

Министерство образования и науки
Российской Федерации
ФГОБОУ ВО «Российский химико-технологический университет
имени Д.И. Менделеева»

Новомосковский институт (филиал)

Основы моделирования в SimInTech

Методическое пособие

Новомосковск
2018

УДК 681.5
ББК 32.965
Т 338

Рецензент:
канд. тех. наук Соболев А.В.
(НИ (филиал) ФГБОУ ВО РХТУ им. Д.И. Менделеева)

Составители: Ляшенко А.И., Маслова Н.В., Вент Д.П.

Т 338 «Основы моделирования в SimInTech. Методическое пособие» / ФГБОУ ВО Российский химико-технологический университет им. Д.И. Менделеева, Новомосковский институт (филиал), Сост.: Ляшенко А.И., Вент Д.П., Маслова Н.В. Новомосковск, 2018. - 42 с.

В пособии представлены основные сведения и указания для работы в моделирующей программе SimInTech, необходимые для выполнения учебных задач по моделированию при изучении различных дисциплин в области кибернетики, теории автоматического управления. Методическое пособие предназначено для студентов профиля 15.03.04 «Автоматизация технологических процессов и производств», а также будет интересным студентам родственных профилей и направлений, аспирантам и инженерам в области систем управления и автоматизации.

Табл. 11 Ил. 16. Библиогр.: 2 назв.

УДК 681.5

© ФГБОУ ВО Российский химико-технологический университет им. Д.И. Менделеева,
Новомосковский институт (филиал), 2018

ОГЛАВЛЕНИЕ

Введение.....	4
1 Назначение программы SimInTech.....	5
2 Установка программы	5
3 Интерфейс пользователя SimInTech.....	5
4 Основные приемы создания расчетных схем в SimInTech.....	9
5 Основные приемы работы с графиками.....	16
6 Основы программирования в SimInTech.....	20
Библиографический список.....	29
Приложение Основные функциональные блоки SimInTech.....	30

Введение

Целью методического пособия является формирование представлений о программном комплексе SimInTech, ознакомление с основными приемами разработки и исследования схем автоматики и управления с использованием имитационного моделирования.

Пособие состоит из шести разделов, в которых вкратце описывается назначение программы SimInTech, последовательность установки программы на компьютер, описаны основные элементы интерфейса программы, даны рекомендации по созданию структурных схем и по работе с графиками. В шестом разделе представлены основы программирования в SimInTech, описаны способы деклараций переменных, представлены основные конструкции.

Предлагаемое методическое пособие предназначено для студентов профиля 15.03.04 «Автоматизация технологических процессов и производств» и смежных профилей.

1 Назначение программы SimInTech

Программа SimInTech («Simulation in technik») - российская система модельно-ориентированного проектирования систем автоматического управления [1-2].

Программа предназначена для исследования нестационарных процессов в различных объектах управления. Разработка математических моделей и алгоритмов управления происходит в виде структурного проектирования логико-динамических систем, описываемых с помощью отношений «вход-выход», в виде систем обыкновенных дифференциальных уравнений и/или дифференциально-алгебраических уравнений.

По своим возможностям SimInTech является отечественной альтернативой таким программным продуктам, как SimuLink (MathWorks), VisSim (Visual Solution), LabView (National Instruments) и т.д. Преимуществами SimInTech является то, что полноценная учебная версия распространяется бесплатно и полностью русифицирована, имеет открытый интерфейс для интеграции с различным ПО, присутствуют инструменты создания интерфейсов управления, имеется возможность моделирования в режиме «реального времени». При этом программа обладает высокой гибкостью и скоростью решения сложных математических задач, обширной справочной системой поддержки.

2 Установка программы

Чтобы установить программу **SimInTech** на своем компьютере, надо зайти на сайт <http://simintech.ru>. В верхнем меню выбрать пункт **Загрузить** → **Русская версия**. В зависимости от установленной операционной системы (Windows XP или Windows 7 и выше, 32 или 64 бита) заполнить нужную форму, указав необходимые данные, помеченные значком *. Затем нажать **Запросить ссылку**. На указанную в форме электронную почту придет письмо со ссылкой для скачивания, действительной в течение 12 часов. Для установки программы скаченный файл следует запустить. Инструкцию по установке можно найти в разделе сайта **Материалы** → **Инструкции** → **Инструкция по установке и регистрации SimInTech**. Регистрацию установленной программы выполнять не обязательно.

3 Интерфейс пользователя SimInTech

Запуск среды **SimInTech** осуществляется одним из следующих способов:

- с помощью ярлыка на рабочем столе или на панели быстрого запуска;

- с помощью стартового меню «Пуск» (Windows XP/7/10);
- с помощью ярлыка на экране приложений (Windows 8/8.1).

После запуска приложения на экране появится **Главное окно SimInTech** (рисунок 1).

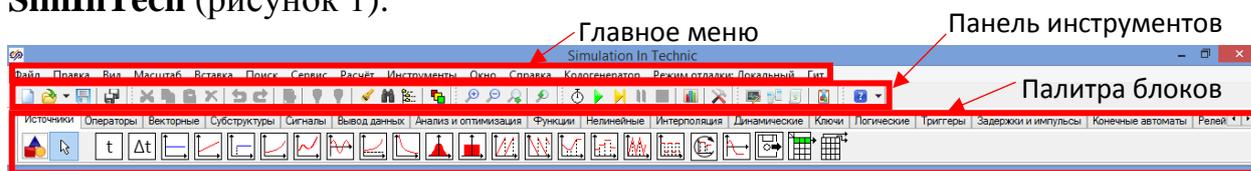


Рисунок 1 – Главное окно SimInTech

Главное меню Главного окна (*Файл, Правка, Вид* и т.д.) отвечает за настройку программы, позволяет редактировать, управлять созданием и расчетом собранных схем, содержит инструменты работы с базой данных.

Панель инструментов Главного окна (*Новый проект, Открыть, Сохранить проект* и т.д.) содержит набор кнопок, отвечающих за наиболее часто используемые команды. Часть кнопок дублирует команды Главного меню и предназначена для ускорения работы с программой.

Палитра блоков Главного окна (*Источники, Операторы* и т.д.) содержит набор библиотек блоков для создания расчетных схем и моделей. «Линейка» типовых блоков состоит из отдельных каталогов, сгруппированных по функциональному признаку. Переключение между ними осуществляется 1-кратным щелчком левой клавиши мыши в поле «закладки» с соответствующим названием. К основным библиотекам относятся: **Источники** входных воздействий; **Операторы** математические; **Векторные** операторы; **Субструктуры**; **Сигналы**; **Вывод данных**; **Анализ и оптимизация**; **Функции** математические; **Нелинейные**; **Динамические**; **Ключи**; **Логические**; **Дискретные**. В Приложении представлено краткое описание основных блоков различных библиотек SimInTech.

Создание и изменение структурной схемы производится при помощи **Схемного окна**. Редактор структурных схем способен загружать несколько независимых проектов. Каждый проект располагается на своем отдельном **Схемном окне**. Для того чтобы создать расчетную схему необходимо (см. рисунок 2):

- либо выбрать с помощью **Главного меню Главного окна**: «**Файл** → **Новый проект** → **Схема модели общего вида**»;
- либо воспользоваться кнопкой **Новый проект** на **Панели инструментов Главного окна**.

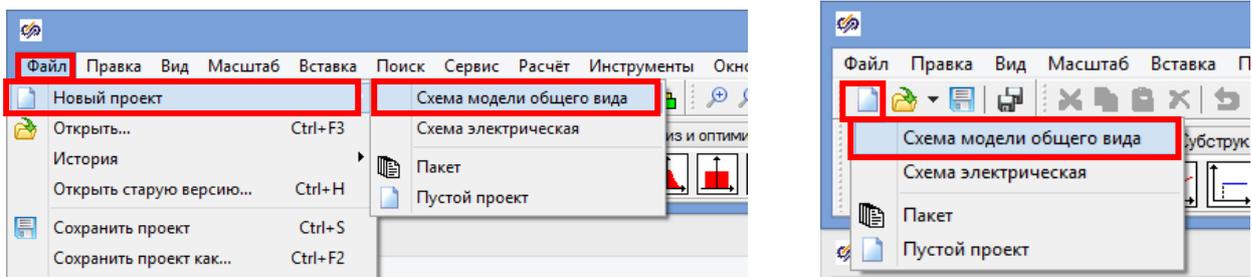


Рисунок 2 – Способы создания нового проекта

После выполнения одного из вышеупомянутых способов на экране появится новое **Схемное окно** (рисунок 3).

Меню окна проекта отвечает за основные функции работы со схемой, за настройку визуального отображения схемы.

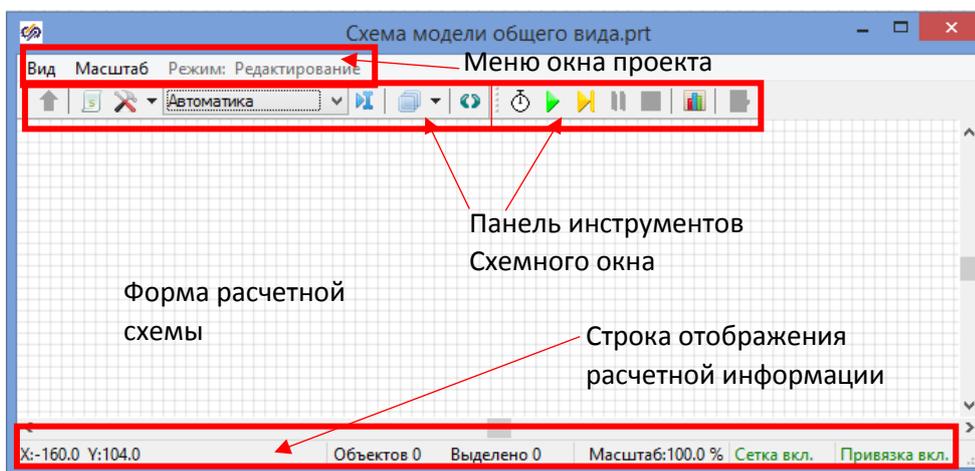


Рисунок 3 - Схемное окно проекта

Панель инструментов Схемного окна содержит следующие основные объекты:

-  **скрипт** – кнопка доступа к редактору скрипта;
-  **параметры расчета** – кнопка доступа к окну изменения расчетных параметров схемы;
-  **инициализация** – инициализирует расчетную схему, но не запускает ее расчет;
-  **пуск** – инициализирует расчетную схему и сразу запускает ее расчет;
-  **сделать шаг** – задача делает один шаг с величиной, заданной пользователем в расчетных параметрах схемы;
-  **пауза** – приостанавливает расчет;
-  **стоп** – останавливает расчет;
-  **менеджер данных** - доступ к окну редактора базы сигналов проекта.

Структурная схема SimInTech строится из блоков, линий связи, являющихся активными элементами схемы, и декоративных элементов (комментариев, рисунков, подписей), не участвующих в расчёте.

Блок - базовый элемент проекта (функциональной диаграммы, расчетной схемы). Основными атрибутами блока являются его графический образ, свойства (задаются пользователем), параметры (вычисляются блоком), математическая модель, входные и выходные порты. В математическом аспекте блоки представляют собой операторы преобразования входных сигналов блока в его выходные сигналы.

Портом называется объект на изображении блока, от которого можно вести линию связи или к которому можно подсоединить линию связи. Порты и линии связи могут быть различных типов. Тип линии связи определяется соответствующим идентификатором и может быть изменён. Линию связи можно подсоединить к порту, только если у них совпадает тип.

Свойство блока - задаваемая пользователем характеристика (константа или переменная величина определённого типа данных) для работы блока.

Параметр блока - формируемая (вычисляемая) блоком переменная, характеризующая работу блока.

Линия связи - служебный блок в виде полилинии, второй базовый элемент расчетной схемы, соединяющий выходной порт одного блока и входной порт другого блока. В общем случае линия связи может соединять множество входных портов с одним выходным портом. В математическом аспекте линии связи являются шинами данных (сигналов) и осуществляют направленную передачу данных от выходов блока к входам других блоков.

Совокупность блоков и соединяющих их порты линий связи образует **алгоритм управления** в схеме автоматике.

Субмодель (макроблок) - блок, математическая модель которого задана в виде структурной схемы, расположенной «внутри» субмодели. Макроблоки позволяют реализовать принцип вложенности структурных схем, являются одним из механизмов создания и сохранения в библиотеке новых типов блоков. Математическая модель макроблока может быть сохранена в отдельном файле (в отдельном проекте) для многократного использования в других проектах.

Проект SimInTech - файл, содержащий структурную схему, предназначенную для расчета тем или иным математическим решателем или расчетным кодом и созданную в окне графического редактора SimInTech,

сохраненный в виде бинарного и/или текстового файла с уникальным именем и расширением «prt» (для бинарного) и «xprt» (для текстового) файла.

Схема автоматки - проект SimInTech, содержащий набор взаимосвязанных алгоритмов управления отдельной подсистемой объекта управления. Может также содержать математические модели объектов управления, реализованные в виде схем типа «вход-выход» или набранные на встроенном языке программирования.

Язык программирования SimInTech - специализированный язык программирования высокого уровня, предназначенный для создания программ, описывающих функционирования типового блока «Язык программирования»; задания глобальных констант и переменных проекта; создания типовых подпрограмм и программ, которые могут производить по мере расчета манипуляции с объектами схемы. Основной отличительной особенностью языка программирования SimInTech является то, что текст программы предназначен для исполнения на каждом расчетном шаге при моделировании схемы.

4 Основные приемы создания расчетных схем в SimInTech

При создании расчетной схемы целесообразно придерживаться следующей последовательности (рекомендуется выполнять процедуру сохранения на жесткий диск после каждого из нижеперечисленных этапов).

Этап 1. Наполнение Схемного окна блоками.

Для размещения того или иного блока нужно перейти на соответствующую вкладку Палитры блоков, 1-кратным щелчком левой кнопкой мыши выбрать интересующий блок (фон станет подсвечен). Далее переместить курсор на Форму расчетной схемы, вид курсора изменится – появится изображение блока и крестообразный указатель места установки блока. Выбираем место установки и 1-кратным щелчком левой кнопки мыши устанавливаем блок на форме.

На рисунке 4 приведен пример размещения на Схемном окне типовых блоков Ступенька (вкладка Источники), Инерционное звено 1-го порядка (вкладка Динамические), Временной график (вкладка Вывод данных).

Чтобы **переместить** блок на схеме для наиболее удобного расположения необходимо нажать левую кнопку мыши на изображении блока, и, не отрывая пальца от кнопки, перетащить блок в нужную точку. После отпускания кнопки мыши блок расположится в том месте, где был курсор.

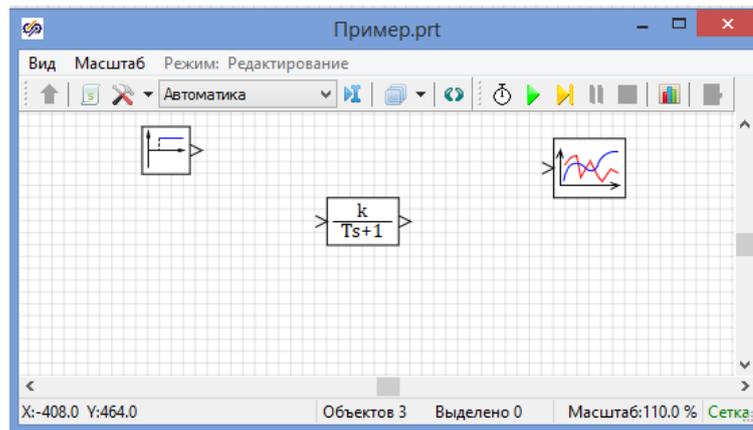
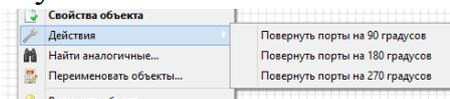


Рисунок 4 – Пример размещения блоков на Схемном окне

Выделение блоков производится следующим образом. Мысленно нужно представить себе прямоугольник на экране, который бы окружал (или хотя бы задевал) только те блоки, которые следует выделить. Далее необходимо нажать левую кнопку мыши в левом верхнем углу этого прямоугольника и, не отпуская ее, начертить этот прямоугольник, проведя мышью с нажатой кнопкой в противоположный, правый нижний, угол. Как только кнопка мыши будет отпущена, блоки окажутся выделенными. Далее с группой блоков можно оперировать как с единым блоком. Например, ее можно перемещать, удалять, копировать и объединять.

Если требуется **развернуть блок** (блоки), необходимо выделить блок (блоки), нажать правую кнопку мыши и в контекстном меню выбрать

«**Действия → Повернуть...**»:



Для **удаления** блоков следует сначала выделить, как описано выше, блок (или блоки) и нажать на клавишу «Delete» клавиатуры.

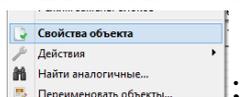
Если на схеме будут находиться несколько групп блоков, похожих по структуре и (или) составу (повторяющиеся элементы схемы), то значительно упростить процесс ее создания поможет **копирование** блоков. Для копирования необходимо выделить копируемую группу блоков и выбрать команду главного меню «**Правка → Копировать**». Группа блоков скопируется в буфер Windows. Далее выбрать команду «**Правка → Вставить**» и мышью указать место, куда будет помещена скопированная группа. Щелчок мыши располагает группу на схеме.

Этап 2. Ввод свойств блоков.

Двигаясь в **Схемном Окне** по направлению «слева-направо» и «сверху-вниз», необходимо задать свойства блоков (коэффициенты усиления, постоянные времени, начальные условия и т.д.).

Вызвать окно редактора свойств можно одним из следующих способов:

1. Выделив блок 1-кратным кликом правой кнопки мыши, вызвать контекстное меню, в котором выбрать пункт Свойства объекта

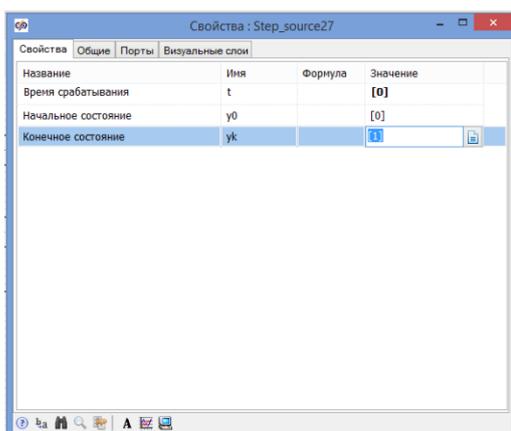


2. Выделив блок, нажать на Панели кнопок кнопку Свойства .
3. Выделив блок щелчком мыши на схеме, а затем произвести по нему двойной щелчок левой кнопкой мыши.

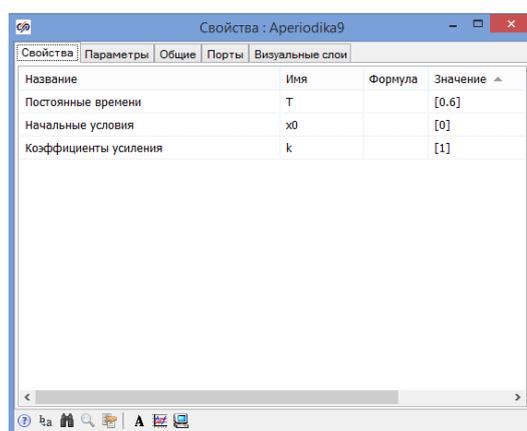
В окне свойств блока наиболее часто используемыми вкладками являются:

- Свойства – расчетные свойства блока, непосредственно влияющие на его исполнение;
- Общие – базовые свойства блока, такие как имя, тип, координаты расположения на листе, видимость и другие;
- Порты – список и редактор параметров входных/выходных портов блока.

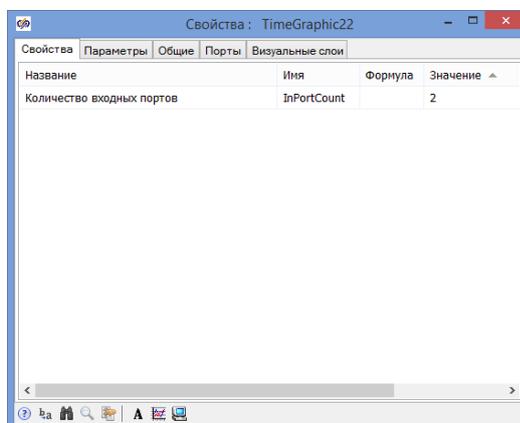
На рисунке 5 приведен пример задания свойств блоков Ступенька, Инерционное звено 1-го порядка, Временной график.



(а)



(б)



(в)

Рисунок 5 - Окно свойств блока, на примере блоков Ступенька (а), Инерционное звено 1-го порядка (б), Временной график (в)

Этап 3. Соединение блоков линиями связи.

Создание линий связи производится левой кнопкой мыши. Чтобы соединить два расположенных на схеме блока необходимо подвести курсор к выходному порту одного блока, сделать 1-кратный щелчок левой кнопкой мыши и, отпустив кнопку (курсор превратится в стрелку вида \uparrow), протянуть линию связи к входному порту другого блока. Как только курсор достигнет входа, нужно снова сделать 1-кратный щелчок левой кнопкой мыши, после чего на входе появится типичная входная стрелка. Если сделать щелчок левой клавишей раньше, чем проводимая связь войдет «в притяжение» входного порта, необходимо дотянуть линию связи до соответствующего входного порта и сделайте щелчок левой кнопкой мыши.

Если требуется сделать поворот на 90 градусов в линии связи, при «протаскивании» необходимо выполнить щелчок левой кнопкой мыши и продолжать проведение линии связи в новом направлении. Если нужно прервать процедуру проведения линии связи, нажмите правой кнопкой мыши в пустое место **Схемного Окна**: линия оборвется (завершится её создание).

Чтобы сделать «ответвление» от линии связи, необходимо навести курсор на линию, нажать на правую кнопку мыши и в появившемся меню выбрать пункт «**Действия** → **Добавить ветвь**», появится новая точка на линии и возможность провести ответвление от данной точки.

Для удаления линий связи необходимо навести курсор на линию, нажать левую кнопку мыши (выделить линию) и нажать «Delete».

На рисунке 6 показано, как выходной порт Ступеньки соединен с 1-ым входным портом Временного графика и одновременно через механизм «ответвления» с входным портом Инерционного звена 1-го порядка, выходной порт которого соединен в свою очередь со 2-ым входным портом Временного графика.

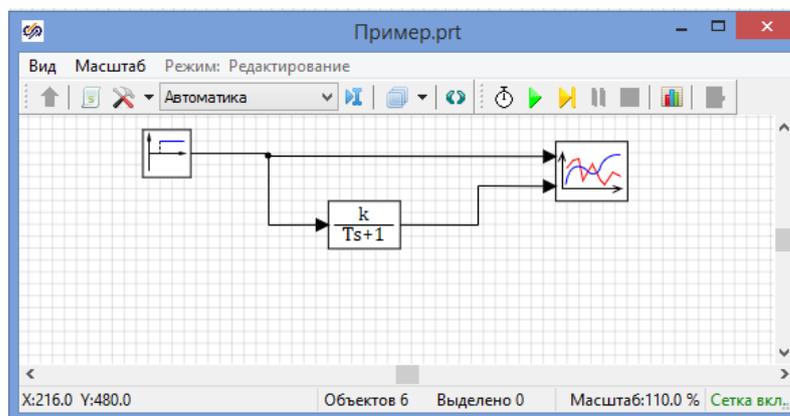


Рисунок 6 – Пример подключения линий связи структурной схемы

Этап 4. Оформление поясняющих подписей.

Для оформления **Схемного Окна** используются подписи. Для добавления подписи к блоку, переместите курсор под блок, найдите положение курсора, при котором он изменит свою форму на стрелку со знаком вопроса и сделайте 2-х кратный щелчок левой кнопкой мыши. Должно появиться временное окно для ввода текста. Переместите курсор в это окно, сделайте щелчок левой кнопкой мыши, после чего введите нужный заголовок (для переноса текста на следующую строку используйте сочетание **ctrl+enter**). Далее переместите курсор на свободное место в **Схемном Окне** и сделайте 1-кратный щелчок левой кнопкой мыши, временное окно закроется и под блоком появится желаемая подпись (рисунок 7).

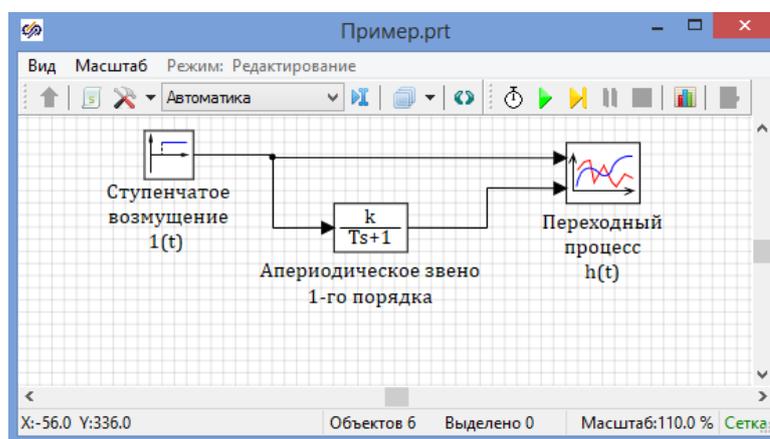
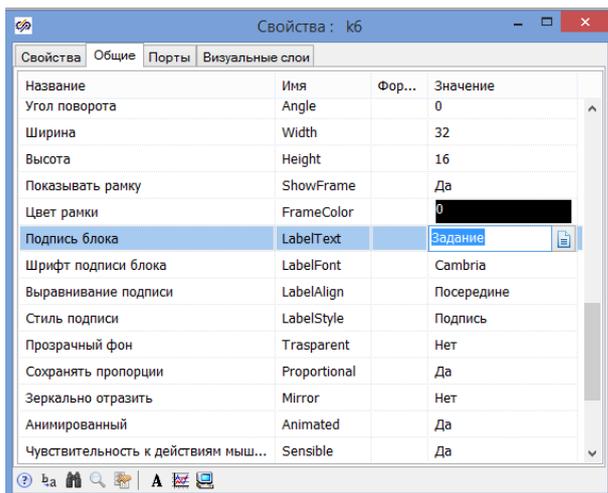
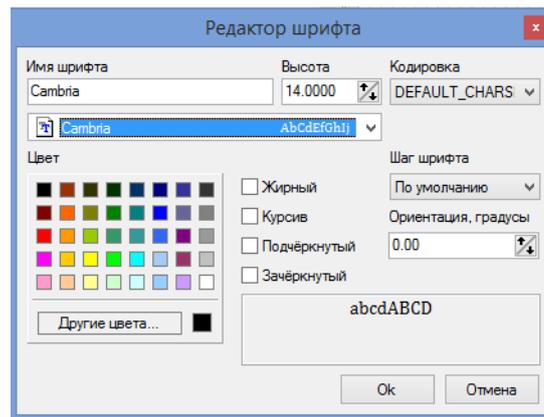


Рисунок 7 – Пример оформленной структурной схемы

Также выполнить подпись блока можно следующим способом. Переместите курсор на нужный блок, сделайте 1-кратный щелчок правой кнопкой мыши и в контекстном меню выберите пункт **Свойства объекта**. Во вкладке **Общие** в пункте *Подпись блока* записывается соответствующий заголовок (рисунок 8а). В пункте *Выравнивание подписи* можно указать требуемый способ выравнивания текста. Выбрав пункт *Шрифт подписи блока*, откроется диалоговое окно Редактор шрифта (рисунок 8б), в котором можно установить шрифт, начертание, цвет размер. Для сохранения изменений необходимо нажать кнопку **Ок**.



(a)

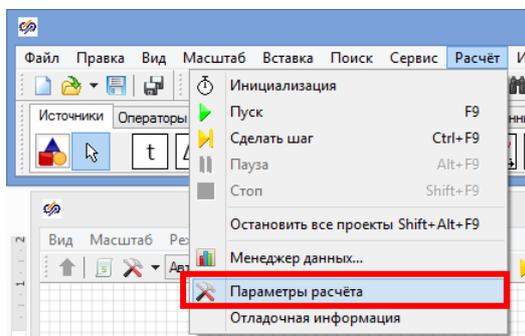


(б)

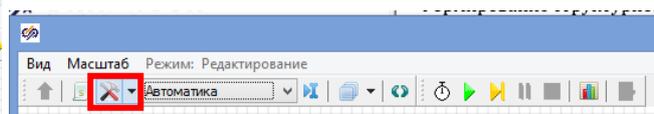
Рисунок 8 – Задание подписи в Свойствах блока

Этап 5. Установка параметров расчета схемы.

Чтобы настроить расчетные параметры схемы в Главном меню необходимо выбрать «Расчет → Параметры расчета» (рисунок 9а), либо нажать кнопку Параметры расчета на Панели инструментов Схемного окна (рисунок 9б).



(a)



(б)

Рисунок 9 – Способы открытия окна настройки параметров расчета схемы

После откроется диалоговое окно с активной закладкой Параметры расчета (рисунок 10), где необходимо выбрать *Метод интегрирования*, установить *Начальный шаг интегрирования* $startstep=0$, *Конечное время расчета* $endtime$ (выбирается из соображения того, чтобы переходные процессы успевали завершиться), *Максимальный шаг* $hmax$ достаточно принять равным 0.1, *Минимальный шаг* $hmin=0.001$, *Относительную ошибку* $relerr=0.001$, остальные параметры оставить по умолчанию или настраивать под конкретные задачи.

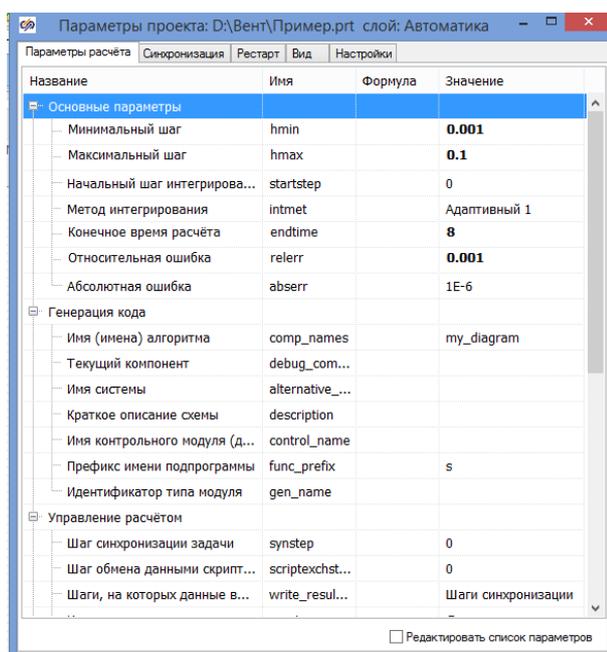


Рисунок 10 – Окно настройки параметров расчета схемы

Закладка **Синхронизация** предназначена для расчета в заданном масштабе времени. Если выбрать пункт *Синхронизировать с реальным временем* и в поле *Коэффициент ускорения* указать 1, то это будет соответствовать расчету в реальном масштабе времени.

Закладка **Рестарт** предназначена для периодического (например, через 1 секунду) сохранения в бинарном формате основных данных расчета, по которым можно продолжить процесс моделирования после завершения расчета;

Закладка **Вид** диалогового окна Параметры проекта предназначена для установки внешнего вида **Схемного Окна** по умолчанию.

Закладка **Настройки** предназначена для дополнительных настроек (связь с базой данных, с плагином решателя схемы и т.д.).

Этап 6. Запуск процесса моделирования.

После того, как структурная схема собрана, заданы свойства всех блоков и введены параметры интегрирования, она может быть запущена на счет. Для этого необходимо нажать на кнопку **Пуск**  на **Панели инструментов** или на клавишу F9. В результате работы схемы выходные сигналы блоков начнут изменяться, их величины будут отображаться на графиках и других индикаторах. Параметры некоторых сигналов и блоков можно изменять в процессе симуляции, другие параметры можно изменить, приостановив процесс работы схемы, нажав кнопку . В строке отображения расчетной информации **Схемного окна** должна появиться надпись: «Конечное время достигнуто...», в противном случае в схеме была допущена ошибка.

5 Основные приемы работы с графиками

Блок **Временной график** (вкладка Палитры блоков: **Вывод данных** → **Временной график**) реализует отображение результатов моделирования в виде временных зависимостей для одной или нескольких переменных.

Число отображаемых в графическом окне сигналов задается в свойствах блока. Для этого необходимо переместить курсор на блок, сделать 1-кратный щелчок правой кнопкой мыши и в контекстном меню блока выбрать пункт **Свойств объекта**. Во вкладке Свойства в пункте *Количество входных портов* InPortCount указывается необходимое число входных сигналов (рисунок 11).

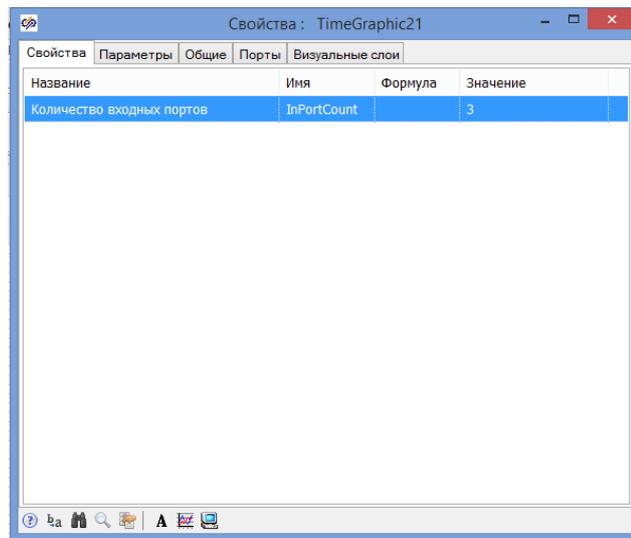


Рисунок 11 – Окно свойств блока Временной график

При 2-х кратном щелчке левой кнопкой мыши по изображению блока в Схемном окне открывается графическое окно (рисунок 12), которому можно придать необходимый размер.

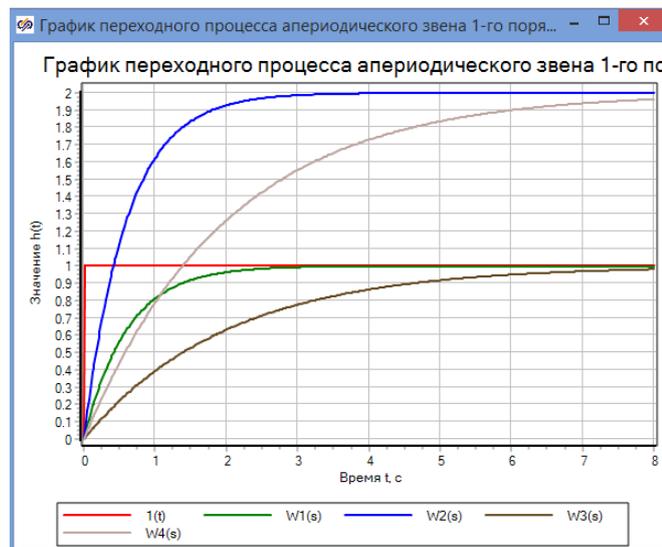


Рисунок 12 – Окно свойств блока Временной график

Чтобы увеличить фрагмент графика необходимо навести курсор в нужную часть графика, сделать двойной щелчок левой кнопкой мыши и, удерживая кнопку нажатой, выполнить движение мышью по направлению «вправо-вверх» или «вправо-вниз». Для восстановления исходного вида графика достаточно повторить ту же процедуру, только движение мышью выполнить по направлению «влево-вверх» или «влево-вниз» с нажатой левой кнопкой мыши. Перемещение видимой части графика производится перемещением мыши с нажатой правой кнопкой мыши.

Если переместить курсор в центральную часть окна графика и сделать 1-кратный щелчок правой кнопкой мыши, должно появиться контекстное меню (рисунок 13).

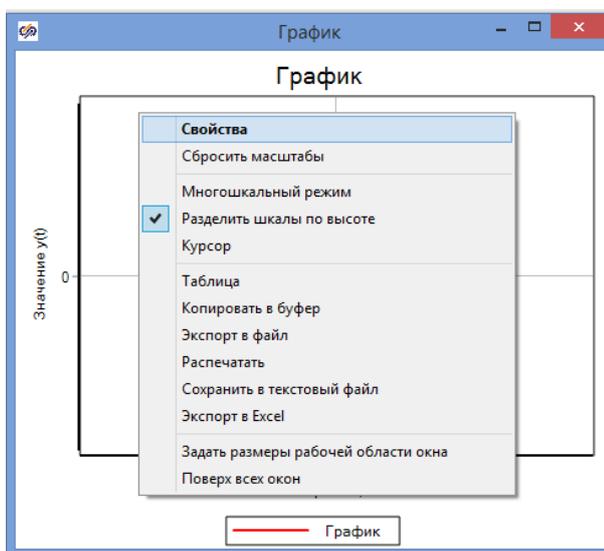


Рисунок 13 - Диалоговое окно настройки графика

Пункт *Копировать в буфер* в контекстном меню позволяет скопировать в буфер обмена изображение графика для последующей вставки его в соответствующие отчетные документы, например, в текстовые документы Word, в графический редактор Paint или в САПР AutoCAD.

Пункт *Распечатать* в контекстном меню позволяет сразу вывести изображение графика на печать, используя доступные принтеры.

Пункт *Таблица* в контекстном меню позволяет сменить способ отображения временной зависимости на табличную, что эффективно, когда требуется высокая оценка точности результатов моделирования (рисунок 14).

№	1(t) - Y	W1(s) - X	W1(s) - Y	W2(s) - X	W2(s) - Y	W3(s) - X	W3(s) - Y
1	0	0	0	0	0	0	0
2	1	0.01367	0.0010062641	0.01367	0.00025158856	0.01367	0.00047353277
3	1	0.052224556	0.013469824	0.052224556	0.0033721144	0.052224556	0.0053586121
4		0.063115507	0.019225971	0.063115507	0.0048180946	0.063115507	0.0074395914
5		0.072924156	0.025134021	0.072924156	0.0063048525	0.072924156	0.0094656328
6		0.12928049	0.069994351	0.12928049	0.01769033	0.12928049	0.022735205
7		0.13783759	0.07813051	0.13783759	0.019775303	0.13783759	0.024889278
8		0.1450102	0.085165268	0.1450102	0.021583373	0.1450102	0.026707686
9		0.19746684	0.1413822	0.19746684	0.036226425	0.19746684	0.040165018
10		0.21084506	0.15674782	0.21084506	0.040294322	0.21084506	0.043600775
11		0.22305929	0.17103382	0.22305929	0.04410408	0.22305929	0.046728654
12		0.29171231	0.25403771	0.29171231	0.066835405	0.29171231	0.064046672
13		0.30822069	0.27428422	0.30822069	0.072553044	0.30822069	0.068130455
14		0.32143574	0.29047103	0.32143574	0.077180303	0.32143574	0.07137447
15		0.33889147	0.31177503	0.33889147	0.083350175	0.33889147	0.075624658
16		0.36942147	0.34866208	0.36942147	0.094265403	0.36942147	0.082961729
17		0.38364099	0.36562038	0.38364099	0.099390446	0.38364099	0.086336982
18		0.39511734	0.37918109	0.39511734	0.1035409	0.39511734	0.089041585
19		0.42755207	0.41590184	0.42755207	0.11531802	0.42755207	0.096501787

Рисунок 14 – Представление временной зависимости в виде таблицы

Если выбрать пункт контекстного меню - *Экспорт в Excel*, то автоматически загрузится таблица MS Excel уже с записанными временными зависимостями SimInTech, которые можно использовать для дальнейших исследований.

Если выбрать пункт *Курсор* контекстного меню, то появится перекрестие, которое можно переместить на точку, координаты которой следует вычислить. Координаты будут отображены в специальном поле (рисунок 15). Чтобы убрать перекрестие, закройте диалоговое окно *Ближайшие к курсору точки*.



Рисунок 15 – Детализация графика

Если выбрать пункт *Свойства* контекстного меню, то откроется диалоговое окно с заголовком *Свойства графика* (рисунок 16), в котором задаются настройки параметров Графического окна, такие как цвет и тип

линий, параметры осей, заголовков графика и подписи по осям координат и т.д.

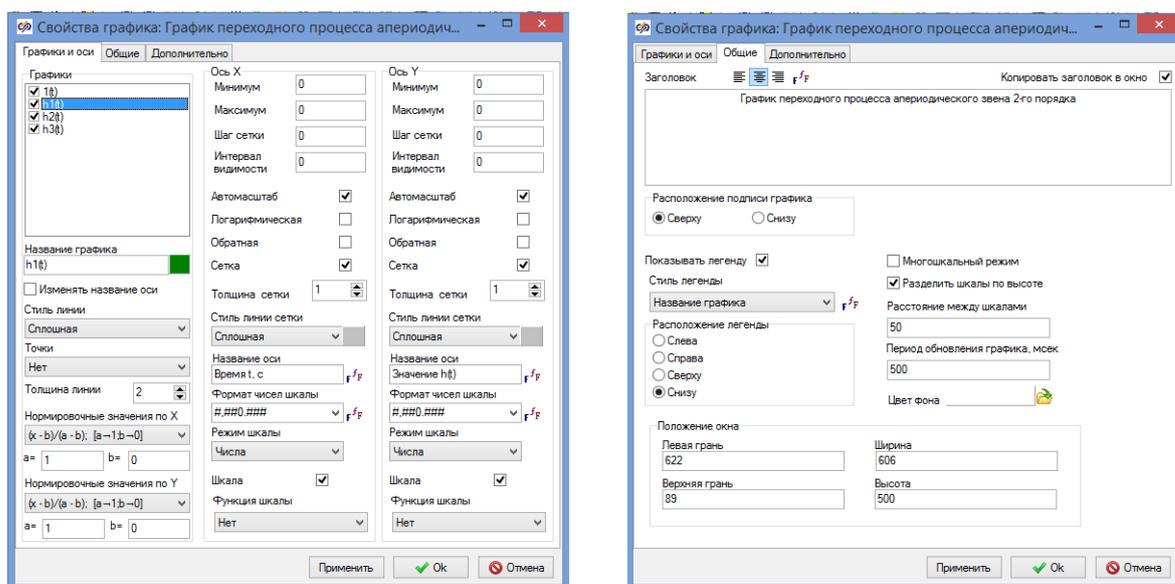


Рисунок 16 – Окно настройки свойств Временного графика

Наиболее часто используемые возможности редактирования:

- *Название графика* - наименование временных зависимостей;
- *Стиль линии* - стиль линии, которой отображается выбранная временная зависимость;
- *Толщина линии* - толщина в пикселях линии выбранного графика;
- *Заголовок* - наименование графика (вверху), можно задать параметры шрифта;
- *Название оси* - строка текста, отображаемого около Оси X и Оси Y;
- *Автомасштаб* - возможность автомасштабирования графика (по умолчанию включен);
- *Период обновления графика* – при исследовании длительных процессов его нужно увеличить, при исследовании быстрых процессов этот параметр лучше уменьшить;
- *Многошкальный режим* - возможность вывода разных шкал для отображения разных величин.

Закончив процедуру редактирования свойств графика, необходимо нажать на кнопку **Ок** и закрыть диалоговое окно.

6 Основы программирования в SimInTech

С помощью языка программирования можно написать программу (скрипт), которая будет выполняться в процессе расчета определенные действия с объектами схемы при наступлении некоторого события. Скрипт можно использовать для задания глобальных или локальных констант и переменных проекта и для описания функционирования типового блока из палитры Динамические – «Язык программирования». Скрипт присутствует на каждом уровне иерархии субмоделей, и для каждой субмодели может быть записан свой скрипт.

Скрипт состоит из **деклараций** и **операторов**. Разделителем деклараций и операторов является точка с запятой «;». В декларациях обязательно должны быть описаны (инициализированы) входные, выходные и динамические переменные. Переменную или константу можно использовать только после того, как она описана в декларации или задана в операторе присваивания. Имена констант, переменных, меток, функций выступают **идентификаторами** и могут содержать буквы латинского, русского алфавита, знак подчеркивания «_» и цифры. Идентификатор должен начинаться с буквы или знака подчеркивания. Прописные и строчные буквы в идентификаторах по умолчанию не различаются. Далее в описании символами <...> обозначены обязательные идентификаторы, а символами {...} – дополнительные идентификаторы. Декларации должны начинаться с **ключевых слов**:

- **const** – константы;

Синтаксис:

```
const <имя переменной 1>{:<тип переменной 1>}{=<иницирующее выражение 1>}{<имя переменной 2>{:<тип переменной 2>}{=<иницирующее выражение 2>}};
```

Описание:

Если начальное значение не задано, то оно принимается равным 0. Если константа потом переопределена, то она будет воспринята как переменная. Константы отличаются от переменных тем, что они рассчитываются только один раз в начале моделирования.

Пример:

```
const a=1e-3, // инициализированная константа вещественного типа  
b=[a, 1, 5.6], // инициализированный массив размерностью 3  
M=[b, [2.5, 1, 3.4]], // инициализированная матрица размерностью 3 на 3  
max_iter: integer=20, // инициализированная константа целого типа
```

```
err: double=1e-4, // инициализированная константа вещественного типа
_I=2+1.5e-5i; // константа комплексного типа
```

- **var** – алгебраические переменные;

Синтаксис:

```
var <имя переменной 1>{:<тип переменной 1>}{=<иницирующее выражение 1>}{,<имя переменной 2>{:<тип переменной 2>}{=<иницирующее выражение 2>}};
```

Описание:

Тип переменных по умолчанию – действительное (вещественное) число. Начальное значение переменной по умолчанию 0. Для декларации действительных массивов и матриц нужно пользоваться конструкцией: X[n] – массив размерностью n; M[n,m] – матрица размерностью n на m.

Пример:

```
var n:double=0.5, // инициализированная переменная вещественного типа
i:integer, // переменная целого типа
k=[2, 3, 6], // инициализированный массив размерностью 3
M[2, 2], // матрица размерностью 2 на 2
M[[1, 2], [3, 4]]; // инициализированная матрица размерностью 2 на 2
```

- **init** – динамические (дифференциальные) переменные;

Синтаксис:

```
init <имя переменной 1>{:<тип переменной 1>}{=<иницирующее выражение 1>}{,<имя переменной 2>{:<тип переменной 2>}{=<иницирующее выражение 2>}};
```

Описание:

Декларации **init** позволяют создавать две переменные одинакового типа. Если x[i] – массив переменных состояния, то x'[i] – производная переменной x[i] этого массива. Переменные состояния позволяют задавать дифференциальные уравнения в форме Коши, т.е. в виде $x'=f(x)$.

Пример:

```
output y1[2], y2[2];
init x1=5,x2=5,x3=5,x4=5.1,x5=5,x6=5;
```

```

a=10; b=28; c=2.6666666; // Ввод коэффициентов и дифференциальных уравнений
x1'=a*(-x1+x2); // Моделирование при н.у. в точке M1
x2'=b*x1-x2-x1*x3;
x3'=-c*x3+x1*x2;
x4'=a*(-x4+x5); // Моделирование при н.у. в точке M2
x5'=b*x4-x5-x4*x6;
x6'=-c*x6+x4*x5;
y1[1]=x1; y1[2]=x4; // Описание выходов
y2[1]=x3; y2[2]=x6;

```

- **input** – входные переменные блока;

Синтаксис:

```

input <имя переменной 1>{:<тип переменной 1>}{=<иницирующее выражение 1>}{,<имя переменной 2>{:<тип переменной 2>}{=<иницирующее выражение 2>}};

```

Описание:

Значения из входов блока будут передаваться в эти переменные. Для массивов и матриц обязательно указываются размерности в квадратных скобках.

Пример:

```

input x1=2, x2=4; // инициализированные входные переменные вещественного типа
var y; // переменная вещественного типа
y=2*x1+(x2-1);

```

В результате выполнения примера переменной блока *y* будет присвоено значение 7.

- **output** – выходные переменные блока;

Синтаксис:

```

output <имя переменной 1>{:<тип переменной 1>}{=<иницирующее выражение 1>}{,<имя переменной 2>{:<тип переменной 2>}{=<иницирующее выражение 2>}};

```

Описание:

Значения этих переменных передаются на выходы блока. Для массивов и матриц обязательно указываются размерности в квадратных скобках.

Пример:

```
var a=2, b=3, // инициализированные переменные вещественного типа
M[2];        // массив размерностью 2
output y1,   // выходная переменная вещественного типа
y2[2];       // выходная переменная массив размерностью 2
M[1]=3;      // инициализация 1-го элемента массива M
M[2]=4;      // инициализация 2-го элемента массива M
y1=a+b^2;
y2[1]=M[2];
```

В результате выполнения примера выходной переменной блока `y1` будет присвоено значение выражения $a+b^2=11$, а выходная переменная `y2` будет представлять собой массив `[4, 0]`, значение 1-го элемента которого равно значению 2-го элемента массива `M`.

- **function** – пользовательские функции;

Синтаксис:

```
function <Имя функции>( <имя входной переменной 1> {:<тип входной переменной 1>} {=<инициирующее выражение 1>} {, out<имя выходной переменной> {:<тип выходной переменной>} {=инициирующее выражение>}}) {:<тип выходного значения>}
<операции выполняемые при вызове функции>
end;
```

Описание:

Задание новой пользовательской функции. Значение функции может быть не присвоено. Выходные переменные декларируются словом `out` – в такую переменную функция записывает значение. Если тип значения функции не указан, то он определяется автоматически по типу выражения, где присваивается выходное значение функции. Если типы параметров функции не указаны, то при создании кода функции типы принимаются как у операндов при соответствующем вызове. При стандартном вызове функции, ее внутренние переменные для всех вызовов будут размещаться в памяти в одном месте. Если необходимо строго разделить использование внутренних переменных функции, тогда можно использовать опциональный модификатор вызова «`inline`», который позволяет каждый раз выделять свои

собственные локальные переменные и не допускать их пересечения для разных вызовов функции.

Пример 1:

```
function PolyF(A, B)
PolyF = A*A + B*B
end;
a = PolyF(2, 5) + 50;      // стандартный вызов функции
b=PolyF(3, 4)|inline + 100; // вызов функции со встраиванием ее кода в место вызова
```

- **if** – условный переход;

Синтаксис:

```
if <условие или логическое выражение>
then <оператор при выполнении условия>
{else<оператор при невыполнении условия>}
```

Описание:

Выполнение операций в зависимости от условия. Условием должно быть логическое или целочисленное выражение. При достижении условия выполняется оператор после ключевого слова **then**, в противном случае выполняется оператор после ключевого слова **else**.

Пример:

```
input x1, x2;      // инициализированные переменные вещественного типа
if x1>x2 then y=x1 else y=x2; // условие
output y;
```

В результате выполнения примера выходной переменной блока будет максимальное значение из двух входных переменных.

- **begin** – операторные скобки;

Синтаксис:

```
begin
<операция 1>;
....
```

```
<операция n>;  
end;
```

Описание:

Операторные скобки объединяют несколько операторов в один. Операторные скобки могут быть вложенными.

Пример:

```
input x1:boolean=1, x2:boolean=0; // инициализированные переменные двоичного типа  
var y, u; // переменные вещественного типа  
if x1=true and x2=true then // условие в виде логического И  
begin  
y=2;  
u=5;  
end  
else begin  
y=0;  
u=0;  
end;
```

В результате выполнения примера $y=0$, $u=0$, т.к. не было выполнено условие в виде логического И.

- **for** – конечный цикл;

Синтаксис:

```
for (<оператор присваивания переменной цикла>, <конечное значение> {, <шаг переменной цикла>}) <оператор цикла>;
```

Описание:

Выполнение конечного цикла по указанной переменной. Операция выполняется, пока значение переменной цикла не достигнет конечного значения. На каждом шаге значение переменной изменяется на величину, равную указанному шагу. Если шаг не указан, то он равен 1 или -1 в зависимости от того, больше конечное значение начального или меньше.

Пример 1:

```
var j=0; // инициализированная переменная вещественного типа
```

```
for (i=1, 10) // условие цикла
j=j+i^2; // оператор цикла
```

В результате выполнения примера переменной *i* будет присвоено значение 11, переменной *j* присвоено значение 385.

Пример 2:

```
var k=0, n=0; // инициализированные переменные вещественного типа
for (i=0.1, 2, 0.2) // условие цикла
begin
k=k+i*2; // операторы цикла
n=n+i^2;
end;
```

В результате выполнения примера переменной *i* будет присвоено значение 2.1, переменной *k* присвоено значение 20, переменной *n* присвоено значение 13.3.

- **while** – цикл с предусловием;

Синтаксис:

```
while <условие цикла> do <оператор цикла>;
```

Описание:

Выполнение операции пока выполняется условие цикла.

Пример:

```
var i = 1; // инициализированная переменная вещественного типа
while i<= 10 do // условие цикла
begin
i = i + 1; // оператор цикла
end;
```

В результате выполнения примера переменной *i* будет присвоено значение 11.

- **repeat** – цикл с постусловием;

Синтаксис:

```
repeat  
<оператор 1>;  
...  
<оператор 2>;  
until <условие выхода>;
```

Описание:

Выполнение операторов, заключенных между `repeat` и `until`, до тех пор, пока не будет выполнено условие выхода из цикла.

Пример:

```
var i=1; // инициализированная переменная вещественного типа  
repeat  
i= i+1; // оператор цикла  
until i=10; // условие цикла
```

В результате выполнения примера переменной `i` будет присвоено значение 10.

- **goto** – безусловный переход;

Синтаксис:

```
goto <имя метки>  
<имя метки>:<операторы назначения>;
```

Описание:

Безусловный переход к операции, помеченной меткой. Безусловные переходы могут быть осуществлены только в пределах одного программного блока (функции или секции алгоритма).

Пример:

```
var i=1;  
Label_1: i=i+1;  
if i<10 then goto Label_1;
```

В результате выполнения примера переменной `i` будет присвоено значение 10.

- **exit** – выход из текущего программного блока;

Синтаксис:

```
exit;
```

Описание:

Выход из текущего программного блока (процедуры или страницы кода). Если **exit** внутри функции, то происходит выход из функции. Если **exit** внутри субмодели, то завершается выполнение скрипта на данном шаге субмодели.

Пример:

```
if time > 5 then exitapp;
```

В результате выполнения примера при достижении модельного времени 5 секунд приложение SimInTech завершит свою работу.

Помимо деклараций и операторов скрипт может содержать комментарии, заключенные в фигурные скобки:

```
{текст комментария}
```

либо в виде:

```
//текст комментария до конца строки.
```

Полный перечень доступных для языка программирования SimInTech системных переменных, возможных операторов и функций можно посмотреть при необходимости непосредственно из программы, выбрав с помощью **Главного меню Главного окна: «Справка → Содержание»** или нажав кнопку F1 (раздел Язык программирования).

Библиографический список

1. Среда динамического моделирования технических систем SimInTech. Свидетельство о регистрации программ для ЭВМ №2010617758 от 23.11.2010 г.

2. Среда динамического моделирования технических систем SimInTech: практикум по моделированию систем автоматического регулирования / Б.А. Карташов, Е.А. Шабаев, О.С. Козлов, А.М. Щекатуров. Москва: Изд-во ДМК Пресс, 2017. 424 с.

Приложение

Таблица П.1 – Основные блоки палитры Источники

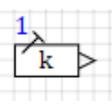
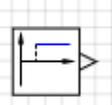
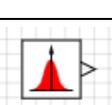
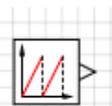
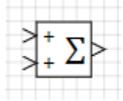
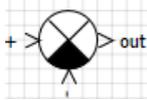
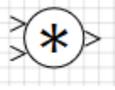
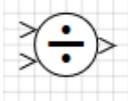
Источники		
В палитре	На схеме	Описание
		<p>Блок Часы формирует на выходе сигнал, численно равный текущему значению модельного времени:</p> $y(t) = t$ <p>где: $y(t)$ – выходной скалярный сигнал. Единица измерения – секунды.</p>
		<p>Блок Константа формирует на выходе заданную пользователем постоянную величину:</p> $y(t) = k$ <p>Для работы блока необходимо задать величину константы k (по умолчанию $k=1$).</p>
		<p>Блок Линейный источник формирует значение выходного сигнала, вычисленное по формуле:</p> $y(t) = a + b \cdot t$ <p>где: a, b – коэффициенты, $y(t)$ – выходной сигнал, t – текущее модельное время в секундах.</p>
		<p>Блок Ступенька формирует значение выходного сигнала, вычисленное по алгоритму:</p> $y(t) = \begin{cases} y_0 & t \leq t_0 \\ y_1 & t > t_0 \end{cases}$ <p>где: t_0 – время срабатывания, y_0 – начальное значение, y_1 – конечное значение, t – текущее модельное время в секундах.</p>
		<p>Блок Синусоида формирует синусоидальный выходной сигнал:</p> $y(t) = a \cdot \sin(\omega \cdot t + f)$ <p>где: a – амплитуда, ω – круговая частота, $1/c$; f – фаза, рад; t – текущее модельное время в секундах.</p>
		<p>Блок Нормальный шум генерирует псевдослучайную последовательность вещественных чисел, нормально распределенных по Гауссу.</p>
		<p>Блок Пила формирует на выходе периодический линейно нарастающий пилообразный сигнал:</p> $y(t) = \text{frac}\left(\frac{t}{T}\right) y_A + y_0$ <p>где: y_A, T, y_0 – свойства блока, $y(t)$ – выходной сигнал, t – текущее модельное время в секундах, $\text{frac}(t/T)$ – дробная часть.</p>

Таблица П.2 – Основные блоки палитры Операторы

Операторы		
В палитре	На схеме	Описание
		<p>Блок Сумматор реализует операцию алгебраического поэлементного суммирования входных сигналов с учетом весовых коэффициентов:</p> $y(t) = a_1 \cdot u_1(t) + a_2 \cdot u_2(t) + a_3 \cdot u_3(t) + \dots + a_n \cdot u_n(t)$ <p>где: $y(t)$ – выходной сигнал блока; $u_1(t), u_2(t), \dots, u_n(t)$ – входные сигналы; $a_1, a_2, a_3, \dots, a_n$ – весовые коэффициенты для каждого входного сигнала блока.</p>
		<p>Блок Сравнивающее устройство реализует (по умолчанию) поэлементное вычитание второго входного векторного сигнала из первого:</p> $\vec{y}(t) = \vec{u}_1(t) - \vec{u}_2(t)$ <p>где: $y(t)$ – выходной сигнал блока; $u_1(t), u_2(t)$ – входные сигналы.</p>
		<p>Блок Перемножитель реализует умножение входных сигналов (для скалярных сигналов на входе):</p> $y(t) = u_1(t) \cdot u_2(t) \cdot \dots \cdot u_n(t)$ <p>где: $y(t)$ – выходной скалярный сигнал; $u_1(t), u_2(t)$ – 1-ый, 2-ой и т.д. входной сигнал; или поэлементное умножение входных величин (для векторных сигналов на входе):</p> $y_i(t) = u_{1i}(t) \cdot u_{2i}(t) \cdot \dots \cdot u_{ni}(t)$ <p>где: $y_i(t)$ – i-ый элемент выходного векторного сигнала; $u_{1i}(t), u_{2i}(t)$ – i-ый элемент 1-го, 2-го и т.д. входного сигнала.</p>
		<p>Блок Делитель реализует деление входных сигналов (для скалярных сигналов на входе):</p> $y(t) = \frac{u_1(t)}{u_2(t)}$ <p>где: $y(t)$ – выходной скалярный сигнал; $u_1(t), u_2(t)$ – входные сигналы; или поэлементное деление входных величин (для векторных сигналов на входе):</p> $y_i(t) = \frac{u_{1i}(t)}{u_{2i}(t)}, i = 1 \dots n$ <p>где: $y_i(t)$ – i-ый элемент выходного векторного сигнала; $u_{1i}(t), u_{2i}(t)$ – i-ые элементы 1-го и 2-го входных сигналов, n – размерность входов и выхода. Обязательным условием является неравенство делителя нулю ($u_{2i}(t) \neq 0$)!</p>

Окончание таблицы П.2

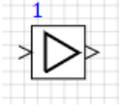
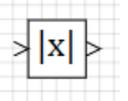
Операторы		
В палитре	На схеме	Описание
		<p>Блок Усилитель производит умножение (усиление) входной величины на заданный коэффициент усиления k (для скалярного сигнала на входе):</p> $y(t) = k \cdot u(t)$ <p>или поэлементное умножение входного векторного сигнала на заданный коэффициент усиления k:</p> $\vec{y}(t) = k \cdot \vec{u}(t) \Leftrightarrow y_i(t) = k \cdot u_i(t), i = 1 \dots n$ <p>где: $u_i(t)$ – i-ый элемент входного векторного сигнала, $y_i(t)$ – элемент выходного векторного сигнала, n – размерность входа и выхода.</p>
		<p>Блок Абсолютное значение реализует вычисление модуля входной величины. При скалярном входном сигнале:</p> $y(t) = u(t) $ <p>где: $u(t), y(t)$ – входной и выходной сигналы. При векторном входном сигнале:</p> $y_i(t) = u_i(t) , i = 1 \dots n, n \geq 2$ <p>где: $u_i(t), y_i(t)$ – i-ые элементы входного и выходного векторов.</p>

Таблица П.3 – Основные блоки палитры Векторные

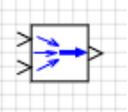
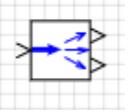
Векторные		
В палитре	На схеме	Описание
		<p>Блок Мультиплексор реализует «сжатие» нескольких скалярных или векторных входных сигналов в один векторный выходной сигнал типа «шина» данных. Размерность выходного вектора равна сумме размерностей входов.</p>
		<p>Блок Демультимплексор реализует «расщепление» векторного входного сигнала в отдельные выходные сигналы, которые могут быть как скалярными, так и векторными. По умолчанию блок реализует демультимплексирование 2-х «жильного» входного векторного сигнала в скалярные сигналы:</p> $y_1(t) = u_{1,1}(t)$ $y_2(t) = u_{1,2}(t)$ <p>где: $y_1(t), y_2(t)$ – 1-ый и 2-ой выходной скалярный сигнал блока; $u_{1,1}(t), u_{1,2}(t)$ – 1-й и 2-й элементы входного векторного сигнала.</p>
		<p>Блок Динамическая выборка выполняет процедуру выбора элемента из вектора по указанному номеру по алгоритму:</p> <pre> Цикл $i = 1 \dots n$ Если $u_1(t) = i$ то $y(t) = u_{2,i}(t)$; Выход из цикла Конец цикла </pre> <p>где: $u_1(t)$ – 1-ый входной скалярный сигнал, определяющий номер выбираемого элемента, начиная с 1; $u_{2,i}(t)$ – i-ый элемент 2-го входного векторного сигнала $u_2(t)$, из которого осуществляется выборка; $n \geq 1$ – размерность 2-го входа.</p>

Таблица П.4 – Основные блоки палитры Субструктуры

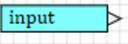
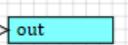
Субструктуры		
В палитре	На схеме	Описание
		Блок Субмодель формирует «оболочку» для создания вложенных схем как из типовых блоков, так и из Субмоделей более «глубокого» уровня вложенности.
		Блок Порт входа используется внутри субмодели и определяет наличие у неё входного порта. Блок организует прием данных от типовых блоков и субмоделей, внешних по отношению к данной субмодели, а также передачу данных для преобразования типовыми блоками (или субмоделями) внутренней структуры данной субмодели.
		Блок Порт выхода используется внутри субмодели и определяет наличие у неё выходного порта. Блок организует прием данных от типовых блоков и субмоделей внутренней структуры данной субмодели, а также передачу данных для преобразования типовыми блоками (или субмоделями), внешними по отношению к данной субмодели.

Таблица П.5 – Основные блоки палитры Вывод данных

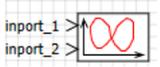
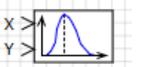
Вывод данных		
В палитре	На схеме	Описание
		Блок Временной график реализует отображение текущих результатов моделирования в виде временных зависимостей для одной или одновременно для нескольких переменных.
		Блок Фазовый портрет реализует отображение текущих результатов моделирования на фазовой плоскости для одной пары или одновременно для нескольких пар фазовых переменных. Блок имеет два векторизованных входа: для каждой точки значение абсциссы берётся с первого входа, значение ординаты – со второго входа.
		Блок График Y от X реализует вывод на график зависимости вектора Y (второй вход) от вектора X (первый вход). Блок предназначен для отображения текущих результатов моделирования в виде графика $Y(t, x)$, реализующего после каждого шага интегрирования построение кривой $y(t) = f(x, t)$, где: x – вектор, содержащий независимые от времени постоянные элементы.

Таблица П.6 – Основные блоки палитры Функции

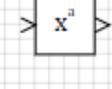
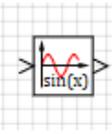
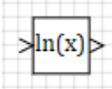
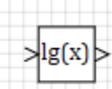
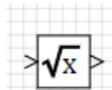
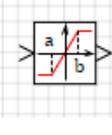
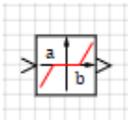
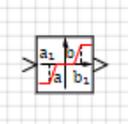
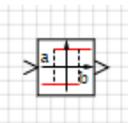
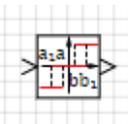
Функции		
В палитре	На схеме	Описание
		Блок Степенная функция возвращает выходное значение, вычисленное по формуле: $y(t) = [x(t)]^a$ где: a – показатель степени, $x(t)$, $y(t)$ – входной и выходной сигналы.
		Блок Синусоидальная функция возвращает выходное значение, вычисленное по формуле: $y(t) = a \cdot \sin(\omega \cdot x(t) + f)$ где: a , ω , f – амплитуда, круговая частота (рад/с) и фаза (рад), $x(t)$, $y(t)$ – входной и выходной сигналы.
		Блок Логарифм натуральный возвращает выходное значение, вычисленное по формуле: $y(t) = a \cdot \ln(\omega \cdot x(t) + f)$ где: a , ω , f – амплитуда, круговая частота и фаза, $x(t)$, $y(t)$ – входной и выходной сигналы.
		Блок возвращает выходное значение, вычисленное по формуле: $y(t) = a \cdot \lg(\omega \cdot x(t) + f)$ где: a , ω , f – амплитуда, круговая частота и фаза, $x(t)$, $y(t)$ – входной и выходной сигналы.
		Блок Корень квадратный реализует вычисление квадратного корня числа: $y(t) = \sqrt{x(t)}$ где: $x(t)$, $y(t)$ – входной и выходной сигналы. Обязательное условие: $x(t) > 0$.

Таблица П.7 – Основные блоки палитры Нелинейные

Нелинейные		
В палитре	На схеме	Описание
		Блок Линейное с насыщением описывает нелинейность типа «насыщение» по алгоритму: $y(t) = \begin{cases} k \cdot x(t), & \text{если } a < x(t) < b \\ y_1, & \text{если } x(t) \leq a \\ y_2, & \text{если } x(t) \geq b \end{cases}$ где: $x(t)$ – вектор входных сигналов, $y(t)$ – вектор выходных сигналов, $k = (y_2 - y_1) / (b - a)$; a , b , y_1 , y_2 – параметры статической характеристики.

Продолжение таблицы П.7

Нелинейные		
В палитре	На схеме	Описание
		<p>Блок Линейное с зоной нечувствительности описывает нелинейность типа «зона нечувствительности» по алгоритму:</p> $y(t) = \begin{cases} k \cdot [x(t) - a], & \text{если } x(t) < a \\ 0, & \text{если } a \leq x(t) \leq b \\ k \cdot [x(t) - b], & \text{если } x(t) > b \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов, $y(t)$ – вектор выходных сигналов, k – коэффициент усиления; a, b – параметры статической характеристики.</p>
		<p>Блок Линейное с насыщением и зоной нечувствительности описывает нелинейность типа «насыщение с зоной нечувствительности» по алгоритму:</p> $y(t) = \begin{cases} y_1, & \text{если } x(t) \leq a_1 \\ k_1(x(t) - a), & \text{если } a_1 < x(t) < a \\ 0, & \text{если } a \leq x(t) \leq b \\ k_2(x(t) - b), & \text{если } b < x(t) < b_1 \\ y_2, & \text{если } x(t) \geq b_1 \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов; $y(t)$ – вектор выходных сигналов; $k_1 = y_1 / (a_1 - a)$; $k_2 = y_2 / (b_1 - b)$; a, b, a_1, b_1, y_1, y_2 – параметры статической характеристики.</p>
		<p>Блок Релейное неоднозначное (гистерезис) описывает нелинейность типа «реле с гистерезисом» по алгоритму:</p> $y(t) = \begin{cases} y_1, & \text{если } x(t) < a \\ y(t - \Delta t), & \text{если } a \leq x(t) \leq b \\ y_2, & \text{если } x(t) > b \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов, $y(t)$ – вектор выходных сигналов, Δt – предыдущий временной шаг интегрирования; a, b, y_1, y_2 – параметры статической характеристики.</p>
		<p>Блок Релейное с зоной нечувствительности описывает нелинейность типа «реле с зоной нечувствительности и гистерезисом» по алгоритму:</p> $y(t) = \begin{cases} y_1, & \text{если } x(t) < a \\ 0, & \text{если } a < x(t) < b \\ y(t - \Delta t), & \text{если } a_1 \leq x(t) \leq a, \text{ или если } b \leq x(t) \leq b_1 \\ y_2, & \text{если } x(t) > b_1 \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов; $y(t)$ – вектор выходных сигналов; Δt – предыдущий временной шаг интегрирования; a_1, a, b, b_1, y_1, y_2 – параметры статической характеристики.</p>

Окончание таблицы П.7

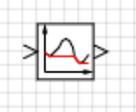
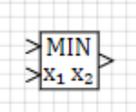
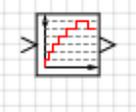
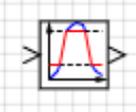
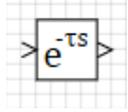
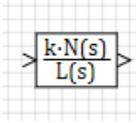
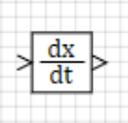
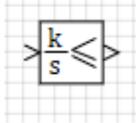
Нелинейные		
В палитре	На схеме	Описание
		<p>Блок Нижний или верхний предел запоминает и выдает на выходной порт экстремальное (минимальное или максимальное) значение входного сигнала (или экстремальные значения в векторе входных сигналов), зафиксированное (-ые) в процессе моделирования на данный момент времени. Алгоритм блока при вычислении максимума:</p> $y_i(t) = \begin{cases} x_i(t), & \text{если } x_i(t) > y_i(t - \Delta t) \\ y_i(t - \Delta t), & \text{если } x_i(t) \leq y_i(t - \Delta t) \end{cases}$ <p>Алгоритм блока при вычислении минимума:</p> $y_i(t) = \begin{cases} x_i(t), & \text{если } x_i(t) < y_i(t - \Delta t) \\ y_i(t - \Delta t), & \text{если } x_i(t) \geq y_i(t - \Delta t) \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов, $y(t)$ – вектор выходных сигналов.</p>
		<p>Блок Минимум по входам производит поэлементное вычисление минимального или максимального из текущих входных значений:</p> $y_i(t) = \min [x_{1,i}(t), x_{2,i}(t), \dots] \text{ или } y_i(t) = \max [x_{1,i}(t), x_{2,i}(t), \dots]$ <p>где: i – номер элемента вектора, $x_1(t)$ – 1-ый входной вектор, $x_2(t)$ – 2-ой входной вектор, $y(t)$ – выходной вектор.</p>
		<p>Блок Квантователь по уровню производит квантование сигнала по дискретным уровням с заданным шагом по алгоритму:</p> $y_i(t) = d_i \cdot \text{sign}(x_i(t)) \cdot \text{floor} \left(\left \frac{x_i(t)}{d_i} \right + 0.5 \right)$ <p>где: $x_i(t)$ – i-ый элемент входного сигнала, d_i – i-ый элемент вектора шагов квантования по уровню, floor – функция округления до меньшего целого числа, sign – функция знака, $y_i(t)$ – элемент выходного сигнала из блока.</p>
		<p>Блок Ограничитель реализует ограничение значения входной величины:</p> $y_i(t) = \begin{cases} x_i^{\min}, & \text{если } x_i(t) < x_i^{\min} \\ x_i(t), & \text{если } x_i^{\min} < x_i(t) < x_i^{\max} \\ x_i^{\max}, & \text{если } x_i(t) > x_i^{\max} \end{cases}$ <p>где: $x_i(t)$ – элемент вектора входных сигналов; $y_i(t)$ – элемент вектора выходных сигналов; x_i^{\min}, x_i^{\max} – i-ые элементы векторов минимальных и максимальных значений блока.</p>
		<p>Блок Задержка на шаг интегрирования реализует задержку входной величины на один шаг моделирования:</p> $y_i(t) = x_i(t - \Delta t)$ <p>где: $x_i(t - \Delta t)$ – i-ый элемент входного сигнала на предыдущем шаге моделирования, $y_i(t)$ – элемент выходного сигнала.</p>

Таблица П.8 – Основные блоки палитры Динамические

Динамические		
В палитре	На схеме	Описание
		Блок Язык программирования позволяет реализовать произвольный алгоритм, написанный в окне текстового редактора на встроенном языке программирования.
		Блок Интегратор реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида: $\frac{dy_i(t)}{dt} = k_i x_i(t)$ где: $x_i(t)$ – i -ый элемент входного сигнала в блок, k_i – элемент вектора коэффициентов усиления, $y_i(t)$ – элемент выходного сигнала из блока. При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией: $W(s) = \frac{k}{s}$
		Блок Инерционное звено 1-го порядка реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида: $T_i \frac{dy_i(t)}{dt} + y_i(t) = k_i x_i(t)$ где: $x_i(t)$ – i -ый элемент входного сигнала, T_i – элемент вектора постоянных времени, k_i – элемент вектора коэффициентов усиления, $y_i(t)$ – элемент выходного сигнала. При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией: $W(s) = \frac{k}{Ts + 1}$ Обязательное условие: $T_i \neq 0$.
		Блок Колебательное звено реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида: $T_i^2 \frac{d^2 y_i(t)}{dt^2} + 2 \cdot \beta_i T_i \frac{dy_i(t)}{dt} + y_i(t) = k_i x_i(t)$ где: $x_i(t)$ – i -ый элемент входного сигнала, T_i – элемент вектора постоянных времени, β_i – элемент вектора коэффициентов демпфирования, k_i – элемент вектора коэффициентов усиления, $y_i(t)$ – элемент выходного сигнала. При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией: $W(s) = \frac{k}{T^2 \cdot s^2 + 2 \cdot T \cdot \beta \cdot s + 1}$ Обязательное условие: $T \neq 0$.

Продолжение таблицы П.8

Динамические		
В палитре	На схеме	Описание
		<p>Блок Идеальное транспортное запаздывание реализует математическую модель звена, динамика которого описывается линейным уравнением (ЛДУ) с запаздыванием:</p> $y_i(t) = x_i(t - T_i)$ <p>где: $x_i(t)$ – i-ый элемент входного сигнала, $x_i(t - T_i)$ – значение i-го элемента входного сигнала, запомненное в блоке T_i секунд назад, по отношению к текущему модельному времени; T_i – элемент вектора времён запаздывания, $y_i(t)$ – элемент выходного сигнала. При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией (для скалярного варианта сигналов):</p> $W(s) = e^{-T \cdot s}$
		<p>Блок Передаточная функция общего вида реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:</p> $L(p)y(t) = N(p) \cdot x(t)$ <p>где:</p> $L(p)y(t) = a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t)$ $N(p)x(t) = b_m \frac{d^m x(t)}{dt^m} + b_{m-1} \frac{d^{m-1} x(t)}{dt^{m-1}} + \dots + b_1 \frac{dx(t)}{dt} + b_0 x(t)$ <p>a_i, b_j – коэффициенты ($i=0 \dots n, j=0 \dots m, n \geq m$); $x(t)$ – входной сигнал (может быть векторным, тогда вектора коэффициентов и выход блока тоже будут векторными той же размерности), $y(t)$ – выходной сигнал. При нулевых начальных условиях динамика блока может быть представлена передаточной функцией:</p> $W(s) = \frac{b_m \cdot s^m + \dots + b_2 \cdot s^2 + b_1 \cdot s + b_0}{a_n \cdot s^n + \dots + a_2 \cdot s^2 + a_1 \cdot s + a_0}$
		<p>Блок Производная реализует численное определение производной на каждом временном шаге как отношение приращений динамической переменной ко времени:</p> $y_i(t) = \frac{dx_i(t)}{dt} = \frac{\Delta x_i(t)}{\Delta t}$ <p>где: $x_i(t)$ – i-ый элемент входного сигнала, $y_i(t)$ – i-ый элемент выходного сигнала.</p>
		<p>Блок Интегратор с ограничением реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:</p> $y_i'(t) = k_i x_i(t)$ <p>при наличии ограничений на значение выходного сигнала:</p> $y_{\min} \leq y_i(t) \leq y_{\max}$ <p>где: $x_i(t)$ – i-ый элемент входного сигнала, k_i – элемент вектора коэффициентов усиления, $y_i(t)$ – элемент выходного сигнала.</p>

Окончание таблицы П.8

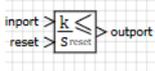
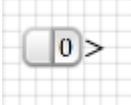
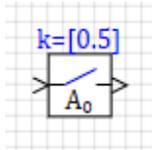
Динамические		
В палитре	На схеме	Описание
		<p>Блок Интегратор с ограничением и сбросом н.у. реализует интегрирование входного сигнала с ограничением на значение интеграла и сбросом состояния:</p> $y_i(t) = k_i \int_{t_1}^{t_2} x_i(t) dt$ <p>где: $x_i(t)$ – элементы вектора входного сигнала, $y_i(t)$ – элементы вектора выходного сигнала, t_1 и t_2 – времена переключения начальных условий, k_i – элемент вектора коэффициентов усиления. Блок реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:</p> $y_i'(t) = k_i x_i(t)$ <p>при наличии ограничений на значение выходного сигнала:</p> $y_{\min} \leq y_i(t) \leq y_{\max}$

Таблица П.9 – Основные блоки палитры Ключи

Ключи		
В палитре	На схеме	Описание
		<p>Блок Кнопка формирует на выходе логический сигнал, равный 0 или 1, в зависимости от своего состояния:</p> $y(t) = \begin{cases} 0, & \text{если } \text{Состояние}(t) = 0 \\ 1, & \text{если } \text{Состояние}(t) = 1 \end{cases}$ <p>Состояние кнопки управляется однократным нажатием на нее левой кнопкой мыши в процессе расчета.</p>
		<p>Блок Ключ-0 реализует функцию управляемого ключа по алгоритму:</p> $y(t) = \begin{cases} y_0, & \text{если } x(t) < K \\ x_2(t), & \text{если } x(t) \geq K \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов, $y(t)$ – вектор выходных сигналов, K – вектор «порогов» (параметров уставок), y_0 – вектор выходных сигналов блока при входных сигналах, меньших пороговых значений.</p>

Окончание таблицы П.9

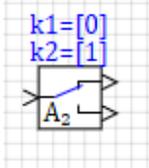
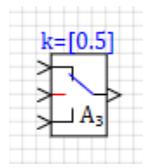
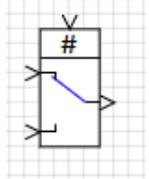
Ключи		
В палитре	На схеме	Описание
		<p>Блок Ключ-2 реализует функцию управляемого ключа по алгоритму:</p> $y_1(t) = \begin{cases} x(t), & \text{если } x(t) \leq K \\ y_{10}, & \text{если } x(t) > K \end{cases}$ $y_2(t) = \begin{cases} x(t), & \text{если } x(t) > K \\ y_{20}, & \text{если } x(t) \leq K \end{cases}$ <p>где: $x(t)$ – вектор входных сигналов; $y_1(t)$ – вектор сигналов на 1-ом выходном порте; $y_2(t)$ – вектор сигналов на 2-ом выходном порте; K_1, K_2 – векторы пороговых значений (уставок), определяющих условия перекоммутации сигналов на выходных портах блока; y_{10} – вектор сигналов на 1-ом выходном порте блока, при входных сигналах, больших пороговых значений K_1; y_{20} – вектор сигналов на 2-ом выходном порте блока, при входных сигналах, меньших пороговых значений K_2.</p>
		<p>Блок Ключ-3 реализует функцию управляемого ключа по алгоритму:</p> $y(t) = \begin{cases} x_1(t), & \text{если } x_2(t) < K \\ x_3(t), & \text{если } x_2(t) \geq K \end{cases}$ <p>где: $x_1(t)$ – вектор сигналов на 1-ом входном порте; $x_2(t)$ – вектор сигналов на 2-ом входном порте; $x_3(t)$ – вектор сигналов на 3-ом входном порте; $y(t)$ – вектор сигналов на выходном порте; K – вектор пороговых значений управляющих сигналов (логических уставок), определяющих условия перекоммутации входных сигналов на выход блока.</p>
		<p>Блок Ключ управляемый перекидной по входам тип 1 реализует функцию управляемого ключа по алгоритму:</p> $y(t) = \begin{cases} x_1(t), & \text{если } \text{Управляющий}(t) = 0 \\ x_2(t), & \text{если } \text{Управляющий}(t) = 1 \end{cases}$ <p>где: $x_1(t), x_2(t)$ – входные сигналы блока (или векторы сигналов), $y(t)$ – выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок сверху. В нормальном положении ключ перекинут на 1-ый вход, при подаче управляющего сигнала ключ перекидывается на 2-ой вход.</p>

Таблица П.10 – Основные блоки палитры Логические

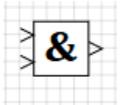
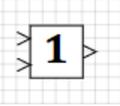
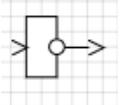
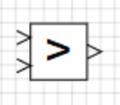
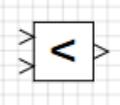
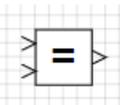
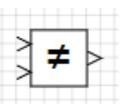
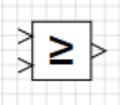
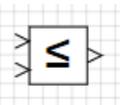
Логические		
В палитре	На схеме	Описание
		Блок Оператор И реализует операцию логического И: $y(t) = \begin{cases} 1, & \text{если } x_1(t) \text{ И } x_2(t) \text{ И } \dots \text{ И } x_n(t) = 1; \\ 0, & \text{в остальных случаях (если хотя бы один из } x_i(t)=0) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Оператор ИЛИ реализует операцию логического ИЛИ: $y(t) = \begin{cases} 1, & \text{если любой из } x_i(t) = 1 \\ 0, & \text{если все } x_i(t)=0 \end{cases}$ где: $x(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Оператор НЕ реализует операцию логического НЕ: $y(t) = \begin{cases} 1, & \text{если } x(t) = 0 \\ 0, & \text{если } x(t) = 1 \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x(t)$ – входной сигнал.
		Блок Операция БОЛЬШЕ реализует логическую операцию БОЛЬШЕ: $y(t) = \begin{cases} 1, & \text{если } x_1(t) > x_2(t) \\ 0, & \text{если } x_1(t) \leq x_2(t) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Операция МЕНЬШЕ реализует логическую операцию МЕНЬШЕ: $y(t) = \begin{cases} 1, & \text{если } x_1(t) < x_2(t) \\ 0, & \text{если } x_1(t) \geq x_2(t) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Операция РАВНО реализует логическую операцию РАВНО: $y(t) = \begin{cases} 1, & \text{если } x_1(t) = x_2(t) \\ 0, & \text{если } x_1(t) \neq x_2(t) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Операция НЕ РАВНО реализует логическую операцию НЕ РАВНО: $y(t) = \begin{cases} 1, & \text{если } x_1(t) \neq x_2(t) \\ 0, & \text{если } x_1(t) = x_2(t) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Операция БОЛЬШЕ ИЛИ РАВНО реализует логическую операцию БОЛЬШЕ ИЛИ РАВНО: $y(t) = \begin{cases} 1, & \text{если } x_1(t) \geq x_2(t) \\ 0, & \text{если } x_1(t) < x_2(t) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.
		Блок Операция МЕНЬШЕ ИЛИ РАВНО реализует логическую операцию МЕНЬШЕ ИЛИ РАВНО: $y(t) = \begin{cases} 1, & \text{если } x_1(t) \leq x_2(t) \\ 0, & \text{если } x_1(t) > x_2(t) \end{cases}$ где: $y(t)$ – выходной сигнал блока, $x_1(t), x_2(t)$ – входные сигналы.

Таблица П.11 – Основные блоки палитры Дискретные

Дискретные		
В палитре	На схеме	Описание
		<p>Блок Запаздывание на период квантования реализует дискретное запаздывание входного сигнала на один период квантования:</p> $y(t) = x(t[k-1])$ <p>где: $y(t)$ – значение выходного сигнала в текущий момент расчетного времени, $x(t[k-1])$ – значение входного сигнала в предпоследний момент дискретизации $t[k-1]$. Квантование происходит с периодом T, задаваемым в свойствах блока.</p>
		<p>Блок Экстраполятор реализует экстраполяцию 0-го порядка:</p> $y(t) = x(t[k])$ <p>где: $y(t)$ – значение выходного сигнала в текущий момент расчетного времени, $x(t[k])$ – значение входного сигнала в момент дискретизации $t[k]$. Квантование происходит с периодом T, задаваемым в свойствах блока.</p>
		<p>Блок Дискретный интегратор используется для выполнения интегрирования в дискретных системах. В качестве аппроксимации передаточной функции $1/s$ используется выражение $T/(z-1)$. Расчет выхода блока происходит по следующему выражению:</p> $y(k) = y(k-1) + T \cdot u(k-1)$ <p>где: y – выход интегратора; u – входной сигнал; T – период квантования; k – номер шага моделирования.</p>
		<p>Блок Дискретная передаточная функция общего вида реализует математическую модель звена, динамика которого описывается дискретной передаточной функцией:</p> $W(z) = \frac{b_m \cdot z^m + \dots + b_2 \cdot z^2 + b_1 \cdot z + b_0}{a_n \cdot z^n + \dots + a_2 \cdot z^2 + a_1 \cdot z + a_0}$ <p>где: a_i, b_j – коэффициенты ($i=0\dots n, j=0\dots m, n \geq m$). Обязательные условия: $m \leq n, a_n \neq 0$.</p>
		<p>Блок Дискретная передаточная функция обратного аргумента реализует математическую модель звена, динамика которого описывается дискретной передаточной функцией:</p> $W(z) = \frac{b_m \cdot z^{-m} + \dots + b_2 \cdot z^{-2} + b_1 \cdot z^{-1} + b_0}{a_n \cdot z^{-n} + \dots + a_2 \cdot z^{-2} + a_1 \cdot z^{-1} + a_0}$ <p>где: a_i, b_j – коэффициенты ($i=0\dots n, j=0\dots m, n \geq m$). Обязательное условие: $a_n \neq 0$.</p>