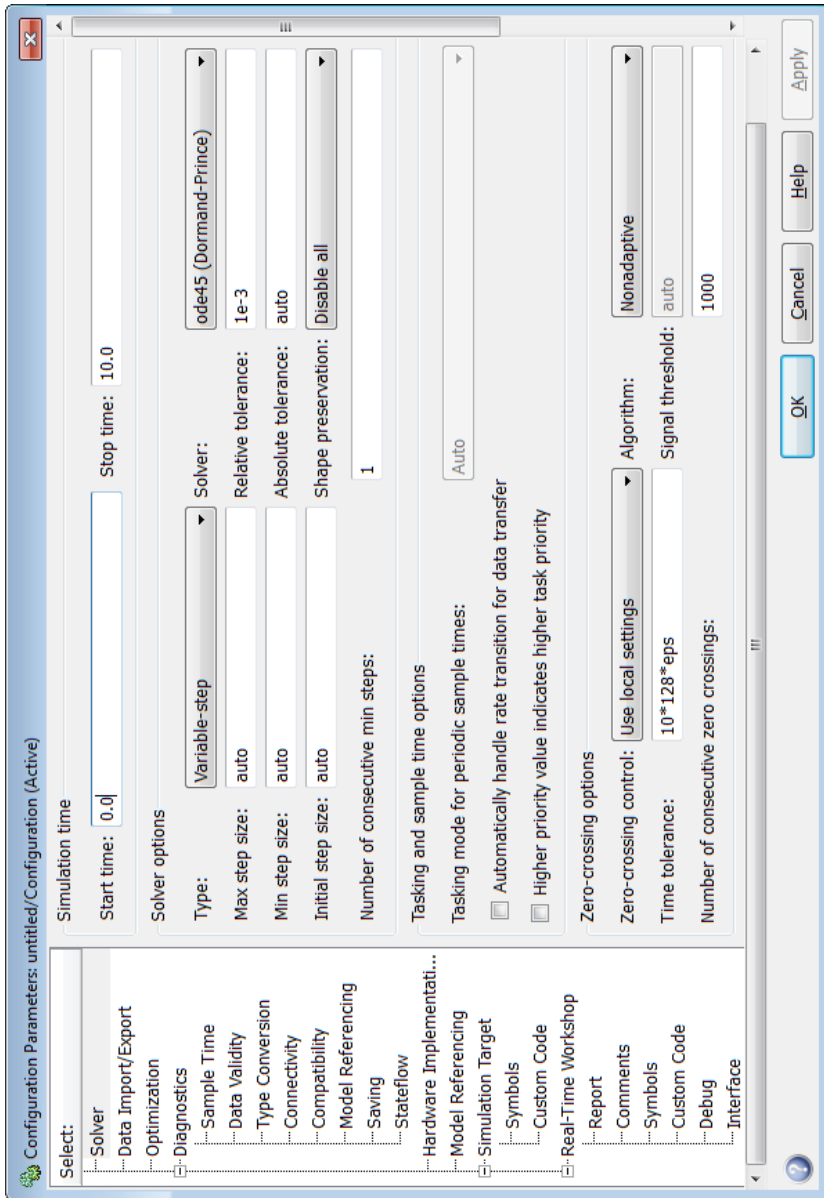


Рис. 20. Окно *Configuration Parameters* вкладки *Solver*

- **Relative tolerance** – относительная погрешность численного интегрирования;
- **Absolute tolerance** – абсолютная погрешность численного интегрирования;
- **Shape preservation** – сохранение переменных состояний
Запретить все \ Разрешить все.
- **Solver** – решатель. В раскрывающемся списке *Solver* решателя с переменным шагом моделирования для аналоговых систем можно выбрать из списка функции (*ode*), реализующие разные методы численного интегрирования, обеспечивающие разную степень точности вычисления. При моделировании дискретных систем выбирается решатель с обобщенным именем *discrete*.

3. В группе *Tasking and time option* (параметры управления задачами и интервалом времени) задаются параметры, контролируемые интервал времени и процесс моделирования. Здесь можно установить режим управления задачами для периодического интервала времени *Tasking mode for periodic sample times*, установив флаги:

- **Automatically handle rate transmission for data transfer** – автоматическое управление преобразованием скорости передачи данных – периода дискретизации;
- **Higher priority value indicates higher task priority** - значение более высокого приоритета указывает на задачу с более высоким приоритетом.

4. В группе *Zero crossing options* (параметры для передачи нуля) выбираются параметры метода «пересечения нуля». При этом решатель обеспечивает контроль монотонности сигнала на каждом шаге моделирования и позволяющий максимально точно определить моменты резкого изменения значений сигнала (скачки). В группе выбираются следующие параметры:

- **Zero crossing control** – контроль точки пересечения нуля с возможными значениями, предлагаемыми из списка;
- **Time tolerance** – показатель точности, контролирующей близость точек пересечения нуля, по умолчанию $10 \cdot 128 \cdot e^{-16}$;

- **Algorithm** – алгоритм для определения местоположение пересечений нуля;
- **Signal threshold** – зона не чувствительности для выбранного алгоритма;
- **Number of consecutive zero crossing** – поле ввода количества точек пересечения нуля, при выполнении которого будет выдаваться предупреждение об ошибке.

4.2. Настройка обмена данными модели с рабочим пространством памяти Workspace

В процессе моделирования обмен данными между моделью и рабочим пространством *Workspace* происходит с помощью окна *Simulation\ Configuration Parameters*.

Настройка модели заключается в задании параметров моделирования в разделе *Data Import/Export* (Импорт/экспорт данных), выделенный в левом верхнем углу окна панели *Select*, представленного на рис. 21.

Параметры моделирования задаются на правой панели:

1. В группе *Load from Workspace* (Загрузка из рабочего пространства) выбираются параметры, контролирующие импорт данных из *Workspace* перед моделированием:

- **Impute** (Входные данные) – флаг, управляющий импортом данных из *Workspace*. При его установке в поле ввода указываются имена импортируемых из *Workspace* данных, которые могут быть представлены в виде матрицы $[t, u]$ (t – вектор столбец возрастающих значений времени, u – матрица, в которой векторы столбцы соответствуют значениям моделируемой величины); отдельно вектор t или матрица u ; выражение *MATLAB* в апострофах.

Импорт данных может происходить и с помощью блока *From Workspace*, в котором устанавливаются имена импортируемых данных.

- **Initial state** (Начальное состояние) – флаг, управляющий импортом начального состояния из *Workspace*. При его установке начальное состояние импортируется из *Workspace* в модель.

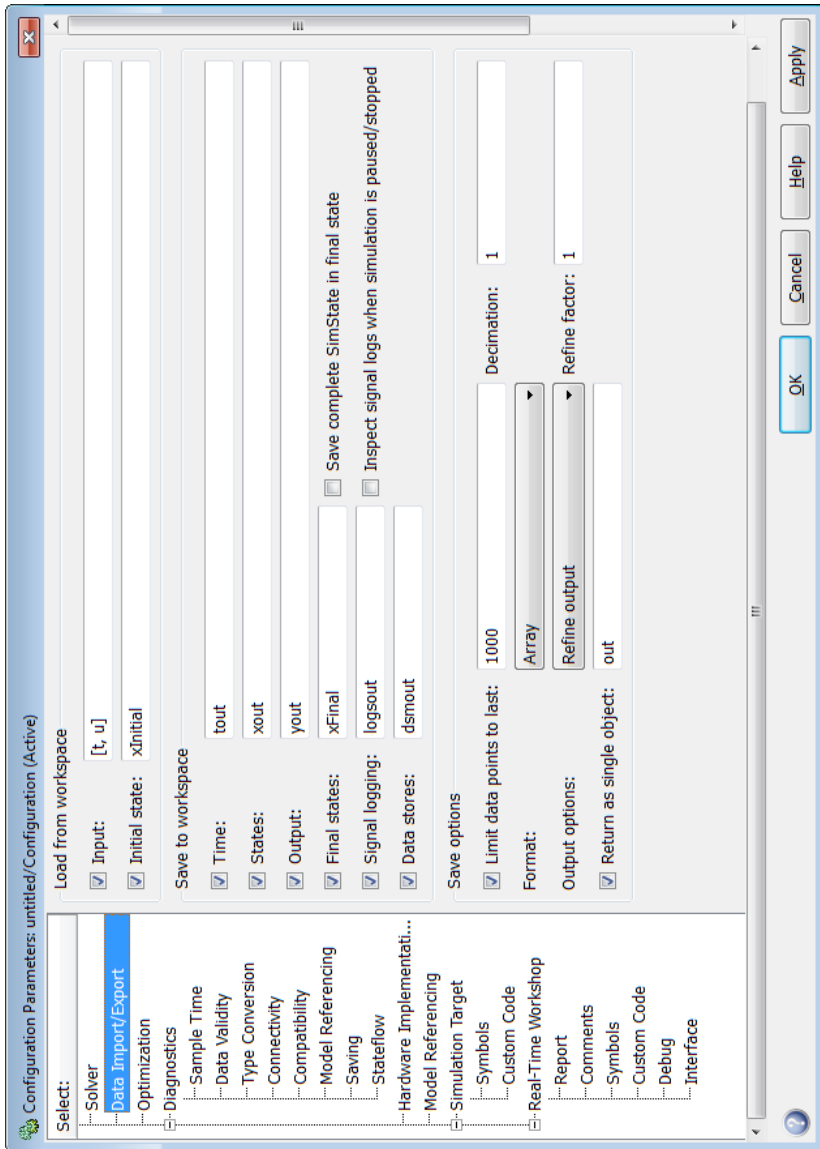


Рис. 21. Окно *Configuration Parameters* вкладки *Data Import / Export*

2. В группе *Save to Workspace* (Сохранение в рабочее пространство) выбираются параметры, управляющие экспортом данных в *Workspace* после моделирования.

- **Time** (Время) – флаг, управляющий экспортом данных в *Workspace* вектора значений времени. При его установке в поле ввода указывается имя вектора с экспортируемыми значениями времени (по умолчанию *tout*);
- **State** (Переменные состояния) – флаг, управляющий экспортом переменных состояний. При его установке в поле ввода указывается имя переменной, посредством которой экспортируются в *Workspace* переменные состояния модели (по умолчанию *xout*). Вид данных указывается в поле ввода *Format* группы *Save options*;
- **Output** (Выходные данные) – флаг, управляющий экспортом выходных данных в *Workspace*. При его установке в поле ввода указывается имя переменной для экспортируемых в *Workspace* данных (по умолчанию *yout*). Вид данных указывается в поле ввода *Format* группы *Save options*;
Экспорт данных может происходить и с помощью блока *To Workspace*, в котором устанавливаются имена экспортируемых данных;
- **Final states** (Конечные состояния) – флаг, управляющий экспортом в *Workspace* переменных состояний на последнем шаге моделирования. При его установке в поле ввода указывается имя переменной, посредством которой экспортируются в *Workspace* конечные состояния модели (по умолчанию *xFinal*). При этом активизируется флаг *Save compute SimState in final state* (сохранить полностью состояние моделирования в последнем положении), управляющий сохранением текущего состояния моделирования;
- **Signal logging** (Регистрация сигнала) – флаг, управляющий регистрацией сигнала в *Workspace* в процессе моделирования. При этом активизируется флаг *inspect signal logs when simulation is pause / stopped* (проверка регистрации сигнала по завершении моделирования или во время паузы), контролирующей регистрацию сигнала в *Workspace* в процессе моделирования и по его завершении.

3. В группе *Save options* (Сохранение параметров) определяются параметры;

- ***Limit date points to last*** (Граничное число данных для сохранения) – флаг, управляющий числом и видом экспортируемых данных. При его установке в поле ввода указывается граничное число значений (по умолчанию *1000*);
- ***Decimation*** (Децимация) – коэффициент прореживания для экспортируемых в *Workspace* сигналов – целое число;
- ***Format*** (Формат) – вид представления экспортируемых в *Workspace* данных (матрица, структура);
- ***Output options*** (Параметры выхода) – параметр для экспортируемых в *Workspace* данных. Используется для решателей с переменным шагом моделирования и установкой значений *Refine factor* (коэффициента коррекции) 1 или 2;
- ***Return as signal object*** (Возвращение в качестве объекта сигнализации). При его установке в поле ввода указывается имя вектора с экспортируемыми значениями данных (по умолчанию *out*).

4.3. Настройка диагностики моделирования

В процессе моделирования возникает необходимость в диагностике моделирования, которую можно настроить с помощью окна *Simulation\ Configuration Parameters*.

Настройка модели заключается в задании параметров диагностики в опции *Diagnostics* (Диагностики), выделенной в левом верхнем углу окна панели *Select*, представленного на рис. 22. Опция содержит:

- *Sample Time* (Интервал времени);
- *Data Validity* (Достоверность данных);
- *Type Conversion* (Преобразование типов данных);
- *Connectivity* (Соединимость);
- *Compatibility* (Совместимость);
- *Model Referencing* (Ссылка на модель);
- *Saving* (Сохранение);
- *State flow* (Состояние потока).

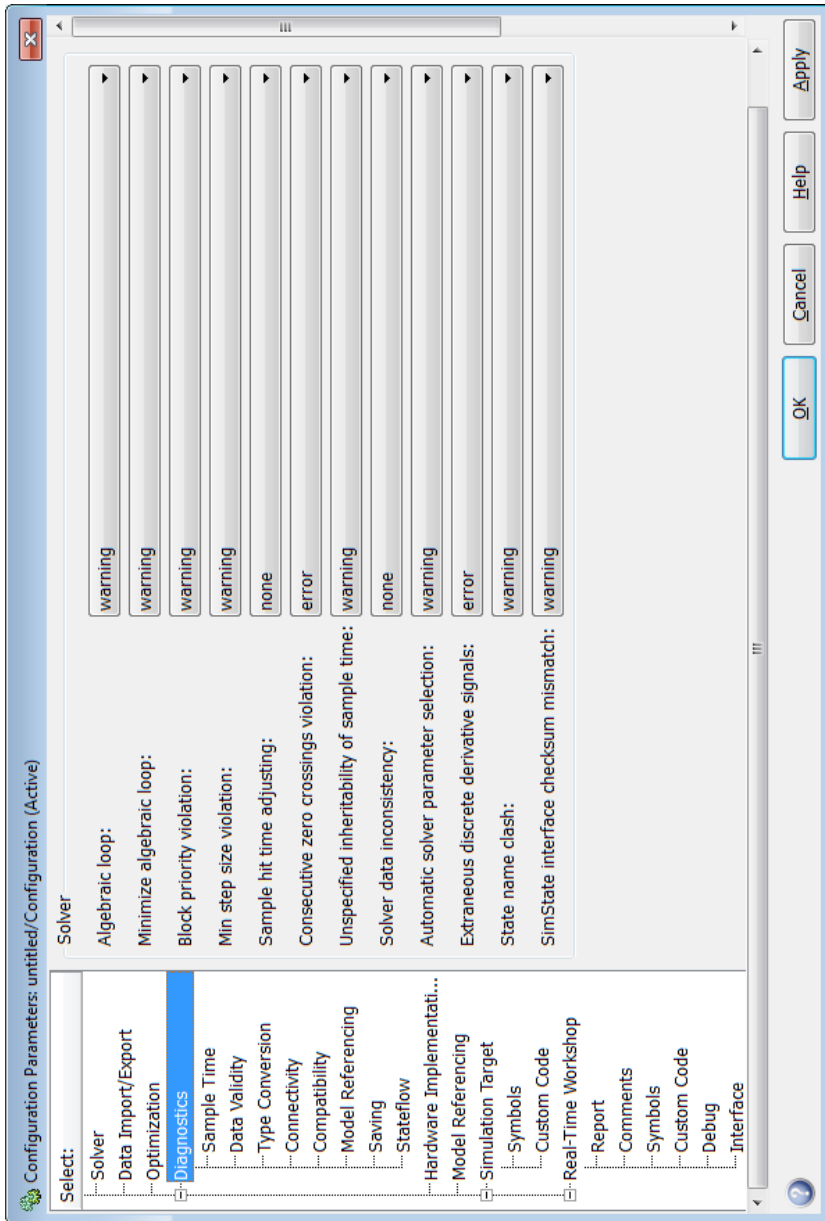




Рис. 22. Окно *Configuration Parameters* вкладки *Diagnostics*

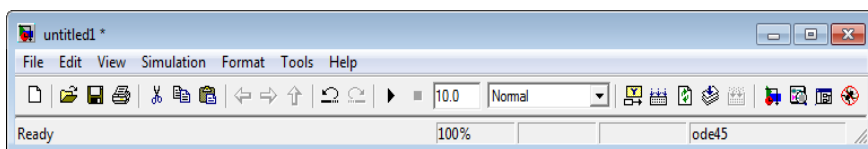
Параметры диагностики выбираются из списка, приведенного на второй половине вкладки *Diagnostics*, который изменяется в зависимости от выделенной опции диагностирования. Реакцию на диагноз – ошибку или проблемную ситуацию при моделировании – пользователь может выбрать из списка стандартных вариантов:

- *none* – игнорировать ситуацию;
 - *warning* – предупреждение и продолжение моделирования;
 - *error* – сообщение об ошибке и остановка моделирования.
- Последние сигналы выдаются в соответствующих окнах.

4.4. Симулирование – запуск и остановка процесса моделирования

Запуск моделирования устройства происходит с помощью кнопки *Start Simulation*  (Запуск моделирования) на панели инструментов окна модели или по команде меню *Simulation/Start*.

Остановка процесса моделирования происходит с помощью кнопки *Stop Simulation*  (Остановка моделирования) на панели инструментов окна модели или по команде меню *Simulation/Stop*.



Содержание отчета лабораторных работ

1. Название работы, цель работы, задание в соответствии с вариантом.
2. Вычисления с ожидаемым результатом в виде таблицы.
3. Модели заданий в соответствии с номером варианта.
4. Результаты расчета и компьютерного моделирования.
5. Выводы.

5. Моделирование систем средствами Simulink

1. Арифметико-логическое устройство

1.1. Подготовка к работе

По указанной литературе изучить:

- понятие полусумматора;
- понятие одноразрядного и многоразрядного сумматора;
- объекты подбиблиотеки *Logic and Bit Operations*, *Sinks*, *Sources*, *Signal Routing* библиотеки *Simulink*.

1.2. Цель работы

Изучить работу полусумматора, одноразрядного и много-разрядного сумматоров.

1.3. Задание и порядок выполнения работы

1. Задание 1. Создать модель цифрового устройства – полу-сумматора, реализующего операцию сложения одноразряд-ных двоичных чисел:

- составить таблицу истинности суммирования двух одно-разрядных двоичных чисел (a, b) ;
- проверить эту таблицу путем моделирования операции суммирования с использованием блоков *Simulink*;
- сравнить результаты моделирования с прогнозируе-мыми значениями;
- скопировать в отчет модель и результаты проверки;
- сделать выводы по проделанной работе.

2. Задание 2. Создать модель одноразрядного двоичного сумматора:

- составить таблицу истинности суммирования трех одно-разрядных двоичных чисел (a, b, c) ;
- проверить эту таблицу путем моделирования операции суммирования с использованием блоков *Simulink*;
- сравнить результаты моделирования с прогнозируе-мыми значениями;
- скопировать в отчет модель и результаты проверки;
- сделать выводы по проделанной работе.

3. Задание 3. На базе одноразрядного сумматора создать модель 4-разрядного двоичного сумматора, выполняющего

операцию суммирования с многоразрядными числами, заданными по варианту в табл. 1.1. Для этого:

- перевести числа, заданные в десятичной системе в двоичный эквивалент,
- выполнить операцию суммирования с этими числами;
- проверить результат с помощью модели многоразрядного сумматора с использованием блоков *Simulink*;
- сравнить результаты моделирования с прогнозируемыми значениями;
- скопировать в отчет полученную модель и результаты проверки;
- сделать выводы по проделанной работе.

Таблица 1.1. Варианты заданий. Операции $a + b$

N	числа		N	числа		N	числа		N	числа	
	a	b		a	b		a	b		a	b
1	4	15	5	12	10	9	8	14	13	10	11
2	5	14	6	13	12	10	9	15	14	11	10
3	6	13	7	14	11	11	10	12	15	12	9
4	7	12	8	15	9	12	11	13	16	13	8

1.4. Технология выполнения Задания 1

1. Арифметико-логическое устройство (АЛУ) является узлом ЭВМ, который выполняет арифметические и логические операции над данными, обрабатываемыми процессором. Основной элемент, используемый в АЛУ, является полусумматором. Функция полусумматора заключается в арифметическом сложении двух двоичных цифр А и В, в результате чего образуется сумма (S) и перенос в старший разряд (P) в соответствии с правилами двоичного сложения:

Таблица истинности работы полусумматора:

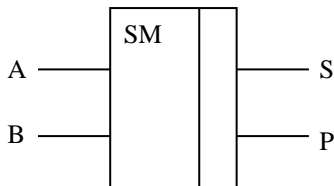
Слагаемые		Результат А+В		
А	В	Р	С	в десятичном коде
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	2

Условно полусумматор на логических схемах изображается следующим образом:

Полусумматор имеет два входа А и В и два выхода S и P.

Как видно из таблицы истинности значения столбца P представляют собой результат логической операции И ($a \wedge b$). Если $a=1$ и $b=1$ образуется перенос из младшего разряда в старший. Столбец S даёт значение суммы двух бинарных чисел и представляет собой логическую операцию «сложение по модулю 2» ($a \oplus b$).

Отсюда следует, что для реализации полусумматора будет достаточно двух логических элементов (AND) и (XOR).



2. В Simulink создайте модель устройства (рис. 1.1), выполняющего заданную арифметическую операцию. В таблице 1.2 указаны названия блоков, папки в которых находятся нужные блоки для моделирования и параметры блоков, которые нужно изменить.

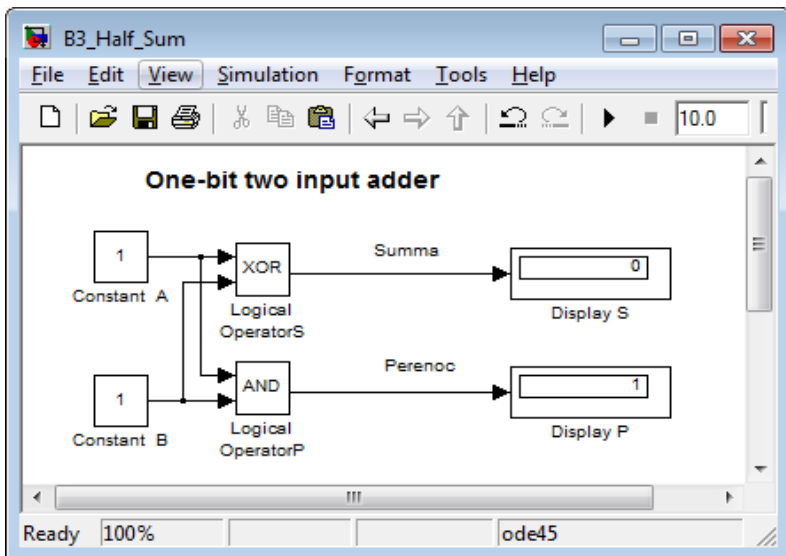


Рис. 1.1. Модель полусумматора

3. Сохраните модель в своей папке под именем *Half_Sum*.

Таблица 1.2. Значения параметров блоков моделирования

Название и назначение блоков	Размещение в браузере	Параметры блоков
Источники: <i>Constant A, B</i>	<i>Simulink / Sources</i>	<i>Constant value</i> // значения <i>a, b</i> из табл. истинности
Логические блоки <i>Logical Operator S, Logical Operator P</i>	<i>Simulink / Logic and Bit Operations</i>	<i>Operator: XOR, Operator: AND</i>
Регистраторы: <i>Display S, Display P</i>	<i>Simulink / Sinks</i>	По умолчанию

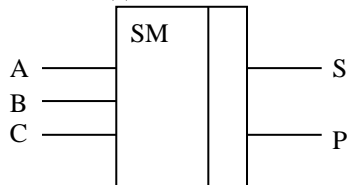
4. Установите поочередно в блоках *Constant A* и *B* бинарные числа в соответствии с таблицей истинности, и, запустив каждый раз модель на исполнение, снимите показания результатов работы полусумматора. Сравните результаты работы модели с таблицей истинности полусумматора.

5. В текстовом редакторе создайте отчет, в который поместите таблицу работы полусумматора и его модель.

1.5. Технология выполнения Задания 2

1. Моделируемому устройству на рис. 2.1 дано название «полусумматор» потому, что оно хотя и даёт значение суммы двух величин и переноса в следующий разряд, однако не учитывает сигнал переноса, получаемый от предыдущего (младшего) разряда. Следовательно, двоичный одноразрядный сумматор должен иметь три входа и два выхода. На логических схемах он условно изображается в виде блока:

На входы *A* и *B* подаются соответственно цифры первого и второго слагаемого, а на вход *C* – цифра переноса из предыдущего разряда. Выходы *S* и *P* соответственно выводят значения суммы и переноса в следующий (более старший) разряд.



Функционирует одноразрядный сумматор в соответствии с таблицей истинности:

Таблица истинности работы сумматора:

Слагаемые			Результат A+B+C		
С перенос	A	B	P	S	в десятич- ном коде
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	2
1	0	0	0	1	1
1	0	1	1	0	2
1	1	0	1	0	2
1	1	1	1	1	3

Сумматор имеет три входа A , B , C и два выхода S и P .

Как видно из таблицы истинности значения столбца P представляют собой результат логических операций ($a \wedge b \vee b \wedge c \vee a \wedge c$). На выходе P формируются переносы из младших разрядов в старшие. Столбец S даёт значение суммы трех бинарных чисел и представляет собой логическую операцию «сложение по модулю 2» ($a \oplus b \oplus c$).

Отсюда следует, что для реализации сумматора необходимы логические элементы AND , OR и XOR .

2. В *Simulink* создайте новую модель устройства (рис. 3.2), выполняющего заданную арифметическую операцию. В таблице 1.2 указаны названия блоков, папки в которых находятся нужные блоки для моделирования и параметры блоков, которые нужно изменить.

Таблица 1.2. Значения параметров блоков моделирования

Название и назначение блоков	Размещение в браузере	Параметры блоков
Источники: <i>Constant</i> A, B, C	<i>Simulink</i> / <i>Sources</i>	<i>Constant value</i> // значения a, b, c из табл. истинности

Логические блоки <i>Logical Operator</i> 1,2,3,4	<i>Simulink / Logic and Bit Operations</i>	<i>Operator: XOR,</i> <i>Operator: AND – 3 блока</i> <i>Operator: OR.</i>
Регистраторы: <i>DisplayS, DisplayP</i>	<i>Simulink / Sinks</i>	По умолчанию

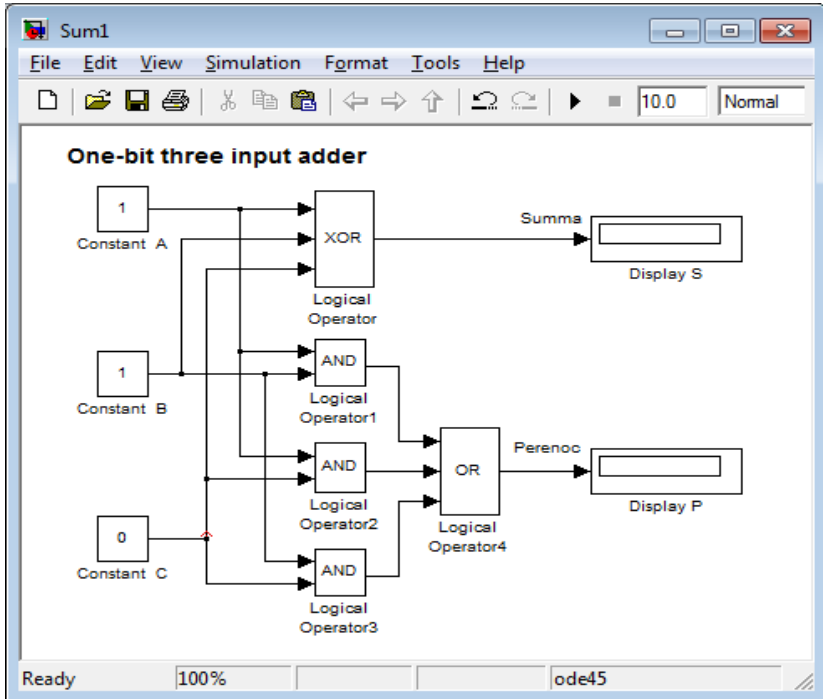


Рис. 1.2. Модель одноразрядного сумматора

3. Сохраните модель в своей папке под именем Sum1.
4. Установите поочередно в блоках *Constant A*, *B* и *C* бинарные числа в соответствии с таблицей истинности, и, запуская каждый раз модель на исполнение (8 раз), снимите показания результатов работы сумматора. Сравните результаты работы модели с таблицей истинности сумматора.
5. В отчет поместите таблицу работы сумматора и его модель.

1.6. Технология выполнения Задания 3

1. Моделируемое устройство на рис. 1.2 даёт значение суммы двух одноразрядных величин и переноса в следующий разряд, учитывает при этом сигнал переноса, получаемый от предыдущего (младшего) разряда. Следовательно, для получения двоичного четырехразрядного сумматора необходимо иметь четыре одноразрядных сумматора. Каждый сумматор имеет по три входа и два выхода. На вход переноса C_0 младшего разряда надо падать 0, а выходы переносов младших разрядов, соединить последовательно со входами следующих старших разрядов.

2. С помощью встроенных функций *MATLAB* представьте два числа a и b , взятые по варианту из табл. 1.1, двоичным эквивалентом. Для этого в командной строке *MATLAB* введите поочередно два числа, например, $a=7$ и $b=13$:

```
>> a=dec2bin(7) % преобразование 10-го числа «7» в 2-е
a =
111
>> b=dec2bin(13) % преобразование 10-го числа «13» в 2-е
b =
1101
```

3. В отчете создайте таблицу, в которую внесите десятичные числа a и b и их двоичные эквиваленты. Оба числа должны быть одинаковой разрядности. Подсчитайте ожидаемый результат в двоичном и десятичном эквивалентах и занесите двоичные разряды в таблицу:

Таблица работы четырехразрядного сумматора

Слагаемые	разряд	0	1	2	3	4
7	A	1	1	1	0	
13	B	1	0	1	1	
перенос	C	0	1	1	1	1
Результат	P	1	1	1	1	
A+B+C=20	S	0	0	1	0	

Ожидаемый результат:

$$S_0+S_1+S_2+S_3+P_3=0*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 1*2^4 = 4+16=20$$

4. В *Simulink* на базе одноразрядного сумматора создайте новую модель четырехразрядного сумматора (рис. 1.6). Поскольку наращивание разрядности сумматора можно достичь многократным повторением схемы одноразрядного сумматора, то целесообразно создать подсистему сумматор (*SM*) на базе одноразрядного сумматора. Для этого:

- 1) скопируйте в новое окно модель одноразрядного сумматора;
- 2) сохраните модель в своей папке под именем *Sum4*;
- 3) мышью выделите блоки *XOR*, *AND* и *OR* (см. рис. 1.3);

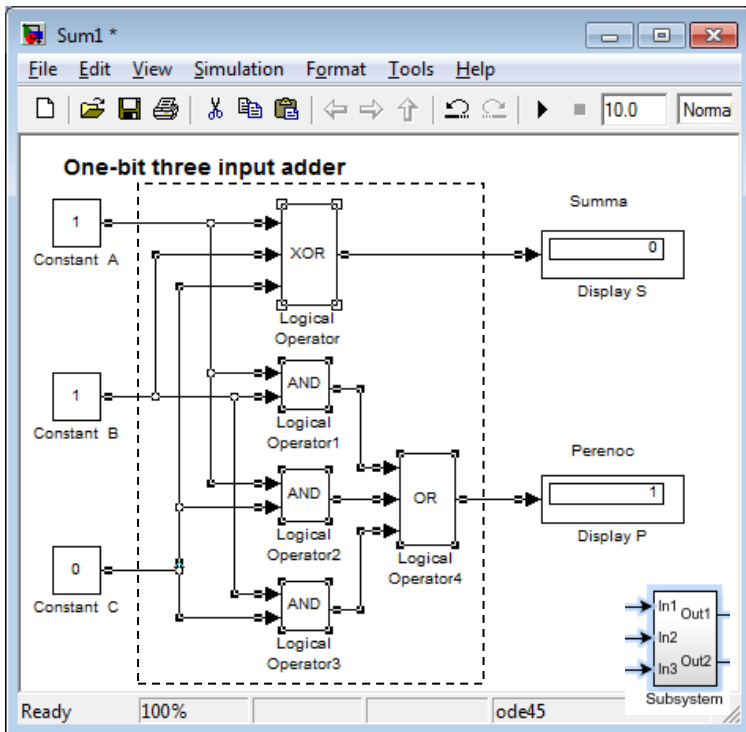
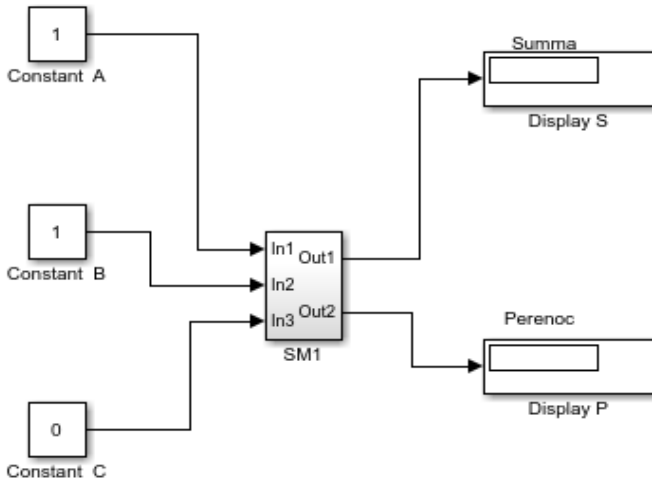


Рис. 1.3. Выделение блоков для подсистемы

4) выберите пункт меню *Edit\ Create Subsystem*;

5) на месте выделенных блоков появится блок подсистемы, имеющий порты ввода *In1,2,3* и порты вывода *Out1,2*. С помощью этих портов подсистема включается в состав основной модели;

One-bit three input adder



б) для вызова подсистемы для просмотра и редактирования (рис. 1.4) достаточно дважды щелкнуть по ней левой кнопкой мыши;

7) переименуйте название блока подсистемы в SM (рис. 1.5);

8) удалите коннекторы, соединяющие подсистему с выходными блоками;

9) скопируйте блок SM трижды в модель и соедините подсистемы последовательно друг с другом как показано на рис. 1.5;

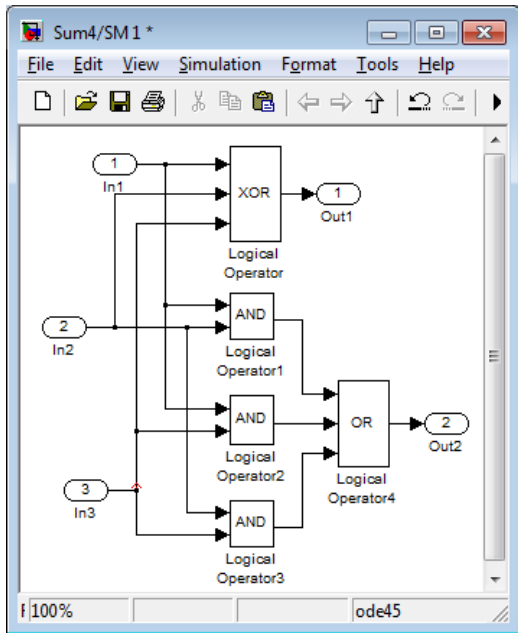


Рис. 1.4. Окно подсистемы SM (сумматора)

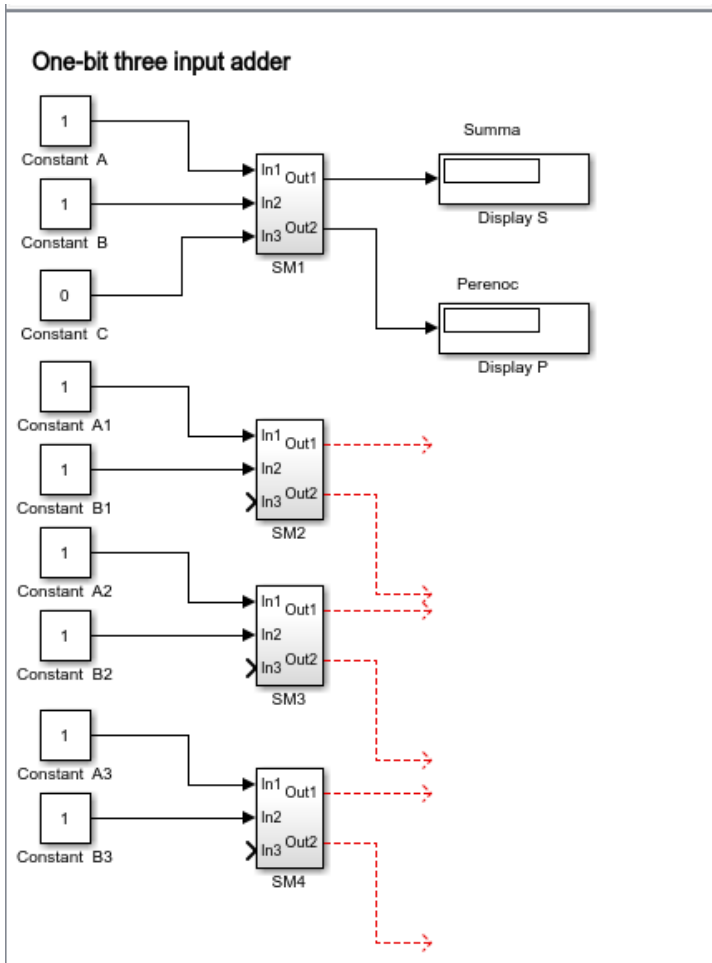


Рис. 1.5. Построение четырехразрядного сумматора

10) добавьте в модель блоки *Constant* для ввода 4-х разрядов чисел a и b и установите в этих блоках значения двоичных эквивалентов. Пример модели приведен на рис. 1.6.

11) для объединения четырех разрядов результата суммирования в вектор установите на выходе сумматора *Mux* – мультиплексор (*Simulink* => *Signal Routing*) с четырьмя входами.

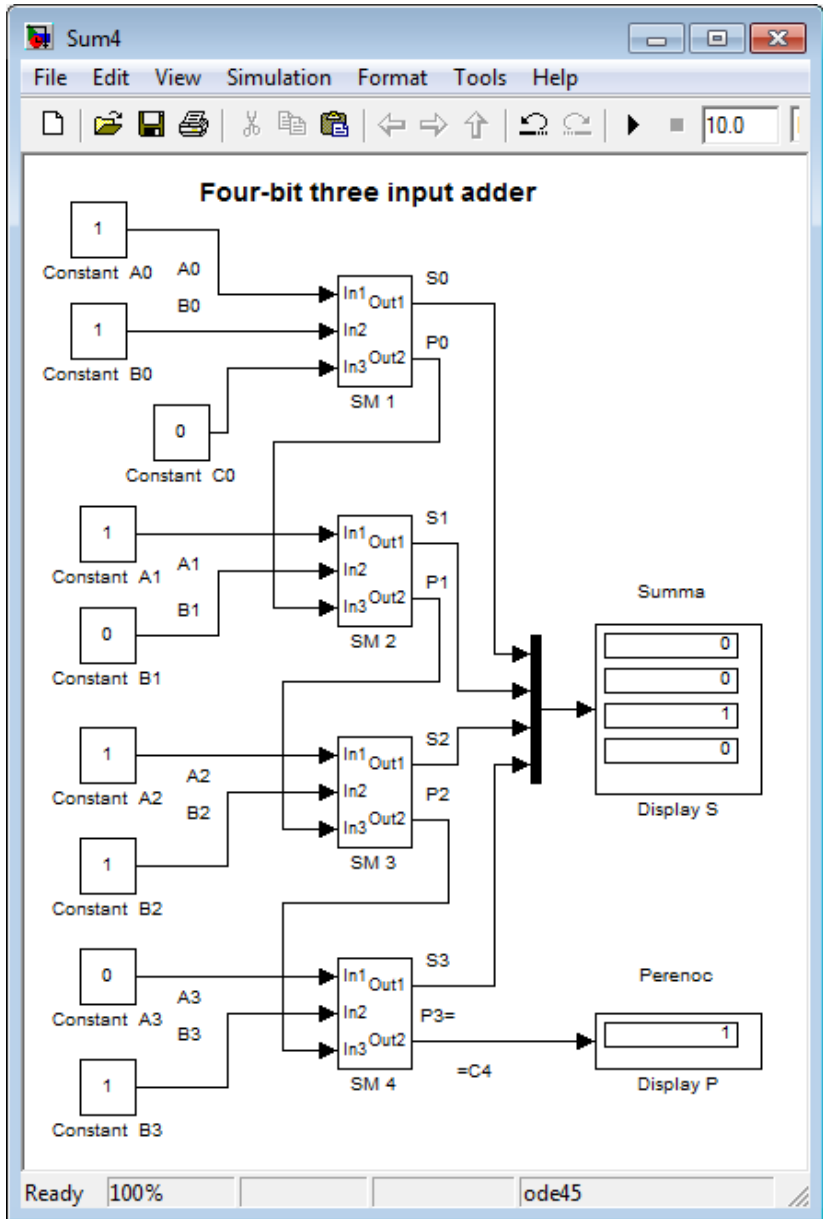


Рис. 1.6. Модель четырехразрядного сумматора

5. Установите в блоках *Constant* бинарные числа a и b в соответствии с вариантом.
6. Сохраните и запустите модель на исполнение.
7. Снимите показания результатов работы сумматора.
8. Сравните результаты работы модели с ожидаемыми значениями сумматора.
9. В отчет поместите таблицу работы многоразрядного сумматора и его модель.
10. Сделайте выводы по проделанной работе.
11. Для увеличения разрядности полученного 4-х разрядного сумматора до 8, необходимо провести аналогичные действия по п. 4. Аналогично, используя подсистемы низшего уровня, можно получить 16-разрядный сумматор (рис. 1.7) и сумматоры большей степени интеграции

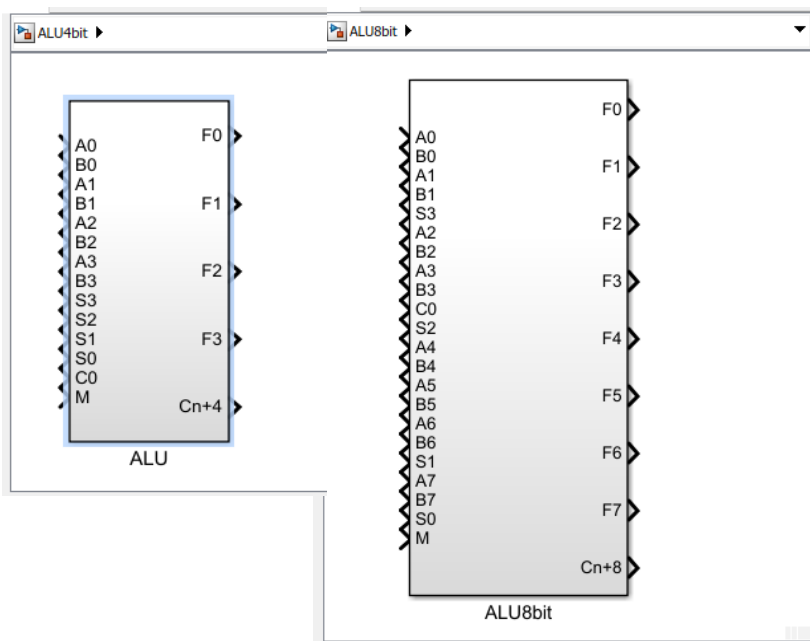


Рис. 1.7. Подсистемы 8 и 16 разрядных сумматоров

Пример контрольного теста «Сумматоры»

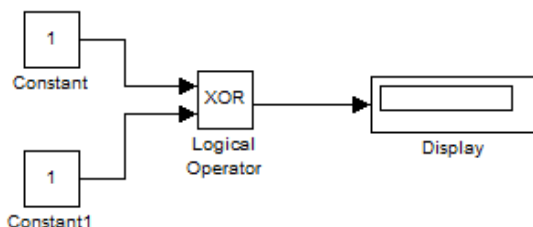
Результат 1_2+1_2 в двоичной системе равен:

Результат арифметической операции $11010+1011$ равен (ввести двоичное число без пробелов):

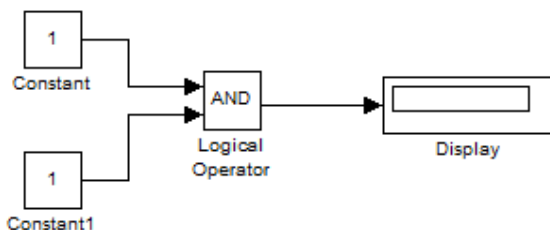
Перенос старшего разряда арифметической операции $1010+1011$ равен:

Результат арифметической операции $A+B$ (при $A=7$ и $B=12$) с двоичными числами равен (ввести двоичное число без пробелов в дополнительном коде):

Введите результат моделирования:



Введите результат моделирования:



Укажите все входы сумматора

Операнд a

Операнд b

Перенос c

Сумма s

Перенос p

Укажите все входы сумматора

2. Моделирование цифрового линейного тракта (ЦЛТ)

2.1. Подготовка к работе

- По указанной литературе изучить:
- свойства блоков *Communication Blockset*;
 - понятие подсистем и управляемых подсистем *Simulink: Ports & Subsystems*;
 - понятие регенерации сигналов в ЦЛТ.

2.2. Цель работы

Создание модели с подсистемой и исследование ее свойств.

2.3. Задание и порядок выполнения работы

1. Задача 1. Создать модель цифрового линейного тракта (ЦЛТ). При моделировании цифровой сигнал, с выхода виртуальной аппаратуры проходит по каналу связи, искажается под воздействием помех, а затем усиливается и восстанавливается в регенераторе до первоначального вида. Провести моделирование системы с визуализацией процессов, протекающих в точках имитационной модели. Варианты цифровых сигнала в виде кодовых комбинаций, вырабатываемых соответствующим кодовым генератором и коэффициент усиления на входе регенератора задаются в табл. 2.1.

Таблица 2.1. Варианты заданий

N	Источник цифрового сигнала	Источник помех	Коэффициент усиления <i>Gain</i>
1	<i>Walsh Code Generator</i>	генератор <i>Gaussian Noise Generator</i>	2
2	<i>PN Sequence Generator</i>		4
3	<i>Bernoulli Binary Generator</i>		6
4	<i>Kasami Sequence Generator</i>		7
5	<i>OVSF Code Generator</i>		3
6	<i>Barker Code Generator</i>		5
7	<i>Walsh Code Generator</i>		7
8	<i>PN Sequence Generator</i>		2
9	<i>Bernoulli Binary Generator</i>		3

10	<i>Kasami Sequence Generator</i>		4
11	<i>OVSF Code Generator</i>		5
12	<i>Barker Code Generator</i>		6
13	<i>Walsh Code Generator</i>		5
14	<i>PN Sequence Generator</i>		3
15	<i>Bernoulli Binary Generator</i>		4
16	<i>Kasami Sequence Generator</i>		2

2. Задача 2. Создать подсистему, в которую поместить рисунок – эмблему регенератора и расположить ее в окне модели устройства. Модель регенератора сохранить под именем *Regenerator* в своей папке и скопировать её в отчет.

3. Задача 3. На базе модели *Regenerator.mdl* создать подсистему с надписью *Regenerator* и поместить ее в отчет.

4. Сделать выводы о возможностях моделирования, с использованием, как простых блоков, так и подсистем.

2.4. Технология выполнения задания 1

Задача 1. Модель цифрового линейного тракта (ЦЛТ) содержит источник цифрового сигнала (ИЦС), источник помех (ИП) и регенератор (Р). Структурная схема модели приведена на рис. 2.1. Регенератор предназначен для восстановления ослабленного искаженного линейного сигнала и формирования цифрового сигнала, аналогичного выходному сигналу источника ИЦС. Осциллограф позволяет в любой момент времени проконтролировать с помощью временных диаграмм работу всего ЦЛТ. Для этого рекомендуется использовать пяти в ходовое устройство.

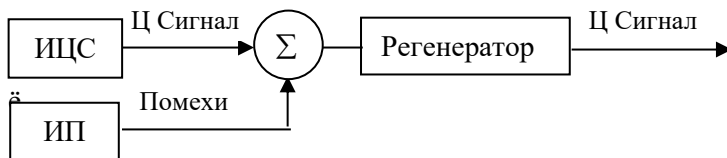


Рис. 2.1. Блок-схема ЦЛТ

На рисунке 2.3 приведено моделируемое устройство. В таблице 2.2 указаны блоки и их параметры, необходимые для моделирования.


Таблица 2.2. Значения параметров блоков моделирования

Название и назначение блоков	Размещения в браузере	Параметры блоков
Генераторы цифр. сигнала: <i>Walsh Code Generator,</i> <i>PN Sequence Generator,</i> <i>Kasami Sequence Generator,</i> <i>OVSF Code Generator,</i> <i>Barker Code Generator,</i> Генератор 2-го сигнала: <i>Bernoulli Binary Generator</i>	<i>Communication Blockset / Comm Sources/Sequence Generator</i> <i>Communication Blockset / Comm Sources/ Random Date Sources</i>	По умолчанию
Источник помех: <i>Gaussian Noise Generator.</i>	<i>Communication Blockset / Comm Sources</i>	По умолчанию
Математический блок вычисления суммы (++)	<i>Simulink / Math Operations</i>	По умолчанию
Усилитель <i>Gain</i>	<i>Simulink/ Math Operations</i>	<i>Gain=4</i> // По варианту
Между в ходовой переключатель <i>Manual Switch</i>	<i>Simulink / Signal Routing</i>	По умолчанию
Ограничитель величины сигнала – <i>Saturation</i>	<i>Simulink / Discontinuous</i>	Пороги ограничения амплитуд: <i>Upper Limit = 1</i> <i>Lower Limit = -1</i>
Численное дифференцирование <i>Derivative</i>	<i>Simulink / Continuous</i>	По умолчанию
<i>Scope</i>	<i>Simulink/Sinks</i>	<i>Number of axes=5</i>
Управляемая подсистема <i>Triggered Subsystem</i>	<i>Simulink/ Ports & Subsystems</i>	<i>Trigger Type = either</i>

Подсистема – это фрагмент модели, оформленный в виде отдельного блока. Связь подсистемы с моделью выполняется с помощью входных (*Impute*) и выходных (*Output*) портов. Одни подсистемы могут включать в себя другие подсистемы. В управляемых подсистемах поведение задается внешним воздействием. Определены три типа управляемых подсистем: *E*-подсистема, *T*-подсистема, *ET* - подсистема. В модели используется *T*-подсистема (рис. 2.2), которая управляется введенным в нее блоком *Trigger* (Переключение).

Этот блок разрешает работу подсистемы только во время заданного события на управляющем входе *Trigger*.

При работе с *T*-подсистемой:

1. в окне библиотеки *Simulink* в разделе *Ports & Subsystems*, найти соответствующий блок *Triggered Subsystem*;
2. с помощью мыши перетащить его в окно новой модели;
3. двойным щелчком по блоку открывается окно структуры *T*-подсистемы (рис.3.2);
4. двойным щелчком по значку  открывается окно параметров блока *Triggered*. В нем определена реакция подсистемы на управляющий сигнал:
 - **rising** (рост) – разрешить работу при перепаде вверх, Γ
 - **falling** (спад) – разрешить работу при перепаде вниз, L
 - **either** (оба) – разрешить работу при любом перепаде, ΓL
 - **function-call** (вызов функции) – по логике функции.
5. установить нужную реакцию, в примере **either**.

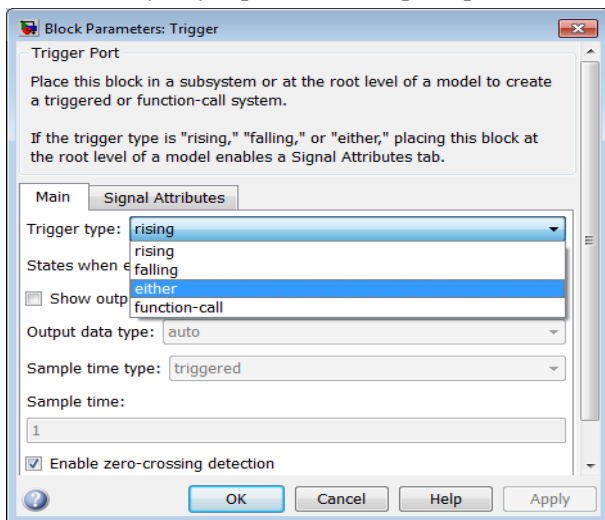
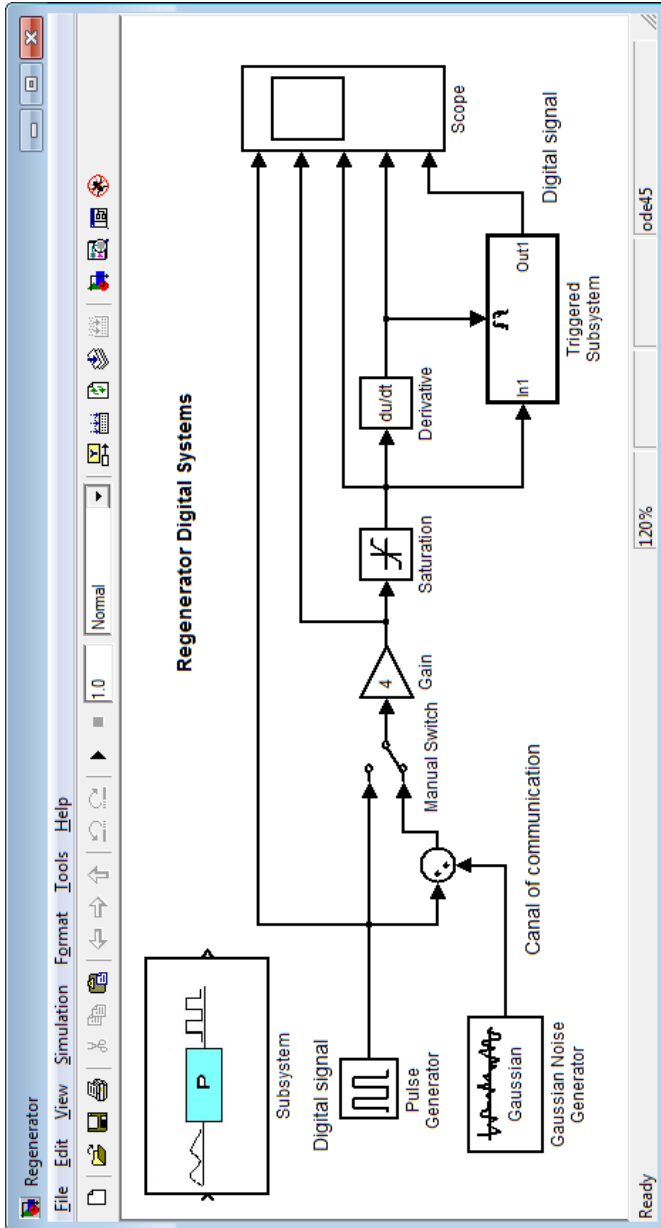


Рис. 2.2. Окно установки параметров блока *Trigger*

Осуществить моделирование ЦЛТ для двух случаев:

- **Помех нет**. Для этого установить переключатель *Manual Switch* в верхнее положение;
- **Помехи есть** (переключатель двойным щелчком мыши переводится в нижнее положение).



2.3. Модель регенератора ЦДТ

Регулируя положением переключателя можно проследить за поведением системы в разных условиях. На рис. 2.4 приведены временные диаграммы осциллографа в случае, когда в линейном тракте отсутствуют (слева) или присутствуют помехи (справа). Вид помехи влияет на восстановление цифрового сигнала

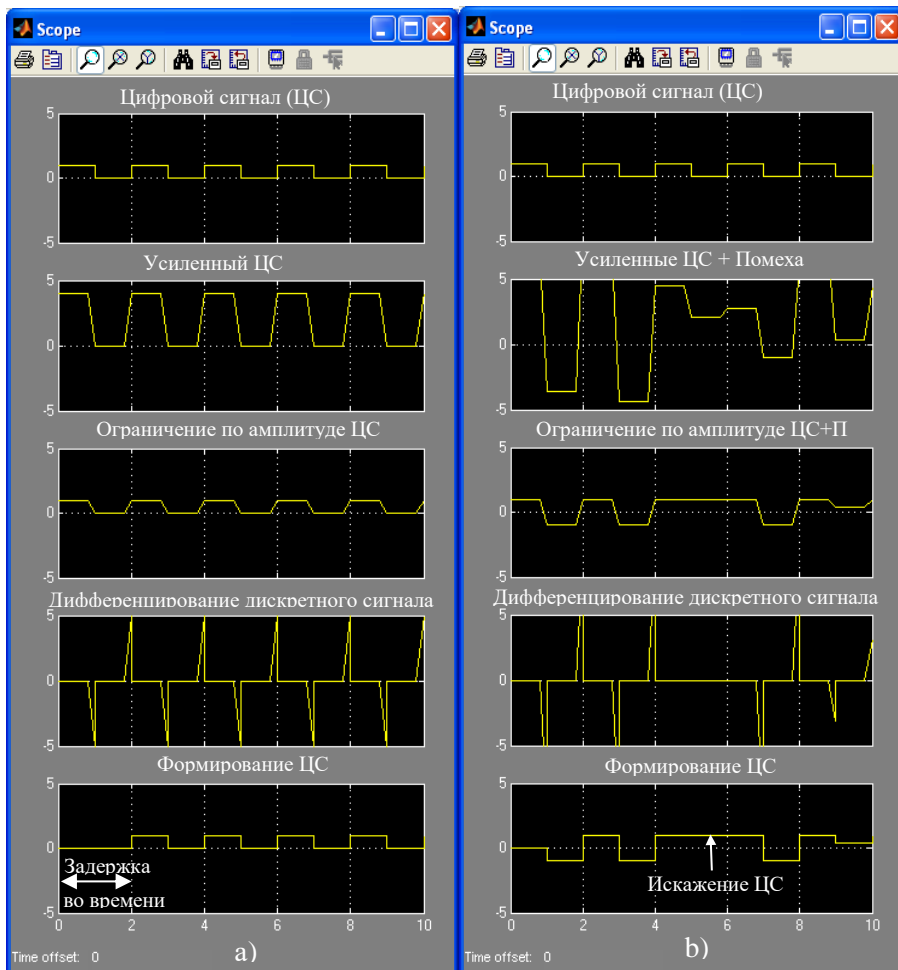


Рис. 2.4. Диаграммы работы регенератора цифрового сигнала в каналах: а) без помех, б) с помехами

2.5. Технология выполнения задания 2

Для создания рисунка (размером, например, 10 x 7см) можно использовать любой графический редактор, например, *Paint*, *Adobe Photoshop*, *Corel Draw*. Сохраняется рисунок в той же папке, что и модель латинскими символами, например, *Reg.JPG*. На рис. 2.5 приведен вариант эмблемы регенератора ЦС.

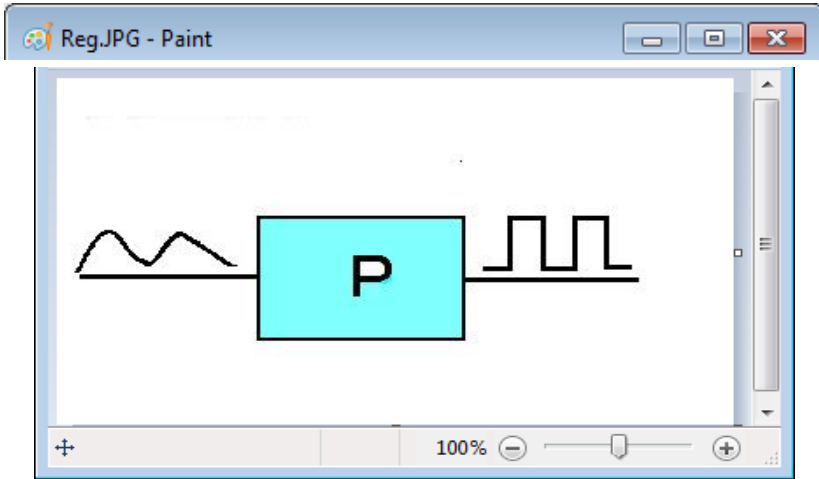


Рис. 2.5. Окно графического редактора Paint

Порядок отображения рисунка на схеме модели следующий:

1) создать пустую подсистему (фрагмент *Simulink*-модели, оформленный в виде отдельного блока). Для этого: в браузере *Simulink* => *Port & Subsystems* найти и перетащить на свободное место уже созданной модели блок *Subsystem* (см. рис. 2.3 левый верхний угол);

2) выполнить ее маскирование (оформить подсистему как отдельный библиотечный блок) с помощью редактора маски *Mask Editor*. Для запуска редактора необходимо выделить *Subsystem* и выполнить команду *Edit* => *Mask Subsystem...* На экран будет выведено окно редактора (рис.2.6), имеющего 4 вкладки: *Icon & Ports* (Пиктограмма и порты), *Parameters* (Параметры), *Initialization* (Инициализация) и *Documentation* (Документация).

Первая из вкладок обеспечивает создание пиктограммы

подсистемы, вторая – дает возможность создать окно диалога для ввода параметров, третья – задать выражения для инициализации блока и четвертая – позволяет ввести описание блока и создать справку. Открыть саму систему для просмотра и ее редактирования можно двойным щелчком мыши;

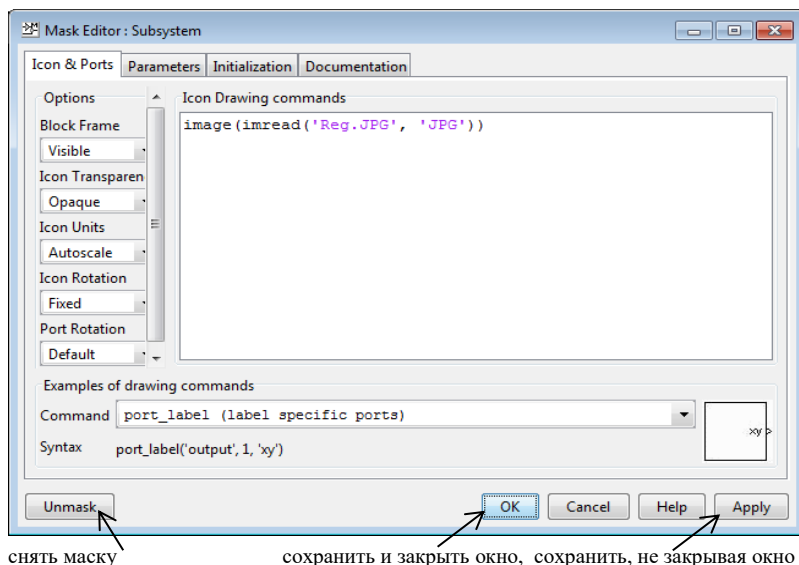


Рис. 2.6. Окно редактора Маски

3) на вкладке *Icon & Ports* в графе *Icon Drawing Commands* ввести команду для считывания из файла и отображения графического образа, например:

```
image(imread('Reg.JPG', 'JPG')),
```

где первый параметр – имя файла, а второй – его тип.

Допустимо также указывать в качестве имени файла полный путь доступа, если файл-рисунок сохранен в отдельной папке от модели, например, 'D:\...Reg.JPG'.

4) нажать на кнопку *OK* или (*Apply*).

На рисунке 2.3 показана модель регенератора с созданным таким способом пиктограммой. Для придания модели законченного вида можно использовать возможности форматирования, команды которого сосредоточены в пункте меню *Format*.

2.6. Технология выполнения задания 3

Для создания своей подсистемы:

1) сохранить модель *Regenerator.mdl* командой *File\Save As...* под именем *Regenerator_Systems.mdl*,

2) удалить все коннекторы, выводящие сигналы на осциллограф, кроме последнего, с выхода которого формируется цифровой сигнал (*Digital signal*),

3) мышью выделить блоки *Gain*, *Saturation*, *Derivative*, *Triggered Subsystem* (см. рис. 2.7),

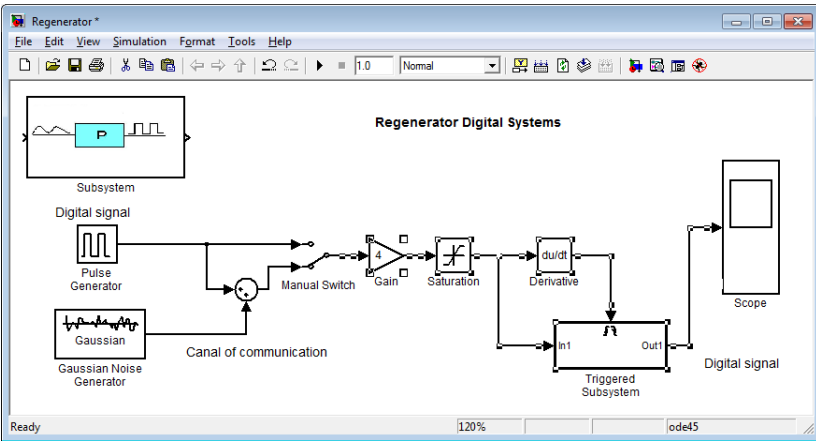
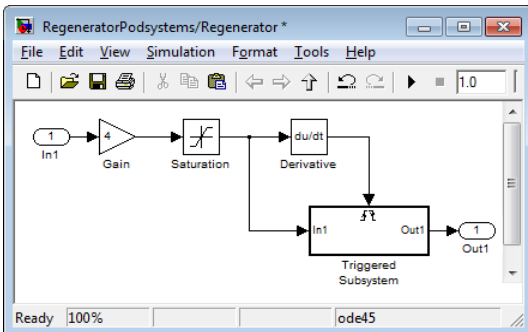


Рис. 2.7. Выделение блоков для подсистемы

4) выбрать пункт меню *Edit\ Create Subsystem*,

5) на месте выделенных блоков появится блок подсистемы, имеющий порт ввода *In1* и порт вывода *Out1*. С помощью этих портов подсистема включается в состав основной модели,

6) переименуйте название блока в *Regenerator* (рис. 2.8).



7) для вызова подсистемы для просмотра и редактирования достаточно дважды щелкнуть по ней левой кнопкой мыши.

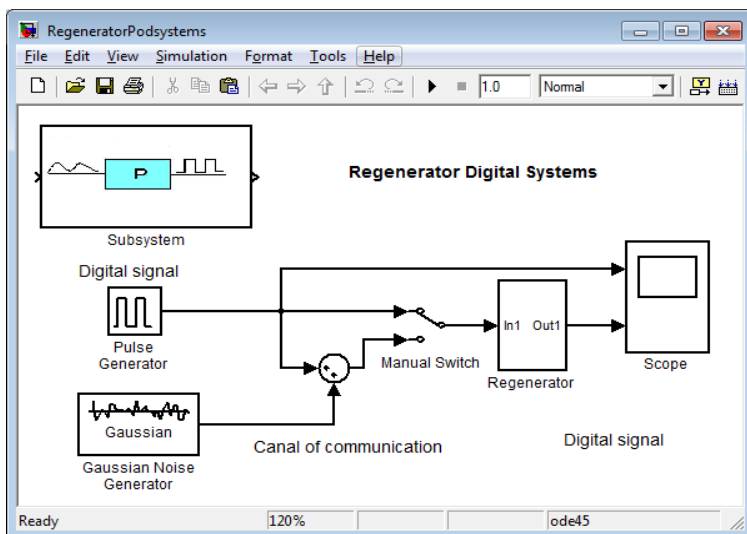


Рис. 2.8. Модель ЦЛТ с подсистемой Regenerator

Запустите модель на исполнение и проконтролируйте ее работу (рис. 2.9), добавив еще один вход и коннектор к осциллографу. Модели и осциллограммы и выводы поместите в отчет.

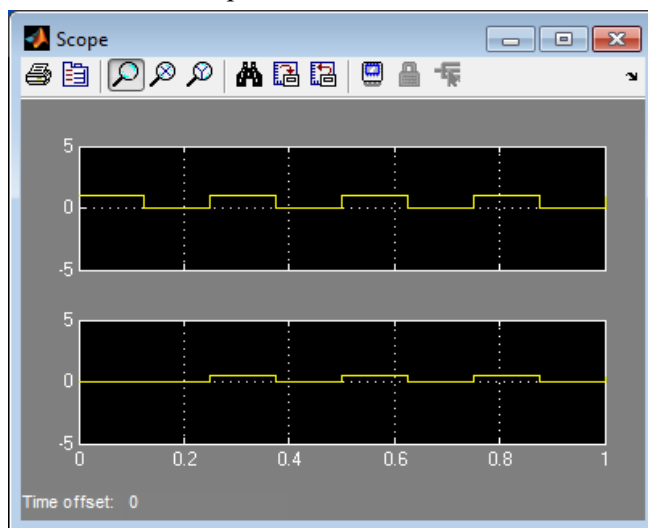


Рис. 2.9. Диаграммы работы подсистемы Regenerator

Пример контрольного теста «Регенераторы»

Укажите все блоки, входящие в состав регенератора:

Sine Wave

Gain

Saturation

Derivative

Sum

Регенератор предназначен для:

генерации аналогового сигнала

деления частоты

генерации цифрового сигнала

усиления сигнала

восстановление цифрового сигнала

Фрагмент модели, оформленный, в виде отдельного блока, является:

преобразователем

формирователем

подсистемой

усилителем

ограничителем

Укажите все типы управляемых подсистем:

E-подсистема,

T-подсистема,

ET-подсистема

R-подсистема

F-подсистема

В управляемых подсистемах поведение системы задается ... воздействием:

Какой порт не входит в триггерную подсистему:

In, Out, Δ , ∇ , $\bar{\nabla}$

Укажите команду считывания графического файла из подсистемы:

3. Модель аналоговой системы передачи

3.1 Подготовка к работе

По указанной литературе изучить:

- свойства блоков *Signal Processing*, *Communications Blockset*;
- принципы построения систем с частотным разделением каналов.

3.2. Цель работы

Создание модели аналоговой системы передачи с частотным разделением каналов и исследовать ее работу.

3.3. Задание и порядок выполнения работы

1. Задача 1. Создать модель трехканальной аналоговой системы передачи (АСП) с частотным разделением каналов (ЧРК). Для каждого из трех каналов создать свою подсистему, входящую в состав системы передачи. Характеристики непрерывных сигналов в виде амплитуды (A) и частоты (F_s) синусоидальных сигналов, а также частоты несущих (F_c) каждого из каналов задаются по варианту в табл. 3.1.

Таблица 3.1. Варианты заданий

N	1 канал			2 канал			3 канал		
	A , В	F_s , Гц	F_c , Гц	A , В	F_s , Гц	F_c , Гц	A , В	F_s , Гц	F_c , Гц
1	2	30	150	3	60	200	4	90	250
2	3	12	70	4	24	110	3	36	150
3	5	14	80	3	28	120	2	42	160
4	6	20	90	5	30	180	2	40	270
5	4	15	85	3	30	125	5	45	165
6	6	17	90	3	34	150	2	51	210
7	4	25	140	5	50	190	6	75	240
8	3	11	60	4	22	90	2	33	120
9	5	13	70	4	26	120	3	39	170
10	2	22	120	3	44	180	5	66	240
11	4	18	100	5	36	180	3	54	260

12	3	16	80	4	32	130	5	48	180
13	5	21	150	6	42	220	7	63	290
14	3	19	140	5	38	210	4	57	280
15	6	10	80	3	25	120	4	40	160
16	4	24	145	5	48	220	3	72	195

- Иллюстрировать работу модели спектрограммами и временными диаграммами.
- Модель и диаграммы сохранить и поместить в отчет.
- Задача 2. Изменить величины несущих частот, уменьшив их все в два раза по сравнению с заданными значениями. Проанализировать полученный результат, и сделать выводы, как величина несущей частоты влияет на группообразование и восстановление исходных сигналов.
- Сделать выводы о возможностях моделирования систем с использованием пакета расширений *Simulink*.

3.4. Методические указания

В многоканальных системах передачи с ЧРК, каждому каналу отводится своя полоса непрерывающихся друг с другом частот. На рисунке 3.1 представлен принцип формирования группового сигнала по частотному признаку. Здесь показаны исходные сигналы в диапазоне 0,3 – 3,4 кГц, спектры которых с помощью соответствующих несущих частот F_c переносятся в более высокую полосу частот для формирования группового сигнала с полосой $\Delta F_1 - \Delta F_N$.

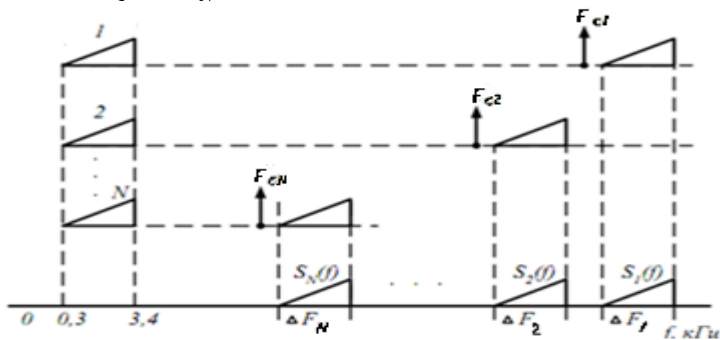


Рис. 3.1. Принцип группообразования в системах с ЧРК

Структурная схема передающей части аналоговой системы передачи с частотным разделением каналов (АСП с ЧРК) изображена на рис. 3.2.

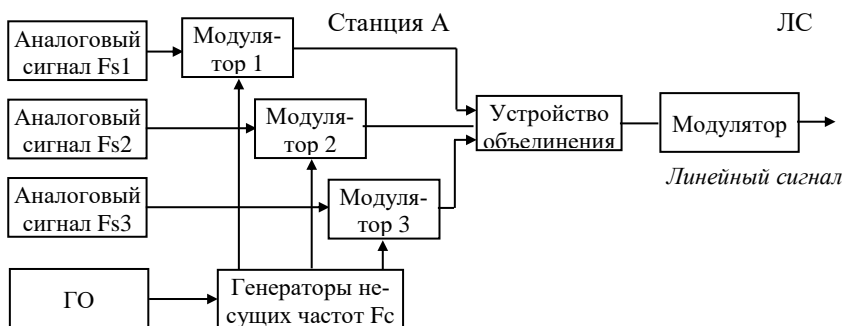


Рис.3.2. Структурная схема передающей части АСП с ЧРК

Аналоговые сигналы (в модели гармонические колебания с частотами F_{s1} , F_{s2} , F_{s3}) поступают на модуляторы, осуществляющие однополосную модуляцию с подавлением несущей частоты F_c и верхней боковой полосы. Совокупность канальных сигналов на выходе суммирующего устройства объединяются в групповой АМ сигнал. Управляет работой модуляторов сигналы, которые выдаются генератором несущих частот передачи. Эти сигналы соответствуют определенным частотам (F_{c1} , F_{c2} , F_{c3}), чем и обеспечивается правильное формирование группового АМ сигнала. Групповой АМ сигнал затем поступают на групповой модулирующий передатчик, который формирует линейный сигнал. Сформированный линейный сигнал поступает в линию связи (ЛС). Поскольку всякая реальная линия связи обладает ограниченной полосой пропускания, то при многоканальной передаче каждому отдельному каналу отводится определенная часть общей полосы пропускания.

Структурная схема построения приемной части аналоговой системы передачи с частотным разделением каналов, изображена рис. 3.3.

На приемной станции линейный сигнал с помощью группового приемника вновь преобразуется в групповой сигнал, из которого выделяются канальные сигналы полосовыми фильтрами

(Ф1, Ф2, Ф3). После детектирования в канальных демодуляторах, работой которых управляет генераторное оборудование, сигналы преобразуются в предназначенные получателем сообщения приемниками сообщений (ПС 1, ПС 2, ПС 3). Сообщения, передаваемые по разным каналам, выделяются с помощью фильтров нижних частот.

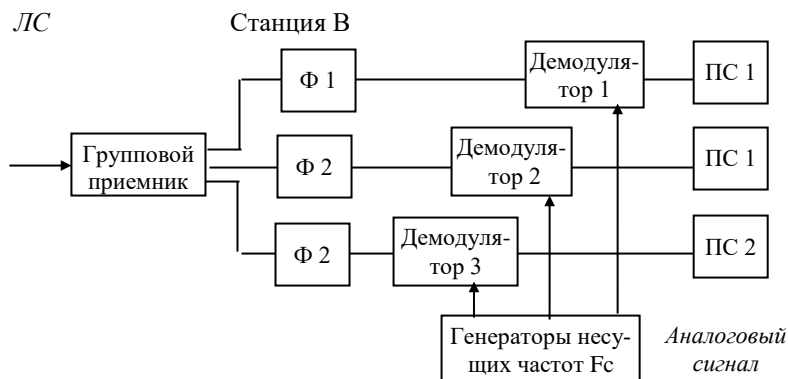


Рис. 3.3. Структурная схема приемной части ЦСП с ВРК

На приемной стороне одновременно действуют сигналы всех каналов, различающиеся положением их частотных спектров на шкале частот. Чтобы без взаимных помех разделить такие сигналы, приемные устройства должны содержать частотные фильтры. Каждый из фильтров должен пропустить без ослабления лишь те частоты, которые принадлежат сигналу данного канала; частоты сигналов всех других каналов фильтр должен подавить.

3.5. Технология выполнения задания 1

Проектируемая модель должна имитировать принцип построения систем передачи с ЧРК, реализовываться в виде блоков Simulink и сопровождаться временными диаграммами и спектрограммами, помогающими разобраться, что происходит в каждой точке, передающей и приемной частей системы.

Процесс моделирования АСП с ЧРК можно условно разбить на несколько этапов:

- I. Моделирование тракта передачи
- II. Моделирование линейного тракта
- III. Моделирование тракта приема.

I. Моделирование тракта передачи

В соответствии с поставленной задачей, надо собрать модель 3-х канальной АСП с ЧРК, которая для упрощения понимания процессов преобразования, будет содержать в качестве источников непрерывных сигналов генераторы синусоидальных напряжений разной частоты и амплитуды (заданных по варианту в табл. 3.1). Модель передающей части приведена на рис. 3.4. В таблице указаны названия блоков, папки в которых находятся нужные блоки для моделирования и параметры блоков, которые нужно изменить при построении каналообразующего оборудования тракта передачи.

Название и назначение блоков	Размещения в браузере	Параметры блоков
Источники сигналов: <i>Sine Wave (DSP) 1...3</i> ,	<i>Signal Processing Blockset / Signal Processing Sources</i>	<i>Amplitude</i> и <i>Frequency (Hz)</i> по варианту (табл. 11.1)
Балансный модулятор блок <i>SSB AM Modulator Pass-band 1...3</i>	<i>Communications Blockset / Modulation / Analog Pass-band Modulation</i>	<i>Carrier frequency (Hz)</i> : // по варианту <i>Sideband to modulate: Lower</i> -нижняя
Блок сложения <i>Add</i>	<i>Simulink/ Math Operations</i>	<i>List of signs: +++</i>
Блок объединения <i>Matrix Concatenate 1, 2</i>	<i>Signal Processing Blockset/Math Functions/ Matrices & Linear Algebra/Math Operations</i>	<i>Number of inputs: 3</i> <i>Mode: Vector</i>
Спектроскоп <i>Spectrum Scope 1...3</i>	<i>Signal Processing Blockset / Signal Processing Sinks</i>	<i>Spectrum type: One-Sided ([0...Fs/2])</i>
Регистратор: <i>Scope</i>	<i>Simulink / Sinks</i>	<i>Number of axes = 4</i>

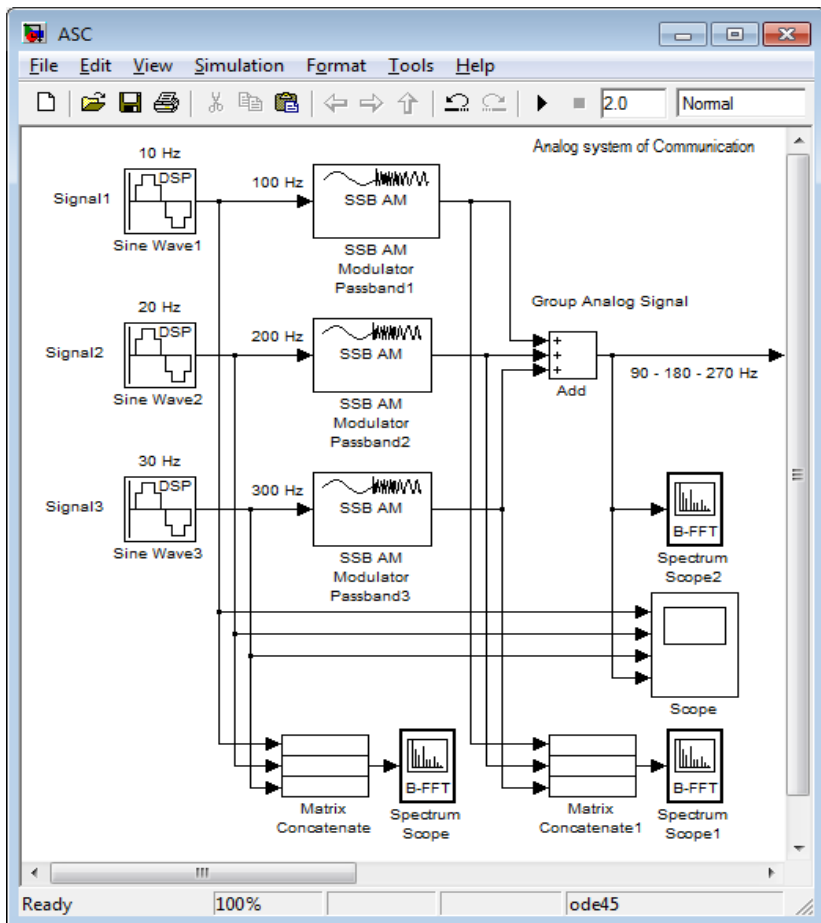


Рис. 3.4. Модель тракта передачи

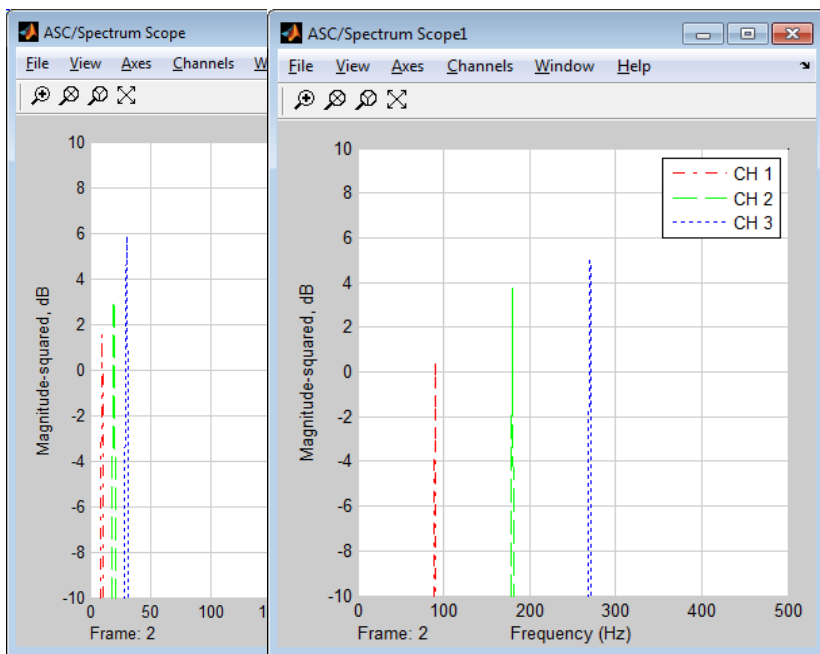
Блок *Sine Wave (DSP)* формирует на своем выходе синусоидальный сигнал с заданной амплитудой, частотой и фазой.

Как известно спектр амплитудно-модулированного колебания состоит из линии несущей частоты F_c и двух боковых полос БП (на рис. 3.1 нижняя БП подавлена). Обе они несут по существу одинаковую информацию о модулирующем сигнале. Удаление одной из боковых полос уменьшает вдвое полосу частот, которую должен иметь приемник сигнала. При этом снижается уровень по-

мех при приеме. Дополнительный выигрыш в помехозащищенности достигается путем удаления несущей частоты, что при заданной мощности передатчика означает существенное повышение уровня сигнала.

Блок *SSB AM Modulator Passband* (балансный модулятор) производит модуляцию синусоидального сигнала по методу балансной модуляции с подавлением несущей частоты и одной из боковых полос. Верхняя *Upper* или нижняя *Lower* боковые полосы формируются с помощью установки их в списке *SideBand to modulate*. В модели остается нижняя БП.

Результаты моделирования представлены спектрами, а) – синусоидальных сигналов рис. 3.5 и б) – нижних боковых составляющих, а также осциллограммами работы модели на рис. 3.6.



а)

б)

Рис. 3.5. Спектрограммы а) источников синусоидальных сигналов (10, 20, 30 Гц) и б) группового сигнала

В модели непрерывные сигналы представлены тремя синусоидами на частотах 10, 20, 30 Гц (рис. 3.5 а). В качестве несущих частот для первого канала принята частота $F_c=100$ Гц, для второго – 200 Гц, для третьего – 300 Гц. В результате модуляции возле трех несущих частот образовались по две боковые составляющие соответственно. Например,

1 частота сигнала $F_{s1}=10$ Гц, $F_{c1}=100$ Гц, $\Delta F_l=100-10=90$ Гц, $\Delta F_u=100+10=110$ Гц (подавляется блоком *SSB1*);

2 частота сигнала $F_{s2}=20$ Гц, $F_{c2}=200$ Гц, $\Delta F_l=200-20=180$ Гц, $\Delta F_u=200+20=220$ Гц (подавляется блоком *SSB2*);

3 частота сигнала $F_{s3}=30$ Гц, $F_{c3}=300$ Гц, $\Delta F_l=300-30=270$ Гц, $\Delta F_u=300+30=330$ Гц (подавляется блоком *SSB3*).

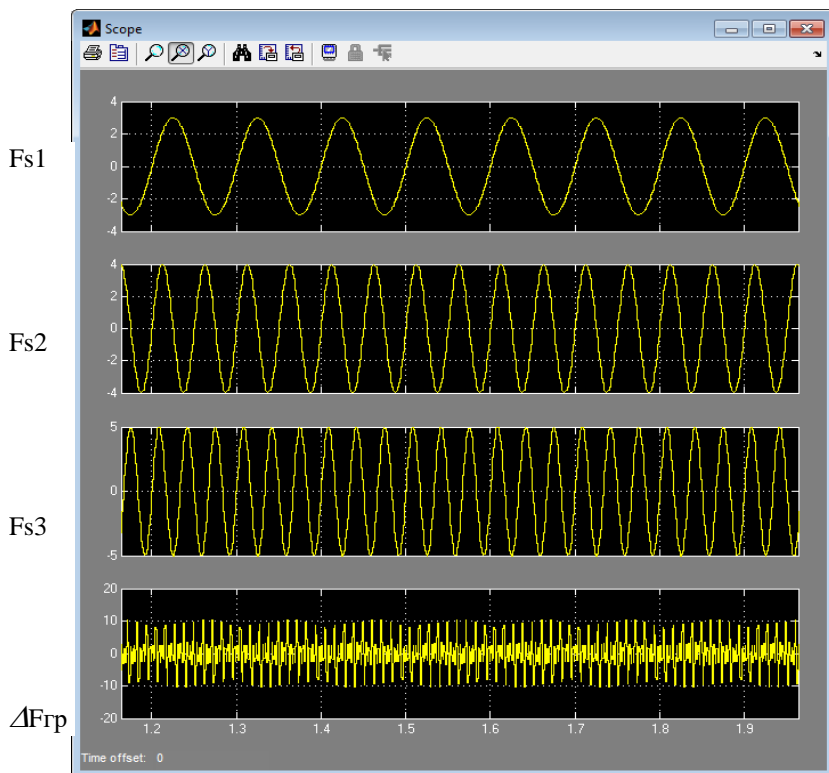


Рис. 3.6. Временные диаграммы работы тракта передачи.

Блок *Matrix Concatenate* объединяет сигналы данных одного типа в единый выходной сигнал. Если все сигналы векторы, то и выходной тоже вектор. В блоке параметров устанавливаются количество входных сигналов (*Number of inputs*) – в модели их 3 по числу объединяемых каналов; и режимы (*Mode*) их объединения: векторное (*Vector*).

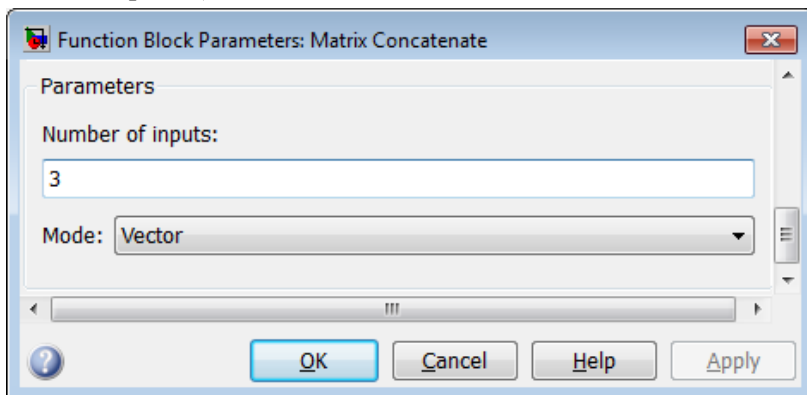


Рис. 3.7. Окно установки параметров блока *Matrix Concatenate*

Для измерения спектральной плотности мощности входных сигналов служит блок *Spectrum Scope*. Окно установки параметров этого блока приведено на рис. 3.8. Как видно окно содержит 4 вкладки:

Scope Properties – позволяет устанавливать:

- тип измерения (спектральную плотность мощности *dBW/Hertz* или среднеквадратичный спектр *Watts*);
- диапазон частот, в котором вычисляется спектр: односторонний ($[0 \dots F_s / 2]$) и двусторонний ($[-F_s / 2 \dots F_s / 2]$) варианты расчета полного спектра мощности;
- подключение входного буфера (флаг *Buffer input*), когда на вход поступают выборки данных и установка при этом размера буфера;
- установка значения числа параметров спектральных средних определяет количество спектров в среднем. Установка этого параметра в 1 эффективно отключает усреднение, 2 включает его.

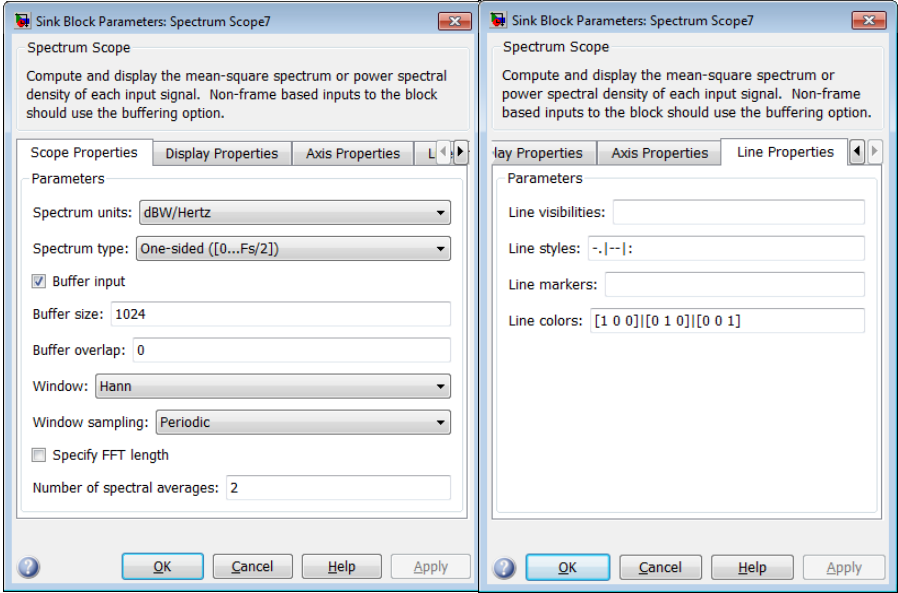


Рис. 3.8. Окно установки параметров блока *Spectrum Scope*

Display Properties – позволяет контролировать, как блок отображает ваши данные (включение сетки, легенды и других параметров).

Axis Properties – позволяет устанавливать свойства осей: смещение по оси частот, диапазон по оси «Y» – *ymin and ymax*.

Line Properties – позволяет форматировать линии спектра:

делая их видимыми (*visibilities*) или нет,

устанавливая стиль линий, например:

- | -- | : | . | (сплошная, длинный штрих,
ch1 ch2 ch3 ch4 точки, штрих-пунктирные);

устанавливая тип маркеров, например:

* | . | x | s | d (снежинки, точки, кресты
ch1 ch2 ch3 ch4 ch5 квадраты, ромбы);

устанавливая цвет линий, например:

[0 0 0] | [0 0 1] | [1 0 0] | [0 1 0] (черный, синий, красный, зелёный).
ch1 ch2 ch3 ch4 (номера линий)

где символ | это знак разделения отображаемых каналов *ch*.

II. Моделирование линейного тракта

Сформированный групповой АМ сигнал представляет собой набор синусоид (рис. 3.6 $\Delta F_{гр}$), которые далее в модели передаются по линейному тракту (рис. 3.9).

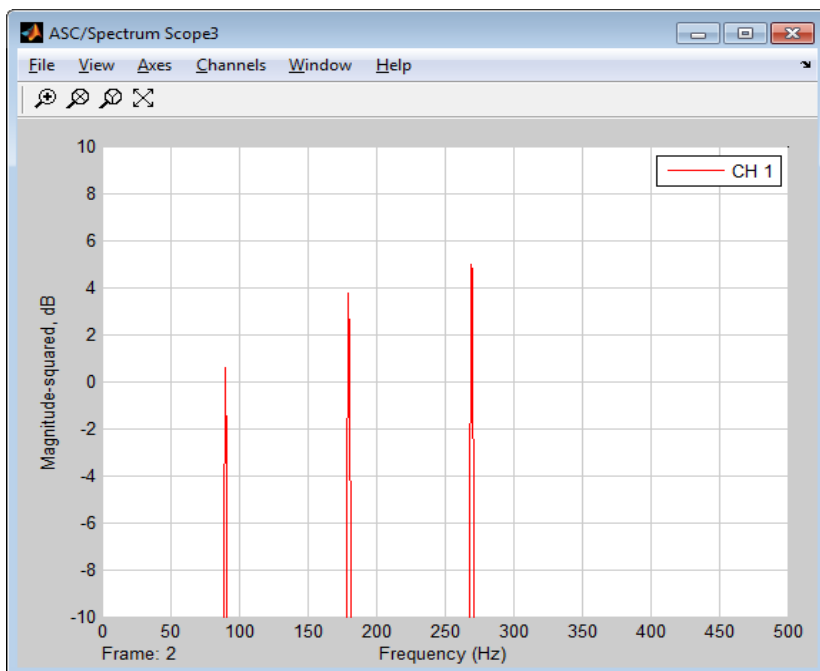


Рис. 3.9. Спектр группового АМ сигнала

В зависимости от протяженности магистрали и используемой аппаратуры ЧРК на том или ином расстоянии магистраль оборудуется обслуживаемыми и необслуживаемыми усилительными пунктами: ОУП и НУП соответственно (в модели они не рассматриваются). Канал включается между передатчиком и приемником сообщений. В качестве канала в модели выберем блок *AWGN Channel*, находящийся в *Communications Blockset /Channels*, который добавляет белый шум к многоканальному линейному сигналу, имитируя помехи в линии связи. Величина шума определяется разными способами. Например:

- E_b/N_0 – отношение энергии бита к спектральной плотности шума.
- E_s/N_0 – отношение энергии сигнала к спектральной плотности шума.
- SNR - отношение мощности сигнала к мощности шума.
- *Variance from masc* – вариации из диалогового окна.
- *Variance from port* – вариации из входного порта.

Блок *AWGN Channel* находится в библиотеке *Communications Blockset /Channels*.

III. Моделирование тракта приема

Многokrратно усиленный линейный сигнал на входе приемной части модели поступает на декодер (в модели для ее упрощения эта часть отсутствует), на выходе которого, формируется групповой АМ-сигнал. Задачей приемной части АСП с ЧКР является восстановление из группового АМ-сигнала синусоидальные сигналы, интерпретируемые как сигналы отдельных 3 каналов. Модель приёмной части приведена на рис. 3.10. В таблице указаны названия блоков, папки в которых находятся нужные блоки для моделирования и параметры блоков, которые нужно изменить при построении оборудования тракта приёма.

Название и назначение блоков	Размещения в браузере	Параметры блоков
Цифровой фильтр <i>Digital Filter Design 1...3</i>	<i>Signal Processing Blockset/Filtering/Filter Implementations</i>	<i>IIR Butterworth Specify order 4</i> Настройка F_s, F_c
Демодулятор <i>SSB AM Demodulator Passband 1...3</i>	<i>Communications Blockset / Modulation/Analog Passband Modulation</i>	<i>Carrier frequency (Hz): // по варианту</i>
Блок объединения <i>Matrix Concatenate 3</i>	<i>Signal Processing Blockset/Math Functions/ Matrices & Linear Algebra/Math Operations</i>	<i>Number of inputs: 3</i> <i>Mode: Vector</i>
Спектроскоп <i>Spectrum Scope 4...7</i>	<i>Signal Processing Blockset / Signal Processing Sinks</i>	<i>Spectrum type: One-Sided ([0...Fs/2])</i>
Регистратор: <i>Scope</i>	<i>Simulink / Sinks</i>	<i>Number of axes = 4</i>

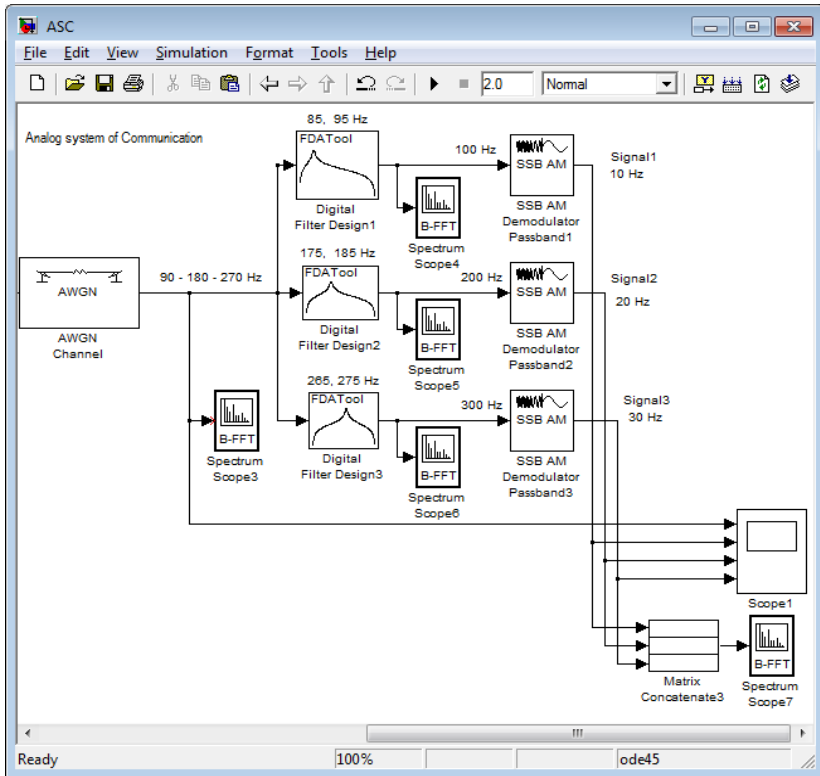


Рис. 3.10. Модель тракта приёма

Назначение фильтра – извлечение или вырезка области спектра входного сигнала или определенной частоты. Поскольку вырезать можно разные области частот, о фильтры подразделяются на 4 типа:

- *Lowpass* – фильтр нижних частот,
- *Highpass* – фильтр верхних частот,
- *Bandpass* – полосовой пропускающий,
- *Bandstop* – полосовой задерживающий.

Для каждого аналогового фильтра возможны методы расчета, использующие математические формулы:

- *Batterworth* с плоской АЧХ в полосе пропускания,
- *Chebyshev 1*. АЧХ с пульсациями в полосе пропускания,

- Chebyshev 2. АЧХ с пульсациями в полосе задерживания,
- Elliptic. АЧХ с пульсациями и в полосе пропускания, и в полосе задерживания.

Цифровые фильтры *Digital Filter Design* могут использовать более сложные алгоритмы для получения наилучших характеристик. Цифровой фильтр имеют две структуры: КИХ и БИХ.

КИХ-фильтры (*FIR*) – с конечной импульсной характеристикой и БИХ-фильтры (*IIR*) – и с бесконечной импульсной характеристикой (Баттерворта, Бесселя, Чебышева или инверсный Чебышева).

Поскольку фильтр цифровой, то на его входе и выходе сигналы представлены в цифровом виде. В окне параметров фильтра необходимо установить **частоту дискретизации, как минимум вдвое превышающую частоту канала.**

Окна параметров цифровых фильтров *Digital Filter Design*, *Digital Filter Design1*, *Digital Filter Design2* приведены на рис. 3.11.

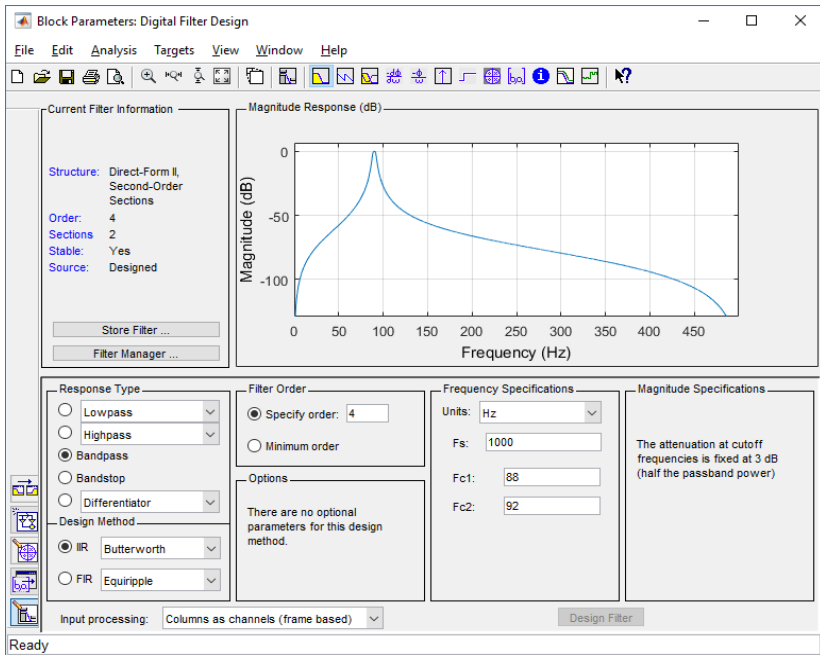


Рис. 3.11 а. Окно установки параметров блока *Digital Filter Design*

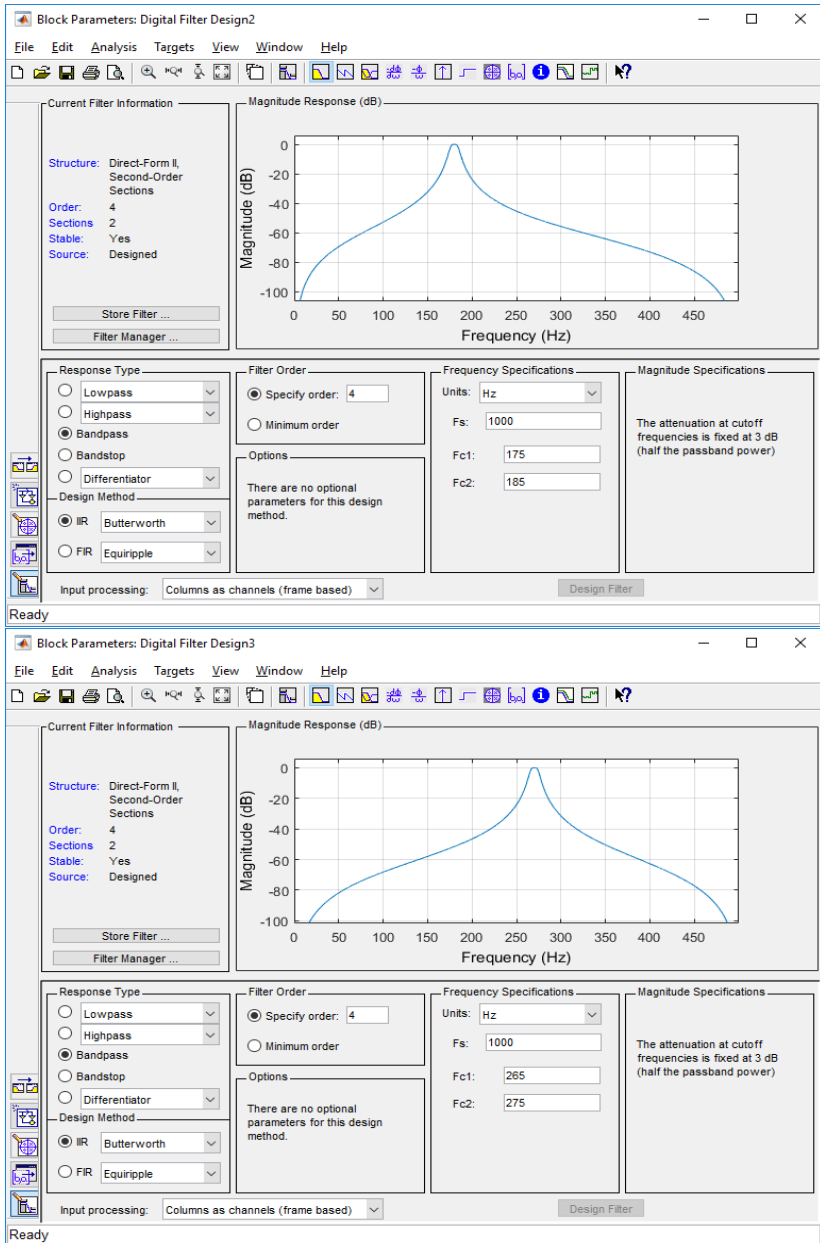


Рис. 3.11. Окна установки параметров блока *Digital Filter Design1,2*

Поскольку групповой сигнал содержит нижние боковые составляющие на частотах 90, 180 и 270 Гц соответственно для 1, 2, 3 каналов, то надо выбрать тип фильтра полосовой, с частотами среза соответственно (рис. 3.11):

- 85 – 95 Гц (для первого канала с $F_c = 90$ Гц),
- 175 – 185 Гц (для второго канала с $F_c = 180$ Гц),
- 265 – 275 Гц (для третьего канала с $F_c = 280$ Гц),

В окне сверху слева указана характеристика полосового фильтра, вырезающего частоту конкретного канала, справа – его частотная характеристика. Снизу слева на право: тип фильтра (*Bandpass* – полосовой) и его реализация (*IIR* - БИХ) – Баттерворта 4 порядка (*Specify order*). Далее, например, для 3 канала, частота дискретизации установлена 1000 Гц ($\gg 2 \cdot 300$ Гц), а частоты среза соответственно 265 – 275 Гц.

Для двух других фильтров необходимо установить частоты среза соответственно 175 – 185 Гц (для 2-го канала) и 85 – 95 Гц (для 1-го каналов). Спектры, вырезанные этими фильтрами, приведены на рис. 3.12. Как видно они полностью совпали со спектрами с выходов АМ модуляторов (см. рис. 3.5 б).

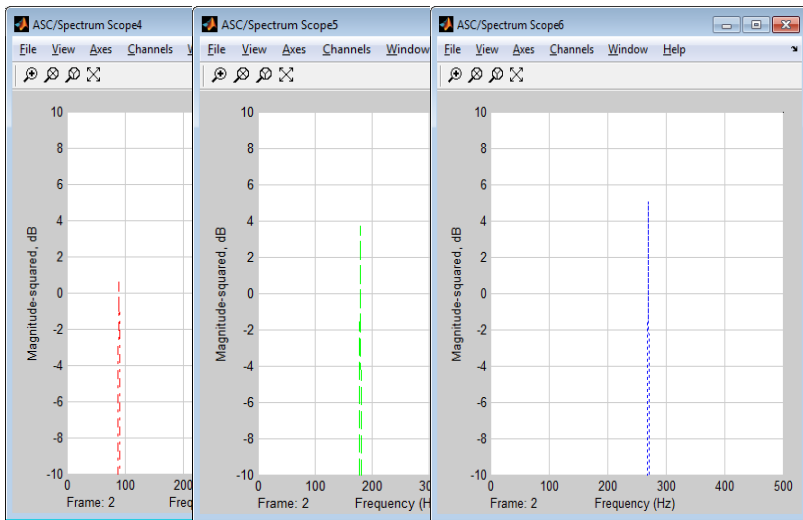


Рис. 3.12. Спектры канальных сигналов

Блок балансного демодулятор *SSB AM Demodulator Passband*

производит демодуляцию канальных сигналов по методу балансной модуляции с подавлением самой несущей частоты и одной из боковых полос. В модели остается нижняя (*lowpass*).

В окне параметров блока *SSB AM Demodulator Passband* (рис. 3.13 б) следует установить:

- номинал несущей частоты соответствующей рабочему каналу по варианту (на рисунке F_{c1} первого канала),
- реализацию фильтра (*Butterworth*) – Баттерворта 4 порядка (*Filter order*)
- частоту среза 100 Гц. Частота отсечки фильтра нижних частот указана в нижних частотах для конструкции фильтра заданным методом в герцах.

В результате демодуляции спектры канальных сигналов переносятся в более низкую область рис. 3.13 а) каждый из которых попадает в свой приемник. Нетрудно заметить по спектрограмме, что спектры приемных сигналов полностью соответствуют спектрам трех источников синусоидальных сигналов 10, 20 и 30 Гц.

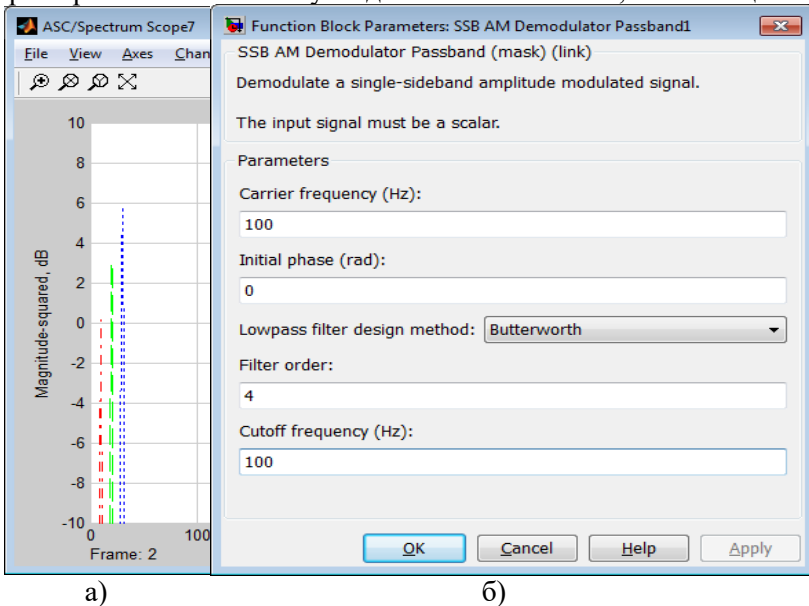


Рис. 3.13. а) Спектры сигналов – приемников, б) окно установки параметров блока *SSB AM Demodulator Passband*.

На временных диаграммах рис. 3.14 изображен групповой амплитудно-модулированный сигнал, представляющий собой совокупность всех трех каналов, из которого в тракте приема цифровые фильтры пропустили без ослабления лишь те частоты, которые принадлежат сигналу данного канала (F_{s1} или F_{s2} или F_{s3}); частоты сигналов всех других каналов фильтры подавили.

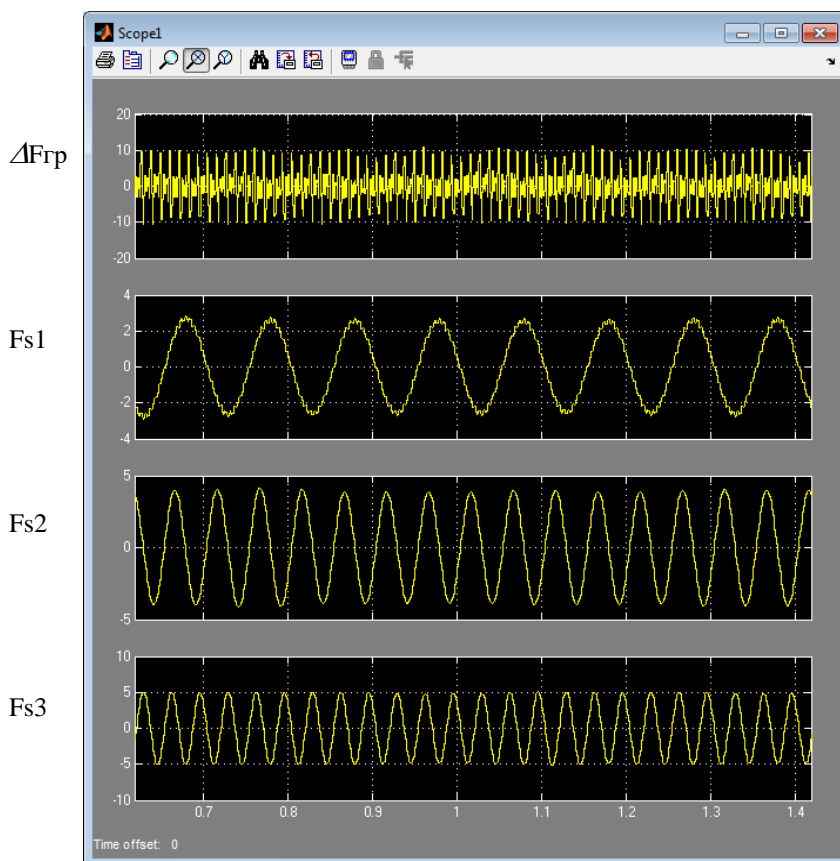


Рис. 3.14. Временные диаграммы работы тракта приема.

3.6. Технология выполнения задания 2

Групповой сигнал представляет собой совокупность всех каналов, различающиеся положением их частотных спектров на

шкале частот. Поэтому местоположение группы зависит от величины несущей частоты и выбора боковой полосы при балансной модуляции. Кроме того, восстановленный каналный сигнал на приемном конце зависит от настройки каналных фильтров. Поэтому, уменьшение величины несущих частот в два раза по сравнению с заданными значениями может привести к перекрытию спектров каналов и невозможности восстановления источников.

На приемной стороне одновременно действуют сигналы всех каналов, различающиеся положением их частотных спектров на шкале частот. Чтобы без взаимных помех разделить такие сигналы, приемные устройства должны содержать частотные фильтры. Каждый из фильтров должен пропустить без ослабления лишь те частоты, которые принадлежат сигналу данного канала; частоты сигналов всех других каналов фильтр должен подавить.

На практике это трудно выполнить, так как возникают взаимные помехи между каналами. Это происходит, во-первых, за счет неполного сосредоточения энергии сигнала k -го канала в пределах заданной полосы частот, во-вторых, за счет не идеальности реальных полосовых фильтров.

Для снижения переходных помех до допустимого уровня вводятся защитные частотные интервалы $\Delta\omega_{\text{защ}}$ (Рис. 3.15).

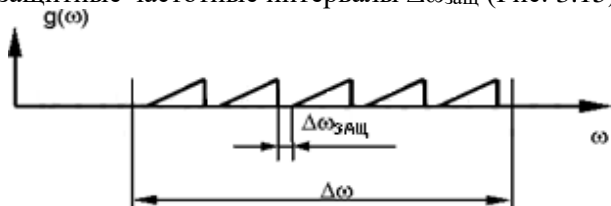


Рис. 3.15. Спектр группового сигнала с защитными интервалами

Так, например, в современных системах многоканальной телефонной связи каждому телефонному каналу выделяется полоса частот 4 кГц, хотя частотный спектр передаваемых звуковых сигналов ограничивается полосой от 300 до 3400 Гц, т.е. ширина спектра составляет 3,1 кГц. Между полосами частот соседних каналов предусмотрены интервалы шириной по 0,9 кГц, предназначенные для снижения уровня взаимных помех при рас- фильтровке сигналов.

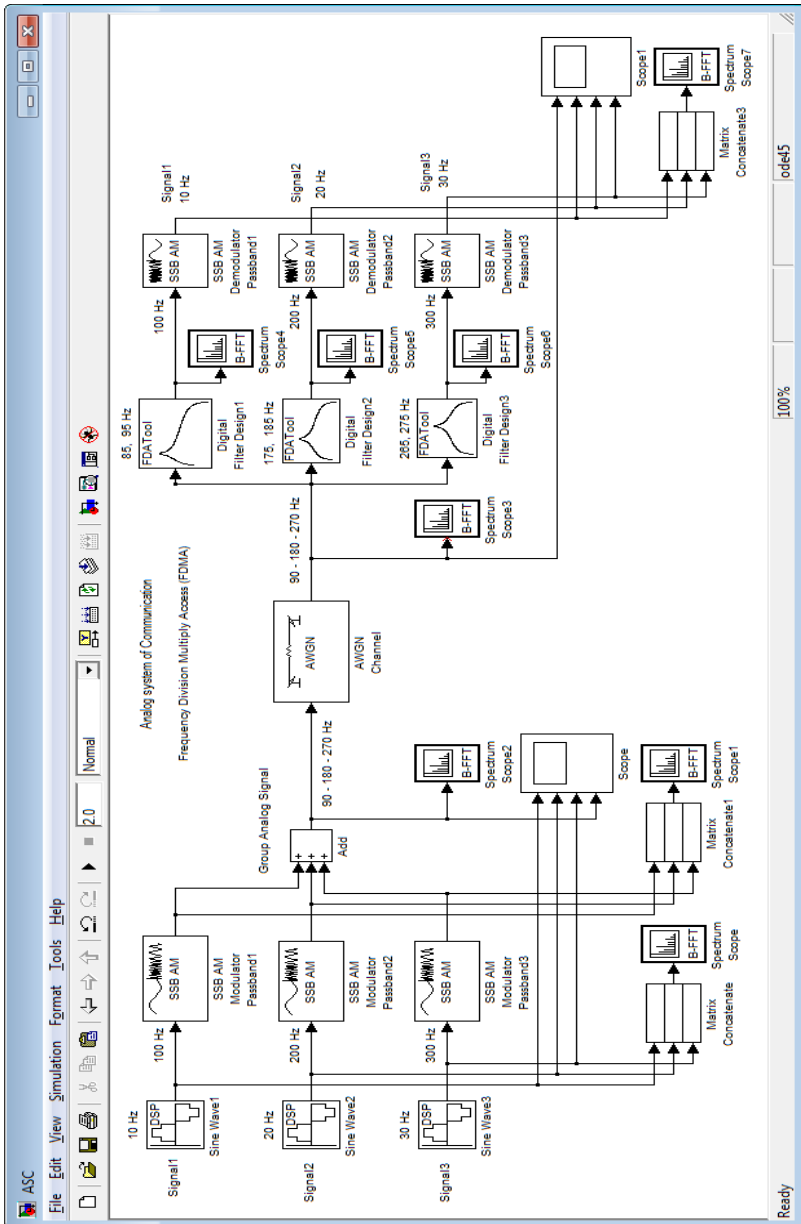


Рис. 3.16. Модель АСП с ЧРК

Поскольку смоделированная система передачи имеет однотипные элементы, то целесообразно создать из них подсистемы. Например, в тракте приема для каждого канала используются полосовые фильтры и демодуляторы, то их можно объединить в подсистему. Фрагмент тракта приема с образованными подсистемами и содержание подсистем приведены на рис. 3,17.

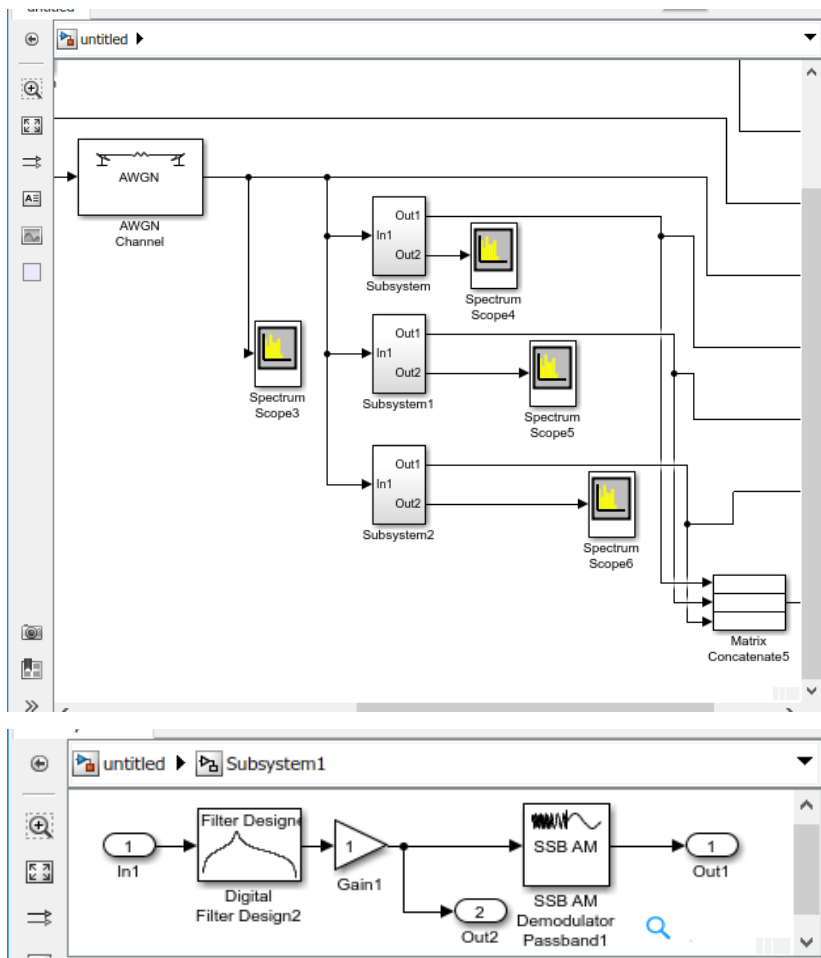


Рис. 3.17. Модель канальных подсистем (Filter+SSB AM) при построении аналоговой системы передачи

Пример контрольного теста «Системы передачи с ЧРК»

Укажите все блоки, относящиеся к тракту передачи:

Sine Wave

Constant

Modulator Passband

Divide

Add

Укажите все блоки, относящиеся к тракту приёма:

Digital Filter Design

Add

AWGN Channel

Demodulator Passband

Sine Wave

Укажите отсутствующий тип фильтра:

Lowpass

Highpass

Bandpass

Bandstop

Averagepass

Укажите нижнюю боковую полосу частот модулированного сигнала, если частота источника 200 Гц, а частота несущего колебания 10 кГц:

10 200 Гц

9 800 Гц

400 Гц

20 000 Гц

500 Гц

Укажите частоту дискретизации цифрового фильтра при восстановлении аналогового сигнала частотой 4 кГц:

4 кГц

2 кГц

6 кГц

9 кГц

8 кГц

4. Модель цифровой системы передачи

4.1. Подготовка к работе

По указанной литературе изучить:

- свойства подбиблиотек *Simulink: Discontinuous, Continuous*;
- принципы построения систем с временным разделением каналов.

4.2. Задание и порядок выполнения работы

1. Задача 1. Создать модель четырехканальной цифровой системы передачи (ЦСП). Характеристики аналоговых сигналов в виде амплитуды (A) и частоты (F) синусоидальных сигналов задаются по вариантам в табл. 4.1. Определить необходимую для 4-х каналов частоту дискретизации.

Таблица 4.1. Варианты заданий

N	Амплитуда и частота синусоидальных сигналов							
	1 канал		2 канал		3 канал		4 канал	
	A	$F1$, Гц	A	$F2$, Гц	A	$F3$, Гц	A	$F4$, Гц
1	2	3	3	4	4	2	5	6
2	3	2	4	4	3	5	1	7
3	5	4	3	2	4	6	2	5
4	6	2	5	3	2	4	3	8
5	4	5	2	4	5	3	4	6
6	6	4	3	5	2	7	4	3
7	3	4	4	5	2	3	3	6
8	5	5	4	3	4	6	5	2
9	2	4	3	3	5	6	4	5
10	4	6	5	2	3	4	6	3
11	3	2	4	5	6	4	4	3
12	5	3	6	4	7	5	2	6
13	4	2	3	4	4	5	5	7
14	5	2	4	3	3	4	2	8
15	6	4	4	5	3	7	5	3
16	2	6	3	2	5	4	4	3

2. Иллюстрировать работу модели временными диаграммами.
3. Модель и диаграммы сохранить и поместить в отчет.
4. Задача 2. Изменить частоту дискретизации, уменьшив ее в два раза по сравнению с расчетным значением. Проанализировать полученный результат, и сделать выводы, как величина частоты дискретизации влияет на группообразование и восстановление исходных сигналов.
5. Сделать выводы о возможностях моделирования систем с использованием пакета расширений *Simulink*.

4.3. Методические указания

Структурная схема модели передающей части цифровой системы передачи с временным разделением каналов (ЦСП с ВРК) изображена на рис. 4.1.

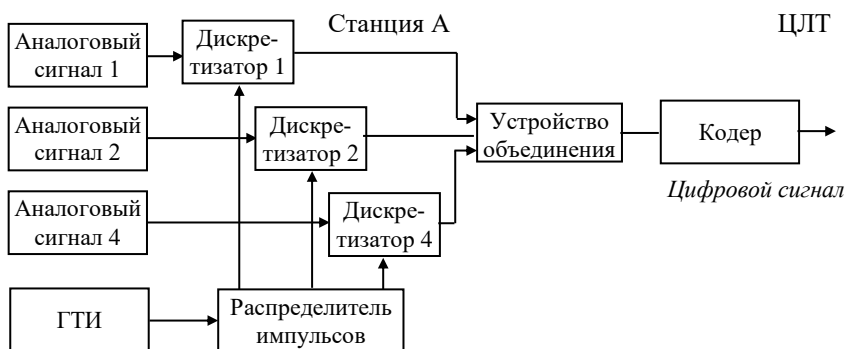


Рис. 4.1. Структурная схема передающей части ЦСП с ВРК

Аналоговые сигналы (в модели гармонические колебания с частотами F_1, \dots, F_4) с ограниченным спектром поступают на электронные ключи, осуществляющие периодически дискретизацию непрерывных сигналов с частотой дискретизации F_d . Дискретные импульсы объединяются в групповой АИМ сигнал. Управляет работой ключей каналные импульсы, поступающие от генераторного оборудования передачи. Эти импульсы подаются поочередно, чем и обеспечивается правильное формирование группового АИМ сигнала. Групповой АИМ сигнал затем поступает на

модулирующее устройство – кодер, который одновременно осуществляет операции квантования по уровню и кодирование. Сформированный импульсно-кодový модулированный (ИКМ) сигнал представляет собой набор двоичных символов, который и передается по ЦЛТ.

Схема имитирует принцип построения систем передачи с ИКМ, реализуется в виде блоков Simulink и сопровождается временными диаграммами, помогающими разобраться, что происходит в каждой точке передающей части модели.

Структурная схема построения приемной части цифровой системы передачи с временным разделением каналов, изображена рис. 4.2.

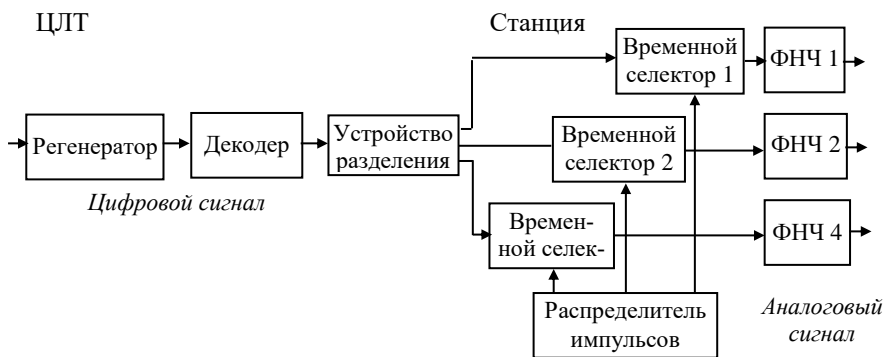


Рис. 4.2. Структурная схема приемной части ЦСП с ВРК

В процессе передачи по линии связи цифровой сигнал периодически восстанавливается (регенерируется) линейными регенераторами. На приемной станции линейный сигнал восстанавливается станционными регенераторами. Восстановленный ИКМ сигнал декодируется в групповой АИМ сигнал и распределяется устройством разделения по своим каналам, работой которых управляет генераторное оборудование. Последовательности канальных импульсов поочередно открывают временные селекторы каналов, обеспечивая выделение отсчетов своего канала из группового АИМ сигнала. Восстановление непрерывного сигнала из последовательности его отсчетов производится с помощью фильтров нижних частот.

4.3. Технология выполнения задания 1

Процесс моделирования ЦСП можно условно разбить на несколько этапов:

- I. Моделирование канального оборудования тракта передачи
- II. Моделирование генераторного оборудования.
- III. Моделирование группового тракта.
- IV. Моделирование тракта приема.

I. Моделирование канального оборудования тракта передачи

В соответствии с поставленной задачей, надо собрать модель 4-х канальной ЦСП с ВРК, которая для упрощения понимания процессов преобразования, будет содержать в качестве источников непрерывных сигналов генераторы синусоидальных напряжений разной частоты и амплитуды (заданных по варианту в табл. 4.1). Модель передающей части индивидуальных каналов приведена на рис. 4.3, а временные диаграммы, подтверждающие ее работоспособность на рис. 4.4.

Синусоидальные сигналы заданной частоты подаются на входы ключей, выполняющих роль дискретизаторов, вторые входы которых управляются импульсами с генераторного оборудования.

В таблице указаны названия блоков, папки в которых находятся блоки для моделирования и параметры блоков, которые нужно изменить при построении тракта передачи.

Название и назначение блоков	Размещения в браузере	Параметры блоков
Источники сигналов: <i>Sine Wave (DSP)</i> ,	<i>Signal Processing Block / Signal Processing Sources.</i>	<i>Amplitude</i> и <i>Frequency (Hz)</i> по варианту (табл. 13.1)
Переключатель <i>Switch 1..4</i>	<i>Simulink / Signal Routing</i>	По умолчанию
Регистратор: <i>Scope</i>	<i>Simulink / Sinks</i>	<i>Number of axes = 4</i>

Блок *Sine Wave (DSP)* формирует на своем выходе синусоидальный сигнал с заданной амплитудой, частотой и фазой.

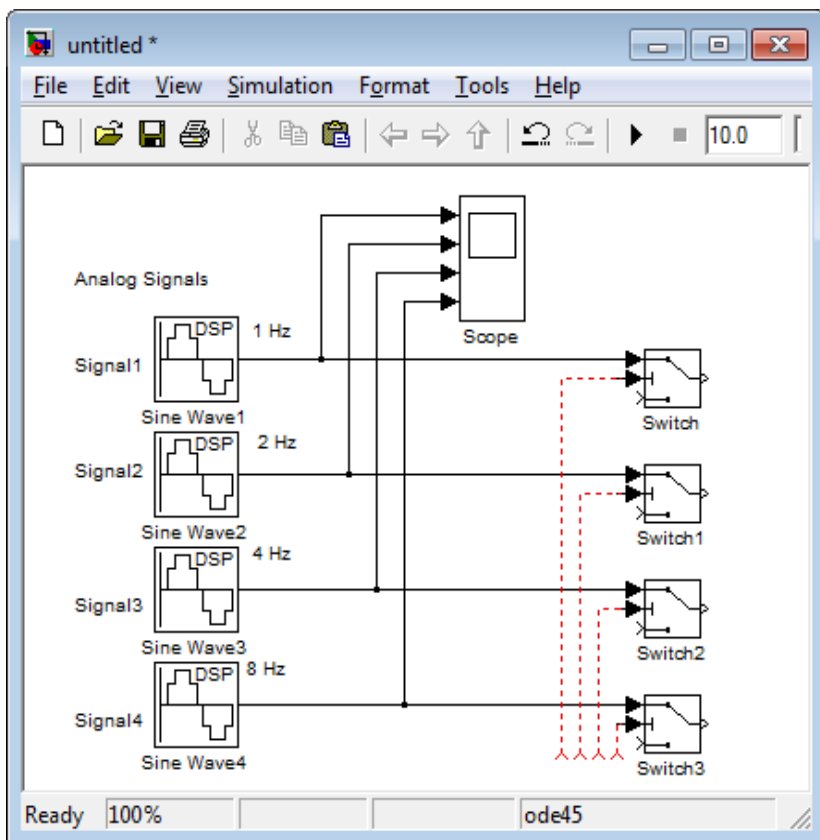


Рис. 4.3. Модель канального оборудования тракта передачи

II. Моделирование генераторного оборудования

Управляет работой ключей тракта передачи канальные импульсы, поступающие от генераторного оборудования передачи поочередно, что обеспечивает правильное формирование группового АИМ сигнала. Импульсы распределены во времени с частотой, согласующейся с теоремой Котельникова $F_d \geq 2 \cdot F$.

В таблице указаны названия блоков, папки в которых находятся блоки для моделирования и параметры блоков, которые нужно изменить при построении генераторного оборудования.

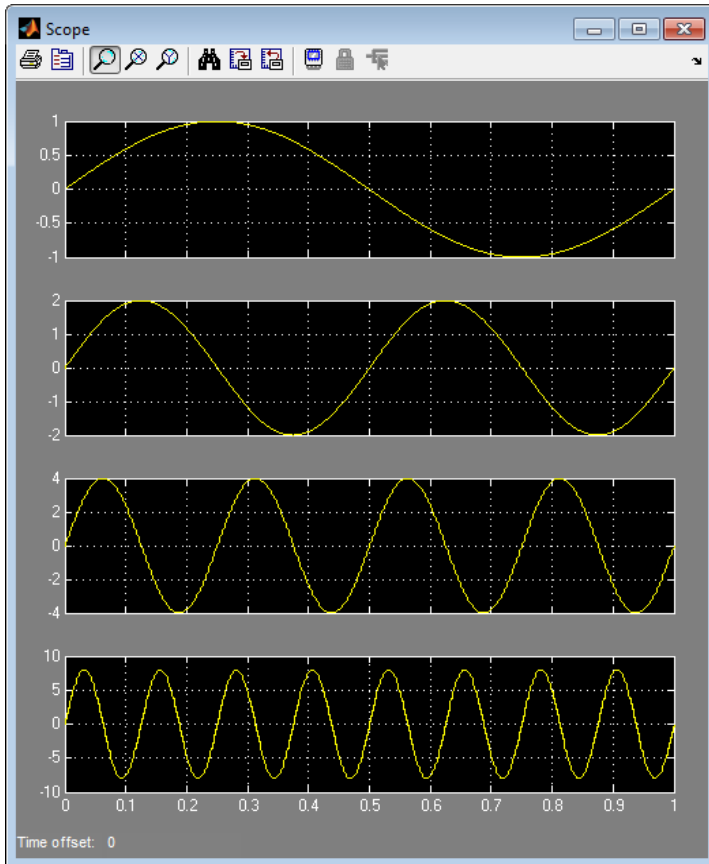


Рис. 4.4. Диаграммы источников аналоговых сигналов

* Например, если по варианту задаются сигналы с частотами 1, 2, 3, 4 Гц, то частоту дискретизации нужно принять для всех четырех сигналов $> 2 \cdot 4$ Гц, как минимум, 8 Гц, а лучше 16 Гц. Тогда период следования источника тактовой частоты равен: $1/F = 1/16 = 0.0625$ с.

Название и назначение блоков	Размещения в браузере	Параметры блоков
Источник тактовой частоты: <i>Pulse Generator</i>	<i>Simulink</i> / <i>Sources</i>	<i>Amplitude 1</i> <i>*Period = 0.0625 (secs)</i> <i>Pulse Width 5 (%)</i>

Делитель частоты на триггерах: <i>D Flip-Flop1..2</i>	<i>Simulink / Extras / Flip-Flops</i>	По умолчанию
Управляющий сигнал: <i>Constant</i>	<i>Simulink / Sources</i>	<i>Constant value: 1</i>
Логический блок: <i>Logical Operator (And 1..4)</i>	<i>Simulink / Logic and Bit Operations</i>	<i>Operator: AND (3-х портовый)</i>
Регистратор: <i>Scope 1</i>	<i>Simulink / Sinks</i>	<i>Number of axes = 5</i>

Модель генераторного оборудования и временные диаграммы его работы приведены на рис. 4.6 и 4.5 соответственно.

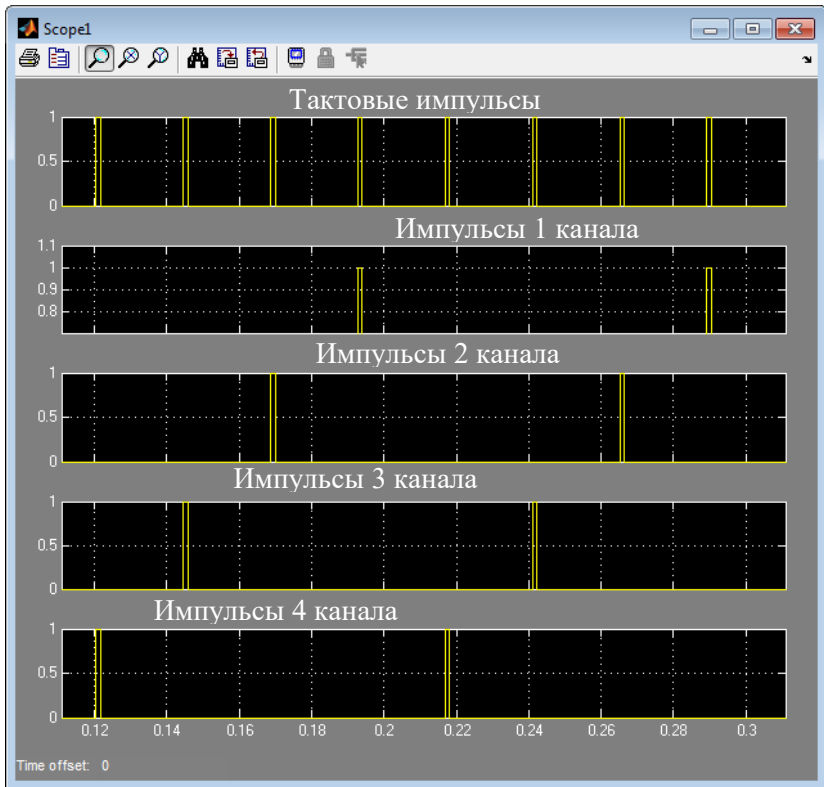


Рис. 4.5. Диаграммы работы генераторного оборудования

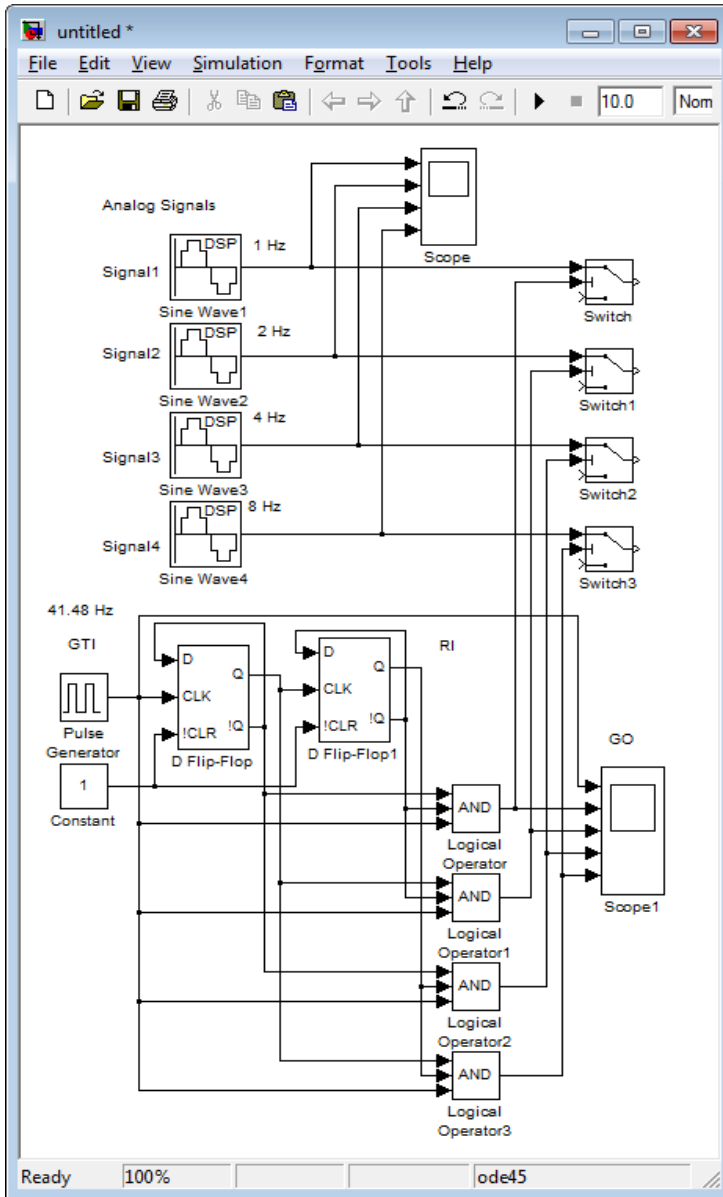


Рис. 4.6. Модель генераторного оборудования и индивидуальной части тракта передачи

Генератор тактовых импульсов формирует частоту дискретизации, которая затем делится с помощью 2-х триггеров на 4 (по числу каналов). Далее происходит формирование импульсов, распределенных во времени с помощью схем И для управления работой ключей-дискретизаторов.

III. Моделирование оборудования группового тракта

На выходе ключей формируются амплитудно-модулированные сигналы (АИМ). Эти сигналы подаются на вход мультиплексора (рис. 4.8), который объединяет их в один, групповой АИМ сигнал (рис. 4.7 *Scope2*).

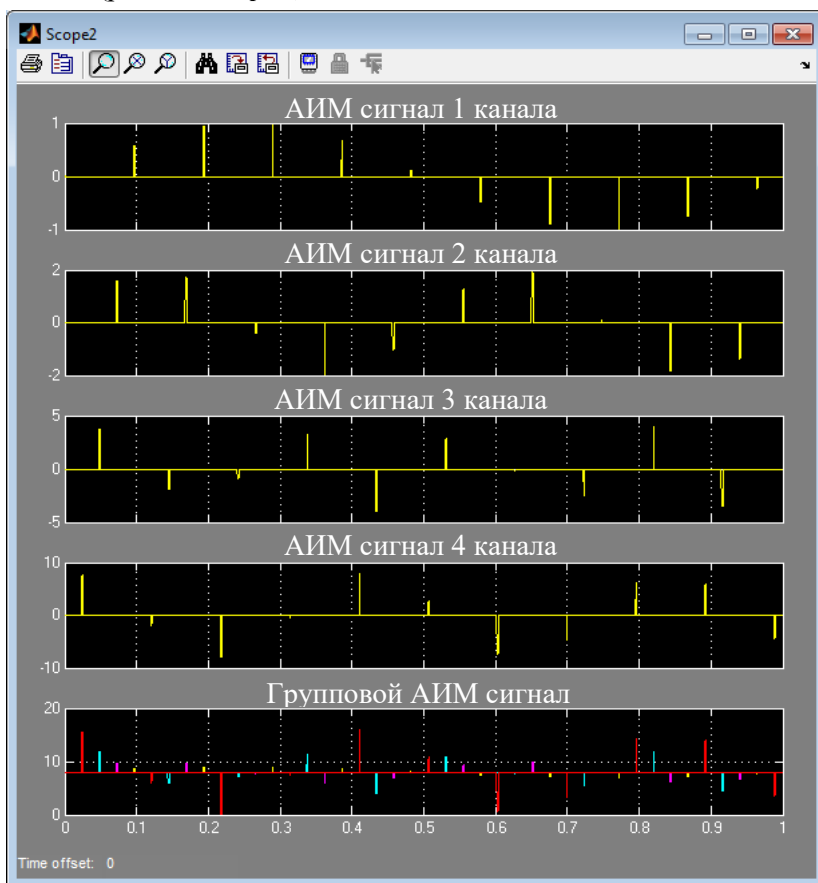


Рис. 4.7. Формирование группового АИМ сигнала.

Поскольку групповой АИМ сигнал является двух полярным, а устройство квантования работает с однополярным сигналом, то с помощью блока *Bias* устраняются отрицательные значения амплитуд синусоид. Если по варианту амплитуды сигнала заданы, например, 1, 2, 4, 8, то смещение следует установить 8 единиц, равное максимально большему значению амплитуды аналогового (синусоидального) источника сигнала.

Групповой АИМ сигнал далее поступает на квантователь (рис. 4.8), который округляет его до ближайшего разрешенного уровня. А затем уже происходит его кодирование. Сформированный ИКМ сигнал представляет собой набор двоичных символов, который далее передается по цифровому линейному тракту.

В таблице указаны названия блоков, папки в которых находятся блоки для моделирования и параметры блоков, которые нужно изменить при формировании модели группового тракта.

Название и назначение блоков	Размещения в браузере	Параметры блоков
Устройство объединения: <i>Mux</i>	<i>Simulink / Signal Routing</i>	Number of inputs 4 // количество объединяемых входов
Математический блок <i>Bias</i>	<i>Simulink/ Math Operations</i>	Bias = 8 (смещение относительно А)
Блок экстраполяции сигнала на интервале дискретизации: <i>Zero-Order Hold</i>	<i>Simulink / Discrete</i>	<i>Sample time</i> = 0.0625 (период следования тактовых импульсов)
Блок квантования: <i>Quantizer</i>	<i>Simulink / Discontinuities</i>	Quantization interval = 0.5
Блок округления: <i>Rounding Function</i>	<i>Simulink / Math Operations</i>	Functions: round
Регистраторы: <i>Scope2, Scope3</i>	<i>Simulink / Sinks</i>	<i>Number of axes</i> = 5 и <i>Number of axes</i> = 2
Регистратор: <i>To Workspace</i>	<i>Simulink / Sinks</i>	По умолчанию

Блок *Zero-Order Hold* 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.

Блок *Quantizer* обеспечивает квантование входного сигнала с одинаковым шагом по уровню.

Кодовые группы в модели представлены в двух вариантах:

- в виде временной диаграммы (рис. 4.9) и
- в виде непосредственной записи в матрице в десятичном эквиваленте (с помощью блока моделирования *Simout*).

Блок *To Workspace* записывает данные, поступающие на его вход в рабочую область системы *MATLAB*. Записанные значения можно просмотреть в окне *Workspace* системы *MATLAB*.

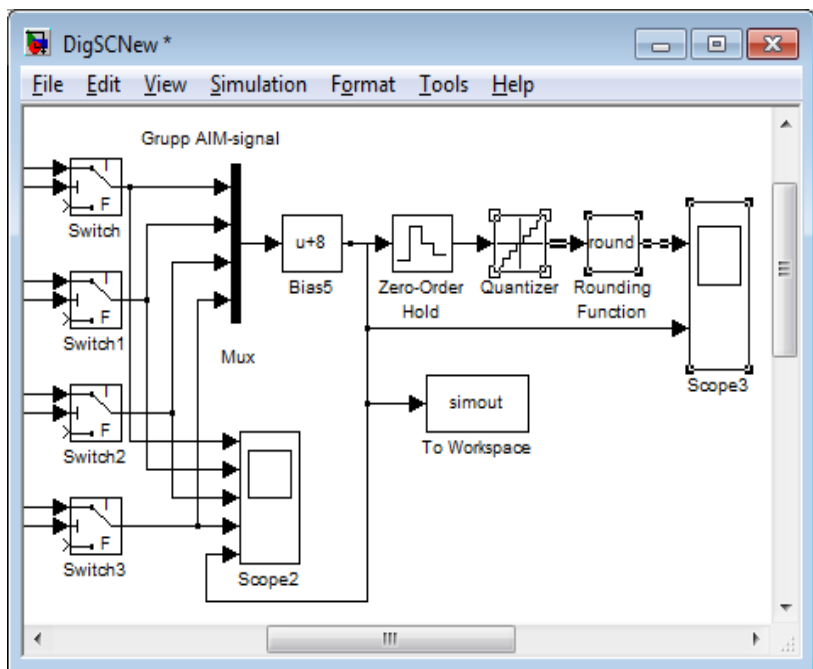
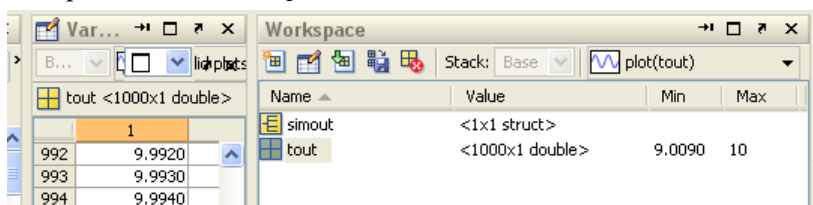


Рис. 4.8. Моделирование группового тракта

Как видно (рис. 4.9), квантованный сигнал представляет собой округленный сигнал до ближайшего разрешенного уровня.

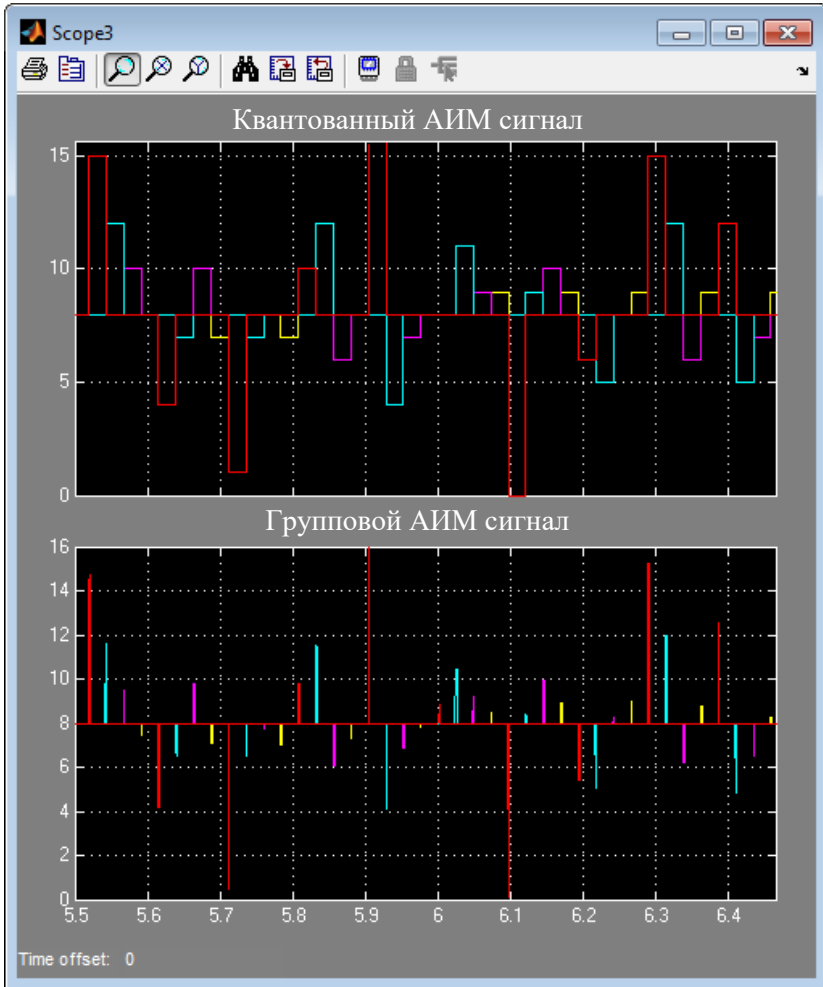


Рис. 4.9. Диаграммы работы групповой части тракта передачи

IV. Моделирование тракта приема

Задача приемной части ЦСП является восстановление из группового цифрового сигнала аналоговых сигналов, интерпретируемых нами как сигналы отдельных 4 каналов.

Линейный сигнал на входе приемной части модели усиливается, затем восстанавливается в стационарном регенераторе и поступает на декодер, на выходе которого, формируется групповой АИМ сигнал. С помощью демультиплексора групповой сигнал разделяется на четыре сигнала, каждый из которых, поступает в свой индивидуальный канал приема. Для формирования из импульсных сигналов аналоговых на выходах каждого из каналов модели устанавливаются сглаживающие фильтры нижних частот. Фильтры выделяют огибающую импульсного сигнала, формируя тем самым непрерывный (в примере синусоидальный) сигнал с заданными частотой и амплитудой.

В таблице указаны названия блоков, папки в которых находятся блоки для моделирования и параметры блоков, которые нужно изменить при формировании тракта приема.

Название и назначение блоков	Размещения в браузере	Параметры блоков
Блок экстраполяции сигнала: <i>Zero-Order Hold</i>	<i>Simulink / Discrete</i>	<i>Sample time = 0.0625</i> (период следования тактовых импульсов)
Математический блок <i>Bias1</i>	<i>Simulink/ Math Operations</i>	<i>Bias1 = -1,25</i> (смещение подбирается индивидуально в каждом варианте)
Усилитель <i>Gain 1..4</i>	<i>Simulink/ Math Operations</i>	<i>Gain</i> //Подбирается при настройке на приеме
Устройство разделения: <i>Demux</i>	<i>Simulink / Signal Routing</i>	<i>Number of outputs: 4</i> // количество выходов
Фильтр: <i>Digital Filter Design 1..4</i>	<i>Signal Processing Blokset / Filtering / Filter Designs</i>	<i>IIR Butterworth</i> // настройки Фильтра приведены на рис. 13.10
Регистратор: <i>Scope 4</i>	<i>Simulink / Sinks</i>	Number of axes = 4

Настройка и установка параметров фильтра происходит в следующей последовательности (см. рис. 4.10):

- выбрать вид фильтра (в примере *Bandpass* – пропускающий конкретную частоту, заданную по варианту),
- выбрать метод расчета (в примере *IIR Butterworth*),
- установить данные о частоте дискретизации цифрового фильтра (вдвое превышающую частоту канала) $F_s = 128$ Гц,
- установить границы среза для заданных по варианту источников сигнала. Например,
 - 0.9 Гц – 1.1 Гц (для первого канала с $F_c = 1$ Гц),
 - 1.9 Гц – 2.1 Гц (для первого канала с $F_c = 2$ Гц),
 - 3.9 Гц – 4.1 Гц (для первого канала с $F_c = 4$ Гц),
 - 7.9 Гц – 8.1 Гц (для первого канала с $F_c = 8$ Гц),
- запустить подпрограмму расчета кнопкой *Design Filter*. После расчета в окне параметров изменится АЧХ фильтра.

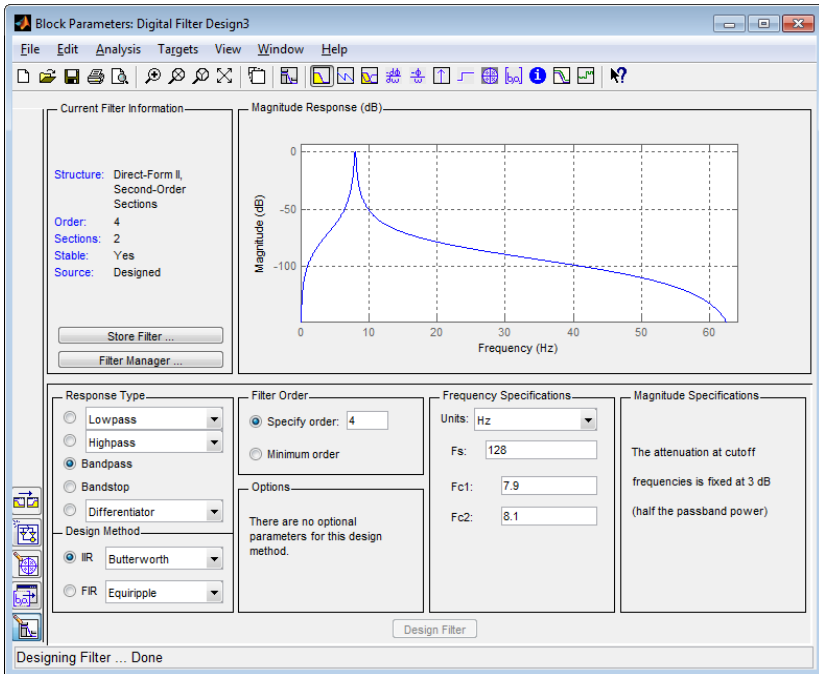


Рис. 4.10. Окно параметров ФНЧ четвертого канала

В окне сверху слева указана характеристика полосового фильтра вырезающего частоту четвертого канала (8 Гц), справа – его частотная характеристика. Снизу слева на право: тип фильтра

(*Bandpass* – полосовой) и его реализация (*IIR* - БИХ) – *Баттерворта* 4 порядка (*Specify order*). Далее частота дискретизации установлена 128 Гц ($\gg 2 \cdot 8$ Гц), а частоты среза соответственно 7.9 – 8.1 Гц.

Процесс работы приемной части модели синхронизируется от того же источника тактовой частоты, что и передающая часть. На рис. 4.11 представлена модель приемной части ЦСП, а на рис. 4.12 диаграммы, ее работы.

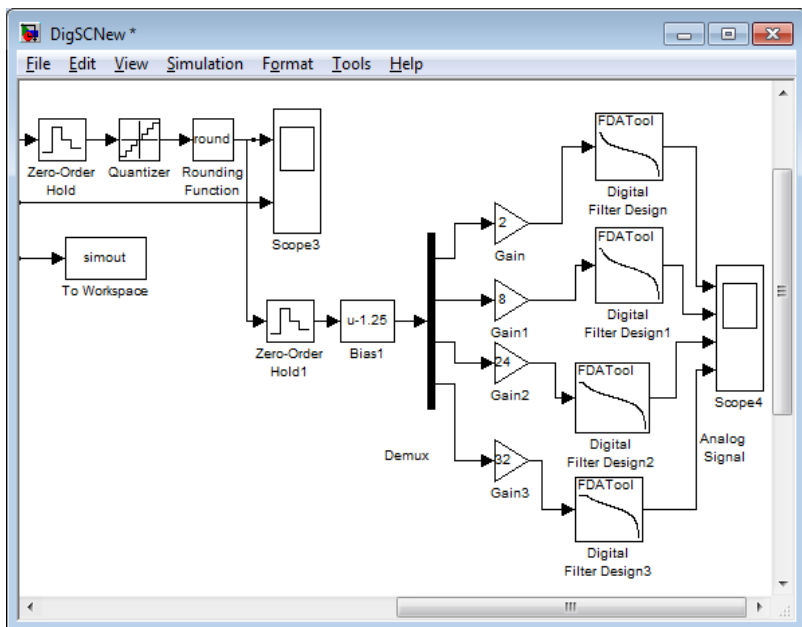


Рис. 4.11. Модель тракта приема

Как показали результаты моделирования, восстановленные 4 непрерывных сигнала (рис. 4.12), полностью соответствуют сигналам источников сообщений (рис. 4.4). Использование цифровых фильтров в тракте приема привело к незначительным флуктуациям по форме восстановленных сигналов.

Если результаты восстановленных сигналов разнятся с исходными значениями, то нужно произвести дополнительную настройку параметров фильтров ФНЧ.

Полная модель ЦСП, построенная поэтапно, приведена на

рис. 4.13. Полученную модель с диаграммами, подтверждающими ее работоспособность поместить в отчет.

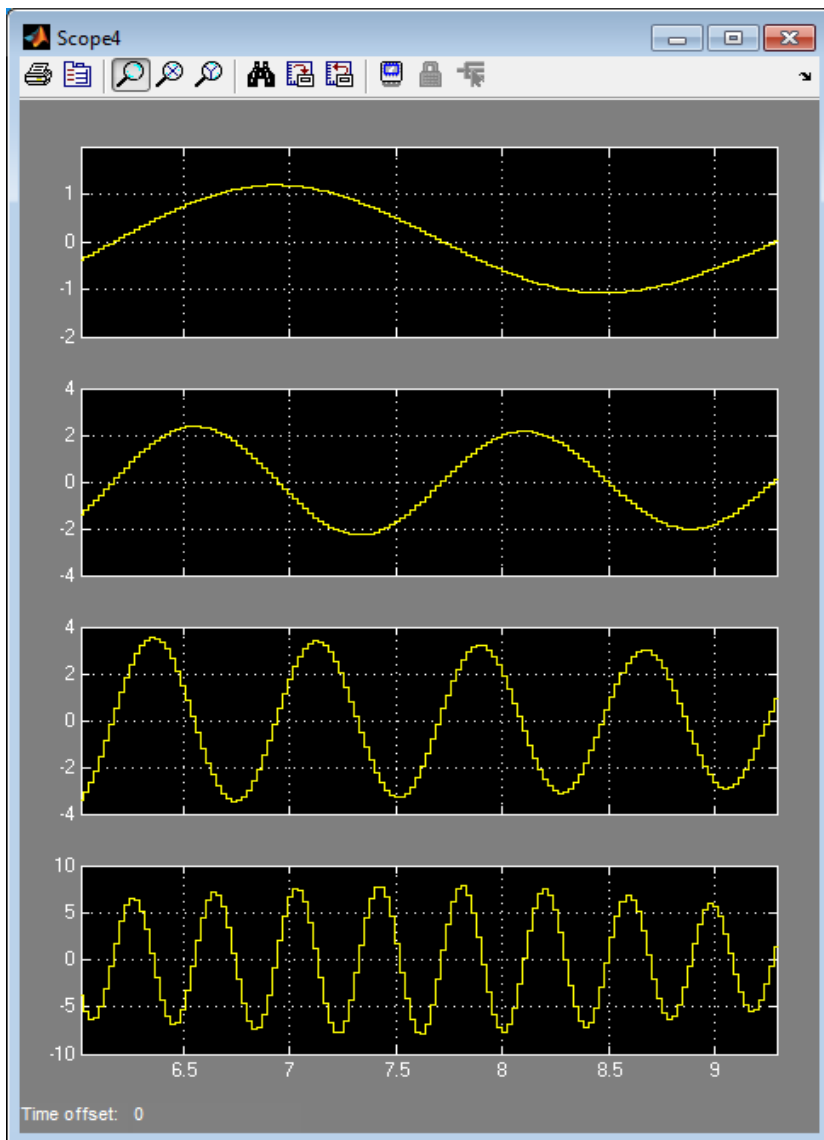


Рис. 4.12. Диаграммы работы тракта приема

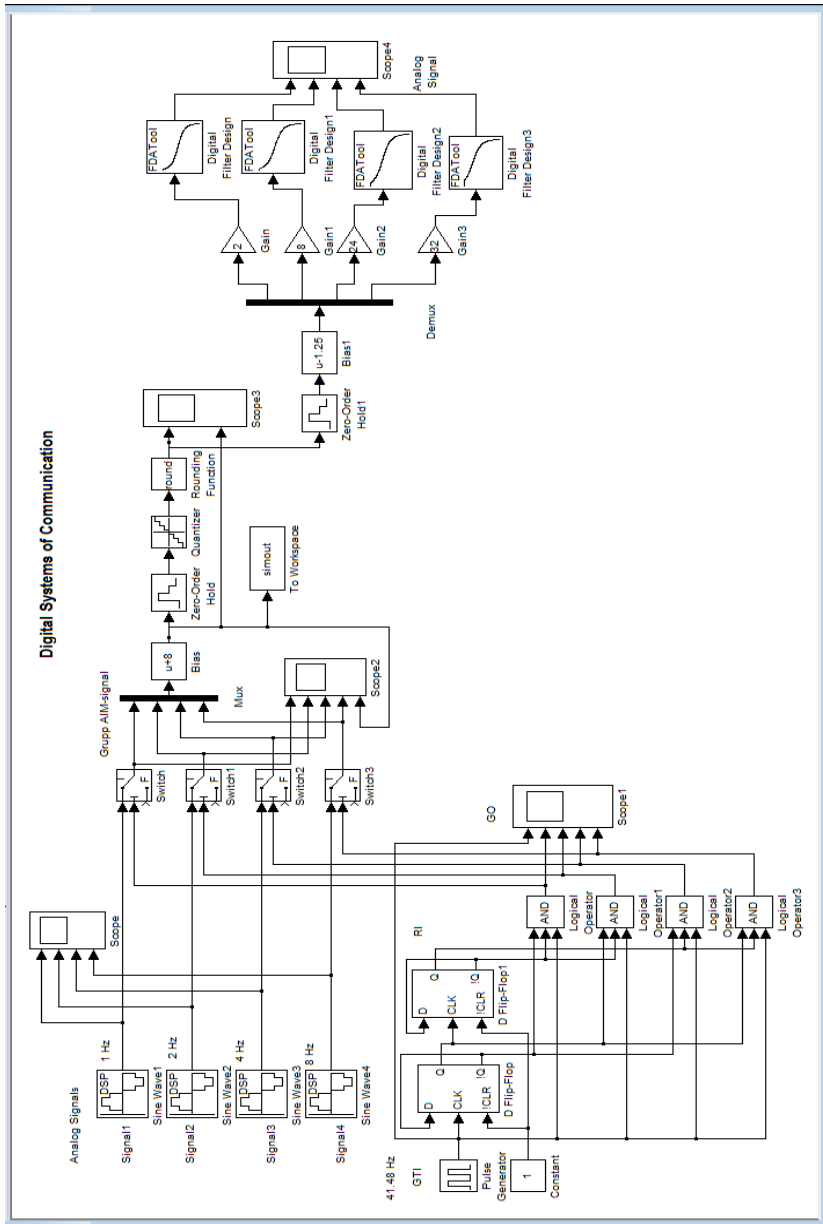


Рис. 4.13 Модель ЦСП

Поскольку смоделированная система передачи имеет однотипные элементы, то целесообразно создать соответствующие подсистемы. Например, в тракте передачи для каждого канала используются каналные ключи - дискретизаторы, их можно объединить в передающие подсистемы. В тракте приема для каждого канала используются усилители и фильтры и их можно объединить в подсистемы приема. Для упрощения представления генераторного оборудования тоже целесообразно создать подсистему. Модель четырех канальной цифровой системы передачи с образованными подсистемами и содержание подсистем приведены на рис. 4.15 и 4.14 соответственно.

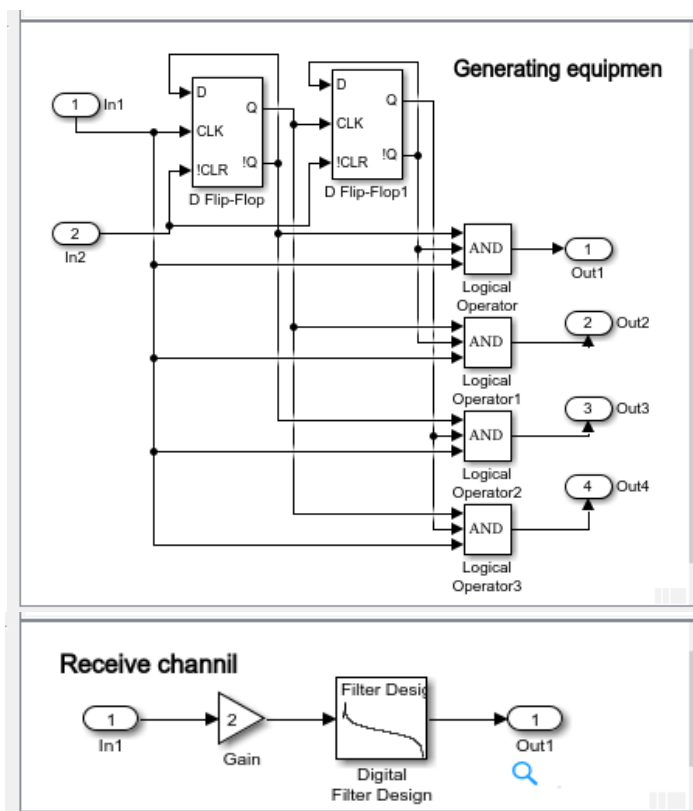


Рис. 4.14. Подсистемы генераторного оборудования и канала приема

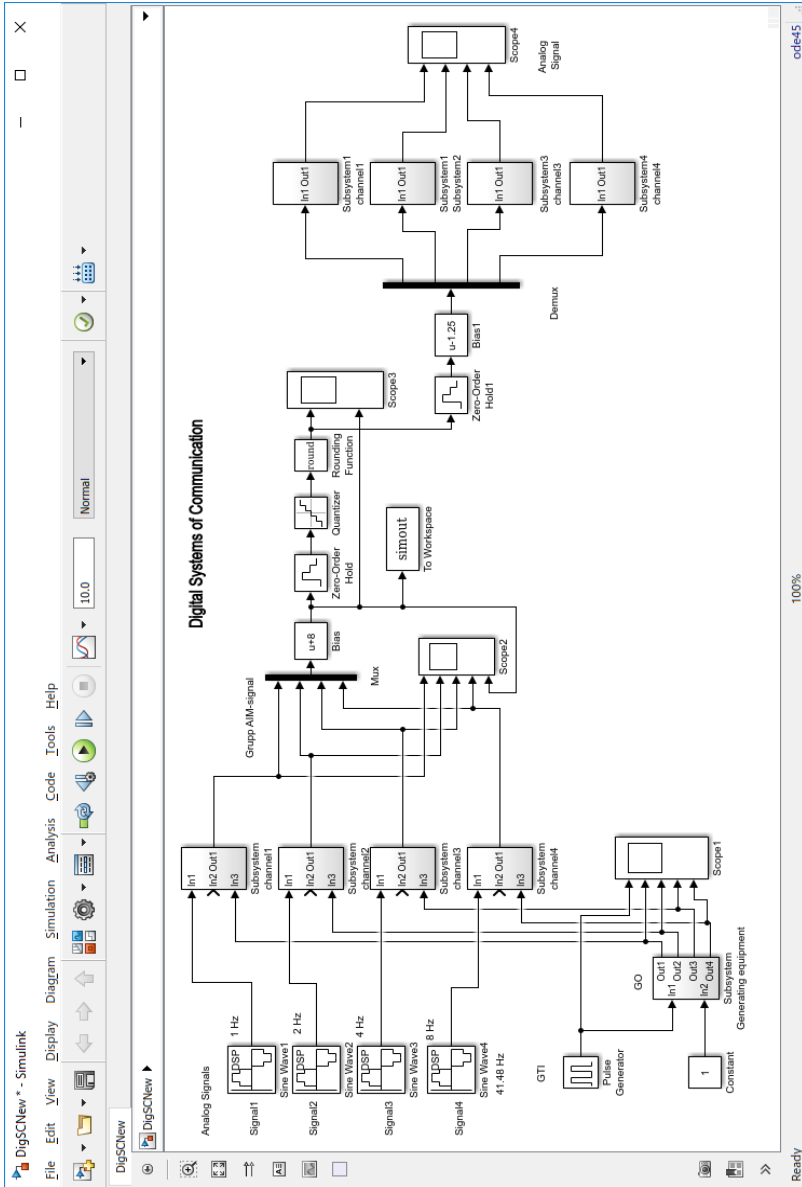


Рис. 4.15. Модель цифровой системы передачи, содержащей подсистемы каналные и генераторного оборудования

Пример контрольного теста «Системы передачи с ВРК»

Укажите все блоки, относящиеся к тракту передачи:

Sine Wave

Constant

Switch

Mux

Add

Укажите все блоки, относящиеся к тракту приёма:

Digital Filter Design

Add

Gain

Demux

Sine Wave

Укажите состав генераторного оборудования:

Генератор тактовой частоты

Делитель частоты

Распределитель импульсов

Усилитель

Выпрямитель

Укажите частоту дискретизации гармонического сигнала частотой 10 кГц для его последующего восстановления:

10 кГц

5 кГц

15 кГц

1 кГц

30 кГц

Укажите частоту дискретизации цифрового фильтра при формировании гармонического сигнала частотой 4 кГц:

1 кГц

2 кГц

10 кГц

4 кГц

8 кГц

Заключение

В настоящем учебном пособии изложены теоретические вопросы компьютерного моделирования сложных систем (§1), имитационное моделирование (§2), основные сведения о системе *MATLAB* и *Simulink* (§ 3), включая сведения о подсистемах *Simulink*, технологии моделирования в *Simulink* (§ 4), методика пошагового создания телекоммуникационных систем с помощью пакета *Simulink* и 4 лабораторные работы (§ 5). Эти разделы составляют часть курса дисциплины «Информатика», посвященные моделированию с использованием математической системы *MATLAB + Simulink* для направлений «Инфокоммуникационные технологии и системы связи» и «Радиотехника».

В практической части пособия содержатся практические рекомендации и приводятся характерные примеры выполнения заданий по каждой лабораторной работе (глава 5). Подробно рассматривается методика выполнения практических заданий, способы решения задач и визуализация работы моделируемых систем в разные моменты времени. Для решения заданий предлагается использование математической системы *MATLAB + Simulink*, широко используемой в решении различных прикладных задач и обладающих для этого соответствующими инструментами.

Первая работа посвящена моделированию арифметико-логических устройств (2-х часовая), вторая – моделированию цифрового линейного тракта, совместно с регенератором цифрового сигнала (2-х часовая), третья – моделированию трехканальной аналоговой системы передачи (4-х часовая), а четвертая – моделированию четырехканальной цифровой систем передачи (6 часовая).

Задания составлены таким образом, что студенты при моделировании используя стандартные блоки системы *MATLAB + Simulink*, могли создать свои собственные субблоки, позволяющие упростить модели систем передачи, наглядно представляя принцип их построения, имитировать однотипные элементы системы – одноразрядный сумматор, регенератор цифрового сигнала, индивидуальные тракты аналоговых и цифровых систем передачи, упрощая при этом моделирование самих систем передачи и цифрового линейного тракта.

В дидактической части пособия сосредоточены таблицы и рисунки, выполненные в виде screenshot диалоговых окон, открываемых в математической среде при настройке и моделировании сложных систем, моделей подсистем и самих систем.

Пособие включает в себя все необходимые части: основную (теоретическую), пояснительную (раскрытие терминов, понятий), методическую (задания, образцы моделей, подробное описание технологии создания моделей, тесты для контроля знаний, литературу), дидактическую (таблицы и рисунки). Это позволит студентам глубоко усвоить содержание разделов курса, аргументировано отвечать на поставленные вопросы, понимать сущность и значимость своей будущей профессии и проявить к ней профессиональный интерес.

Выполнение данного цикла лабораторных работ позволит студентам более углубленно освоить приемы работы по моделированию сложных устройств и систем телекоммуникаций в современных математических средах.

Процесс моделирования систем телекоммуникаций позволяет наиболее эффективно организовать процесс обучения. Студенты и магистранты на практических занятиях большую часть времени смогут посвятить изучению ключевых компонентов, составляющих систему связи и их влияние на ее характеристики, а также заниматься совершенствованием навыков программирования. Кроме того, моделирование дает студентам и магистрантам более наглядное представление об исследуемой системе, дает возможность как бы «потрогать» и понять, как она устроена, «прочувствовать» все ее составляющие элементы.

Лабораторные занятия позволяют студентам и магистрантам не только выполнять моделирование системы связи, но и проверить на сколько собранная ими модель соответствует действительности на уже реальных сигналах и при работе с реальным оборудованием.

Оглавление

<i>Введение</i>	<u>3</u>
<i>1. Компьютерное моделирование сложных систем</i>	<u>5</u>
<i>2. Имитационное моделирование</i>	<u>10</u>
<i>3. Основные сведения о системе MATLAB+ Simulink</i>	<u>13</u>
3.1. Применение подсистем	<u>17</u>
3.2. Создание подсистемы из части основной модели	<u>17</u>
3.3. Библиотека Ports & Subsystems	<u>20</u>
3.4. Особенности применения подсистем	<u>21</u>
3.5. Создание подсистемы на основе блока <i>Subsystem</i>	<u>22</u>
3.6. Управляемые подсистемы	<u>25</u>
3.7. Маскирование подсистем	<u>26</u>
3.8. Пакет расширения Signal Processing	<u>31</u>
<i>4. Технология моделирования в Simulink</i>	<u>33</u>
<i>5. Моделирование систем средствами Simulink</i>	<u>42</u>
1. Арифметико-логическое устройство	<u>42</u>
<i>Пример контрольного теста «Сумматоры»</i>	<u>54</u>
2. Моделирование цифрового линейного тракта	<u>55</u>
<i>Пример контрольного теста «Регенераторы»</i>	<u>65</u>
3. Модель аналоговой системы передачи	<u>66</u>
<i>Пример контрольного теста «Системы передачи с ЧПК»</i>	<u>87</u>
4. Модель цифровой системы передачи	<u>88</u>
<i>Пример контрольного теста «Системы передачи с ВПК»</i>	<u>107</u>
<i>Заключение</i>	<u>108</u>
<i>Рекомендуемая литература</i>	<u>1101</u>

Рекомендуемая литература

Основная

1. **Васильев К.К.** Математическое моделирование систем связи. [Текст]: учебное пособие. / К.К. Васильев, М.Н. Служивый. – Ульяновск: УЛГТУ, 2008, 179 с.
2. **Дьяконов, В. П.** MATLAB и SIMULINK для радиоинженеров [Текст] / В. П. Дьяконов – М.: ДМК, 2011, 976 с.
3. **Дьяконов, В. П.** MATLAB 6.5 SP1/7+Simulink 5/6 в математике и моделировании [Текст] / В. П. Дьяконов, – М.: СОЛОН-Пресс, 2005. – 576 с. – (Библ. профессионала)
4. **Дьяконов, В. П.** MATLAB 6.5 SP1/7+ Simulink 5/6. Основы применения [Текст] / В. П. Дьяконов, – М.: СОЛОН-Пресс, 2005. – 800 с. – (Библиотека профессионала).
5. **Павловский, Ю. Н.** Имитационное моделирование. [Текст]: учебник / Ю. Н., Павловский, Н. В. Белотелов, Ю. И. Бродский, М.: Академия, 2008. – 236 с.
6. **Солонина А. И.** Цифровая обработка сигналов. Моделирование в Simulink [Текст] / А. И. Солонина, – СПб.: БХВ-Петербург, 2012. – 430 с. – (Учебное пособие).
7. **Стефанова И. А.** Моделирование устройств телекоммуникаций в системе MATLAB+ Simulink. [Текст]: Учебное пособие /И.А. Стефанова. – Самара: ИУНП ПГУТИ, 2016 - 148 с.
8. **Стефанова И. А.** Программирование в системе MATLAB [Текст]: задания и методические указания к лаб. работам по информатике и программированию / И. А. Стефанова.: ПГУТИ, Кафедра ИВТ. – Самара: ИУНЛ ПГУТИ, 2014. – 66 с.
9. **Стефанова И. А.** Применение MATLAB+SIMULINK в технике связи [Текст]: задания и методические указания к лаб. работам по информатике / И. А Стефанова.; ПГУТИ, Кафедра ИВТ. – Самара: ИУНЛ ПГУТИ, 2015. – 72 с.

Дополнительная

10. **Беспалов, Е. С.** Simulink–модели радиотехнических устройств [Текст]: учеб. пособие / Е. С., Беспалов, М. И Мусьянков, А. П Пирхавка; Московский государственный

институт радиотехники, электроники и автоматики (технический университет), ГОУ ВПО МИРЭА. – М.: [Б. и.], 2010. - 56 с.

11. **Черных, И. В.** Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink. [Текст]/И. В. Черных – М.: ДМК, 2014. 296 с.
12. <https://studfiles.net/preview/2789341/> – электронный ресурс. Моделирование. StudFiles (Файловый архив студентов)