

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Ю.С. Захаревич, А.В. Захаревич

**ЛОГИЧЕСКОЕ УПРАВЛЕНИЕ И
ТЕХНОЛОГИЧЕСКИЕ ЗАЩИТЫ**

Методические указания
к лабораторным работам

Издательство
«АлКом»
2021

УДК 681.51.01(076.5)

ББК 32.965-01я73

З -38

Захаревич Ю.С.

З - 38 Логическое управление и технологические защиты: методические указания к лабораторным работам / Ю.С. Захаревич, А.В. Захаревич Томский политехнический университет. – Томск: Изд-во «АлКом», 2021. – 56 с.

В пособии представлены лабораторные работы по дисциплине «Логическое управление и защиты». Лабораторные работы выполняются с использованием ПО MatLab Simulink и содержат, как ознакомительные задания, предназначенные для освоения интерфейса и библиотек нового программного продукта, так и более сложные задачи синтеза систем технологических и аварийных защит, моделирования схем комбинационных цифровых устройств, логических законов управления, способных обеспечить высокие качественные показатели ведения технологического процесса с применением графических средств пакета Fuzzy Logic Toolbox.

В лабораторных работах кратко изложены теоретические сведения, разобраны примеры выполнения и представлены варианты для самостоятельной индивидуальной проработки изученного материала.

Предназначено для магистров, обучающихся по направлению 13.04.01 «Теплоэнергетика и теплотехника».

УДК 681.51.01(076.5)

ББК 32.965-01я73

Рецензенты

Доктор физико-математических наук, профессор ТПУ

П.А. Стрижак

Доктор технических наук, декан факультета инновационных технологий ТГУ

С.В. Шидловский

© ФГАОУ ВО НИ ТПУ, 2021

© Захаревич Ю.С., Захаревич А.В. 2021

© Оформление. Издательство «АлКом» 2021

Содержание

Лабораторная работа № 1	4
Знакомство с ПО MATLAB Simulink	
Лабораторная работа № 2	28
Синтез логических устройств в заданном базисе логических элементов	
Лабораторная работа № 3	33
Проектирование комбинационных схем конечных логических автоматов в Simulink	
Лабораторная работа № 4	42
Логическое управление на базе теории нечетких множеств с применением FIS редактора в Simulink	
Лабораторная работа № 5	51
Построение систем логического управления и защиты в Simulink	

Лабораторная работа № 1

Знакомство с ПО MATLAB Simulink

1 Цель работы

Познакомиться с возможностями ПО MATLAB Simulink, изучить его интерфейс и основные библиотеки. Получить первые практические навыки моделирования работы основных логических функций в Simulink.

2 Теоретические сведения

2.1 Общие сведения о ПО MATLAB Simulink

Simulink – среда динамического междисциплинарного моделирования сложных технических систем и основной инструмент для модельно-ориентированного проектирования. Его основным интерфейсом является графический инструмент для построения диаграмм и настраиваемый набор библиотек блоков. Simulink тесно интегрирован с остальной средой MATLAB и может либо использовать MATLAB, либо создавать сценарии из него.

Simulink широко используется в автоматическом управлении и цифровой обработке сигналов для многодоменного моделирования и проектирования на основе моделей. Он предназначен для моделирования и симуляций на системном уровне, что позволяет проводить всестороннее исследование разрабатываемой системы в единой среде проектирования.

Моделирование и симуляции позволяют провести проверку поведения системы в критических условиях или аварийных сценариях. Тем самым происходит снижение затрат на дорогостоящие физические прототипы. Проверка системы осуществляется с помощью полунатурного моделирования и быстрого прототипирования.

Ключевые возможности Simulink включают:

1) Графический редактор для моделирования всех компонентов системы.

2) Библиотека готовых блоков для моделирования алгоритмов и физических систем.

3) Блоки крупномасштабного моделирования для создания многоразовых системных компонентов и библиотек.

4) Механизм моделирования с решателями ODE для проверки совместной работы всех частей системы.

5) Инструменты визуализации для анализа и сравнения результатов нескольких симуляций.

2.2 Начало работы в Simulink

Для начала работ в Simulink необходимо запустить приложение MATLAB, установленное на вашем ПК, либо воспользоваться удаленным рабочим столом в домене ТПУ.

Для подключения к удаленному рабочему столу необходимо в строке поиска web-браузера ввести: **var.tpu.ru**. В появившемся окне ввести **домен tpu\имя пользователя** и **пароль**, как показано на рис. 1.

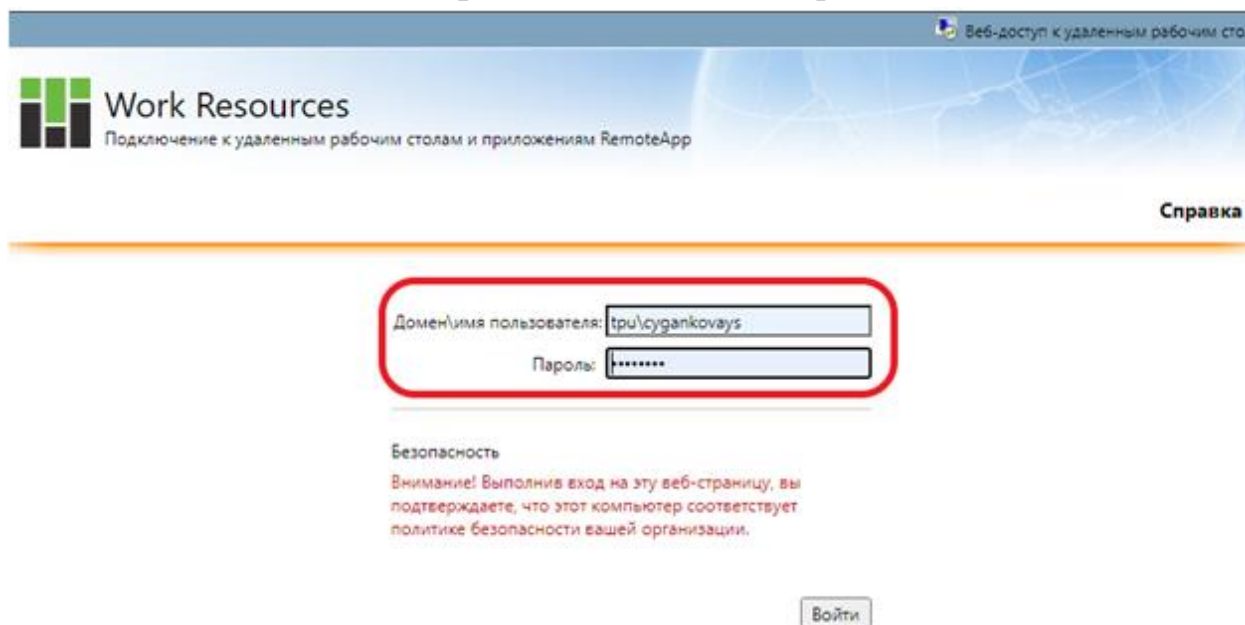


Рисунок 1 – Вход на удаленный рабочий стол ТПУ

Затем в папке «Математические» найти значок MATLAB R2020a (рис. 2) и скачать на свой ПК приложение для подключения к удаленному рабочему столу.

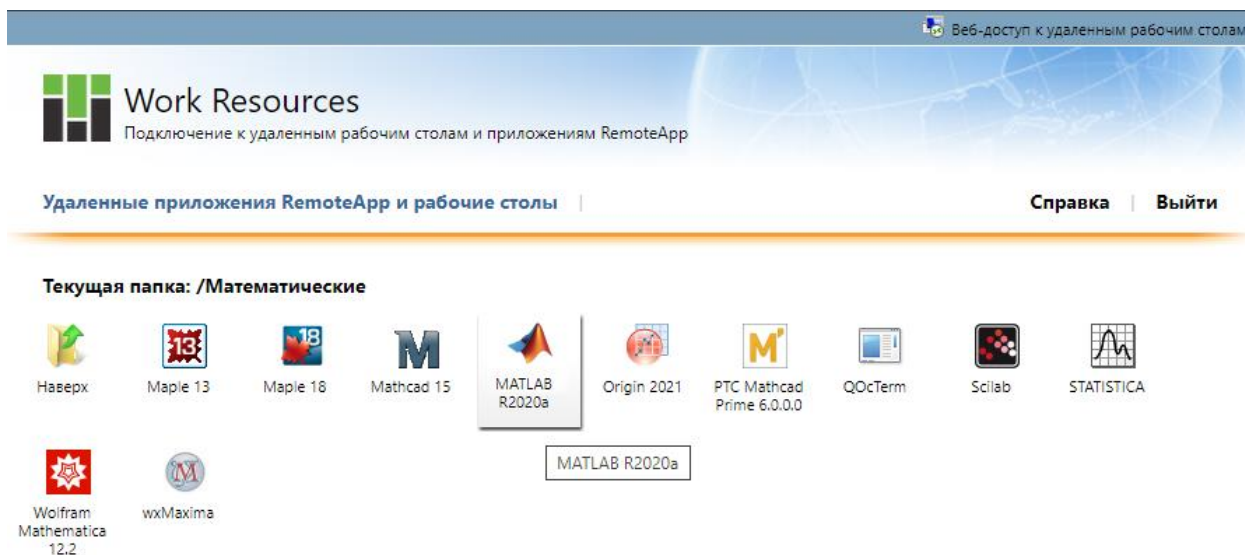


Рисунок 2 – Удаленный MATLAB R2020a

Найти на ПК в папке с загрузками файл удаленного приложения (рис. 3) и запустить его, введя по запросу имя пользователя и пароль, зарегистрированные в домене ТПУ. В дальнейшем можно открывать MATLAB, сразу запуская файл удаленного приложения, сохраненный на вашем ПК.

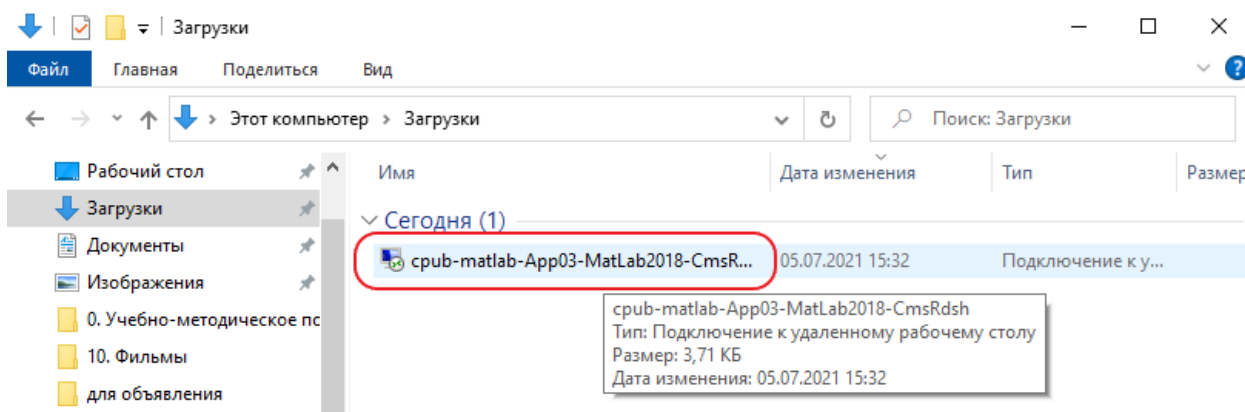


Рисунок 3 – Запуск удаленного приложения MATLAB

Перед вами откроется главное окно ПО MATLAB. Для перехода в среду графического моделирования Simulink необходимо в открывшемся окне MATLAB на панели инструментов нажать кнопку Simulink, как показано на рис. 4.

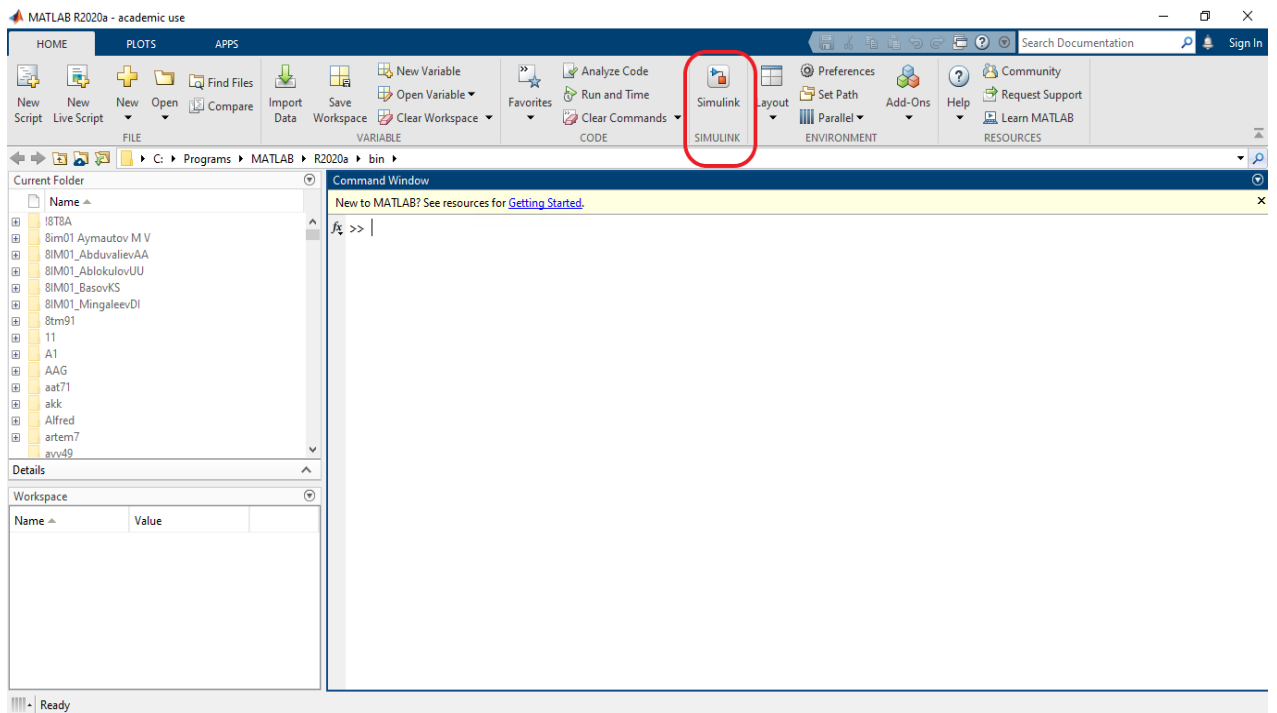


Рисунок 4 – Запуск среды графического моделирования Simulink

Далее выбираем Blank Model (рис. 5).

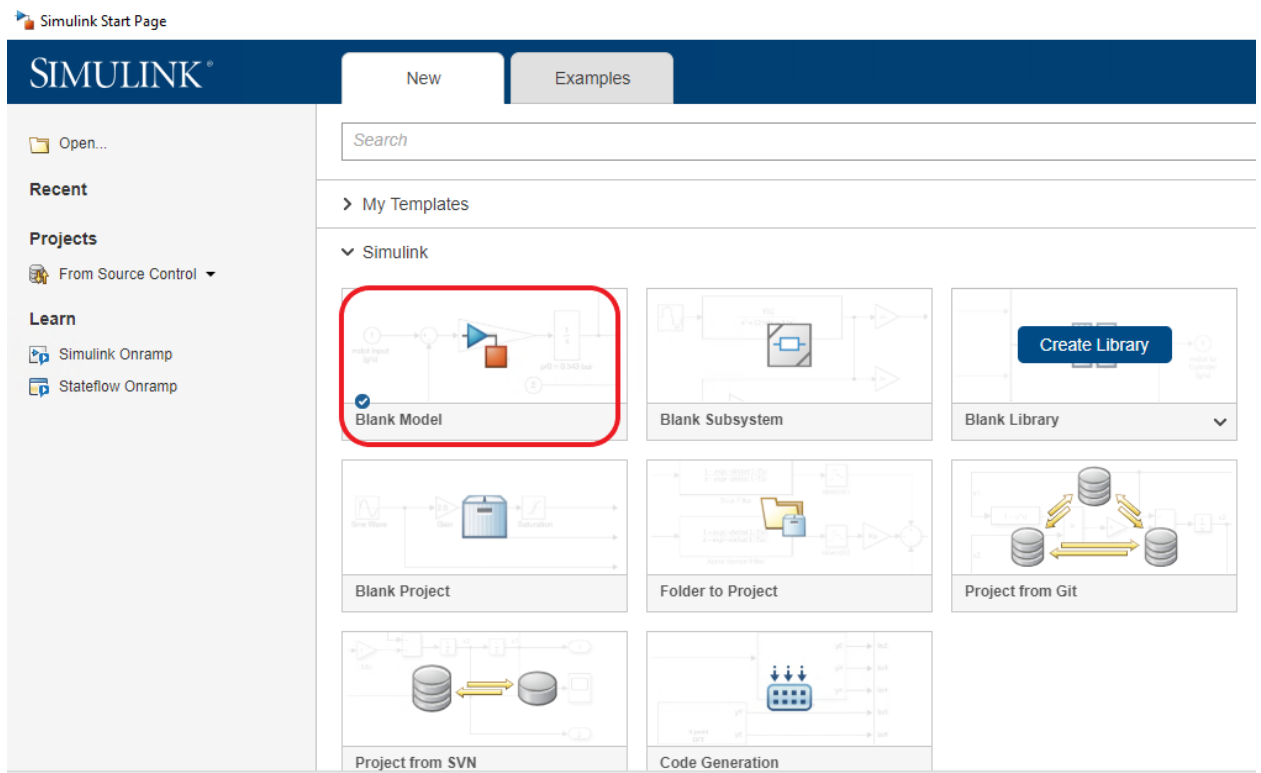


Рисунок 5 – Создание нового проекта в Simulink

Для создания и моделирования динамических систем в Simulink существует 2 основных компонента: браузер библиотеки и редактор модели. Главное окно редактора модели Simulink представлено на рис. 6. Из

редактора модели можно вызвать браузер библиотек, нажав на значок Library Browse на панели инструментов, как показано на рис. 6.

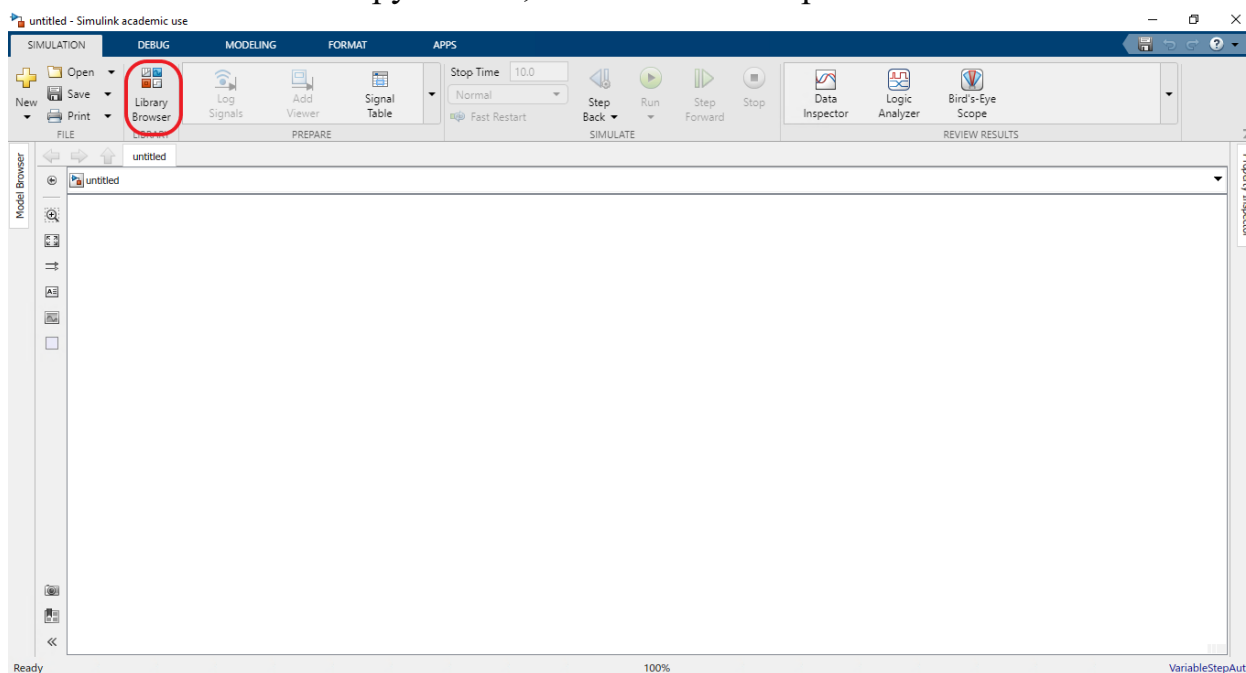


Рисунок 6 – Главное окно редактора модели в Simulink

В библиотеке браузера (рис. 7) уже есть готовые блоки, которые вы можете выбрать для создания модели динамических систем: дискретных, непрерывных и гибридных (могут иметь как непрерывные, так и дискретные состояния). Рекомендуется самостоятельно ознакомиться с библиотекой Simulink, воспользовавшись навигацией, расположенной в левой части окна. Это поможет в дальнейшем лучше ориентироваться в блоках библиотеки и быстрее находить нужные элементы.

Чтобы найти необходимый блок в браузере библиотеки, можно воспользоваться поиском по ключевым словам в строке поиска (рис. 7). Чтобы получить более подробную информацию о том, как этот блок работает, можно нажать на нем правой кнопкой мыши и из выпадающего меню выбрать «Help for the (наименование блока)» (рис. 8). Откроется документация на английском языке, описывающая принцип работы блока и примеры его использования.

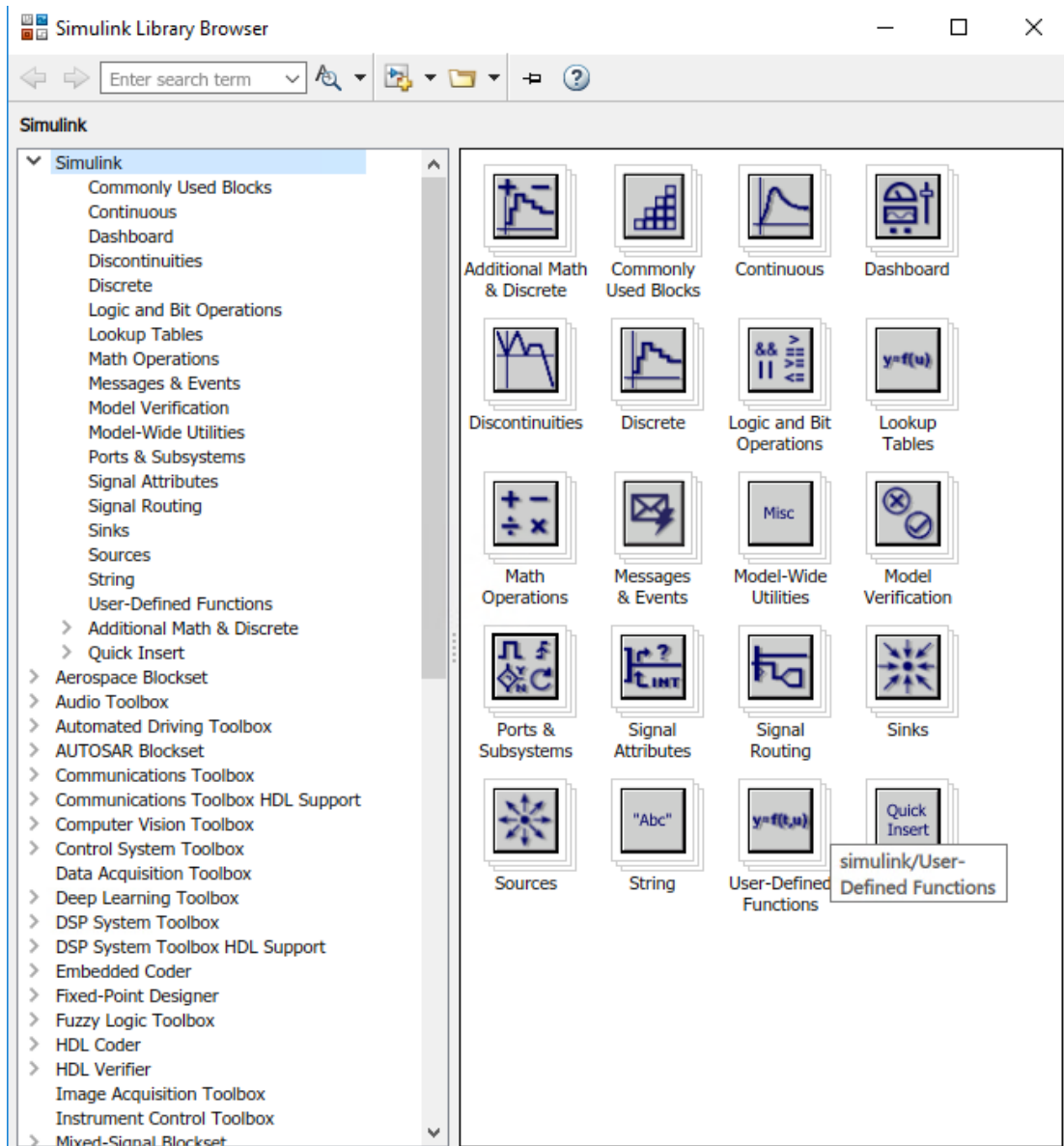


Рисунок 7 – Окно браузера библиотеки

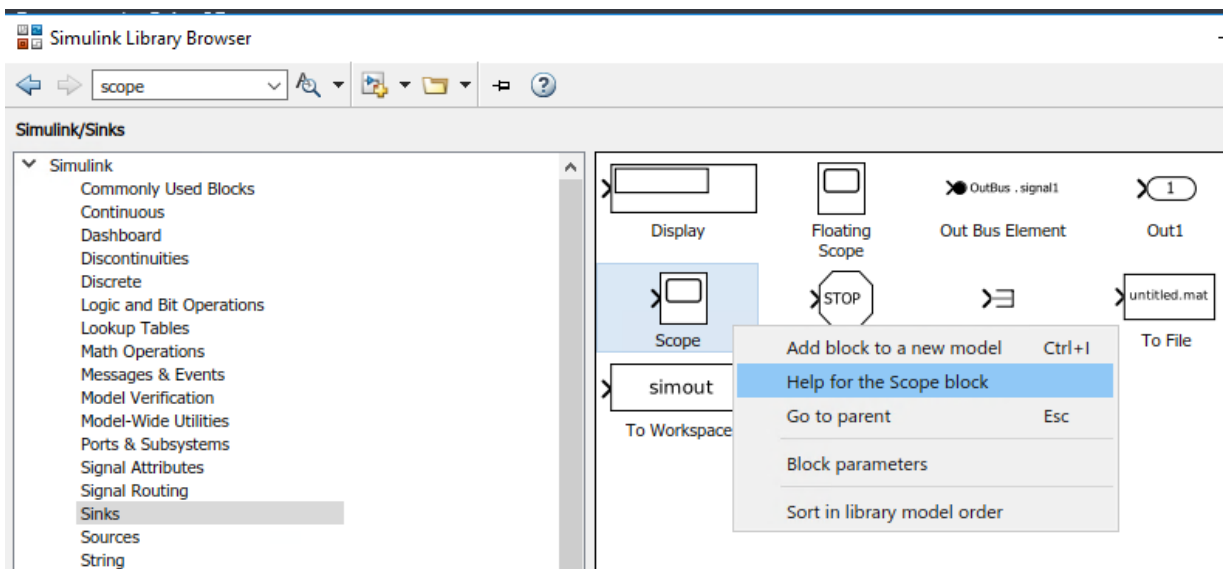


Рисунок 8 – Вызов справки о работе блока

Опишем функционал некоторых, наиболее часто применяемых блоков библиотеки.

Библиотека **Источники (Sources)** содержит набор источников сигнала, используемых для моделирования. Размещение указателя мыши на иконке Sources и двойное нажатие левой кнопки открывают библиотеку. Изменение свойств любого элемента осуществляется двойным нажатием левой кнопки мыши на изображении элемента.

Элемент **Постоянная (Constant)** позволяет задавать постоянный сигнал в системе.

Элемент **Дисплей (Display)** показывает значение сигнала в некоторой точке схемы моделирования.

Элемент **Дискретный генератор импульсов (Discrete Pulse Generator)** генерирует периодический импульсный сигнал.

Элемент **Осциллограф (Scope)** позволяет визуально наблюдать постоянные или изменяющиеся во времени сигналы в схеме моделирования.

Элемент **Часы (Clock)** является своего рода секундомером, производящим отсчет времени в системе и позволяющим синхронизировать все процессы в схеме моделирования.

Элемент **Буфер (To Workspace)** позволяет запоминать изменяющиеся значения переменной в рабочем пространстве в виде вектора с соответствующим названием.

Более углубленные знания о функционировании и настройке остальных блоков, требуемых для моделирования некоторых динамических систем вы будете получать в процессе выполнения лабораторных работ по данной дисциплине.

Знакомство со средой моделирования Simulink начнем с рассмотрения широко используемого для мониторинга и анализа результата работы любой системы средства визуализации – **осциллографа (блок Scope)**. Для этого построим простую модель, которая получает синусоиду и усиливает ее на заданный коэффициент. Результат усиления будем регистрировать на блоке осциллографа.

Создадим новую модель в Simulink, нажав на значок New -Blank model, либо комбинацией клавиш Ctrl + N (рис. 9).

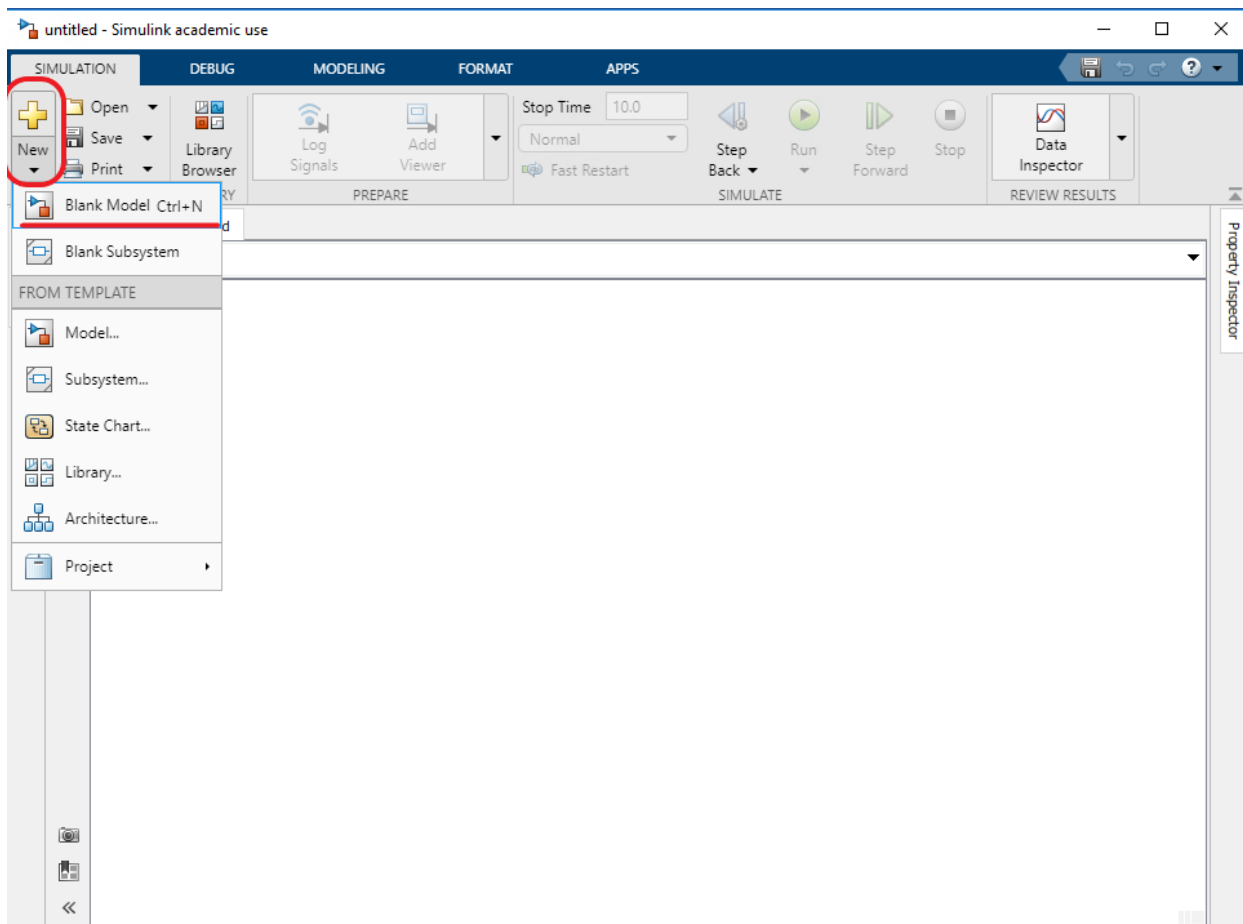


Рисунок 9 – Создание новой модели в Simulink

Сохраним новую модель под именем «PrimerScore» на своем ПК, воспользовавшись кнопкой Save на рабочем столе редактора модели Simulink.

Затем откроем браузер библиотеки и перенесем в редактор модели два блока: блок синусоиды (Sine Wave) и блок усиления (Gain). Блок Sine Wave находится в разделе Sources (источники). Блок Gain - в разделе Math Operations (математические операции). Для визуализации результатов моделирования добавим в модель блок осциллографа (Scope), который находится в разделе выходных сигналов (Sinks) (рис. 10). Напомним, что для поиска блоков в библиотеке также можно воспользоваться строкой поиска, находящейся вверху окна. Еще один вариант поиска - щелкнуть левой кнопкой мыши на свободном месте в редакторе модели и начать набирать на клавиатуре название необходимого блока. В выпадающем списке появятся подходящие варианты наименований блоков, из которых можно выбрать искомый блок.

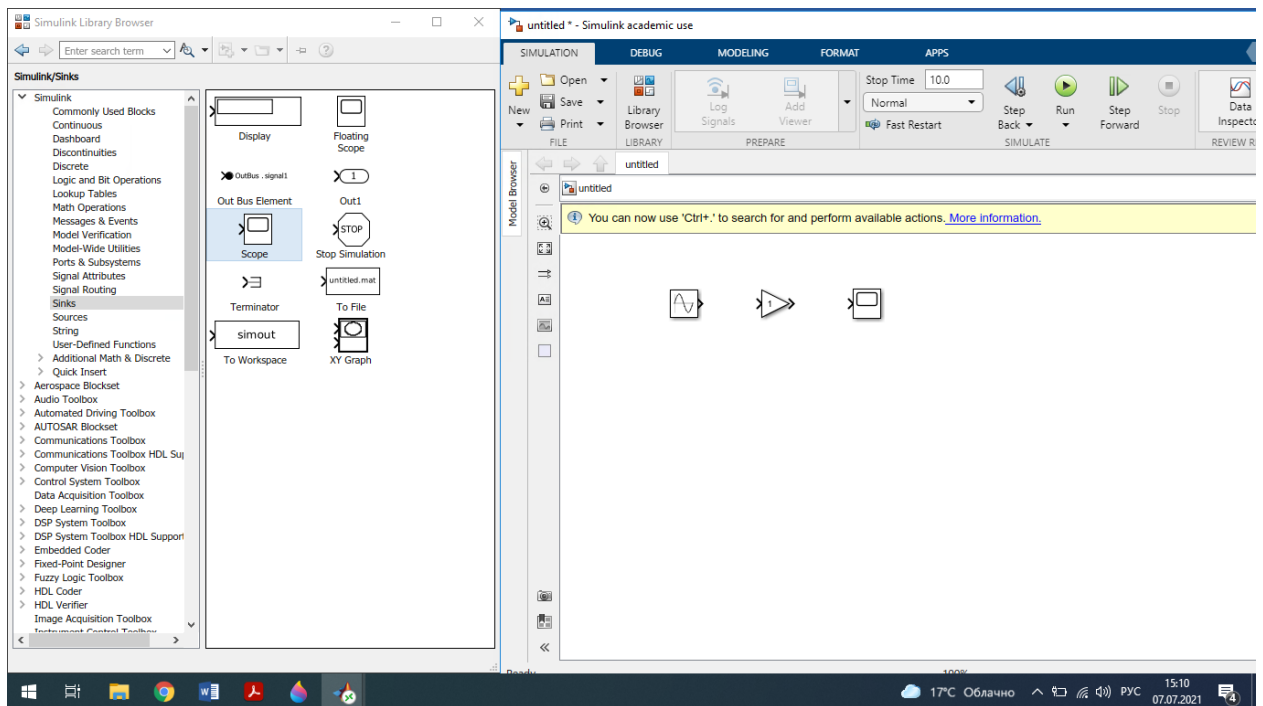


Рисунок 10 – Перенос блоков из библиотеки в редактор модели

При переносе и размещении блоков в редакторе Simulink вы можете увидеть, как среда помогает вам выровнять их относительно друг друга для улучшения внешнего вида модели.

Теперь для получения сигнала в виде синусоиды и его усиления необходимо соединить блоки. Создать сигнал можно двумя способами. Либо просто перетащив связующие линии между выходным портом одного блока и входным портом другого блока. Либо нажать на один блок, зажать кнопку Ctrl и нажать на другой блок, в результате получится простая модель, представленная на рис. 11.

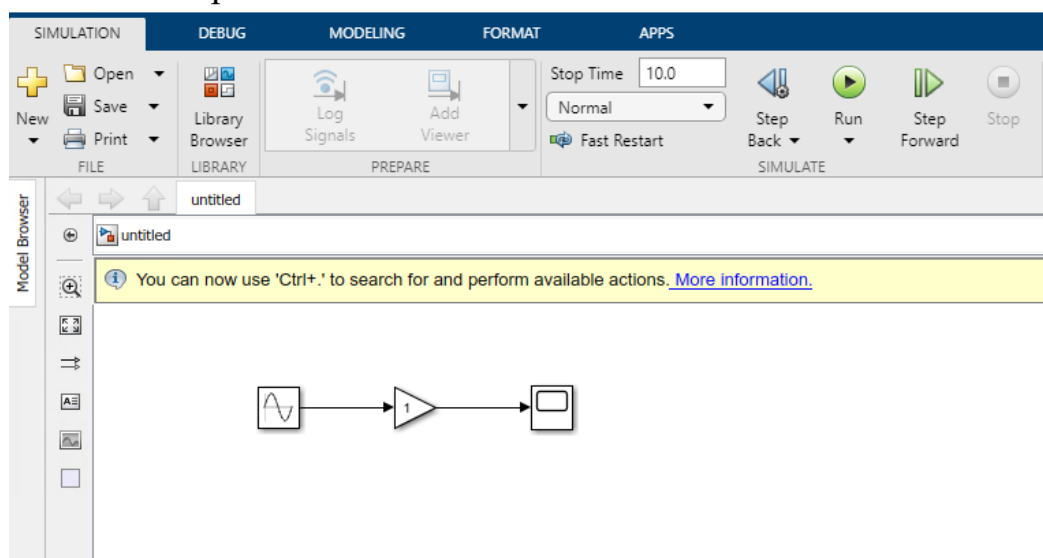


Рисунок 11 – Модель усиления синусоиды

Перейдем к запуску модели с настройками блоков по умолчанию и посмотрим результаты симуляции на сигнальном дисплее блока Scope. Для этого двойным щелчком левой кнопки мыши на блоке Scope откроем его окно. Затем для запуска модели нажмем на кнопку воспроизведения «Run» на рабочем столе Simulink. На сигнальном дисплее Scope появится синусоида (рис. 12).

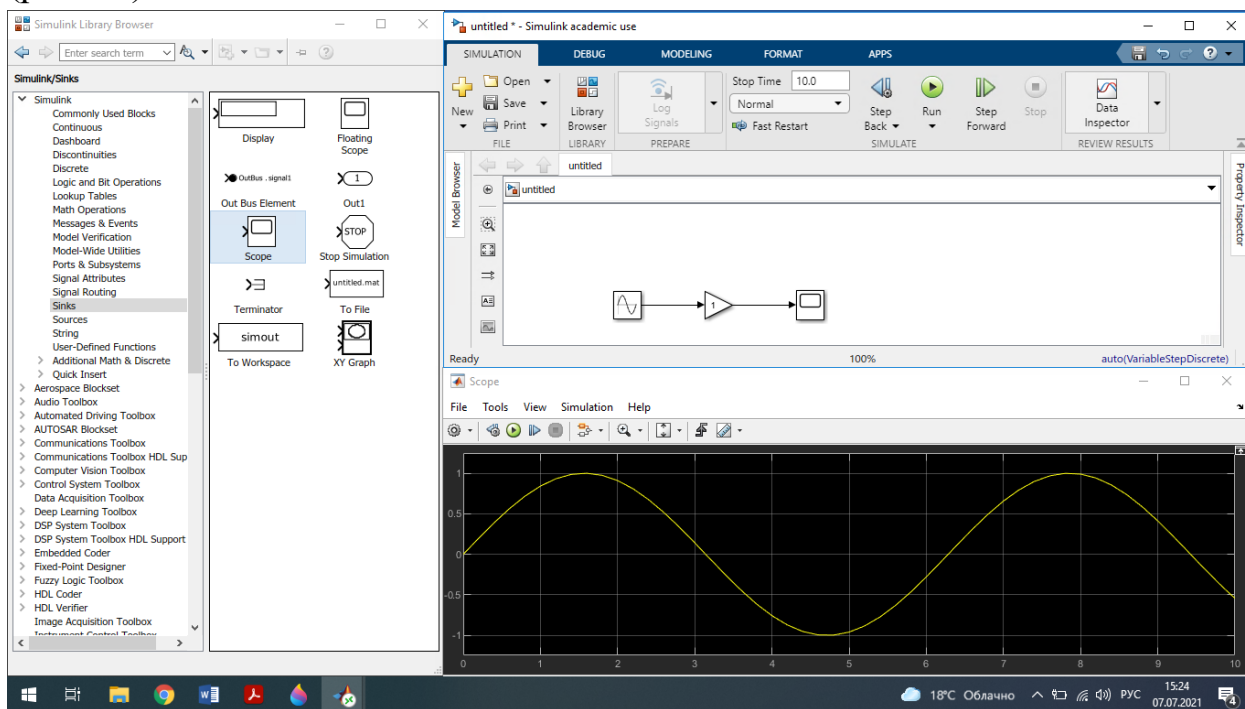


Рисунок 12 – Построение синусоиды в окне блока Scope

Теперь попробуем выбрать другой коэффициент усиления сигнала. Для этого двойным щелчком мыши на блоке Gain перейдем к его настройке. Изменим коэффициент усиления на «3». Нажмем Ок (рис. 13).

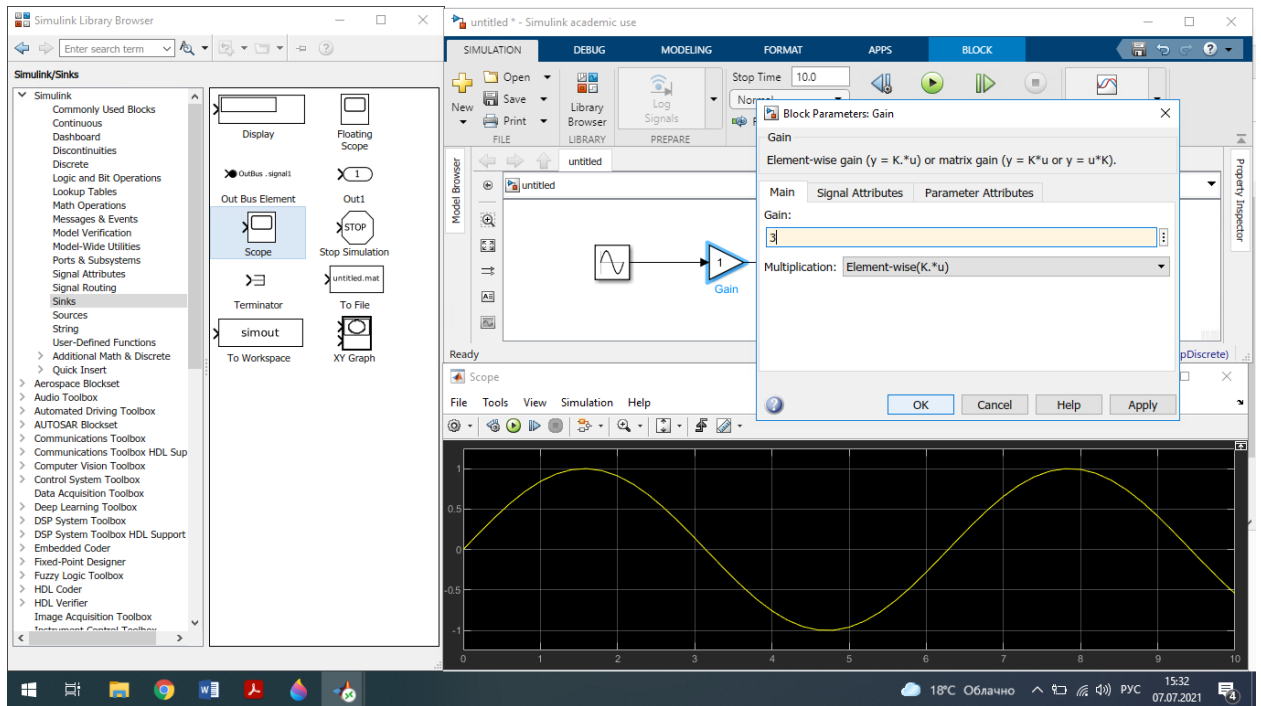


Рисунок 13 – Настройка блока Gain

Simulink также поддерживает векторные значения сигналов. Например, можно создать несколько синусоид, задавая векторные значения для одного из параметров в блоке синусоид *Sine Wave*. Для этого двойным щелчком мыши на блоке *Sine Wave* в открывшемся окне настроек введем векторные значения, как показано на рис. 14.

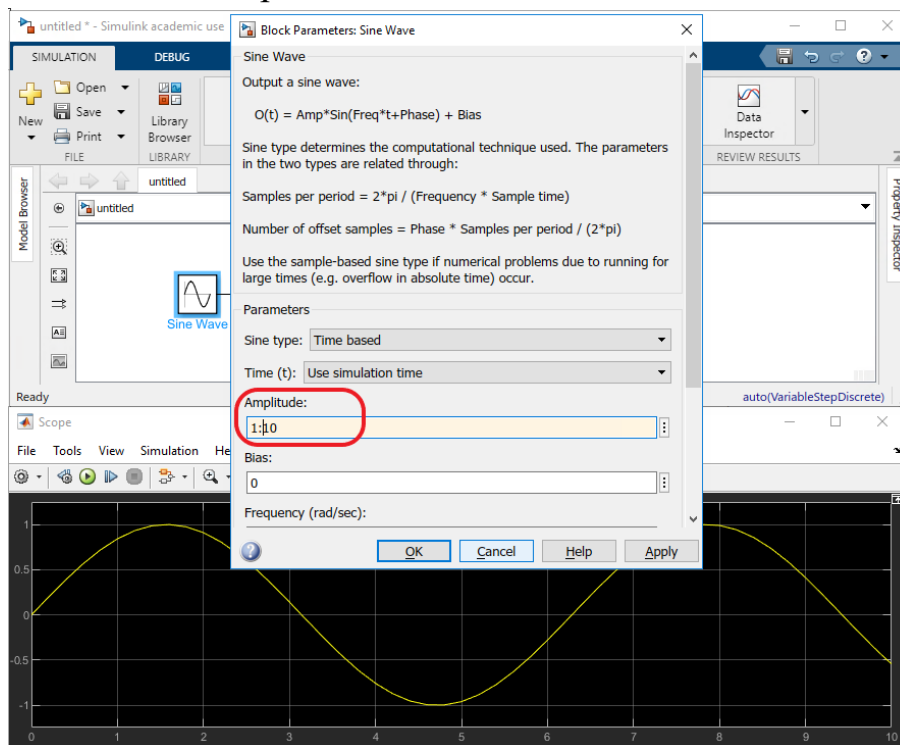


Рисунок 14 – Ввод векторных значений в блоке Sine Wave

Посмотрим на результат моделирования с новыми параметрами, запустив симуляцию нажатием кнопки Run. Если график в окне осциллографа выходит за границы видимой области, необходимо вписать его в лимиты осей, нажав на кнопку автомасштаба, как показано на рис. 15.

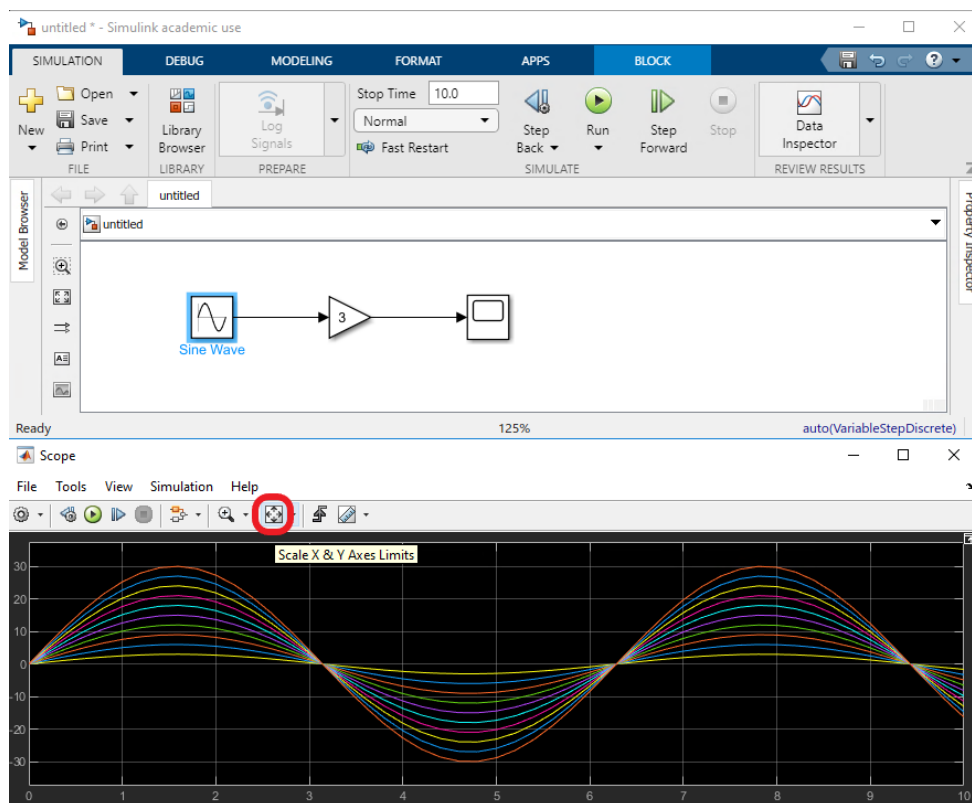


Рисунок 15 – Результат симуляции усиления синусоид

Добавим в модель блок экстраполяции нулевого порядка для получения дискретного сигнала на выходе. В разделе браузера библиотеки Discrete, либо поиском находим блок Zero- Ordre Hold и перетаскиваем его на сигнальную линию, соединяющую синусоиду с блоком усиления, как показано на рис. 16. Устанавливаем интервал дискретизации для блока дискретизации, например, вводим значение «0.2». Нажимаем Ок.

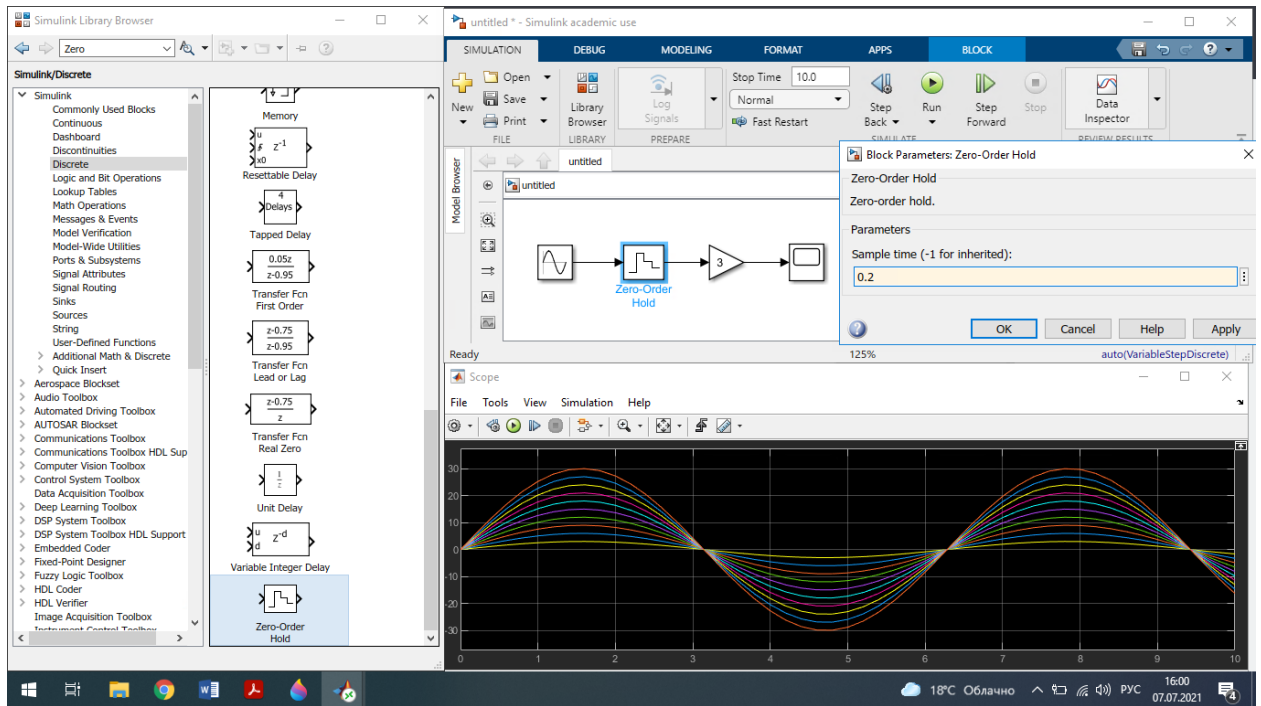


Рисунок 16 – Настройка экстраполятора нулевого порядка (блок Zero- Ordre Hold)

Запускаем симуляцию и наблюдаем результат моделирования на сигнальном дисплее осциллографа Scope (рис. 17). Заметим, что данные на сигнальном дисплее обновляются только после запуска симуляции модели кнопкой Run.

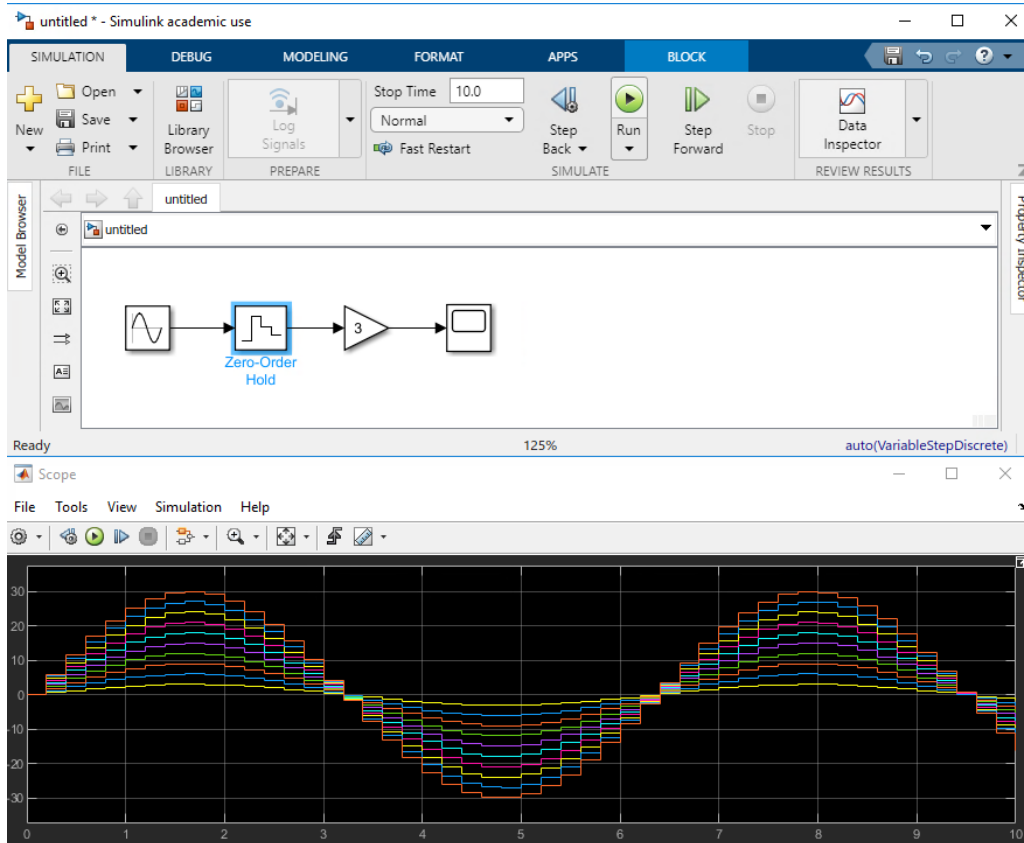


Рисунок 17 – Результат дискретизации синусоидального сигнала

Теперь потренируемся в настройках визуализации осциллографа. Для удобства работы уменьшим количество синусоид, ограничившись тремя графиками. Изменим векторное значение в настройках блока синусоид (рис. 14) на «1:3».

Добавим в модель еще один источник сигнала – генератор дискретных импульсов Pulse Generator. В его настройках изменим значение амплитуды на «10» и соединим с осциллографом. Запустим симуляцию (рис. 18).

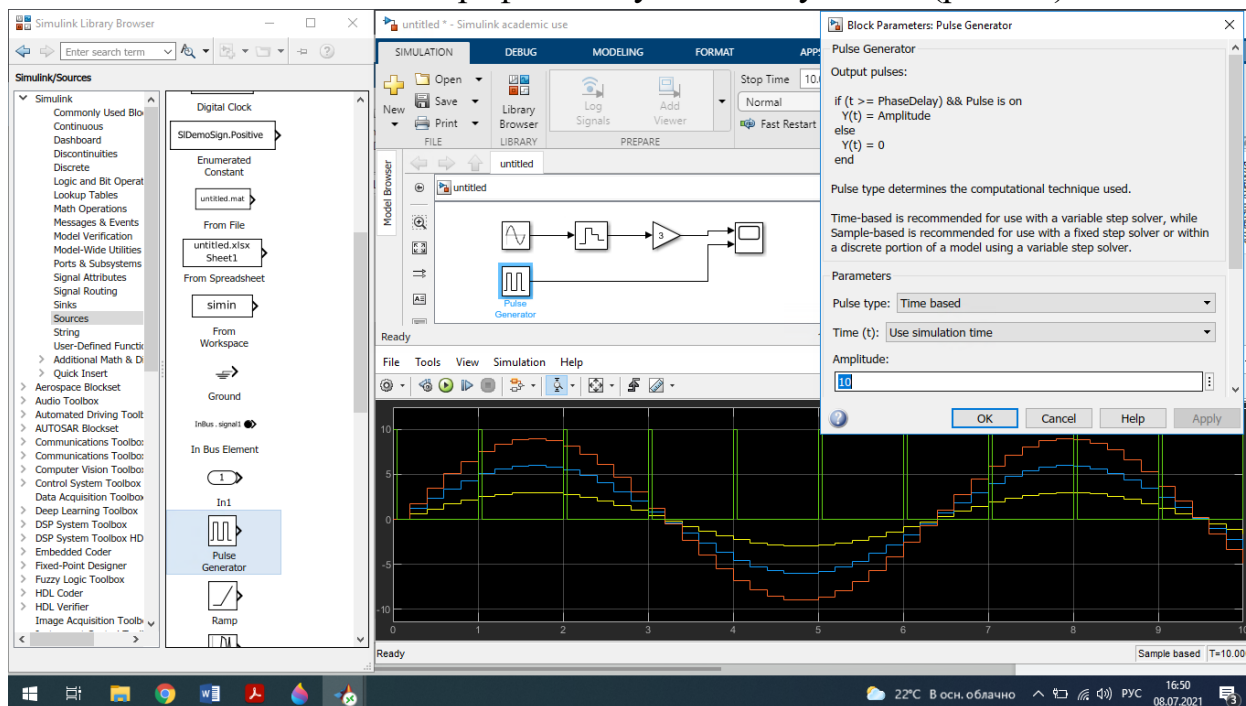
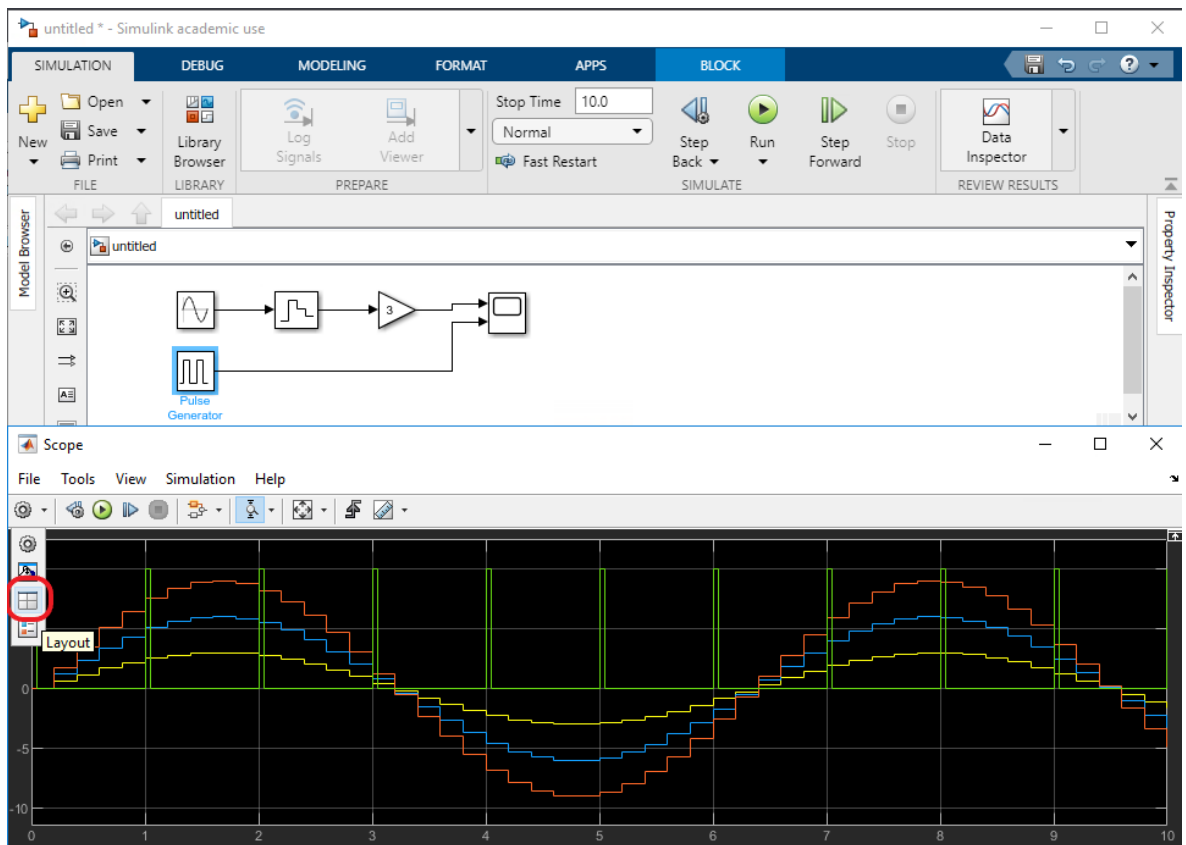


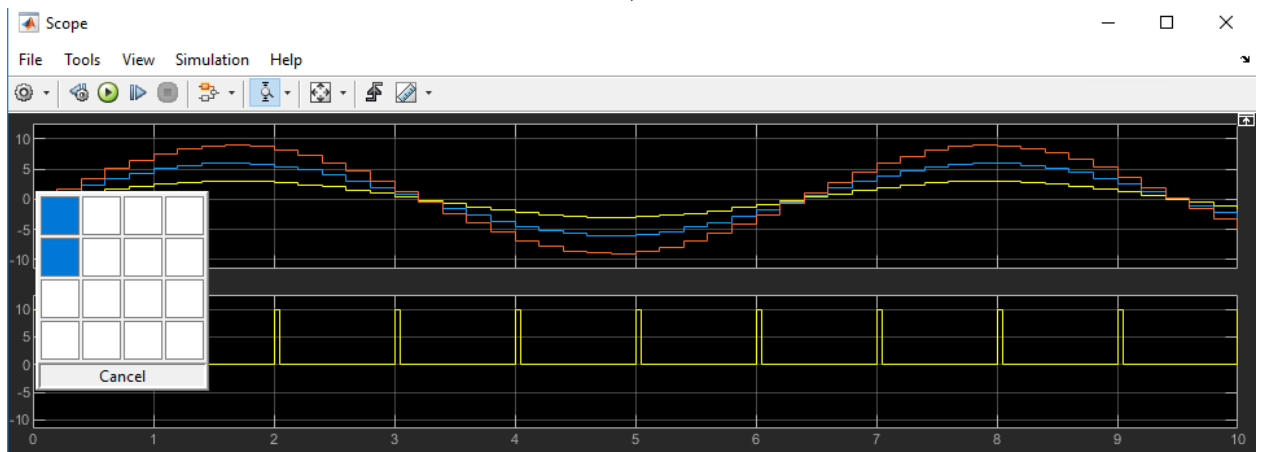
Рисунок 18 – Добавление нового источника сигнала

Теперь разнесем полученные графики, добавив дополнительные оси на сигнальном дисплее. Откроем выпадающий список настроек на панели инструментов сигнального дисплея и выберем кнопку Layout (рис.19а).

Заметим, что новые оси можно добавлять сбоку, снизу, как удобно для работы и последующего анализа. Мы выберем горизонтальную компоновку двух осей и получим два графика, расположенных друг под другом (рис. 19б).



a)



б)

Рисунок 19 – Добавление осей на сигнальном дисплее

Выполним настройку визуального представления графиков, как показано на рис. 20. Для этого в выпадающем списке настроек сигнального дисплея нажмем кнопку Style. В появившемся окне настроек поменяем цвет фона с черного на белый, сетку сделаем черной, изменим цвета линий графиков на более контрастные (переключение между линиями осуществляется в строке Properties for line). Настройки необходимо сделать для обоих дисплеев. Переключиться между дисплеями можно в строке Active display.

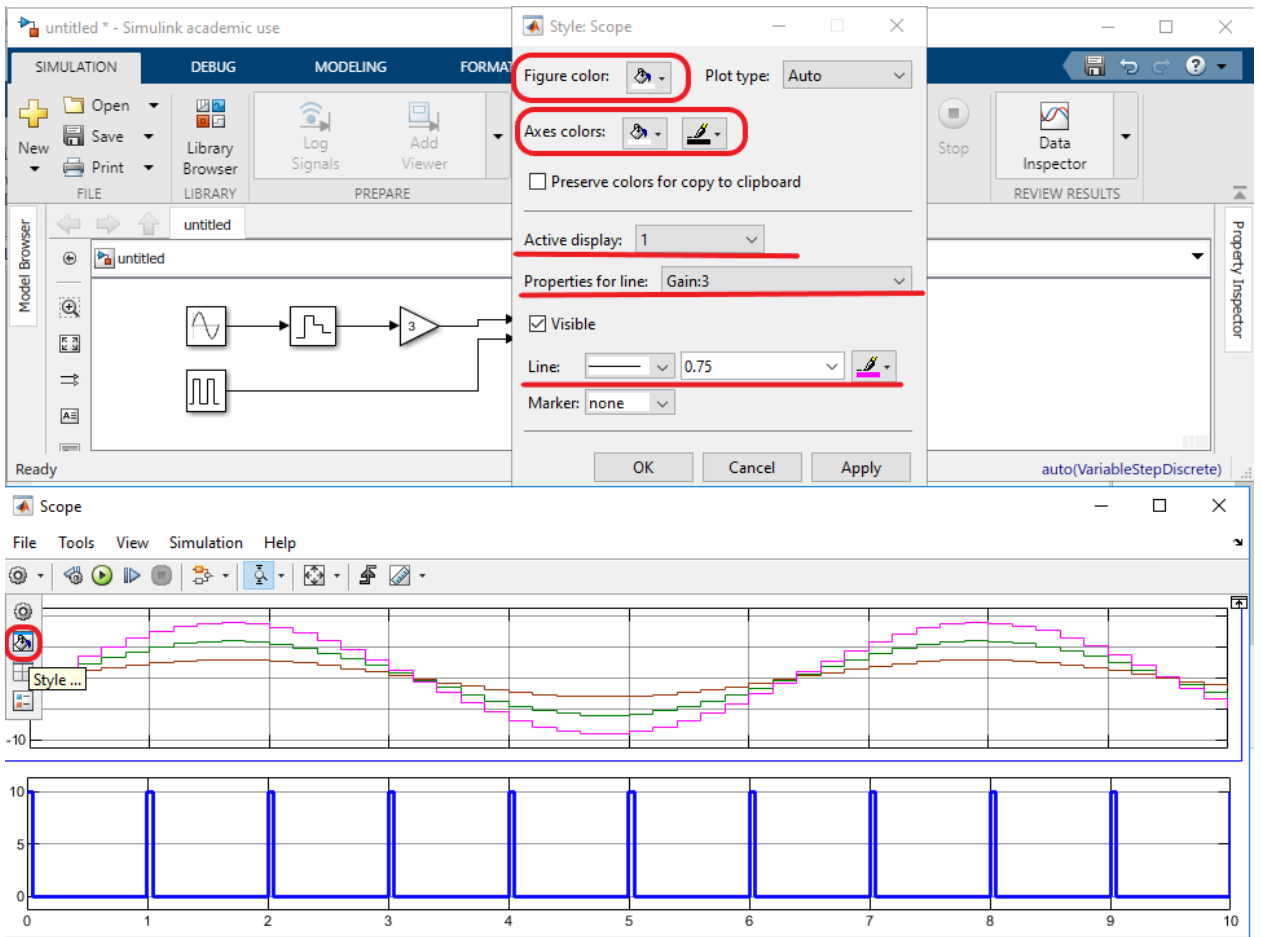


Рисунок 20 – Настройка визуализации

Затем добавим на графики легенды (рис. 21). Назовем наши зависимости Синус1, Синус2, Синус3 и Импульс.

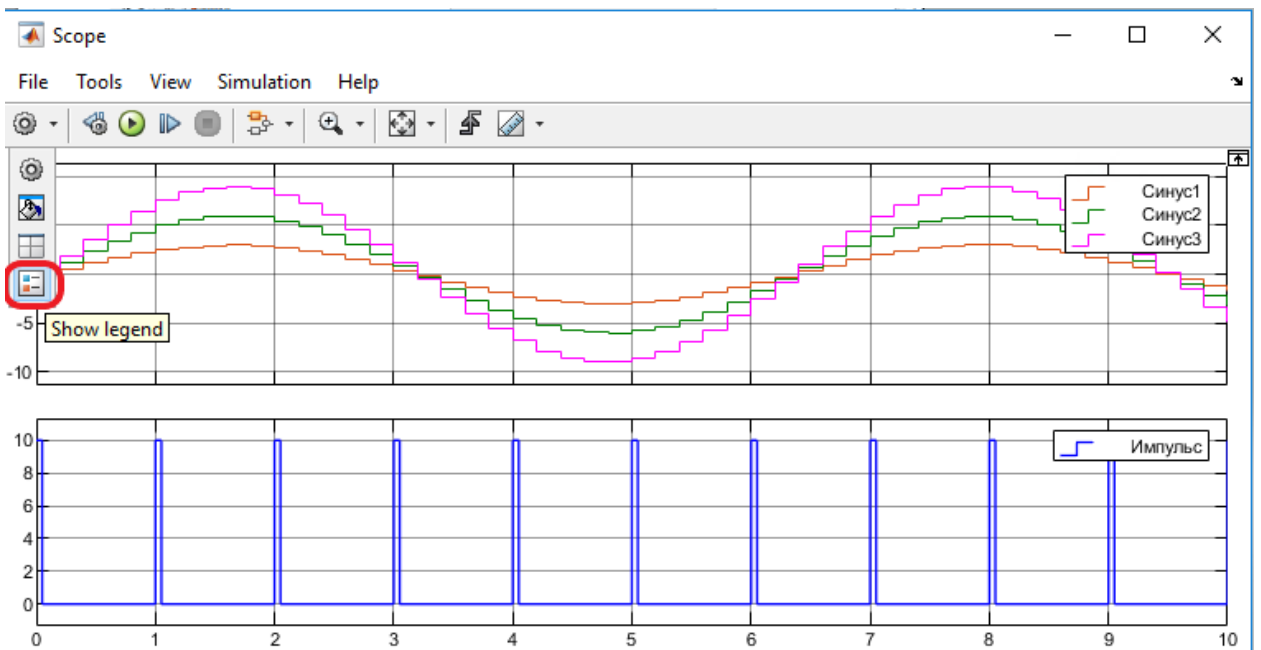
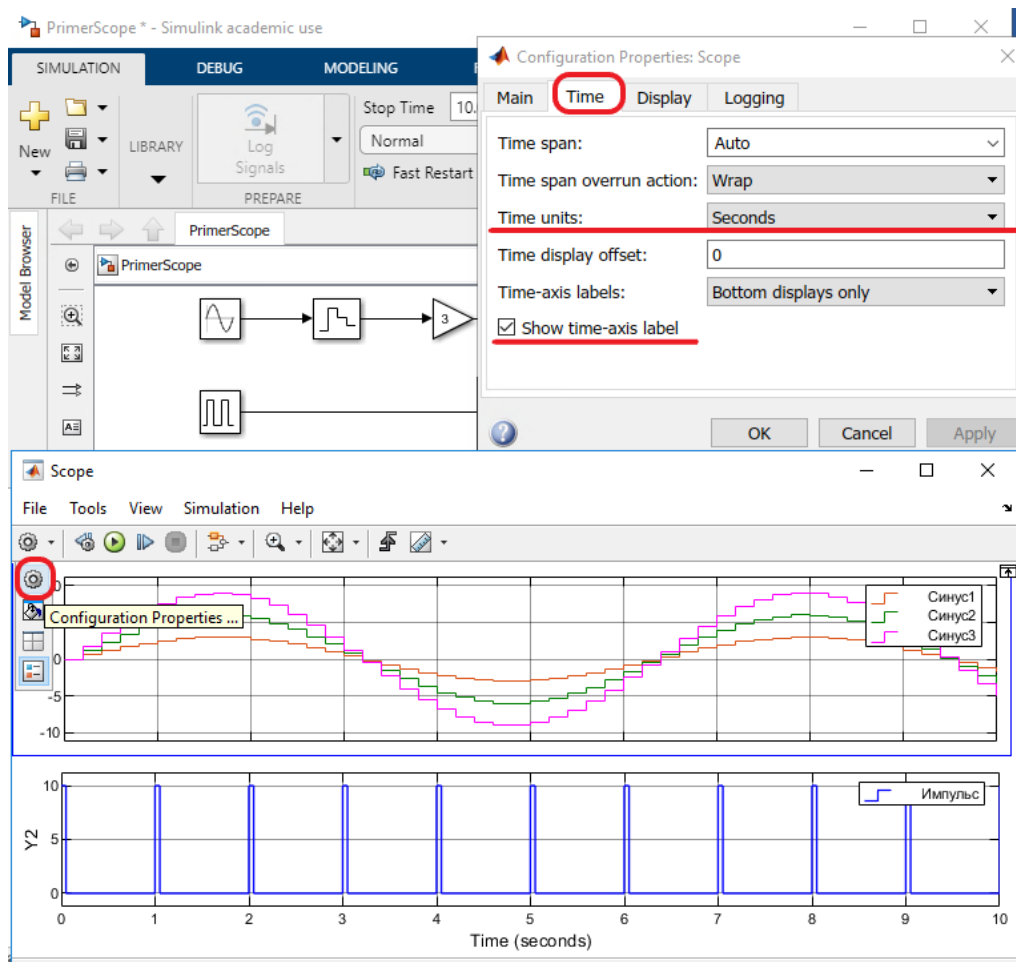
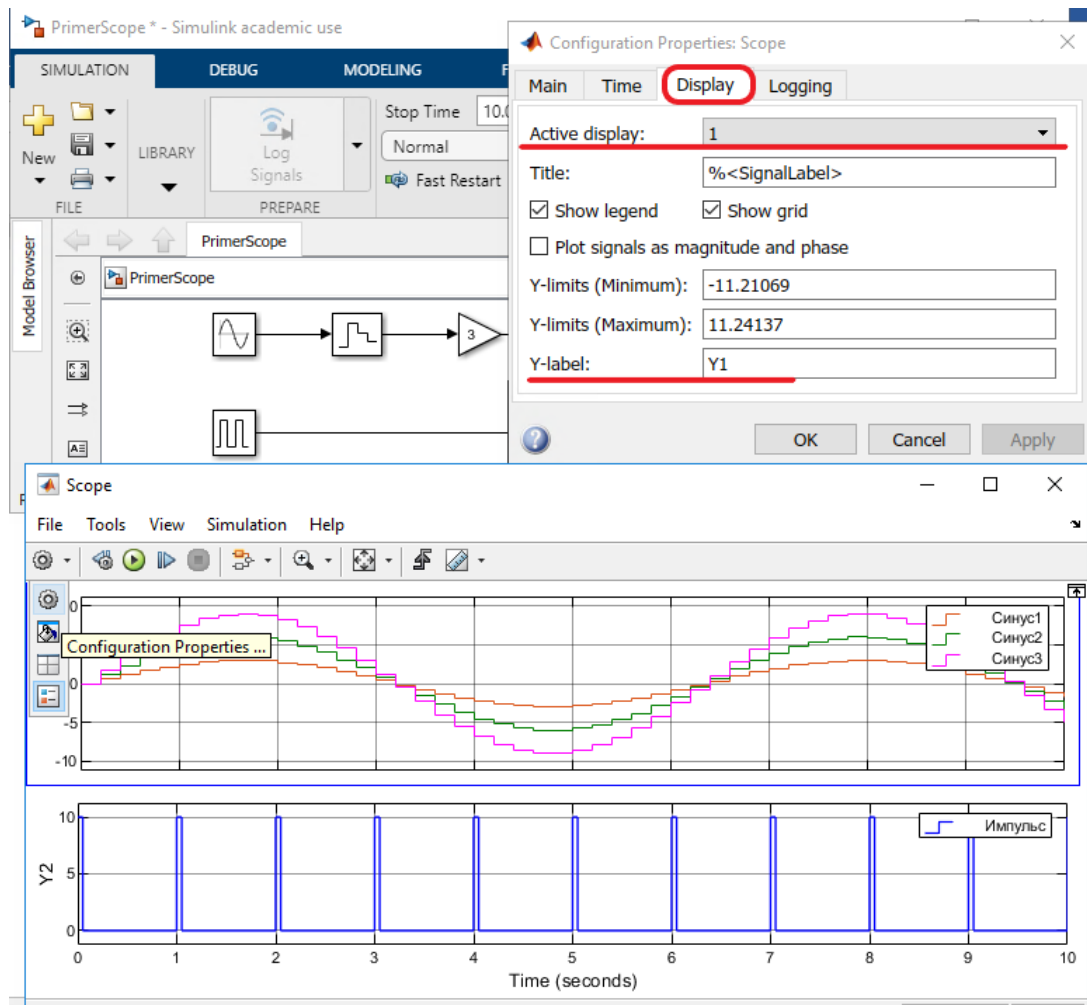


Рисунок 21 – Добавление легенд на графике

Добавим подписи осей. На вкладке Time в окне Configuration Properties поставим галочку «Show time-axis label» и в строке «Time units» выберем Second (рис. 22а). Чтобы подписать ось Y, перейдем на вкладку Display и введем обозначение осей для обоих сигнальных дисплеев, как показано на рис. 22б.



а)



б)

Рисунок 22 – Добавление подписей осей

Отметим еще несколько кнопок, которые могут пригодиться в дальнейшем при работе с блоком Scope. Кнопка «Лупа», позволяет приближать линию зависимости для более подробного изучения. В выпадающем списке лупы доступны разные варианты приближения (по оси Y, по оси X, и одновременно по X и Y). И кнопка «Измерения», позволяет проводить измерения прямо на сигнальном дисплее в Simulink (рис. 23).

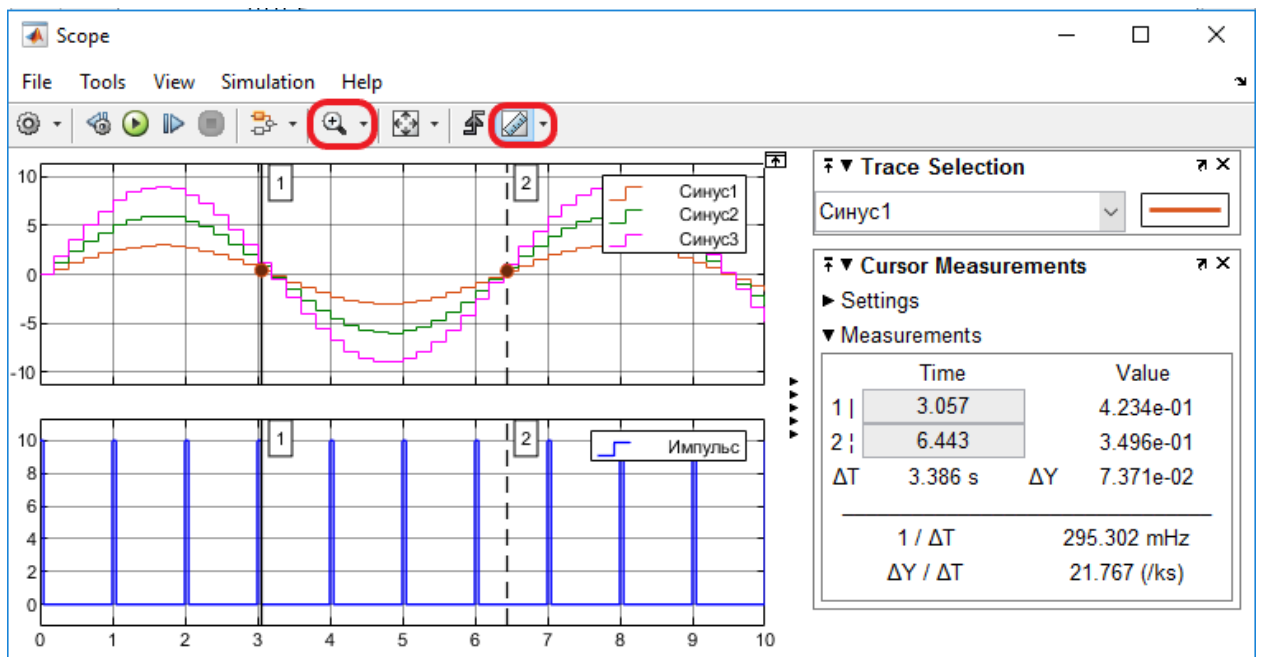


Рисунок 23 – Кнопки «Лупа» и «Измерения»

2.3 Моделирование основных логических функций в Simulink

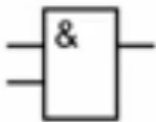

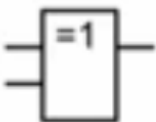



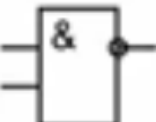
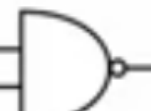
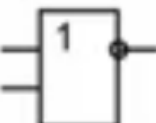

Прежде чем перейти к моделированию основных логических функций в Simulink, вспомним их условные обозначения на схемах и соответствующие им таблицы истинности, описывающие работу каждой логической функции (табл.1).

Примечание: Схемы, реализующие основные логические функции, называются *вентильми*. В англоязычной литературе они называются *GATE* (ворота). Например, схема (или вентиль) ИЛИ-НЕ в Simulink будет называться NOR GATE (см. табл. 1).

Таблица 1 – Обозначение основных логических функций в Simulink

Логическая функция	Изображение на схемах		Таблица истинности															
	МЭК (в России и Европе)	ANSI (в Америке)																
НЕ (NOT GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>\bar{A}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	\bar{A}	0	1	1	0									
A	\bar{A}																	
0	1																	
1	0																	
ИЛИ (OR GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

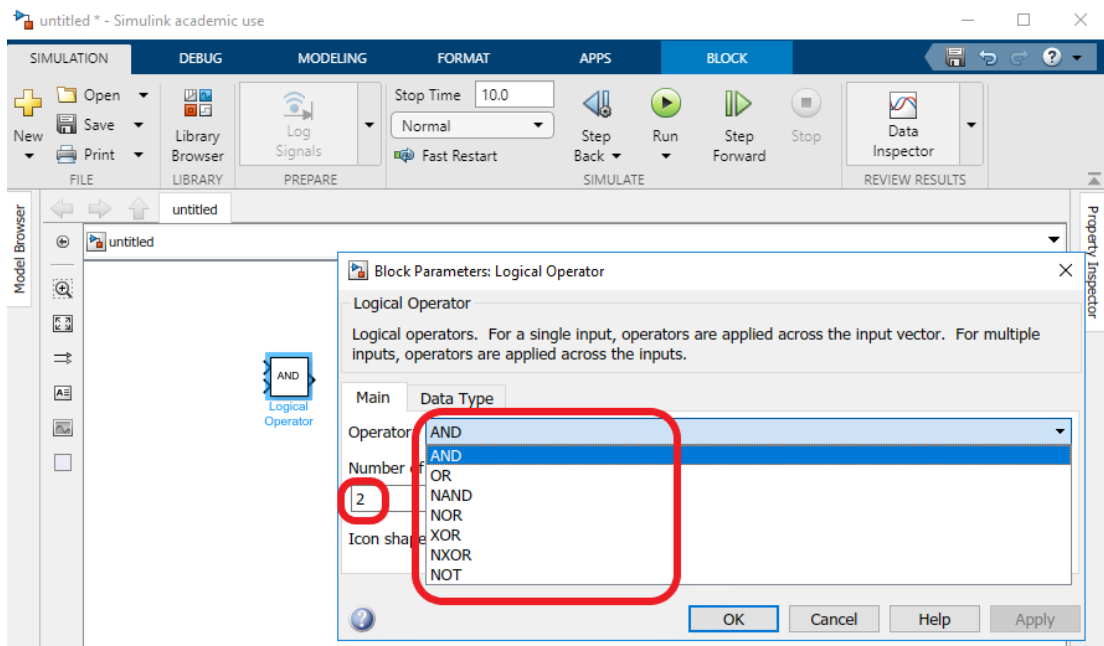
Продолжение таблицы 1

И (AND GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
Исключающее ИЛИ (XOR GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Исключающее И (NXOR GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
Штрих Шеффера (И-НЕ) (NAND GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
Стрелка Пирса (ИЛИ-НЕ) (NOR GATE)			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

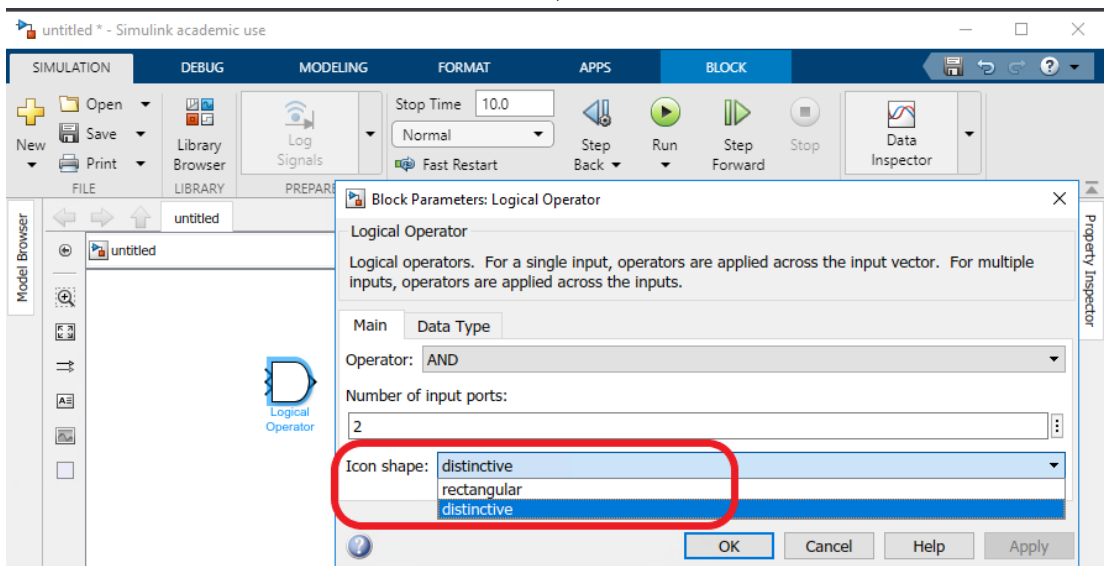
Все базовые логические операции в Simulink реализуются при помощи блока Logical Operator, который находится в браузере библиотеки в разделе Logic and Bit Operations. Откроем новую модель в Simulink. Сохраним ее под именем BasicLog. Добавим на рабочий стол редактора модели блок логических операций и перейдем в его настройки.

В окне настроек на основной вкладке Main из выпадающего списка в строке Operator можно выбрать любую логическую функцию, описанную в табл.1, как показано на рис. 24а. В строке Number of input ports ввести требуемое количество входов логического блока. В строке Icon shape – изменить вид значка, отображаемого на схеме по отечественному стандарту

МЭЖ (в виде прямоугольников – rectangular), либо по американскому стандарту ANSI (distinctive) (рис. 24б).



а)



б)

Рисунок 24 – Окно настроек блока Logical Operator

Теперь подадим на вход блока логических операций логические «1» и «0». Для этого добавим на схему два источника постоянных значений - блок Constant (находится в разделе библиотеки Commonly Used Blocks). Укажем в настройках блоков Constant на первой вкладке Main значения «1» и «0» - для разных блоков, на второй вкладке Signal Attributes изменим тип данных на Boolean – для обоих блоков (рис. 25).

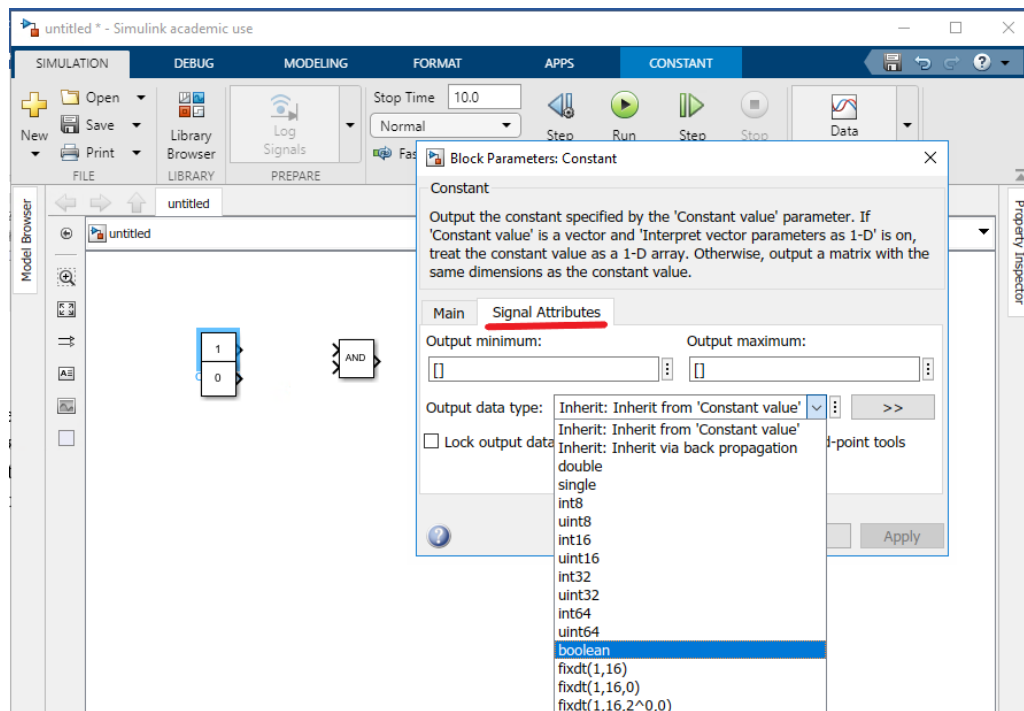


Рисунок 25 – Настройка блоков Constant

Между блоками Constant будем переключаться вручную, используя блок Manual Switch (находится в разделе библиотеки Signal Routing). Соединим блоки, как показано на рис. 26. Примечание: чтобы получить бинарный сигнал на втором входе блока Logic Operator, достаточно скопировать верхнюю часть схемы и соединить ее со вторым входом блока. Выходной сигнал с блока Logic Operator будем фиксировать на Display (находится в разделе библиотеки Sinks).

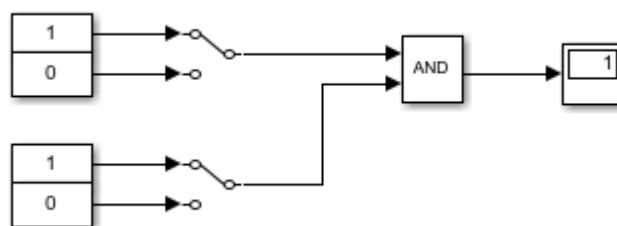


Рисунок 26 – Схема проверки работы логической функции AND

Запустим симуляцию схемы и с помощью ручных ключей Manual Switch проверим правильность работы логического блока AND. Единица на дисплее должна появляться только при двух единицах на входе.

Примечание: для запуска симуляции и проверки работы схемы в строке *Stop time* на панели инструментов редактора Simulink должно стоять значение *inf* – симуляция с бесконечной длительностью.

Задание 1.

Самостоятельно проверьте работу схемы OR (ноль на дисплее должен быть только при двух нулях на входе). При этом форму значка выберете по стандарту ANSI. Сохраните все полученные схемы.

Задание 2.

Соберите в редакторе модели схему, показанную на рисунке 27. Добавьте из библиотеки элемент Relational Operator, который позволяет сравнивать как булевы сигналы, так и любые численные значения. С помощью этого элемента сравните два выходных бинарных сигнала после логических элементов ИЛИ и НЕ так, чтобы работа элемента сравнения была эквивалентна работе логического элемента «Исключающее И» (NXOR). Добавьте на схему сам элемент «Исключающее И» и проверьте правильность настройки элемента Relational Operator путем сравнения сигналов. Сохраните работу.

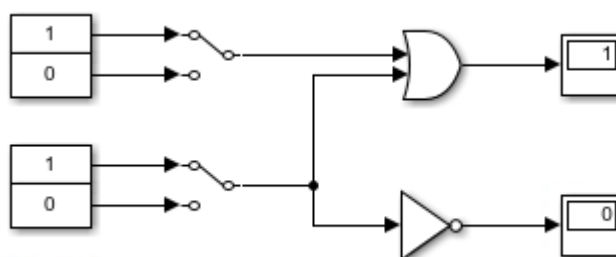


Рисунок 27 – Схема к заданию 2

Задание 3.

Составить в Simulink модель для проверки работы всех логических элементов из таблицы 1. По результатам проверки заполнить таблицу 2. Проверить ее достоверность по таблице истинности этих логических элементов. Сохранить модель в Simulink.

Таблица 2 - Результаты симуляции в модели

Комбинации аргументов на входах	NOT GATE	OR GATE	AND GATE	XOR GATE	NXOR GATE	NAND GATE	NOR GATE
00							
01							
10							
11							

Контрольные вопросы:

1. Для каких систем и в каких случаях применяется Simulink?
2. Перечислите основные возможности ПО Simulink.
3. Какие стандарты для изображения логических элементов на схемах вы знаете и в чем их принципиальное отличие?
4. Сколько входов может иметь элемент инвертор (NOT GATE)?
5. Какую функцию в модели выполняет элемент Scope? В каком разделе браузера библиотеки он находится?

Содержание отчета:

1. Титульный лист по ГОСТ ТПУ.
2. Цель работы.
3. Краткие теоретические сведения.
4. Результаты моделирования с пошаговыми пояснениями, скриншотами схем из Simulink, таблицами.
5. Ответы на контрольные вопросы.
6. Выводы по работе.

Лабораторная работа № 2

Синтез логических устройств в заданном базисе логических элементов

1 Цель работы

Научиться синтезировать логические устройства с учетом требования сокращения номенклатуры логических элементов на схемах и моделировать полученные схемы устройств в ПО Matlab Simulink.

2 Теоретические сведения

Структуру любого логического устройства можно составить на базе трех основных логических элементов И, ИЛИ, НЕ. Это называется *функционально полный логический базис*. Недостатком такой схемы является отсутствие унификации элементов на схеме.

На практике, с целью уменьшения номенклатуры используемых микросхем, часто пользуются функционально полной системой логических элементов в составе двух: штрих Шеффера (операция И-НЕ) и стрелка Пирса (операция ИЛИ-НЕ). Обозначение на схемах и таблицу истинности см. табл. 1 лабораторная работа №1. Напомним их математическую запись и условное обозначение:

- 1) *штрих Шеффера* отображает операцию И-НЕ и обозначается символически вертикальной черточкой \mid :
$$y = x_1 \mid x_2 = \overline{x_1 \cdot x_2}$$

Мнемо-правило: выход $y = 1$ тогда и только тогда, когда хотя бы на одном входе действует "0".
- 2) *стрелка Пирса* отображает операцию ИЛИ-НЕ и обозначается символически вертикальной стрелкой \downarrow :
$$y = x_1 \downarrow x_2 = \overline{x_1 + x_2}$$

Мнемо-правило: выход $y = 0$ тогда и только тогда, когда хотя бы на одном входе действует "1".

Таким образом, любую логическую схему можно выразить как через соединения только элементов И-НЕ, так и через соединения только элементов ИЛИ-НЕ. Для цифровой электроники это означает, что для реализации любой цепи преобразования сигнала достаточно одного типового элемента. В таком случае достигают унификации элементов на схеме и говорят, что логическое устройство реализовано в базисе элементов И-НЕ или ИЛИ-НЕ (рис. 1).

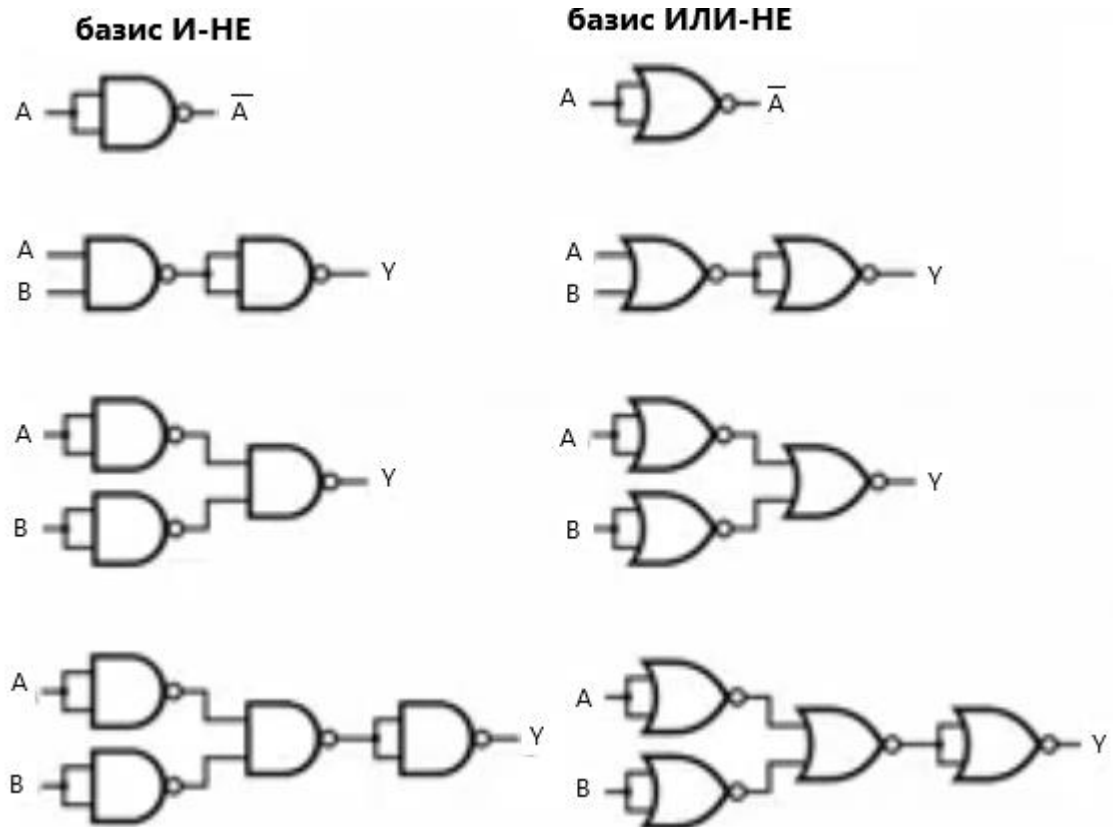


Рисунок 28 – Примеры реализации логических устройств в базисах И-НЕ, ИЛИ-НЕ

Существуют специальные приемы для перехода к базисам И-НЕ и ИЛИ-НЕ.

Если необходимо перейти в базис И-НЕ, то исходную функцию алгебры логики (ФАЛ) записывают в виде ДНФ, затем, путем двойного инвертирования выражения ДНФ функции или его части и применения теорем де Моргана логическая функция приводится к виду, содержащему только операции логического умножения и инвертирования.

Если же необходимо перейти в базис ИЛИ-НЕ, то исходную логическую функцию записывают в КНФ, затем, аналогично, применяя двойное инвертирование функции или ее части и теорем де Моргана приводят ее к виду, содержащему только операции логического сложения и инверсии. Далее логическое выражение записывается через условные обозначения выбранных операций.

Пример 1.

Перевести исходную ДНФ функции в базис И-НЕ:

$$\begin{aligned}
 y &= x_1 \cdot x_3 + \bar{x}_0 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_0 \cdot \bar{x}_1 \cdot \bar{x}_2 = \overline{\overline{x_1 \cdot x_3 + \bar{x}_0 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_0 \cdot \bar{x}_1 \cdot \bar{x}_2}} = \\
 &= \overline{(\overline{x_1 \cdot x_3}) \cdot (\overline{\bar{x}_0 \cdot \bar{x}_2 \cdot \bar{x}_3}) \cdot (\overline{x_0 \cdot \bar{x}_1 \cdot \bar{x}_2})} = (x_1 | x_3) | (\bar{x}_0 | x_2 | \bar{x}_3) | (x_0 | \bar{x}_1 | \bar{x}_2).
 \end{aligned}$$

Аналогично, функция КНФ в базисе ИЛИ-НЕ имеет вид:

$$y = (x_0 + x_3) \cdot (\overline{x_0} + x_1 + x_2) \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3}) = \overline{\overline{(x_0 + x_3)} \cdot \overline{(\overline{x_0} + x_1 + x_2)} \cdot \overline{(\overline{x_1} + \overline{x_2} + \overline{x_3})}}$$

$$\overline{(x_0 + x_3) + (\overline{x_0} + x_1 + x_2) + (\overline{x_1} + \overline{x_2} + \overline{x_3})} = (x_0 \downarrow x_3) \downarrow (\overline{x_0} \downarrow x_1 \downarrow x_2) \downarrow (\overline{x_1} \downarrow \overline{x_2} \downarrow \overline{x_3}).$$

Пример 2.

Пусть логическая функция задана выражением:

$$y = x_0 \cdot x_3 + (\overline{x_0} + x_2 + \overline{x_3}) \cdot (x_1 + \overline{x_2});$$

Требуется привести функцию к базису И-НЕ, ИЛИ-НЕ.

Решение:

Сложные функции удобно переобозначать через более простые.

Пусть:

$$y_1 = x_0 \cdot x_3, y_2 = \overline{x_0} + x_2 + \overline{x_3}, y_3 = x_1 + \overline{x_2};$$

Тогда:

$$y = y_1 + \overline{y_2} \cdot y_3;$$

а) приводим функцию к базису И-НЕ

$$y = y_1 + \overline{y_2} \cdot y_3 = y_1 + \overline{\overline{\overline{y_2}} \cdot \overline{y_3}} = y_1 + \overline{\overline{y_2}} \mid \overline{\overline{y_3}} = y_1 + \overline{\overline{y_2}} \mid \overline{\overline{y_3}} = \overline{\overline{y_1}} \cdot \overline{\overline{\overline{y_2}} \mid \overline{\overline{y_3}}} = \overline{\overline{y_1}} \mid (\overline{\overline{y_2}} \mid \overline{\overline{y_3}});$$

$$\overline{\overline{y_1}} = \overline{\overline{x_0 \cdot x_3}} = x_0 \mid x_3;$$

$$\overline{\overline{y_2}} = \overline{\overline{\overline{x_0} + x_2 + \overline{x_3}}} = x_0 \cdot \overline{x_2} \cdot x_3 = x_0 \cdot \overline{x_2} \cdot x_3 = x_0 \mid \overline{x_2} \mid x_3;$$

$$y_3 = x_1 + \overline{x_2} = \overline{\overline{\overline{x_1} + \overline{x_2}}} = \overline{\overline{x_1}} \cdot \overline{x_2} = \overline{x_1} \mid x_2;$$

$$y = (x_0 \mid x_3) \mid ((x_0 \mid \overline{x_2} \mid x_3) \mid (\overline{x_1} \mid x_2));$$

б) приводим функцию к базису ИЛИ-НЕ

$$y = y_1 + \overline{y_2} \cdot y_3 = y_1 + \overline{\overline{\overline{y_2}} \cdot \overline{y_3}} = y_1 + (\overline{y_2} + \overline{y_3}) = y_1 + (\overline{\overline{\overline{y_2}} + \overline{\overline{\overline{y_3}}}}) = y_1 \downarrow (\overline{\overline{\overline{y_2}} + \overline{\overline{\overline{y_3}}}});$$

$$y_1 = x_0 \cdot x_3 = \overline{\overline{\overline{x_0 \cdot x_3}}} = \overline{\overline{x_0} + \overline{x_3}} = \overline{x_0} \downarrow \overline{x_3};$$

$$y_2 = \overline{x_0} + x_2 + \overline{x_3} = \overline{\overline{\overline{\overline{\overline{x_0} + x_2 + \overline{x_3}}}}} = \overline{\overline{x_0}} \downarrow \overline{x_2} \downarrow \overline{\overline{x_3}};$$

$$y_3 = x_1 + \overline{x_2} = \overline{\overline{\overline{x_1} + \overline{x_2}}} = \overline{x_1} \downarrow \overline{\overline{x_2}} \Rightarrow \overline{y_3} = x_1 \downarrow \overline{x_2};$$

$$y = (\overline{\overline{\overline{x_0} \downarrow \overline{x_3}}}) \downarrow ((\overline{\overline{\overline{x_0} \downarrow \overline{x_2} \downarrow \overline{x_3}}}) \downarrow (x_1 \downarrow \overline{x_2})).$$

Задание.

В соответствии со своим вариантом задания в табл. 1 (порядковый номер по списку группы) синтезировать логическую функцию четырех аргументов x_1, x_2, x_3, x_4 в заданном базисе логических элементов (по возможности, упростив). Собрать схему в заданном базисе в Simulink и промоделировать ее работу. Результаты моделирования занести в табл. 2.

Таблица 1 – Варианты заданий

Номер варианта	ФАЛ	Базис
1	$y = x_1\bar{x}_2 + x_3x_2 + \bar{x}_1x_2x_4$	И-НЕ
2	$y = (\bar{x}_1 + x_2)(\bar{x}_3 + x_2)(x_1 + x_2 + x_4)$	ИЛИ-НЕ
3	$y = x_1 + x_1x_2x_3 + \bar{x}_1x_2x_3\bar{x}_4$	И-НЕ
4	$y = \bar{x}_1(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3 + x_4)$	ИЛИ-НЕ
5	$y = x_1\bar{x}_4 + x_3x_1 + \bar{x}_1x_3x_4$	И-НЕ
6	$y = (x_1 + x_4)(\bar{x}_3 + \bar{x}_1)(x_1 + x_3 + x_4)$	ИЛИ-НЕ
7	$y = x_2x_4 + \bar{x}_3 + x_1 + x_3\bar{x}_4$	И-НЕ
8	$y = \bar{x}_3x_1(\bar{x}_2 + x_4)(x_3 + x_4)$	ИЛИ-НЕ
9	$y = x_2 + \bar{x}_1\bar{x}_2\bar{x}_3 + x_1x_2x_4$	И-НЕ
10	$y = x_2(x_1 + x_2 + x_3)(\bar{x}_1 + x_2 + x_4)$	ИЛИ-НЕ
11	$y = \bar{x}_2\bar{x}_4 + x_4x_2x_3 + \bar{x}_1x_2x_3$	И-НЕ
12	$y = (x_2 + x_4)(\bar{x}_4 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3)$	ИЛИ-НЕ
13	$y = \bar{x}_2 + x_1x_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_4 + x_3$	И-НЕ
14	$y = x_2\bar{x}_3(x_1 + \bar{x}_2 + x_3)(x_1 + x_2 + x_4)$	ИЛИ-НЕ
15	$y = x_1x_2 + x_3x_2 + x_2\bar{x}_4$	И-НЕ
16	$y = (x_1 + \bar{x}_2)(x_3 + x_2)(x_2 + x_4)$	ИЛИ-НЕ
17	$y = \bar{x}_4x_2 + \bar{x}_3x_2\bar{x}_4 + x_1x_4$	И-НЕ
18	$y = (x_4 + \bar{x}_2)(x_3 + x_2 + \bar{x}_4)(\bar{x}_1 + x_4)$	ИЛИ-НЕ
19	$y = \bar{x}_4 + x_3x_2\bar{x}_1 + x_1x_4$	И-НЕ
20	$y = x_4(\bar{x}_3 + x_2 + x_1)(\bar{x}_1 + x_4)$	ИЛИ-НЕ
21	$y = x_1 + x_2 + \bar{x}_3x_1 + \bar{x}_1x_3x_4$	И-НЕ
22	$y = \bar{x}_1\bar{x}_2(x_3 + x_1)(\bar{x}_1 + x_3 + x_4)$	ИЛИ-НЕ
23	$y = x_1x_2 + x_3x_4 + \bar{x}_2\bar{x}_3x_4$	И-НЕ
24	$y = (\bar{x}_1 + \bar{x}_2)(x_3 + x_4)(x_2 + x_3 + x_4)$	ИЛИ-НЕ
25	$y = x_2x_4 + \bar{x}_3\bar{x}_4 + x_2 + \bar{x}_1$	И-НЕ

Таблица 2 – Результаты симуляции в модели

Комбинации аргументов x_1, x_2, x_3, x_4	Значения функции y
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Контрольные вопросы:

1. Что такое функционально полный логический базис?
2. В чем особенность логических элементов штрих Шеффера и стрелка Пирса?
3. Как перейти в базис И-НЕ и ИЛИ-НЕ?

Содержание отчета:

1. Титульный лист по ГОСТ ТПУ.
2. Цель работы.
3. Краткие теоретические сведения.
4. Номер варианта;
5. Исходные данные;
6. Результаты моделирования с пошаговыми пояснениями, скриншотами схем из Simulink, таблицами.
7. Ответы на контрольные вопросы.
8. Выводы по работе.

Лабораторная работа № 3

Проектирование комбинационных схем конечных логических автоматов в Simulink

1 Цель работы

Познакомиться с комбинационными и последовательностными цифровыми устройствами. Научиться проектировать комбинационные схемы конечных логических автоматов в среде ПО Matlab Simulink.

2 Теоретические сведения

Все цифровые устройства по принципу функционирования разделяются на два типа: *комбинационные* и *последовательностные*.

Комбинационные цифровые устройства (КЦУ) или **логические автоматы без памяти** называются устройства, в которых текущие значения на выходах полностью определяются текущими значениями на входах.

Последовательностные цифровые устройства (ПЦУ) или **логические автоматы с памятью** называются устройства, в которых состояния выходов определяются не только текущими входными сигналами, но и входными сигналами, действующими в предыдущие моменты времени.

Логика функционирования КЦУ чаще всего задается таблицей истинности. Рассмотрим некоторое КЦУ, которое имеет 3 входа и 1 выход и работает в соответствии с таблицей истинности (рис. 1).

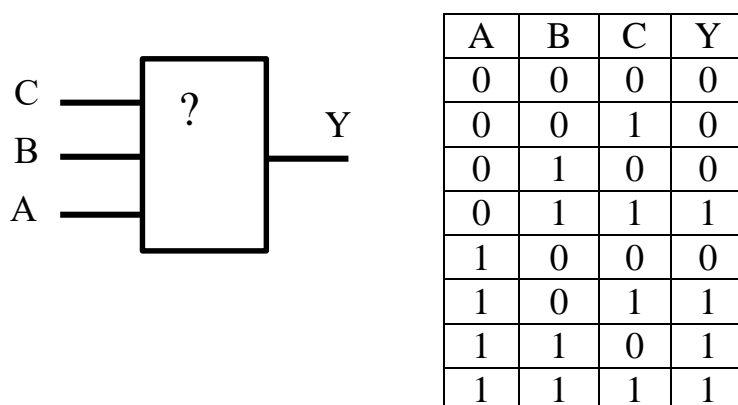


Рисунок 1 – КЦУ и его таблица истинности

Входные комбинации из 3-х бит можно рассматривать как бинарное представление целого числа, где сигнал А соответствует старшему биту, а

сигнал С – младшему. Таким образом, каждую строчку в таблице истинности можно обозначить одним целым числом от 0 до 7:

первая верхняя строка $000_2 = 0_{10}$,
 вторая строка $001_2 = 1_{10}$,

 последняя строка $111_2 = 7_{10}$.

Согласно заданной таблице истинности (рис.1) в строках 3, 5, 6 и 7 $Y=1$. Значит в КЦУ при таких комбинациях на входе, на выходе будет высокий логический сигнал, т.е. «1», а при остальных комбинациях – низкий, т.е. «0».

Теперь нужно составить схему такого КЦУ из логических элементов. Вариантов реализации этой логики может быть несколько, но лучше выбирать наиболее экономичный, т.е. состоящий из минимально возможного количества логических элементов.

Задача поиска минимальной рабочей схемы КЦУ называется *синтез*. При решении задачи синтеза КЦУ выполняется минимизация логического выражения, описывающего его работу. Минимизировать можно любыми известными методами: как с помощью различных Теорем и Аксиом алгебры логики, так и с помощью карт Карно, диаграмм Вейтча, метода Квайна или метода Квайна-Мак-Класки.

Для минимизации схемы заданного КЦУ воспользуемся изученным на практических занятиях методом карт Карно. Для этого по заданной таблице истинности составим карту Карно и выполним склейку по единицам, получим МДНФ (рис. 2).

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0			1	
	1		1	1	1

Рисунок 2 – Карта Карно для минимизации схемы КЦУ

По карте Карно составим минимизированную логическую функцию в ДНФ, описывающую работу КЦУ:

$$Y = A \cdot B + A \cdot C + B \cdot C \tag{1}$$

Таким образом, выходной сигнал с КЦУ будет равен «1», если будут включены сигналы А и В или А и С или В и С или все одновременно. Действительно, согласно таблице истинности (рис.1) в тех строках, где стоит две или более единиц, на выходе Y - единица.

Представим полученную в МДНФ функцию в базисе И-НЕ, взяв двойную инверсию и применив правило де Моргана:

$$Y = \overline{\overline{A \cdot B} \cdot \overline{A \cdot C} \cdot \overline{B \cdot C}} \quad (2)$$

По полученным формулам (1) и (2) составим схемы КЦУ (рис. 3). Выглядеть они будут по-разному, но работа их эквивалентна.

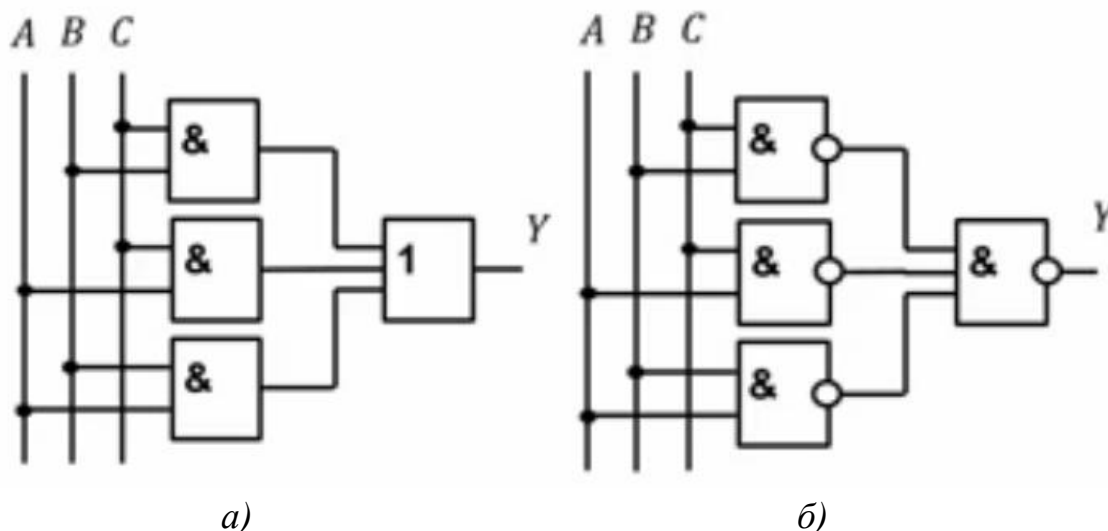


Рисунок 3 – Эквивалентные схемы КЦУ:
a) базисе И, ИЛИ, НЕ, *б)* в базисе И-НЕ

Схема рис. 3*а* составлена из 3-х элементов И и 1-го трехвходового элемента ИЛИ.

Схема рис. 3*б* составлена из 4-х однотипных элементов И-НЕ, говорят, что она реализована в базисе И-НЕ.

Смоделируем эти схемы в Simulink.

Запускаем Simulink. Создаем и сохраняем новую модель под именем ModelKTSY.

Соберем в редакторе модели Simulink схему, как показано на рис. 4. В модели присутствует источник входных сигналов - блок Constant, сама схема, собранная на элементах AND/OR, эквивалентная рис. 3*а* и блок Display для контроля результата работы модели.

Для выделения 3-х младших битов целого числа на входе используются блоки Extract Bit. Блоки Extract Bit необходимо настроить, как показано на рис. 5. Проставляя в строке Bit Indices для 1-го, 2-го и 3-го блоков соответственно значения 0, 1, 2.

Для преобразования входного сигнала в логический тип boolean используются блоки Data Type Conversion. Блоки Data Type Conversion необходимо настроить в соответствии с рис. 6.

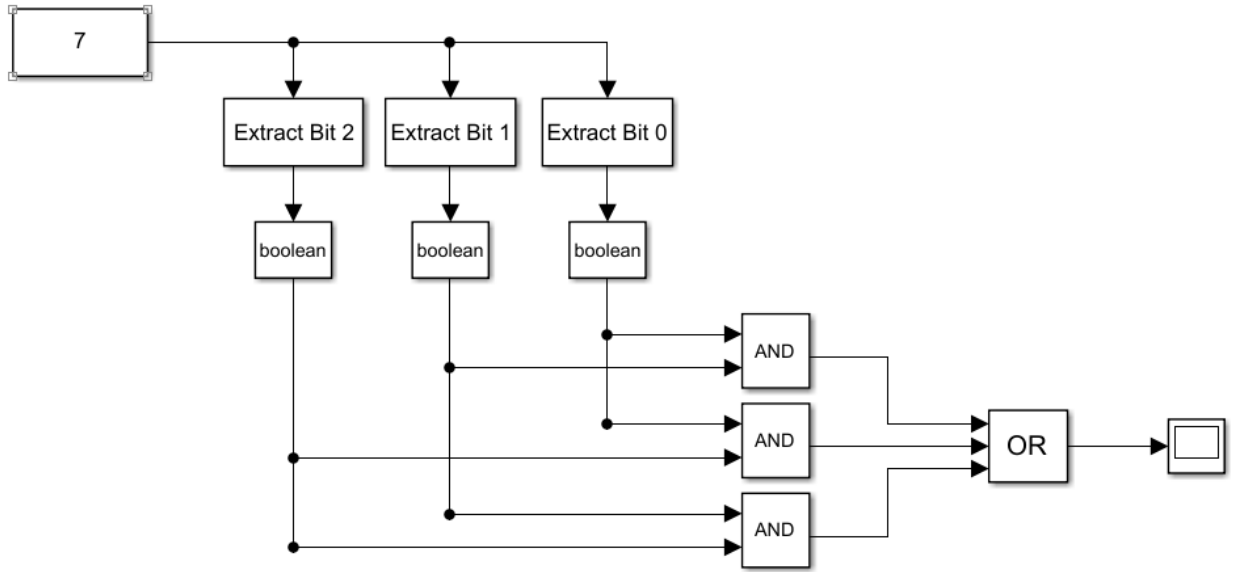


Рисунок 4 – Схема КЦУ в Simulink

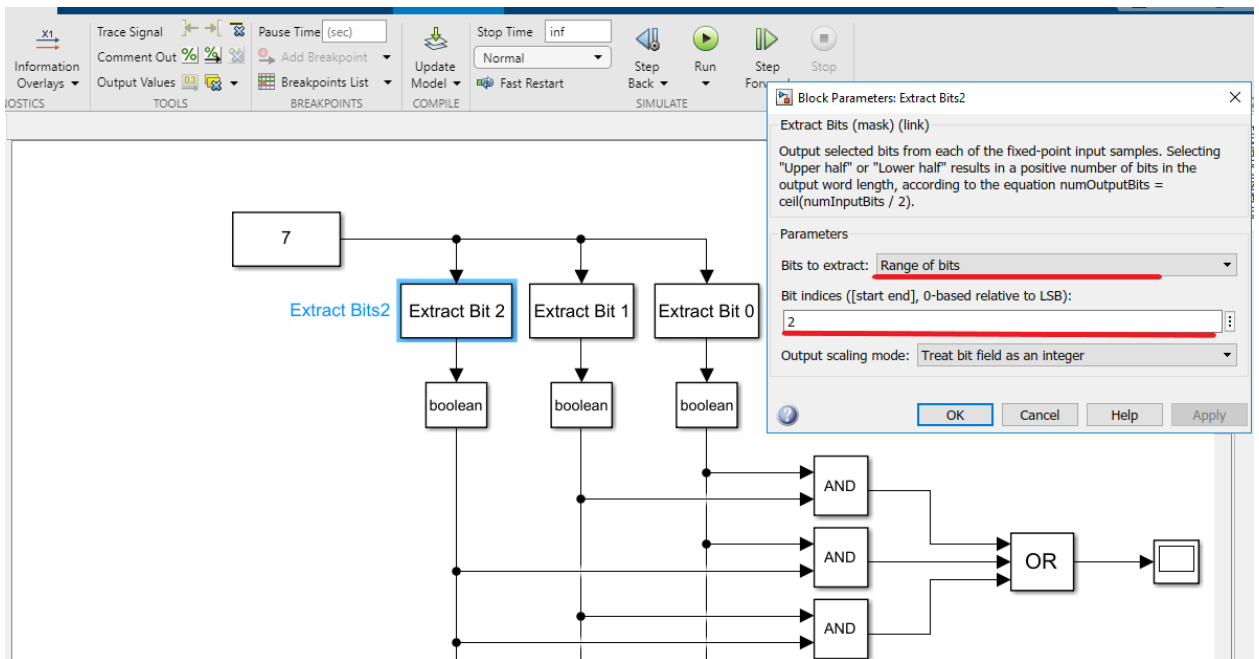


Рисунок 5 – Настройка блоков Extract Bit

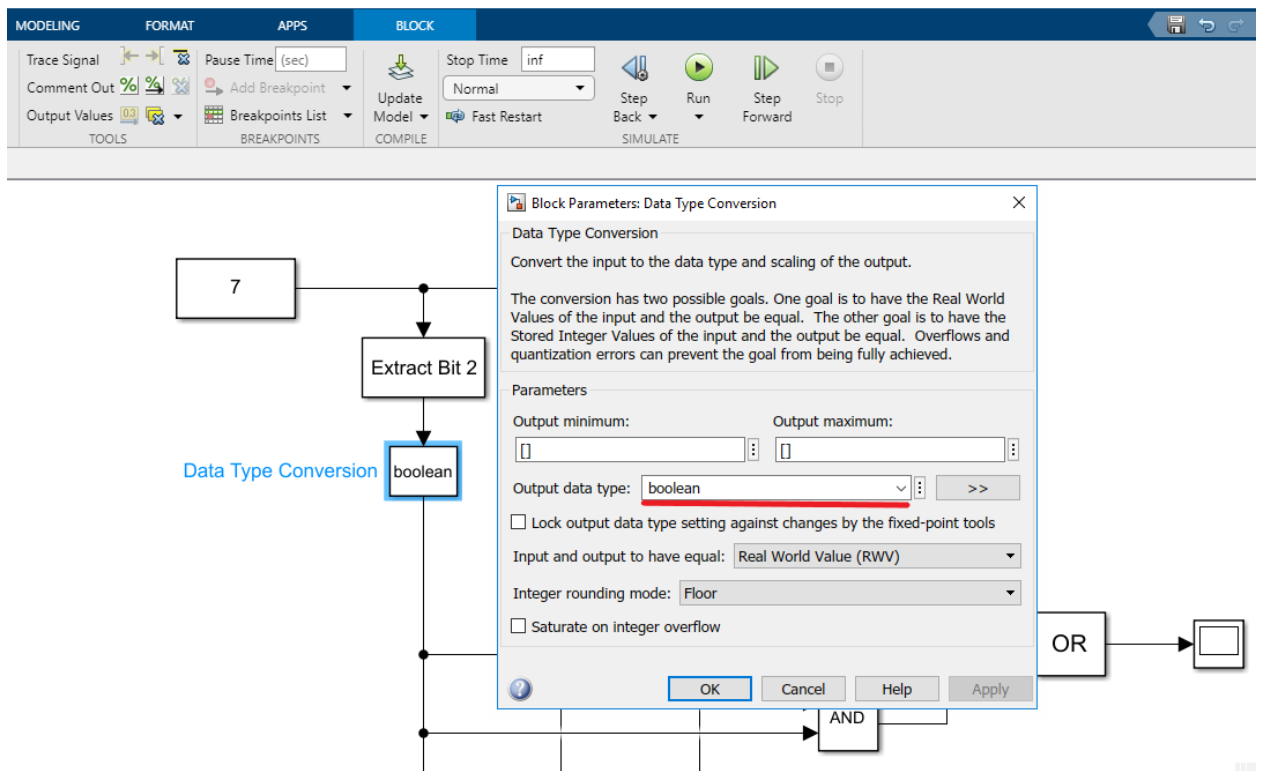


Рисунок 6 – Настройка блоков Data Type Conversion

Когда модель собрана, подсветите типы данных, как показано на рис. 7, чтобы проверить корректность модели.

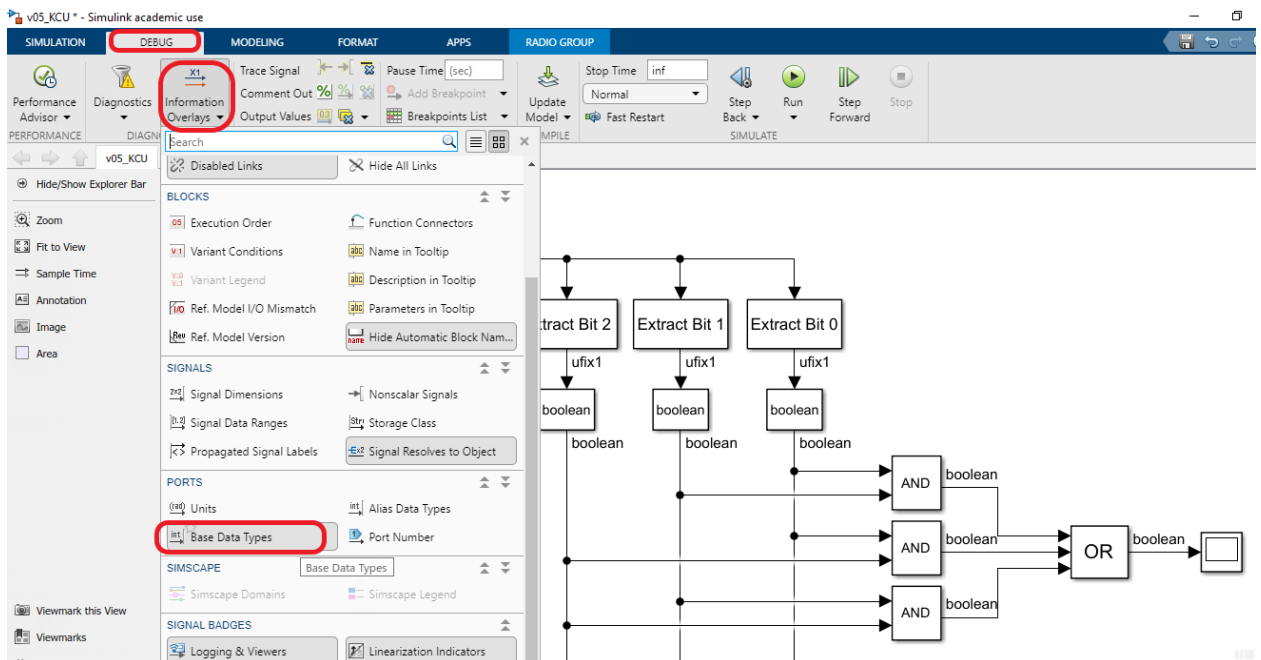


Рисунок 7 – Визуализация типов данных на модели

Блок Constant выдает в модели целочисленные положительные значения, из которых блоками Extract Bit выделяется три младших бита и преобразовываются в тип boolean блоками Data Type Conversion. Логические

значения битов подаются в схему сравнения, а на ее выходе на дисплее фиксируется логические «0» или «1».

Кнопкой Run в Simulink запустим симуляцию модели и, изменяя значения входной константы от 0 до 7, проверим правильность работы схемы, сравнивая результаты с таблицей истинности на рис. 1.

Для перебора дискретных входных значений удобно использовать элементы Simulink из библиотеки Dashboard – это графические блоки для визуализации и интерактивного изменения параметров других блоков (различные кнопки, поля ввода, индикаторы и тд.). Выберем виджет Radio Button. Присоединим его к блоку Constant и введем числовые значения от 0 до 7. Настроим внешний вид, добавляя подписи рядом с кнопками. Настройки виджета Radio Button показаны на рис. 8.

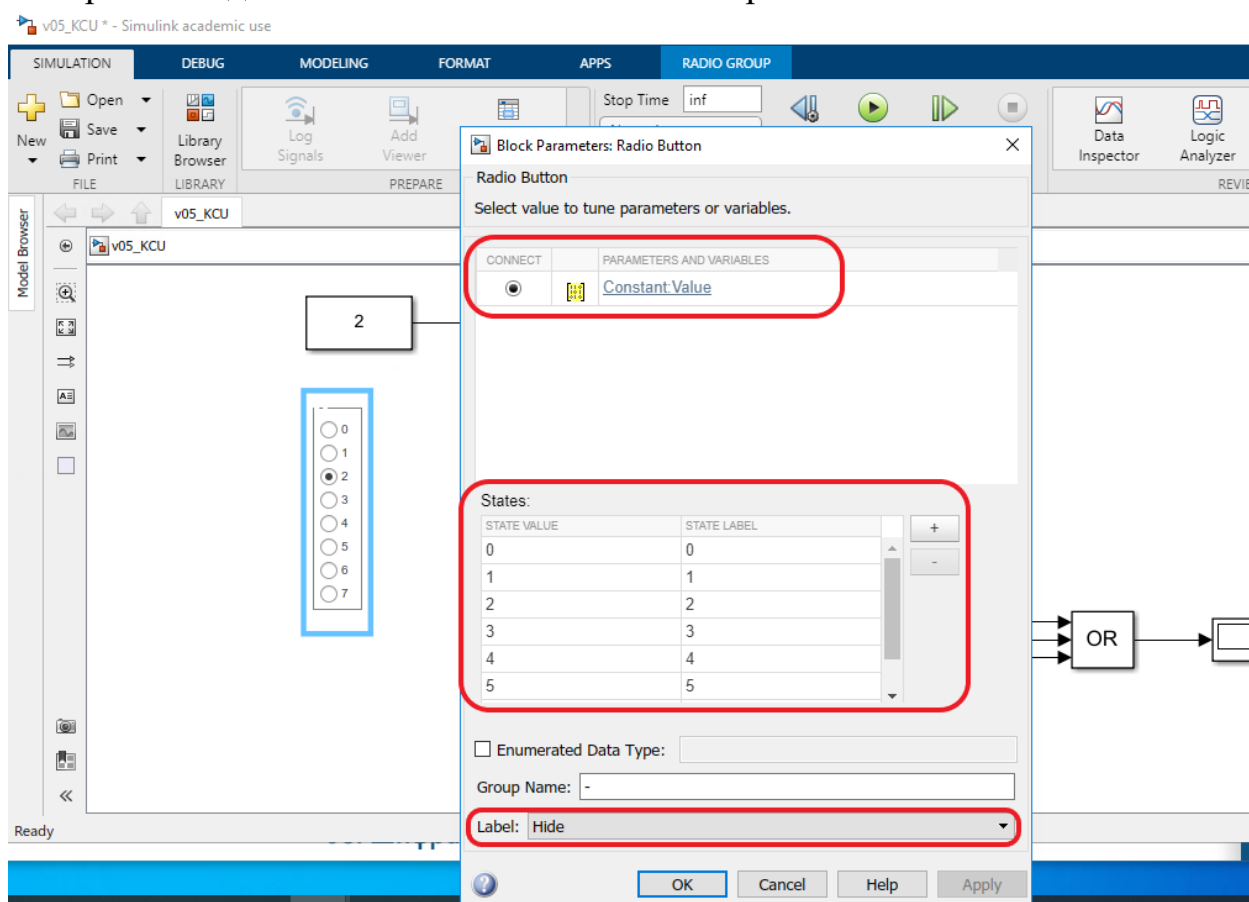


Рисунок 8 – Настройки виджета Radio Button

Проверьте работу виджета, запустив симуляцию и сравнив результаты работы схемы, фиксируемые на дисплее, с таблицей истинности (рис.1).

Теперь построим модель в Simulink, реализующую другую форму (2) записи логического выражения в базисе И-НЕ. Схема приведена на рис. 3б. *Примечание: для построения схемы в базисе И-НЕ выберите форму представления логических элементов по стандарту ANSI.*

В результате должна получиться модель, представленная на рис. 9.

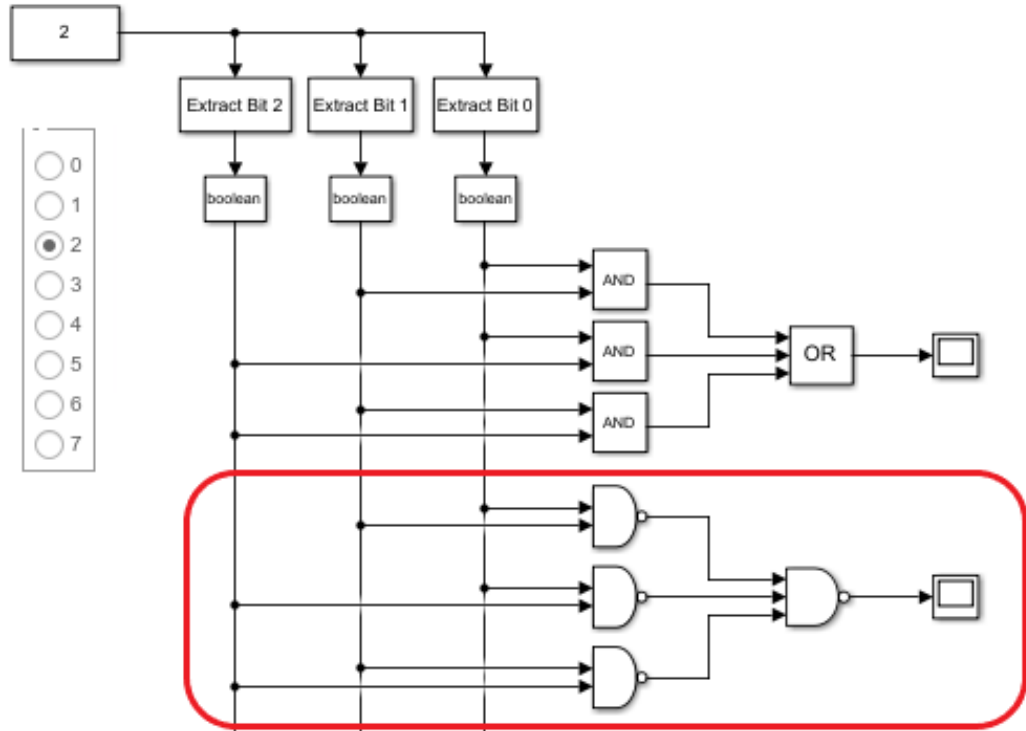


Рисунок 9 – Модель Simulink в базе И-НЕ

Запустите симуляцию и проверьте правильность функционирования обеих схем, сравнивая значения результатов на дисплеях.

Для реализации любого КЦУ в Simulink не обязательно создавать схему из логических элементов. Для любого КЦУ с любым числом входов и выходов можно использовать блок Combinatorial Logic. Подадим ему на вход вектор из трех бит через блок мультиплексора (Mux), который нужен для объединения скалярных сигналов в векторный. Результат будем фиксировать также на дисплее (рис. 10). Затем двойным щелчком левой кнопки мыши на блоке Combinatorial Logic перейдем в его настройки и введем в строке Truth table выходной столбец таблицы истинности (рис.1), как показано на рис. 10.

Запустите симуляцию и проверьте идентичность результатов на трех дисплеях при изменении дискретных входных сигналов.

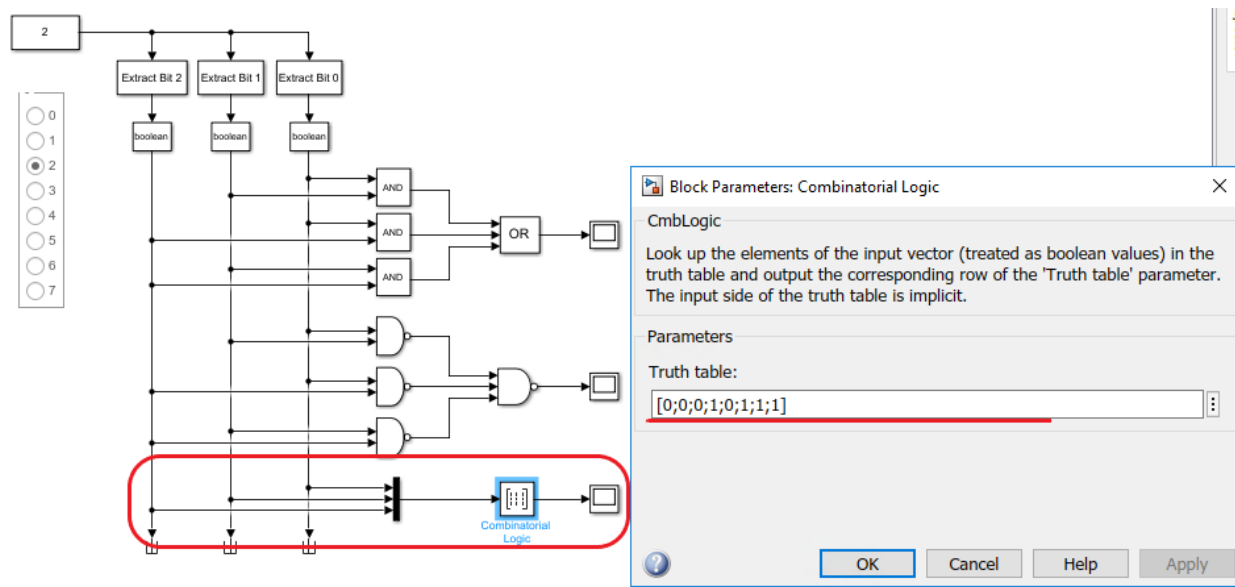


Рисунок 10 – Построение схемы КЦУ на основе блока Combinatorial Logic

Задание.

Для ФАЛ из табл.1, лабораторной работы № 2 построить в Simulink модель КЦУ, реализующего заданную логику с учетом: 1. критерия минимизации элементов на схеме; 2. унификации номенклатуры используемых логических элементов. Переключение входного сигнала на схеме выполнить с применением виджета Radio Button. Для проверки правильности функционирования собранных вариантов схем добавить в модель блок Combinatorial Logic. Вариант задания взять по принципу: (№п/п+1).

Контрольные вопросы:

1. Что такое КЦУ и ПЦУ в чем различие?
2. Что такое синтез КЦУ?
3. Запишите десятичное число 13 в двоичном коде, используя 4 бита.
4. Для чего в модели используется блок Extract Bit?
5. С помощью какого блока в модели Simulink осуществить преобразование типа сигнала?

Содержание отчета:

1. Титульный лист по ГОСТ ТПУ.
2. Цель работы.
3. Краткие теоретические сведения.

4. Номер варианта;
5. Исходные данные;
6. Результаты моделирования с пошаговыми пояснениями, скриншотами схем из Simulink, таблицами.
7. Ответы на контрольные вопросы.
8. Выводы по работе.

Лабораторная работа № 4

Логическое управление на базе теории нечетких множеств с применением FIS редактора в Simulink

1 Цель работы

Получить представление о синтезе нестандартных логических законов управления, способных обеспечить высокие качественные показатели ведения технологического процесса, основанных на теории нечетких множеств с применением пакета Fuzzy Logic Toolbox ПО Simulink.

2 Теоретические сведения

2.1 Общие сведения о пакете Fuzzy Logic Toolbox:

Для рассмотрения результатов разработки и функционирования системы нечёткого вывода будем использовать графические средства пакета Fuzzy Logic Toolbox. Эти же средства используются и при разработке систем нечёткого вывода как графический объектно-ориентированный язык автоматического программирования.

В состав этих средств входят:

- редактор систем нечёткого вывода FIS Editor (FIS);
- редактор функций принадлежности систем нечёткого вывода Membership Function Editor (MFE);
- редактор правил систем нечёткого вывода Rule Editor;
- программа просмотра правил системы нечёткого вывода Rule Viewer;
- программа просмотра поверхности нечёткого вывода Surface Viewer.

2.2 Система нечёткого вывода

Система нечёткого вывода (СНВ) реализует процесс получения результата по исходным данным посредством нечётких продукций, иначе говоря, по правилам нечётких продукций.

Целью нечёткого вывода может быть получение результата либо в виде нечёткой величины с оценкой степени её принадлежности, либо чёткой количественной оценки результата.

Правило-продукция состоит из двух частей: условия (антецедента) или посылки и заключения (консеквента) или вывода. Продукция основывается на правиле заключения при исчислении предикатов — *modus ponens*: если А

и $A \supset B$ (импликация) произвольно выводимые формулы, т.е. истинные выражения, то B — так же выводимая формула.

В правилах нечётких продукций и условная часть, и заключения описываются как нечёткие высказывания. Таким образом, нечёткая продукция является аналогом правила *modus ponens* в нечёткой среде высказываний, реализуя немонотонный вывод.

Для описания нечётких высказываний используются нечёткие *лингвистические переменные (ЛП)*.

ЛП — это именованная переменная, которая принимает свои значения из множества лингвистических термов, т.е. символьных величин.

Для нечёткой ЛП терм-множество задаётся как нечёткое множество. Этот процесс называется *фаззификацией*.

Фаззификация является одной из проблемных задач описания нечёткого вывода и отражает индивидуальные эмпирические знания автора.

Нечёткие высказывания в условной части нечёткой продукции могут быть составными, соединёнными связками “И” и/или “ИЛИ”. Эти связки при исчислении высказываний реализуются логическими или арифметическими операциями пересечения или объединения, соответственно.

Операции исчисления высказываний в условной части нечётких продукций с определением степени их истинности называют *агрегированием*.

При получении результата по каждому правилу необходимо дать оценку степени его истинности. Эта оценка зависит от степени истинности высказываний условной части правила, степени истинности отношения, положенного в основу правила, между исходными утверждениями (посылкой) и заключением, т.е. степени истинности импликации, и степени истинности высказывания относительно значения из терм-множества возможных результатов, приведенного в правиле.

Получение оценки степени истинности заключения, полученного по правилу, называют *активизацией*.

В случае необходимости получения чёткого количественного значения результата оно может быть получено на основании функции принадлежности терм-множества результата различными способами по алгоритмам, названным по именам их авторов (Мамдани, Сугено, Цукамото и т.д.), что определяет тип системы нечёткого вывода. Эта операция называется *дефаззификацией*.

2.3 Редактор систем нечёткого вывода FIS

Редактор FIS является основным средством, которое используется для создания и редактирования систем нечёткого вывода в графическом режиме.

Вызов редактора FIS для создания СНВ производится с помощью ввода функции fuzzy в окне команд системы MATLAB.

Редактор даёт возможность описать основные свойства СНВ: тип и структуру, способы реализации операций агрегирования, активизации и дефаззификации, а также описание лингвистических переменных, используемых при описании модели реального мира.

Описание выполняется с помощью графического интерфейса через окно редактора. Интерфейс позволяет так же вызывать все другие редакторы и программы просмотра.

На схеме, в верхней части окна редактора FIS (рис. 1), приведена схема СНВ, установленная по умолчанию. Левый и правый прямоугольники – входная и выходная ЛП, соответственно, именованные по умолчанию. Центральный прямоугольник отображает процессор нечетких правил (именованный по умолчанию Untitled). Совокупность этих правил описывается в БЗ и определяет функционирование СНВ.

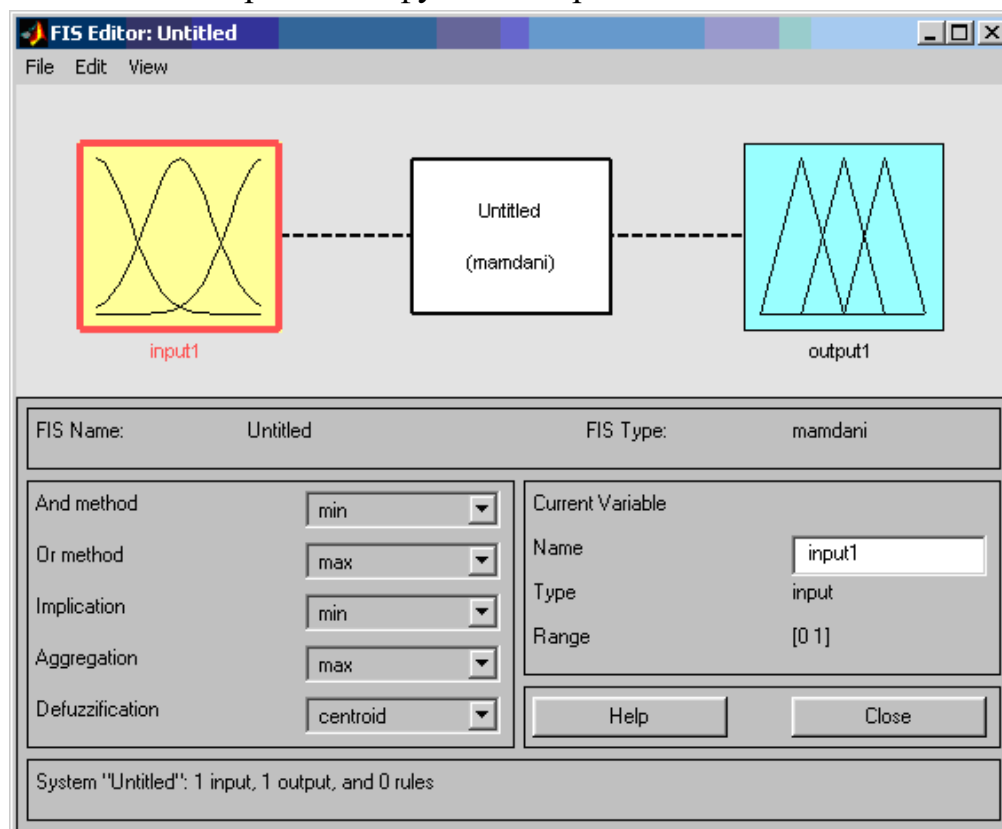


Рисунок 1 - Графический интерфейс редактора FIS

По умолчанию так же задается целый ряд параметров: тип СНВ по алгоритму Мамдани, нечеткие логические операции, методы импликации, агрегирования и дефаззификации.

Изменение начальных установок может производиться следующим образом.

Для добавления входной переменной следует выполнить команду основного меню Edit > Add Variable... > Input. После этого число левых прямоугольников увеличится на один, именованный по умолчанию «input».

Для удаления входной переменной следует выделить её щелчком по изображению и нажать клавишу Delete.

Для методов выполнения операций нечеткого вывода изменение производится выбором нужных пунктов из пяти всплывающих меню в левой нижней области окна.

2.4 Редактор функций принадлежности (MFE)

Редактор функций принадлежности в графическом режиме обеспечивает задание и изменение функции принадлежности любых термов ЛП СНВ.

Для фаззификации лингвистической переменной СНВ следует выделить ее изображение – именованный прямоугольник в левой верхней части окна редактора (рис. 2).

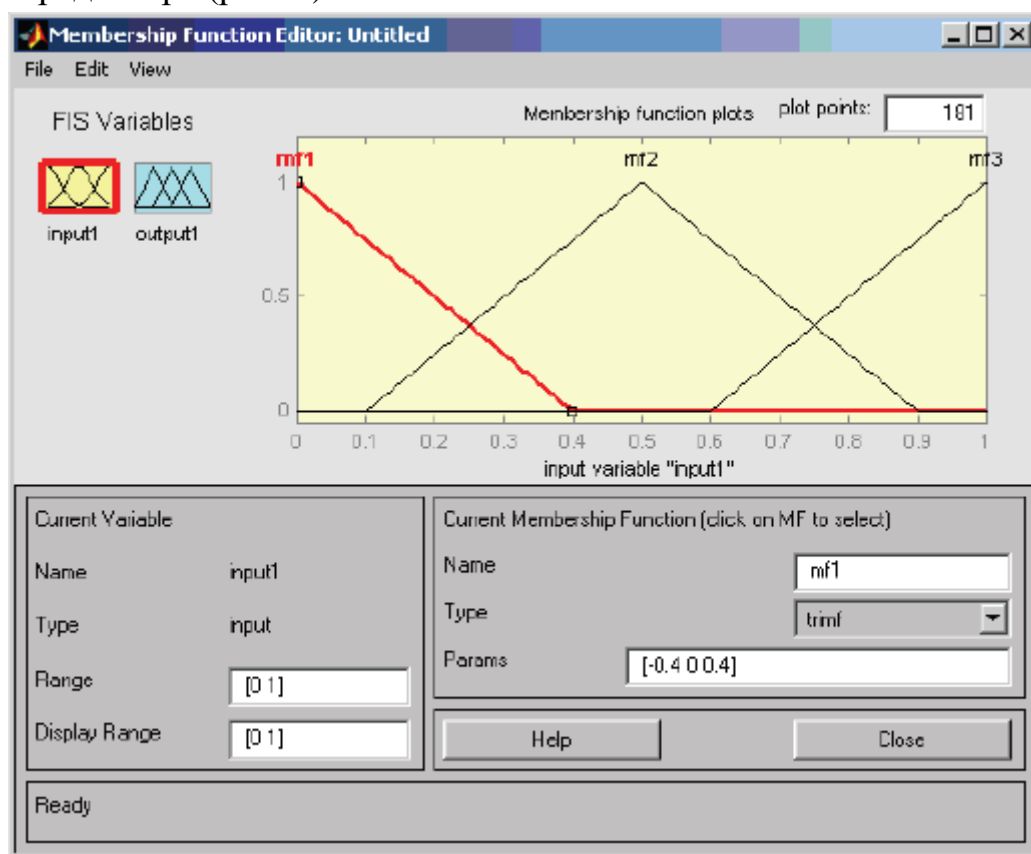


Рисунок 2 – Графический интерфейс редактора MFE

В окне редактора выводятся графики функций принадлежности для всех значений выделенной ЛП (по умолчанию для трёх значений).

Для описания функции принадлежности каждого значения ЛП используются три поля: Name, Type и Params. Описываемая функция выделяется щелчком по её графику. В поле Name устанавливается значение ЛП. В поле Type, выбором элемента меню, устанавливается имя нужной функции принадлежности (одной из 11-ти встроенных). В поле ввода Params указываются необходимые параметры функции принадлежности, которые определяют положение ее модальных значений на числовой шкале, диапазон изменения которой указывается в полях ввода Range и Display range.

Эти операции выполняются над всеми значениями из терм-множеств лингвистических переменных СНВ.

Добавление нового значения ЛП со встроенной функцией принадлежности производится по команде основного меню Edit > Add MF.

Удаление ненужного значения ЛП производится нажатием клавиши Delete, после выделения графика функции принадлежности этого значения.

2.5 Редактор правил системы нечеткого вывода

Редактор правил СНВ (рис. 3) обеспечивает описание правил системы в графическом режиме. Основой языка представления знаний (ЯПЗ) являются априорные элементарные нечеткие высказывания относительно значений ЛП вида:

<имя ЛП is значение ЛП>

Сложные нечеткие высказывания в условных частях нечетких продукций соединяются связками «and» и/или «or».

Условная часть нечеткого правила вводится словом «If» и отделяется от заключения словом «then».

После заключительной части в правиле в скобках указывается значение весового коэффициента правила (поле ввода Weight).

Правила нумеруются.

Таким образом база знаний СНВ, описанная средствами пакета Fuzzy Logic ToolBox, представляется линейной последовательностью нумерованных нечетких продукций, описанных изложенным образом.

Тексты введённых правил размещаются на поле в верхней части окна редактора правил.

Описание новых или изменение нечетких продукций производится только после описания всех необходимых ЛП средствами редакторов FIS и MFE. После этого могут выполняться три вида операций над правилами: удаление (кнопка Delete rule), добавление (кнопка Add rule) и изменение (кнопка Change rule).

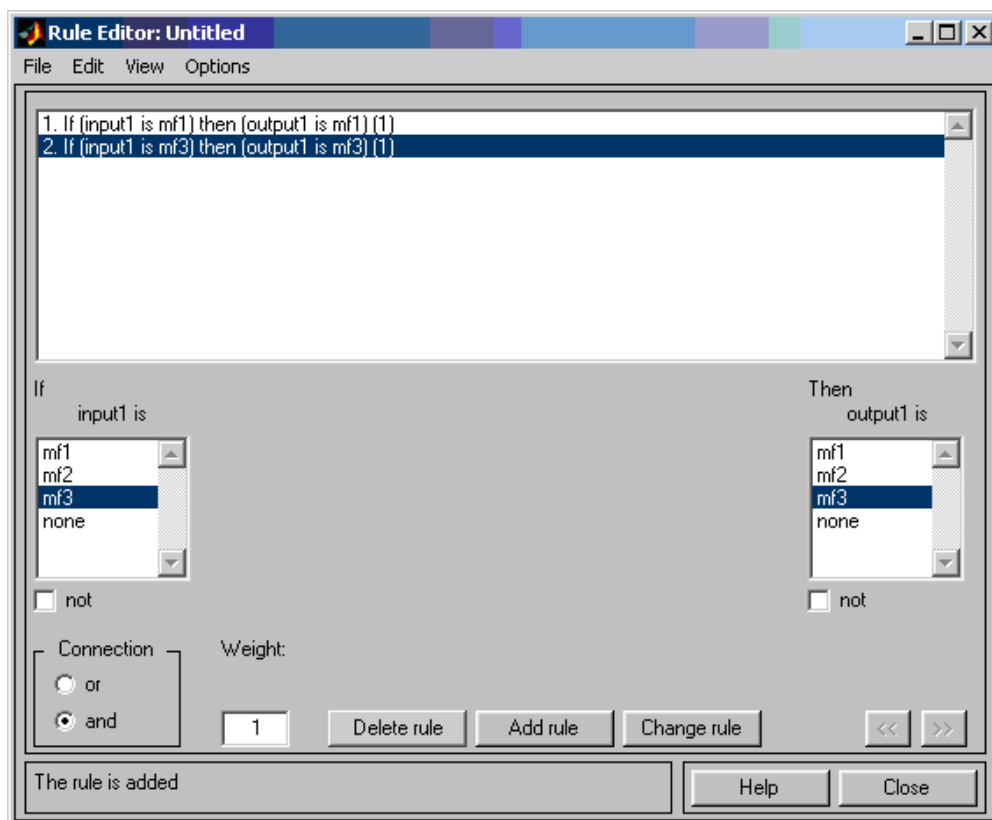


Рисунок 3 - Графический интерфейс Rule Editor

Удаляемое или изменяемое правило предварительно должно быть выделено. При изменении или добавлении правила набор нужных значений ЛП, для описания априорных высказываний, производится выделением их из списков, размещённых в полях нижней части окна редактора.

Тип связки для сложных высказываний устанавливается переключателем Connection.

Величина весового коэффициента правила устанавливается в поле ввода Weight.

Перечисленные операции над правилами выполняются после щелчка по соответствующей клавише внизу окна.

3 Порядок работы

1. Построить модель системы автоматического управления согласно рис. 4 и 5.

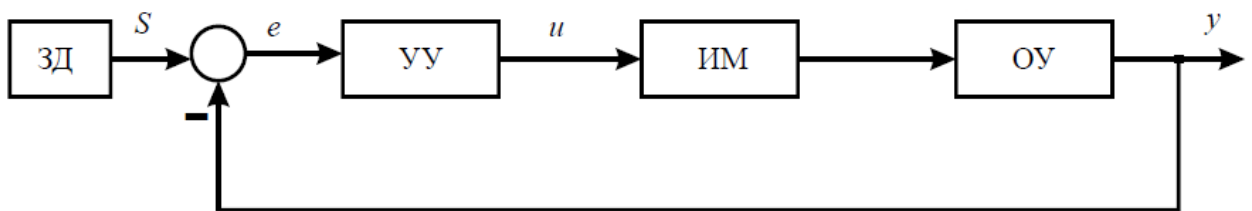


Рисунок 4 - Одноконтурная система автоматического регулирования:

ОУ – объект управления; ИМ – исполнительный механизм;

УУ – устройство управления; ЗД - задатчик

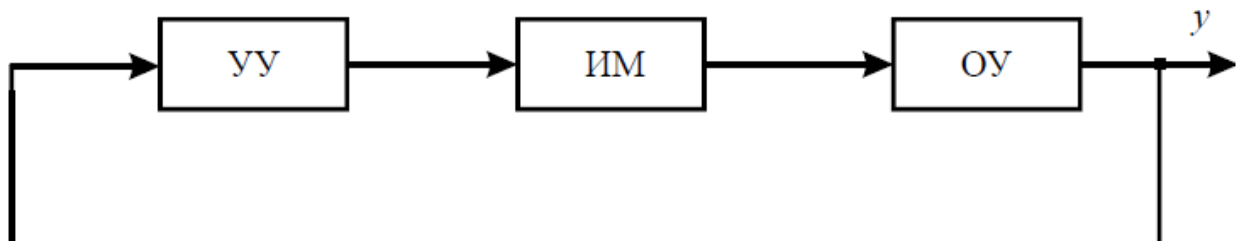


Рисунок 5 - Система автоматического регулирования

2. Передаточные функции УУ и ИМ взять из таблицы 1 согласно своему варианту (№ п/п посписку группы).

3. В качестве устройства управления использовать нечеткий регулятор. Для каждой из структур синтезировать функции принадлежности входа и выхода нечеткого регулятора.

4. Разработать базу правил нечетких регуляторов.

5. Построить модель одноконтурной системы автоматического регулирования (рис. 4), используя в качестве УУ двухпозиционный закон регулирования вида:

$$u(e) = \begin{cases} k, & \text{если } e > 0, \\ -k, & \text{если } e < 0. \end{cases}$$

* параметр k выбрать самостоятельно.

6. Построить переходные характеристики в полученных системах автоматического регулирования и оценить его качество.

7. По найденным прямым оценкам качества переходных процессов сравнить синтезированные системы автоматического регулирования.

Таблица 1 - Варианты заданий

№ п/п	ОУ	ИМ	S
1	$W_{Oy}(p) = \frac{2}{20p + 1}$	$W_{ИМ}(p) = \frac{1}{10p}$	2
2	$W_{Oy}(p) = \frac{1}{40p + 1}$	$W_{ИМ}(p) = \frac{1}{20p}$	1
3	$W_{Oy}(p) = \frac{6}{80p + 1}$	$W_{ИМ}(p) = \frac{1}{5p}$	2
4	$W_{Oy}(p) = \frac{8}{67p + 1}$	$W_{ИМ}(p) = \frac{1}{40p}$	1
5	$W_{Oy}(p) = \frac{2}{20p + 1}$	$W_{ИМ}(p) = \frac{1}{50p}$	2
6	$W_{Oy}(p) = \frac{2}{20p + 1}$	$W_{ИМ}(p) = \frac{1}{30p}$	2
7	$W_{Oy}(p) = \frac{3}{120p + 1}$	$W_{ИМ}(p) = \frac{1}{20p}$	2
8	$W_{Oy}(p) = \frac{6}{43p + 1}$	$W_{ИМ}(p) = \frac{1}{50p}$	2
9	$W_{Oy}(p) = \frac{0.6}{20p + 1}$	$W_{ИМ}(p) = \frac{1}{10p}$	2
10	$W_{Oy}(p) = \frac{1}{50p + 1}$	$W_{ИМ}(p) = \frac{1}{20p}$	2
11	$W_{Oy}(p) = \frac{5}{120p + 1}$	$W_{ИМ}(p) = \frac{1}{40p}$	2
12	$W_{Oy}(p) = \frac{2}{70p + 1}$	$W_{ИМ}(p) = \frac{1}{50p}$	2
13	$W_{Oy}(p) = \frac{1}{90p + 1}$	$W_{ИМ}(p) = \frac{1}{20p}$	2
14	$W_{Oy}(p) = \frac{3}{29p + 1}$	$W_{ИМ}(p) = \frac{1}{10p}$	2
15	$W_{Oy}(p) = \frac{8}{40p + 1}$	$W_{ИМ}(p) = \frac{1}{30p}$	2
16	$W_{Oy}(p) = \frac{2}{55p + 1}$	$W_{ИМ}(p) = \frac{1}{40p}$	2
17	$W_{Oy}(p) = \frac{1}{32p + 1}$	$W_{ИМ}(p) = \frac{1}{20p}$	2
18	$W_{Oy}(p) = \frac{7}{134p + 1}$	$W_{ИМ}(p) = \frac{1}{10p}$	2

Контрольные вопросы:

1. Что такое лингвистическая переменная?
2. Что такое фаззификация и на чем она основывается?
3. В каком редакторе пакета Fuzzy Logic Toolbox выполняется фаззификация ЛП?
4. В каком редакторе пакета Fuzzy Logic Toolbox формируется база знаний СНВ?

5. В чем вы видите преимущества регулятора нечеткой логики, для регулирования какими объектами его целесообразно применять?

Содержание отчета:

1. Титульный лист по ГОСТ ТПУ.
2. Цель работы.
3. Номер варианта;
4. Исходные данные;
5. Структурную схему и порядок синтеза моделей;
6. Результаты моделирования по п.1-6 с пошаговыми пояснениями, скриншотами схем из Simulink;
7. Ответы на контрольные вопросы.
8. Выводы по работе.

Лабораторная работа № 5

Построение систем логического управления и защиты в Simulink

1 Цель работы

Получить навыки и способности синтеза дискретных систем и систем технологических и аварийных защит в ПО Matlab Simulink.

2 Порядок работы

1. Построить в редакторе Simulink модель системы логического управления, согласно рис. 1.

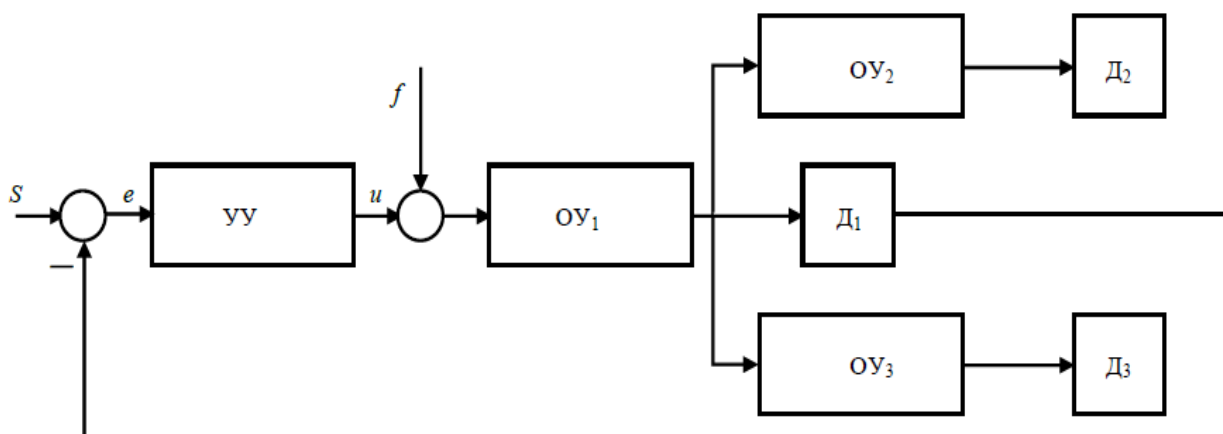


Рисунок 1 - Структурная схема системы автоматического управления:

УУ - устройство управления; ОУ – объект управления;

Д – датчик; f – сигнал возмущения; S – сигнал задания;

e – ошибка регулирования; u – управляющее воздействие

$$u(e) = \begin{cases} k, & \text{если } e > 0, \\ -k, & \text{если } e < 0. \end{cases}$$

2. Реализовать логическую защиту технологического объекта по параметрам, получаемых с Д₁, Д₂, Д₃ согласно своего варианта, табл.1 (№ п/п по списку группы).

3. Построить переходные процессы в системе:

3.1. при нормальной работе ($f = 0$ или $f < f_{кр}$);

3.2. при аварийной ситуации, вызванной влиянием возмущающего воздействия $f \geq f_{кр}$.

Уровень возмущения $f_{кр}$ и предельные значения параметров, получаемых с Д₁, Д₂, Д₃, выбрать самостоятельно.

Таблица 1 - Варианты заданий

№ п/п	OY_1	OY_2	OY_3	S	Схема защиты	k
1	$W_{OY1}(p) = \frac{2}{20p+1}$	$W_{OY2}(p) = \frac{5}{10p+1}$	$W_{OY3}(p) = \frac{4}{15p+1}$	2	"2 из 3"	1,5
2	$W_{OY1}(p) = \frac{1}{40p+1}$	$W_{OY2}(p) = \frac{3}{13p+1}$	$W_{OY3}(p) = \frac{5}{20p+1}$	1	"1 из 3"	5
3	$W_{OY1}(p) = \frac{6}{80p+1}$	$W_{OY2}(p) = \frac{15}{17p+1}$	$W_{OY3}(p) = \frac{3}{22p+1}$	2	"2 из 2" (D_2, D_3)	0,5
4	$W_{OY1}(p) = \frac{8}{67p+1}$	$W_{OY2}(p) = \frac{5}{26p+1}$	$W_{OY3}(p) = \frac{4}{10p+1}$	1	"3 из 3"	0,3
5	$W_{OY1}(p) = \frac{7}{90p+1}$	$W_{OY2}(p) = \frac{4}{10p+1}$	$W_{OY3}(p) = \frac{3}{20p+1}$	2	"2 из 3"	0,5
6	$W_{OY1}(p) = \frac{2}{95p+1}$	$W_{OY2}(p) = \frac{15}{12p+1}$	$W_{OY3}(p) = \frac{6}{10p+1}$	2	"1 из 3"	1,5
7	$W_{OY1}(p) = \frac{3}{20p+1}$	$W_{OY2}(p) = \frac{4}{10p+1}$	$W_{OY3}(p) = \frac{5}{15p+1}$	2	"3 из 3"	1,5
8	$W_{OY1}(p) = \frac{8}{150p+1}$	$W_{OY2}(p) = \frac{3}{10p+1}$	$W_{OY3}(p) = \frac{4}{15p+1}$	2	"2 из 3"	0,3
9	$W_{OY1}(p) = \frac{1}{80p+1}$	$W_{OY2}(p) = \frac{4}{10p+1}$	$W_{OY3}(p) = \frac{19}{12p+1}$	2	"2 из 2" (D_2, D_3)	5
10	$W_{OY1}(p) = \frac{2}{200p+1}$	$W_{OY2}(p) = \frac{25}{40p+1}$	$W_{OY3}(p) = \frac{4}{55p+1}$	2	"2 из 3"	1,5
11	$W_{OY1}(p) = \frac{2}{20p+1}$	$W_{OY2}(p) = \frac{5}{10p+1}$	$W_{OY3}(p) = \frac{4}{15p+1}$	1	"3 из 3"	0,3
12	$W_{OY1}(p) = \frac{1}{40p+1}$	$W_{OY2}(p) = \frac{3}{13p+1}$	$W_{OY3}(p) = \frac{5}{20p+1}$	1	"2 из 3"	1,5
13	$W_{OY1}(p) = \frac{6}{80p+1}$	$W_{OY2}(p) = \frac{15}{17p+1}$	$W_{OY3}(p) = \frac{3}{22p+1}$	2	"1 из 3"	5
14	$W_{OY1}(p) = \frac{8}{67p+1}$	$W_{OY2}(p) = \frac{5}{26p+1}$	$W_{OY3}(p) = \frac{4}{10p+1}$	1	"1 из 3"	1,5
15	$W_{OY1}(p) = \frac{7}{90p+1}$	$W_{OY2}(p) = \frac{4}{10p+1}$	$W_{OY3}(p) = \frac{3}{20p+1}$	2	"3 из 3"	1,5
16	$W_{OY1}(p) = \frac{2}{95p+1}$	$W_{OY2}(p) = \frac{15}{12p+1}$	$W_{OY3}(p) = \frac{6}{10p+1}$	2	"2 из 3"	5

Продолжение Таблицы 1

№ п/п	OY_1	OY_2	OY_3	S	Схема защиты	k
17	$W_{OY1}(p) = \frac{3}{20p + 1}$	$W_{OY2}(p) = \frac{4}{10p + 1}$	$W_{OY3}(p) = \frac{5}{15p + 1}$	2	“2 из 2” (Д2, Д3)	0,3
18	$W_{OY1}(p) = \frac{8}{150p + 1}$	$W_{OY2}(p) = \frac{3}{10p + 1}$	$W_{OY3}(p) = \frac{4}{15p + 1}$	1	“2 из 2” (Д1, Д3)	1,5
19	$W_{OY1}(p) = \frac{1}{80p + 1}$	$W_{OY2}(p) = \frac{4}{10p + 1}$	$W_{OY3}(p) = \frac{19}{12p + 1}$	2	“2 из 3”	0,5
20	$W_{OY1}(p) = \frac{2}{200p + 1}$	$W_{OY2}(p) = \frac{25}{40p + 1}$	$W_{OY3}(p) = \frac{4}{55p + 1}$	1	“2 из 3” (Д2, Д3)	0,5

Контрольные вопросы:

1. Назначение технологических защит?
2. Перечислите основные отличия системы ТЗиБ от АСУ ТП?
3. Какие возможны результаты воздействия системы ТЗиБ на защищаемое оборудование?
4. Какую выбрать структуру защиты для уменьшения вероятности ложного срабатывания, если датчики не обладают достаточной надежностью?
5. Какую выбрать структуру защиты для повышения надежности ее действия и отсутствия отказов в работе?

Содержание отчета:

1. Титульный лист по ГОСТ ТПУ.
2. Цель работы.
3. Номер варианта;
4. Исходные данные;
5. Структурную схему и порядок синтеза моделей;
6. Результаты моделирования по п.1-3 с пошаговыми пояснениями, скриншотами схем из Simulink;
7. Ответы на контрольные вопросы.
8. Выводы по работе.

Литература

1. www.exponenta.ru
2. Бирюков И.И. Теория автоматов. Часть 1. Логическое проектирование комбинационных схем: Учебное пособие. – М. – 2010. – 62 с.

Учебное издание

Захаревич Юлия Сергеевна
Захаревич Аркадий Воадимирович

ЛОГИЧЕСКОЕ УПРАВЛЕНИЕ И ТЕХНОЛОГИЧЕСКИЕ ЗАЩИТЫ

Методические указания к лабораторным работам

Издано в авторской редакции