|  |  |
| --- | --- |
|  | ООО «3В Сервис»  РФ, 115191, Москва, Гамсоновский пер., д.2, стр.1  телефон/факс: +7 (495) 221-22-53 [www.3v-services.com](http://www.3v-services.com/) |

**Утверждаю**

Генеральный директор

ООО «ЗВ Сервис»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Петухов В.Н.

****

**Среда динамического моделирования технических систем SimInTech™**

**Руководство оператора**

# Реферат

Руководство на 96 стр., 19 рис., 20 таблиц.

АНИМАЦИЯ, БИБЛИОТЕКА, БЛОК, ГРАФИЧЕСКИЙ РЕДАКТОР, МОДЕЛИРОВАНИЕ, СТРУКТУРНОЕ МОДЕЛИРОВАНИЕ.

В документе приводится описание командного меню, основных опций и командных кнопок графической среды SimInTech.

Представлены сведения о параметрах и алгоритмах работы блоков, входящих в состав библиотеки стандартных блоков SimInTech.

# Содержание

[Реферат 2](#_Toc454536808)

[Содержание 3](#_Toc454536809)

[Обозначения и сокращения 4](#_Toc454536810)

[1 Общие сведения 5](#_Toc454536811)

[2 Опции и команды главного окна 7](#_Toc454536812)

[3 Опции и команды схемного окна 15](#_Toc454536813)

[4 Опции и команды менеджера данных 26](#_Toc454536814)

[5 Опции и команды окна анимации 28](#_Toc454536815)

[6 Описание библиотеки блоков 41](#_Toc454536816)

[6.1 Закладка «Источники» 41](#_Toc454536817)

[6.2 Закладка «Операторы» 44](#_Toc454536818)

[6.3 Закладка «Векторные» 47](#_Toc454536819)

[6.4 Закладка «Динамические» 51](#_Toc454536820)

[6.5 Закладка «Нелинейные» 57](#_Toc454536821)

[6.6 Закладка «Субструктуры» 65](#_Toc454536822)

[6.7 Закладка «Данные» 66](#_Toc454536823)

[6.8 Закладка «Ключи» 71](#_Toc454536824)

[6.9 Закладка «Логические» 76](#_Toc454536825)

[6.10 Закладка «Триггеры» 79](#_Toc454536826)

[6.11 Закладка «Задержки и импульсы» 80](#_Toc454536827)

[6.12 Закладка «Релейные» 82](#_Toc454536828)

[6.13 Закладка «Дискретные» 83](#_Toc454536829)

[6.14 Закладка «Функции» 86](#_Toc454536830)

[6.15 ЗАКЛАДКА «ИССЛЕДОВАНИЯ» 90](#_Toc454536831)

[6.16 Закладка «СТАТИСТИКА» 91](#_Toc454536832)

# Обозначения и сокращения

|  |  |  |
| --- | --- | --- |
| ПО | - | программное обеспечение |
| САПР | - | система автоматизированного проектирования |
| SDK | - | Software Developement Kit – набор для разработки ПО |

# 1 Общие сведения

Среда SimInTech относится к системам автоматизированного проектирования (САПР) логико-динамических систем, описываемых во входо-выходных отношениях.

Объектом САПР может быть любая техническая система, устройство или физический процесс, математическая модель динамики которого описывается системой дифференциально-алгебраических уравнений и может быть реализована методами структурного моделирования.

Под автоматизированным проектированием применительно к среде SimInTech понимается интерактивный процесс работы в *Графической среде*, включающий:

* разработку модели объекта исследования в виде структурной схемы;
* формирование сценариев переходных процессов;
* формирование средств отображения результатов и управления параметрами модели в процессе моделирования;
* исследование модели с использованием встроенных алгоритмов интегрирования системы дифференциально-алгебраических уравнений;
* анализ полученных результатов исследования.

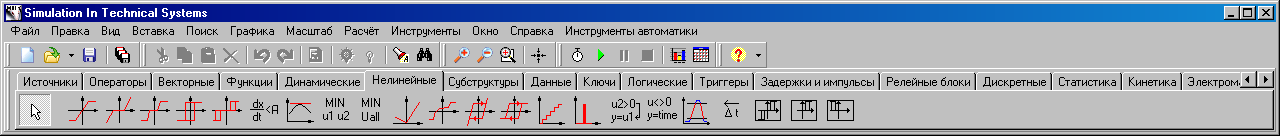
Запуск SimInTech осуществляется посредством файла Mstarter.exe, расположенного в каталоге (папке) BIN. Открытие существующего или нового проекта выполняется из *Главного Окна* посредством соответствующих командных кнопок или командного меню.

Необходимо заметить, что если запущен файл Mstarter.exe, а проект не открыт (или не создан), то большинство командных кнопок в *Главном Окне* неактивны (бледно-серый цвет).

В состав *Графической среды* входят следующие компоненты:

* главное окно;
* редактор библиотеки блоков;
* схемное окно;
* менеджер данных;
* графический редактор.

На рисунке 1 представлена экранная копия проекта при работе в среде SimInTech. В верхней части рисунка расположено главное окно графической оболочки, а в нижней части – схемное окно с набранной в нем структурной схемой.



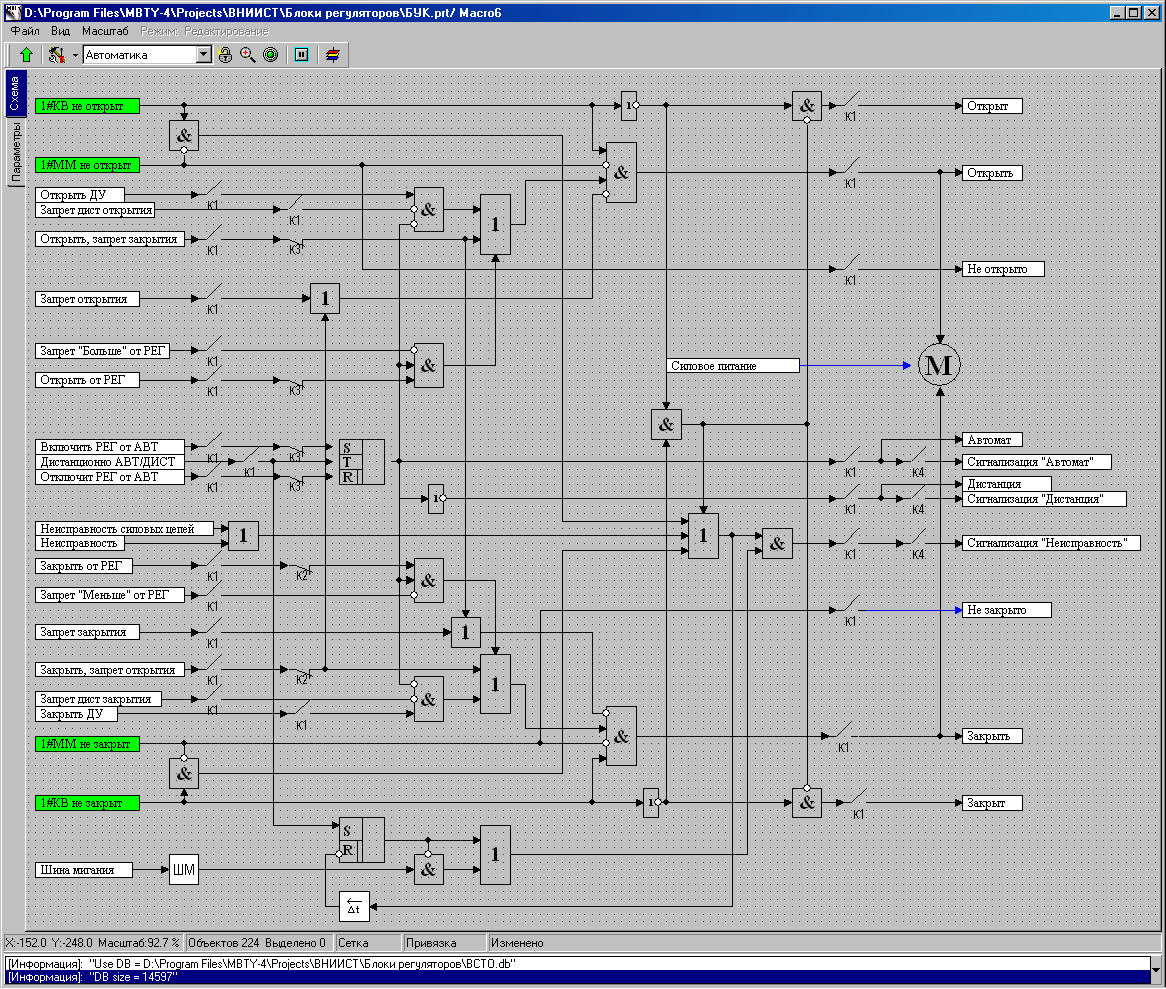


Рисунок 1. – Экранная копия проекта в среде SimInTech.

# 2 Опции и команды главного окна

Главное окно (см. рисунок 2) появляется при запуске среды SimInTech и содержит: ***палитру компонентов***, ***главное меню*** и ***панели инструментов***.

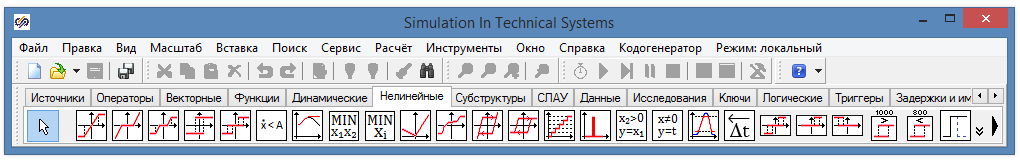


Рисунок 2 – Главное окно *Графической среды*

Описания команд главного меню программы приведено в таблицах 1..10. Меню *«Инструменты*» является настраиваемым, оно позволяет вызывать дополнительные внешние программы. Меню «*Окно*» позволяет вызывать или скрывать существующие в программе схемные окна. Пункты главного меню частично продублированы в панелях инструментов.

Таблица 1 – Опции меню **Файл**

|  |  |  |  |
| --- | --- | --- | --- |
| **Опция** | | **Описание** | |
| **Новый проект** | | Создание нового схемного окна (проекта) | |
| **Открыть…** | | Загрузка нового проекта из файла | |
| **История** | | Вывод списка последних открываемых проектов | |
| **Открыть старую версию** | | Выбрать номер старой версии и сохранить ее как текущую | |
| **Сохранить проект** | | Сохранение текущего проекта в файл | |
| **Сохранить проект как…** | | Сохранение текущего проекта в файл с новым именем | |
| **Сохранить страницу** | | Сохранение текущей страницы проекта в файл | |
| **Сохранить страницу как…** | | Сохранение текущей страницы проекта в файл с новым именем | |
| **Сохранить все страницы** | | Сохранение всех изменённых страниц в файлы | |
| **Закрыть** | | Закрыть активный проект | |
| **Перегрузить страницы** | | Перезагрузить страницы, загруженные по ссылкам | |
| **Согласовать страницы** | | Приведение страниц, загружаемых из одного и того же файла, к общему виду | |
| **Задать пароль страницы** | | Задание пароля для доступа к содержимому текущей страницы | |
| **Сохранить все проекты** | | Сохранение изменений всех открытых проектах в файлы | |
| **Закрыть все** | | Закрыть все открытые проекты | |
| **Печать…** | | Печать содержимого окна текущей субмодели проекта | |
| **Печать с субмоделями…** | | Печать содержимого окон всех субмоделей проекта | |
| **Настройки печати…** | | Настройка печати (отступы, область печати, наличие сетки и т.д.) | |
| **Загрузить библиотеку…** | | Загрузить новую библиотеку элементов из файлов | |
| **Редактирование библиотеку…** | | Запуск редактора библиотеки элементов | |
| **Сохранить в библиотеку…** | | Сохранить выделенный блок или часть структурной схемы в библиотеку элементов | |
| **Сохранить в файл…** | | Сохранить параметры выделенного графического контейнера в файл (с расширением «.elt») | |
| **Загрузить группы из файла** | | Заменить параметры выделенных графических контейнеров на параметры, сохраненные в файле | |
| **Заменить набор параметров** | | Заменить параметры выделенных блоков на параметры, сохраненные в файле (включая значения параметров) | |
| **Обновить набор параметров** | | Заменить параметры выделенных блоков на параметры, сохраненные в файле (значения параметров не изменяются) | |
| **Сохранить набор параметров** | | Сохранить параметры выделенного блока в файл (с расширением «ps») | |
| **Добавить в список обновлений** | | Добавить выделенный блок в список обновлений. При загрузке проектов, в которых присутствуют блоки данного типа, набор параметров для этих блоков заменяется на сохраненные в файле (значения параметров не изменяются) | |
| **Параметры** | | Настройка основных параметров программы (вид, рабочие папки и файлы) | |
| **Выход** | | Выход из программы | |

Таблица 2 – Опции меню **Правка**

|  |  |  |  |
| --- | --- | --- | --- |
| **Опция** | **Описание** | | |
| **Вырезать** | Удаление выделенных объектов в буфер обмена | | |
| **Копировать** | Копирование образов выделенных объектов в буфер обмена | | |
| **Вставить** | Вставка объектов из буфера обмена в окно ***Графический редактор*** | | |
| **Удалить** | Удаление выделенных объектов | | |
| **Отмена** | Отмена выполненного действия | | |
| **Возврат отмены** | Выполнение отменённого действия | |
| **Включить объекты** | Включить выделенные на схеме объекты в расчетную схему | |
| **Исключить объекты** | Исключить выделенные на схеме объекты из расчетной схемы | |
| **Снять выделение** | Снять выделение со всех объектов, находящихся в окне ***Графический редактор*** | |
| **Выделить все** | Выделение всех объектов в окне ***Графический редактор*** | |
| **Выделить по имени** | Выделение примитивов по выбранному имени | |
| **Выделить по типу** | Выделение примитивов заданного типа | |
| **Переименовать объекты** | Переименование выделенных объектов | |
| **Повернуть** | Поворот выделенных объектов (для поворота необходимо выделить нужные объекты, затем выбрать данный пункт меню и далее указать «мышью» центр и вторую точку вектора поворота) | |
| **Переместить** | Перемещение выделенных объектов (сначала необходимо выделить нужные объекты, затем выбрать данный пункт меню и далее указать начальную и конечную точки вектора смещения) | |
| **Масштабировать** | Пропорциональное изменение размеров выделенных объектов (для масштабирования необходимо выделить нужные объекты, затем выбрать данный пункт меню и далее указать «мышью» центр масштабирования и вторую точку) | |
| **Масштабировать с коэффициентом** | Пропорциональное изменение размеров выделенных объектов с некоторым коэффициентом | |
| **Растянуть-сжать** | Непропорциональное изменение размеров выделенных объектов (последовательность действий аналогична предыдущей команде) | |
| **Зеркальное отражение** | Зеркальное отражение графического образа блока относительно указанной оси | |
| **Выровнять по сетке** | Выравнивание выделенных объектов по узлам сетки | |
| **Выдвинуть вперед** | Перемещение выделенного объекта в конец списка объектов (объект будет перерисован *поверх* всех остальных) | |
| **Поместить назад** | Перемещение выделенного объекта в начало списка объектов (объект будет перерисован *позади* всех остальных) |

Продолжение таблицы 2

|  |  |
| --- | --- |
| **Сортировать слева направо** | Сортировка выделенных объектов на схеме проекта с направлением от верхнего левого угла экрана к нижнему правому. |
| **Порядок перерисовки** | Изменение порядка перерисовки объектов |
| **Собрать в группу** | Создание из выделенных примитивов группы (для объединения примитивов в группу необходимо предварительно их выделить) |
| **Собрать в блок** | Создание экземпляра «стандартного блока», для которого в качестве графического образа используются выделенные на схеме графические примитивы |
| **Разобрать** | Разрушение выделенной группы (для выполнения данного действия группу необходимо выделить) |
| **Свойства** | Изменение свойств выделенного объекта или общих свойств (по типу и названию) нескольких выделенных объектов |
| **Изменить блок** | Изменение описание типа выделенного объекта |

Таблица 3 – Опции меню **Вид**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Режим разработчика** | Включение расширенного набора опций проекта |
| **Панорамирование по левой кнопке мыши** | Включение режима панорамирования страницы при помощи левой кнопки мыши |
| **Выделение группы** | Если установлена эта опция, то выделение объектов производится в стиле AutoCAD, если нет – то в стиле Windows. |
| **Автоматически скрывать список сообщений** | Скрывать список сообщений проекта при отсутствии новых |
| **Не перемещать объекты** | Включение/отключение возможности перемещения объектов в схемном окне |
| **Редактировать подписи** | Включение/отключение возможности редактирование подписи блока непосредственно в схемном окне |
| **Не изменять размеры блоков** | Включение/отключение возможности изменения размеров объектов в схемном окне |

Продолжение таблицы 3

|  |  |
| --- | --- |
| **Показывать номера блоков** | Активирует отображение свойства «Индекс блока». |
| **Одно окно оперативного просмотра** | Если установлена эта опция, то открытие нового окна оперативного просмотра расчетных значений приводит к закрытию предыдущего окна |
| **Трассировка линий связи** | Опция включает/отключает алгоритм автоматического определения координат промежуточных точек для линий связи |
| **Сетку вперед** | Отображать сетку поверх изображений блоков |
| **Цвет линий** | Установка цвета линий по умолчанию |
| **Цвет заливки** | Установка цвета заливки по умолчанию |
| **Шрифт текста** | Установка шрифта текста по умолчанию |
| **Использовать стиль слоя** | Если включена эта опция, то при вычерчивании линий связи и блоков будут использованы установки, указанные для текущего слоя, которому принадлежит элемент схемы |
| **Панели инструментов** | Управление видимостью отдельных частей палитры инструментов |
| **Вид по умолчанию** | Восстановление вида главного окна по умолчанию |

Таблица 4 – Опции меню **Масштаб**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Увеличить в 2 раза** | Увеличение масштаба изображения в 2 раза. |
| **Уменьшить в 2 раза** | Уменьшение масштаба изображения в 2 раза |
| **Увеличить фрагмент** | Увеличение выделенного фрагмента изображения во все окно (для увеличения – сначала выбрать пункт меню, затем выделить фрагмент изображения) |
| **Показать все** | Вписывает структурную схему в размеры схемного окна |
| **Исходное положение** | Установка центра координат изображения по центру окна (центр координат в режиме показа с сеткой изображается крестиком) |
| **Исходный масштаб** | Установка масштаба изображения 1:1 |
| **Стандартный масштаб** | Установка стандартного масштаба изображения |
| **Увеличить по выделенному объекту** | Установка видимой области схемного окна по габаритам выделенного объекта (обычно применяется при необходимости отмасштабировать по какой-либо «рамке» листа на схемном окне). |

Таблица 5 – Опции меню **Вставка**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Панель примитивов** | Показать панель графических примитивов |
| **Повтор вставки блока** | Вставляет на схему последний ранее вставленный объект. |
| **Режим замены блоков** | При активации данного пункта новый вставляемый на схему блок будет заменять предварительно выделенный на схеме блок. Все линии связи при этом останутся на схеме без изменений (при условии совпадения количества портов входа-выхода у блоков). После осуществления замены режим замены блоков сбрасывается автоматически. |
| **Синхронизировать свойства при замене** | При активации данного пункта в режиме замены блока в случае, когда у заменяемого блока и заменяющего блока совпадают поля в колонке «Имя» в свойствах, то происходит сохранение поля «Значение» для совпадающих свойств. |
| **Базовые блоки** | Вставка на схему пустых базовых блоков, из которых делаются все остальные. |

Таблица 6 – Опции меню **Поиск**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Поиск блока…** | Поиск произвольного графического объекта на схеме в соответствии с заданными критериями. |
| **Поиск в память-из-памяти…** | Поиск блоков типа «В память» и «Из памяти» по имени переменной. |
| **Поиск в базе данных** | Поиск объекта в базе данных |
| **Настройки индексатора** | Открытие окна настройки свойств поискового индексатора |

Таблица 7 – Опции меню **Сервис**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Сигналы…** | Вызов редактора сигналов проекта |
| **Генерация сигналов и связей** | Автоматическая генерация сигналов, связей и *скрипта* по шаблонам для выделенных объектов |
| **Связи…** | Вызов редактора связей и дерева проекта данного окна |
| **Скрипт…** | Вызов редактора глобальных параметров страницы (в нём можно записывать и графические операции). |
| **Скриншот в буфер** | Копирует в буфер видимую на экране часть схемного окна |
| **Сохранить изображение в файл** | Сохраняет в файле выбранного формата (.bmp, .emf, .wmf, .dxf) видимую на экране часть схемного окна |

Таблица 8 – Опции меню **Расчёт**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Инициализация** | Инициализация расчёта текущего проекта |
| **Пуск** | Запуск расчёта текущего проекта |
| **Сделать шаг** | Выполнение одного шага расчета |
| **Пауза** | Приостановка расчёта текущего проекта |
| **Стоп** | Полная остановка расчёта текущего проекта |
| **Менеджер данных** | Вызов менеджера данных текущего проекта |
| **Инспектор параметров** | Вызов окна выбора параметров объекта |
| **Параметры расчёта** | Открытие окна настройки параметров расчета проекта |
| **Отладочная информация** | Вызов диалогового окна с диагностической информацией и отладочными функциями (актуально только на этапе проведения расчета или после выполнения расчета) |

Таблица 9 – Опции меню **Инструменты**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Настройка** | Настройка вызова внешних программ из редактора схем. Позволяет добавлять в меню инструменты дополнительные пункты. |

Таблица 10 – Опции меню **Окно**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Показать все** | Показать все схемные окна |
| **Скрыть все** | Минимизировать все схемные окна |

Таблица 11 – Опции меню **Справка**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Содержание** | Вызов главного файла справки программы (если он есть) |
| **Контекстная справка** | Вызов контекстной справки для выделенного блока на схеме |
| **Поиск обновлений** | Поиск, загрузка и установка обновления из Интернета. Для поиска обновлений вызывается программа update.exe. |
| **Написать в службу поддержки…** | Выведение формы для составления сообщения в службу поддержки |
| **О программе** | Выводит заставку программы |

# 3 Опции и команды схемного окна

Создание и изменение структурной схемы производится при помощи ***схемного окна***. Редактор структурных схем способен одновременно загружать несколько независимых проектов. Каждый проект располагается на своём отдельном схемном окне. Проекты могут рассчитываться независимо друг от друга, за счёт использования многозадачности операционной системы Windows. Вид схемного окна представлен на рисунке 3.

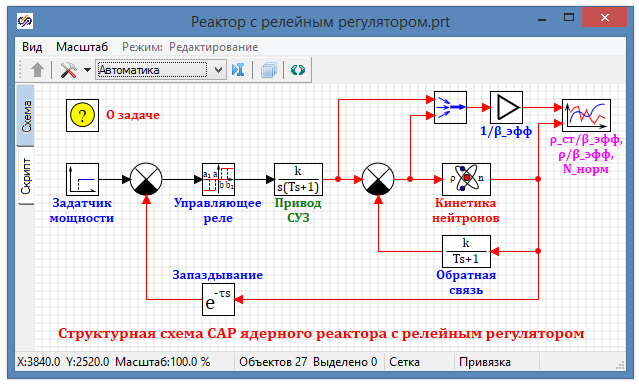


Рисунок 3 – Схемное окно проекта

Структурная схема состоит из следующих элементов: блоки, линии связи и декоративные элементы. Декоративные элементы – это объекты, не участвующие в составлении расчётного задания (комментарии, рисунки), они являются простыми графическими примитивами и добавляются на схемное окно так же, как и на окно графического редактора, описанное ниже. В составлении расчётного задания принимают участие блоки и линии связи – они являются активными элементами схемы.

***Блок*** – это элемент схемы, который имеет порты.

***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.

***Линия связи*** – это элемент схемы, соединяющий два или более портов блока.

И блоки, и линии связи являются графическими примитивами и поддерживают все функции обычных графических примитивов, но, кроме того, они имеют дополнительные возможности.

Активные элементы схемы имеют настраиваемые списки свойств и параметров. Они имеют специальные поля, необходимые для расчёта – имя расчётного класса и расчётный шаблон.

В редакторе схем предусмотрена также возможность создания ***комплексных блоков***.

***Комплексный блок*** – это несколько блоков, объединённых по принципу родитель-наследник, т.е. когда один блок может принадлежать другому. Блоки могут принадлежать также и линиям связи, что даёт возможность наглядного построения схем со сложной неоднородной структурой. Чтобы создать блок, который может вставляться в другие элементы схемы, необходимо установить у него значение свойства «*Вставлять в другие*» равным «*Да*» и сохранить этот блок в библиотеку классов. После этого блок будет вставляться из библиотеки только в другие блоки.

***Свойство*** элемента схемы – это некоторый элемент данных, значение которого может быть изменено при помощи редактора свойств элемента схемы. Для изменения значения свойств активного элемента схемы нужно выделить его щелчком мыши на схеме, а затем или произвести по нему двойной щелчок, или выбрать пункт меню «*Редактирование – Свойства объекта*». При этом появится окно редактора свойств элемента схемы (см. рисунок 4). Свойства некоторых типов (вещественные, целые, массивы, матрицы) могут быть заданы в виде алгебраического выражения, при этом в качестве операндов могут использоваться имена общих сигналов проекта, свойств и параметров макромодели. Эта возможность используется для параметризации субмоделей.

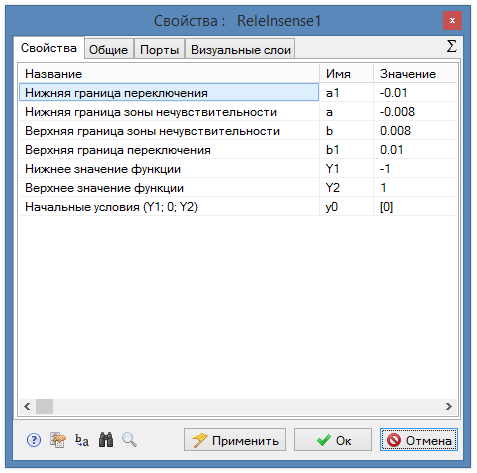


Рисунок 4 – Окно редактора свойств элементов схемы

***Параметром*** называется элемент данных, который не может быть изменён при помощи редактора свойств элементов схемы. Параметры предназначены для записи в них результатов расчёта схемы. Свойства и параметры элементов схемы имеют название, внутреннее имя и описание типа данных. Они доступны по именам в скрипте и списке связей схемного редактора, аналогично другим свойствам графических примитивов.

***Имя расчётного класса*** – это название типа элемента в расчётном модуле, соответствующего данному блоку или линии связи на структурной схеме.

***Расчётный шаблон*** – это описание элемента в расчётном модуле, соответствующего данному блоку или линии связи на структурной схеме.

Изменение состава списков свойств, параметров, количества и типа портов и описания имени расчётного класса и расчётного шаблона блоков производится при помощи редактора блоков. Для линий связи изменять описание можно при помощи редактора библиотеки классов. При помощи него можно создавать свои блоки из пустых или из уже имеющихся в библиотеке классов.

Для изменения описания какого-либо блока необходимо:

1. Поместить исходный блок на схемное окно проекта. Для этого надо или выбрать пункт меню «*Вставка – Базовые блоки – Стандартный блок*» или вставить нужный исходный блок из палитры.
2. Выбрать пункт меню «*Правка – Изменить блок*». При этом появится редактор блоков, окно которого показано на рисунке 5.

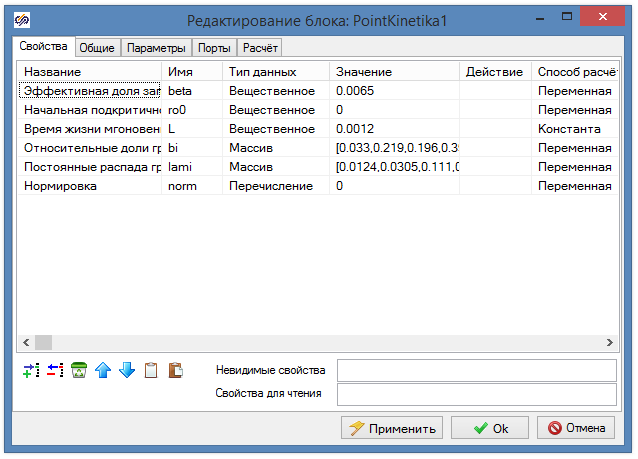


Рисунок 5 – Редактор блоков

При помощи этого инструмента можно добавлять, удалять, изменять имена настраиваемых свойств (закладка «*Свойства*»), параметров (закладка «*Параметры*»), портов (закладка «*Порты*»), имя расчётного класса и расчётного шаблона (закладка «*Расчёт*»).

После того как выделенный блок был изменён при помощи редактора блоков, его можно сохранить в библиотеке классов. В библиотеку классов можно сохранять не только одиночные блоки, но и отдельные фрагменты схемы. Для сохранения одного или нескольких элементов схемы в библиотеку классов необходимо выбрать пункт меню «*Файл – Сохранить в библиотеку*» и в появившемся диалоге ввести имя записи в библиотеке.

Библиотека блоков может быть изменена при помощи специального редактора. Окно редактора библиотеки блоков показано на рисунке 6.

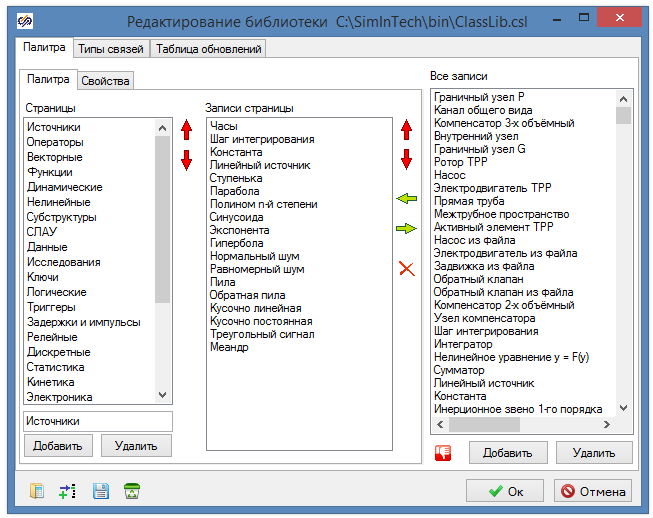


Рисунок 6 – Окно редактора библиотеки блоков

При помощи данного инструмента можно изменить состав библиотеки и настроить внешний вид палитры компонентов редактора схем. Палитра компонентов делится на отдельные страницы. Каждая страница может содержать несколько записей. Чтобы добавить страницу, необходимо нажать кнопку «*Добавить*» в разделе «*Страницы*». Чтобы добавить в страницу записи из общего списка, необходимо мышкой перетащить соответствующее название из списка «Все записи» в список «*Записи страницы*». Для каждой записи можно задать изображение, которое будет выводиться на соответствующей кнопке в палитре компонентов, и подсказку для этой кнопки.

Редактор библиотеки классов позволяет создавать и изменять типы линий связи. Для линий связи аналогично блокам задаются свойства, параметры, имя расчётного класса и расчётный шаблон, а также параметры внешнего вида линии.

Перед составлением расчётного задания структурную схему проекта необходимо преобразовать в файл входных данных расчётного модуля.

Первым этапом преобразования является декомпозиция структурной схемы проекта, при которой из полного описания удаляется описание геометрических параметров элементов и различного рода служебных элементов.

Редактор схем поддерживает два метода декомпозиции проекта:

- блочно-иерархическая декомпозиция;

- декомпозиция на расчётные слои.

Блочно-иерархическая декомпозиция подразумевает наличие в проекте субмоделей.

***Субмодель*** – это специальный блок, который содержит внутри фрагмент схемы. Для создания субмодели необходимо выбрать пункт меню «*Вставка –Базовые блоки – Субмодель*». Для просмотра или изменения содержимого субмодели необходимо произвести двойной щелчок мыши по изображению субмодели. Пример схемы с субмоделью приведен на рисунке 7, а содержимое субмодели показано на рисунке 8.

У субмодели можно определить порты. Для изменения количества портов в субмодели используются специальные блоки типа «*Порт субмодели*». Для добавления порта у субмодели необходимо: произвести двойной щелчок мыши по изображению субмодели, а затем выбрать пункт меню «*Вставка –Базовые блоки – Порт субмодели*», а для определения типа порта необходимо изменить соответствующие свойства у данного блока.

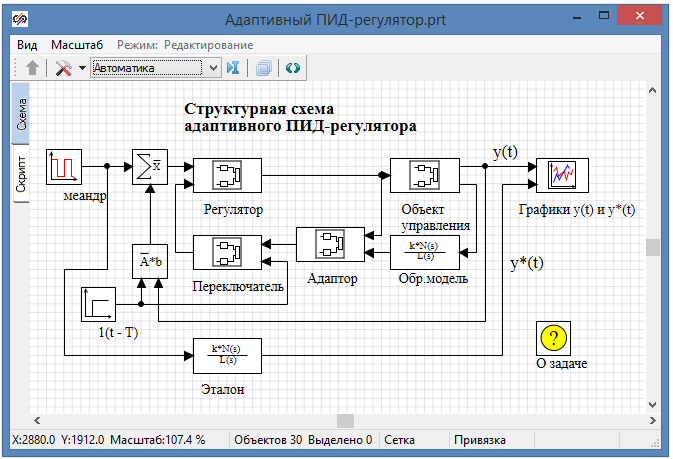


Рисунок 7 – Пример структурной схемы с субмоделью «Регулятор»

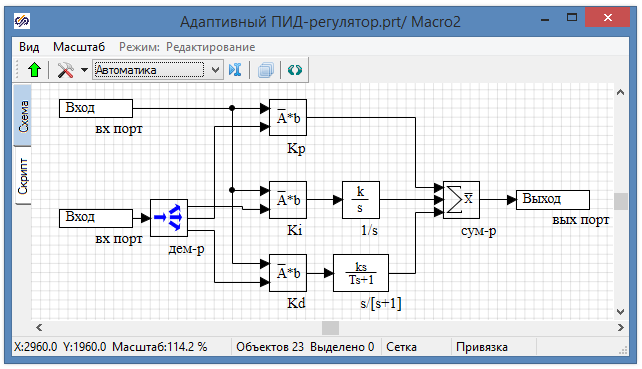


Рисунок 8 – Внутренняя структура субмодели «Регулятор»

В редакторе схем предусмотрен механизм параметризации субмодели – у каждой субмодели, как и у остальных блоков, можно задать список свойств. Свойства любых элементов, находящихся внутри субмодели, могут ссылаться на свойства субмодели, используя имена свойств субмодели. Таким образом, из простых элементов можно собирать более сложные узлы и сохранять их в библиотеку классов.

При формировании расчётного задания внутренняя структура субмодели распаковывается так, чтобы структура проекта стала одноуровневой. Предусмотрено два режима субмоделей – описание субмодели может вноситься в расчётное задание, а может и не вноситься. Эти режимы определяются свойством «*Активная модель*» блока – если это свойство имеет значение «*Да*», то описание субмодели вносится в расчётное задание, а все блоки, упакованные в ней, вносятся в список подчинённых блоков расчётного элемента.

Декомпозиция на расчётные слои подразумевает, что у каждого блока и линии имеется идентификатор слоя.

***Слой*** – это объект, содержащий некоторое множество элементов схемы и общие свойства расчётного задания. Каждый расчётный слой имеет индивидуальные настройки и формирует отдельное расчётное задание для какой-либо расчётной программы. Слои могут добавляться и удаляться при помощи кнопок «*Новый слой*» и «*Удалить слой*» в панели инструментов схемного окна. Текущий слой каждой страницы проекта выбирается при помощи переключателя «*Выбор слоя*». Слой может быть изменён при помощи специального редактора слоёв, окно которого изображено на рисунке 9.

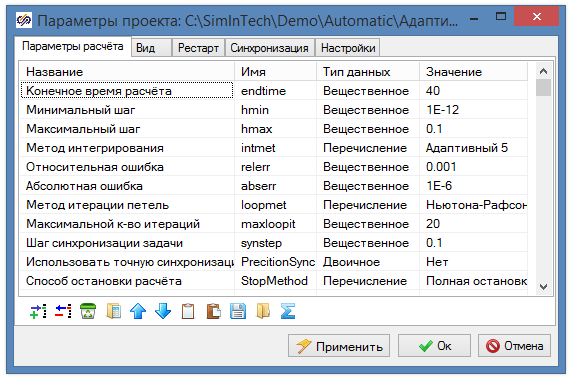


Рисунок 9 – Окно редактора свойств расчётного слоя

Для каждого слоя настраиваются список параметров расчёта (закладка «*Параметры расчёта*»), визуальные свойства (закладка «*Общие*»), и устанавливается настройки расчёта – имя переходной библиотеки для соединения редактора схем с расчётной программой. Для каждого слоя можно настроить список доступных групп классов – это те классы, которые могут быть вставлены в текущий слой. Группа каждого класса имя группы настраивается в редакторе классов.

В одном проекте может быть несколько слоёв, каждый из которых может рассчитываться в своей расчётной программе. Это даёт возможность создания моделей устройств с процессами различной физической природы.

Редактор схем позволяет создавать проекты, состоящие из нескольких независимых файлов. Для этого предусмотрена возможность загрузки субмодели из внешнего файла проекта. Субмодели имеют специальное свойство «*Имя файла*». Если это свойство не пустое, то при загрузке проекта данные о содержании субмодели будут считываться из соответствующего файла проекта (\*.prt), иначе субмодель будет считываться из основного файла проекта. Файлы страницы могут загружаться с использованием специальных путей файлов:

$(BIN)\ - путь к программной директории;

$(ROOT)\ - настраиваемая рабочая директория;

$(DATABASE)\ - корневая директория базы данных;

Таким образом, при изменении одного проекта, все субмодели, которые ссылаются на этот проект в другом проекте, будут изменены при его перезагрузке. При загрузке субмоделей из внешнего файла в основной проект добавляются недостающие сигналы, а в список свойств субмодели добавляются недостающие свойства. Для перезагрузки страниц проекта необходимо выбрать пункт меню «*Файл – Перегрузить страницы*», при этом будут перезагружены все субмодели, ссылающиеся на внешний файл. Редактировать такие субмодели можно в режиме отдельного проекта или в режиме общего проекта, во втором случае для сохранения изменений в странице необходимо выбрать пункт меню «*Файл – Сохранить страницу*» или «*Файл – Сохранить все*» или выделить субмодель правой кнопкой мыши и выбрать пункт «*Действия – Сохранить и связать с файлом*».

В одном проекте могут быть несколько субмоделей, внутренняя структура которых загружается из одного файла. Редактор схем позволяет согласовывать содержание таких субмоделей без их перезагрузки из файла, т.е. при изменении одной из них остальные будут также изменены. Согласование моделей может производиться автоматически при выходе из субмодели и запуске расчёта или вручную. Для автоматического согласования необходимо выбрать пункт меню «Файл – Автоматически согласовывать страницы». Для ручного согласования текущей страницы с её аналогами необходимо выбрать пункт меню «Файл – Согласовать страницы».

Редактор схем имеет систему контроля версий файлов проекта. Опции контроля версий доступны в виде списка опций, расположенных в окне параметров среды (Файл - Параметры) на вкладке «Файлы и папки». Список опций контроля версий описан в таблице 12.

Таблица 12 – Опции контроля версий проектов

|  |  |
| --- | --- |
| Сохранять старые версии файлов менеджера данных | При сохранении проекта, выполняется сохранение резервной копии менеджера данных (.mgr - файл) |
| Сохранять старые версии проектов | При сохранении проекта выполняется сохранение резервной копии предыдущей версии проекта (.prt - файл) |
| Ограничение количества резервных копий | Максимальное количество резервных копий, которые могут быть сохранены |
| Не сохранять при наличии атрибута «Только для чтения» | Если файл имеет соответствующий атрибут, то сохранение резервных копии не происходит |

Если задействованы опции контроля версии, то при сохранении старый файл страницы проекта или менеджера данных будет переименован – к нему в качестве расширения будет дописан номер версии файла, начиная с единицы. Загрузка предудущих версий возможна с помощью команды «Файл – Открыть старую версию».

# 4 Опции и команды менеджера данных

Для сбора рассчитанных данных в редакторе схем предусмотрен специальный программный модуль – менеджер данных. Менеджер данных позволяет структурировать отображение результатов расчёта и представить их в виде дерева. Окно менеджера данных представлено на рисунке 10.

Менеджер данных позволяет отображать результаты расчёта в виде графиков или в виде таблицы значений (окна просмотра). Графики и окна просмотра могут объединяться в ***категории***, что позволяет структурировать результаты расчёта при их большом количестве. В каждую категорию можно добавлять и графики, и окна просмотра. Для создания нового графика необходимо сначала создать или указать категорию.

Категорию можно создать, нажав кнопку «*Добавить категорию*» в окне менеджера данных. Затем надо выделить категорию в дереве менеджера данных и нажать кнопку «*Добавить график*» в окне менеджера данных.

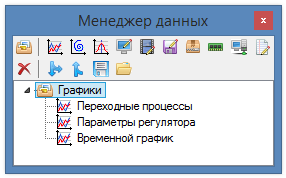


Рисунок 10 – Окно менеджера данных

Графики и окна просмотра могут отображать значения свойств и параметров объектов. Для отображения параметра на графике необходимо добавить в график ссылку этого параметра. Ссылка параметра может иметь своё задаваемое имя, отличное от имени параметра.

Добавление на график или окно просмотра ссылку параметра осуществляется с помощью специального инструмента – инспектора параметров, который позволяет просмотреть и добавить в менеджер данных параметры или свойства любого объекта.

Для вызова инспектора параметров необходимо выбрать пункт меню «*Расчёт – Инспектор параметров*». На рисунке 11 приведено окно инспектора параметров объекта.

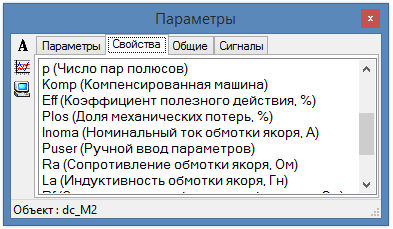


Рисунок 11 – Инспектор параметров объекта

При выделении соответствующего объекта в схемном окне в инспекторе параметров будут появляться его параметры, свойства, визуальные свойства, а также список сигналов проекта. Для добавления ссылки параметра необходимо перетащить мышкой параметр из инспектора параметров в окно графиков или окно просмотра или непосредственно в дерево менеджера данных. Для удаления ссылки параметра из графика или окна просмотра необходимо выделить её название в дереве менеджера данных и нажать кнопку «*Удалить*» в окне менеджера данных.

Конфигурация менеджера данных каждого проекта сохраняется отдельно от файла проекта в файле с расширением «\*.mgr», расположенном в той же директории, что и файл проекта. Это необходимо для обеспечения правильной работы программы при создании распределённых проектов. При загрузке проекта для редактирования проверяется наличие файла менеджера данных. Если файл присутствует, то из него загружается менеджер данных проекта. Отсутствие данного файла не приводит к ошибке.

# 5 Опции и команды окна анимации

Для формирования графического изображения блоков используется специальный ***графический редактор*** (см. рисунок 13). Графический редактор поддерживает некоторые стандартные объекты ОС Windows и технологию OLE, что позволяет вставлять внешние объекты (например, диаграммы, видеофрагменты и т.п.). Редактор технологических схем полностью поддерживает все функции, реализованные в графическом редакторе, поэтому многие элементы управления редактора технологических схем и графического редактора совпадают по названию и назначению.

Изображение формируется из отдельных элементов – графических примитивов. Список поддерживаемых графических примитивов представлен в специальном окне ***Примитивы*** (см. рисунок 12).

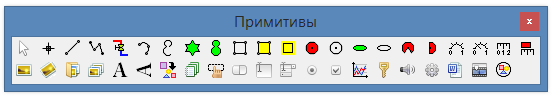


Рисунок 12 – Панель графических примитивов

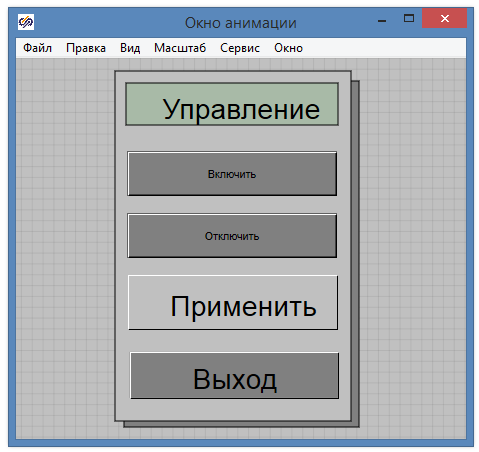


Рисунок 13 – Окно графического редактора

Окно ***Примитивы*** (*Панель графических примитивов*) содержит около 40 элементов, причем ряд элементов относится к категории *базовых* (прямая, полилиния, дуга, круг, эллипс, прямоугольник и др.) и может быть использован для создания визуального образа (чертеж, мнемосхема, рисунок) *новых* более сложных устройств.

Ряд примитивов предназначен для «конструирования» стрелочных и цифровых приборов. В *Панели графических примитивов* присутствуют и специализированные элементы, например, обеспечивающие функционирование ряда специальных опций *Графического редактора*.

Для добавления нового элемента изображения в окно ***Графический редактор*** необходимо щелкнуть по пиктограмме нужного элемента в окне ***Примитивы***, а затем щелчком «мыши» указать место вставки в окне ***Графический редактор***. Для редактирования вставленного элемента необходимо выделить его щелчком «мыши», а затем изменять его положение, размеры, цвет и другие свойства.

После создания визуального образа объекта в окне ***Графический редактор*** и последующего закрытии этого окна щелчком «мыши», в Схемном окне проекта появится подобный визуальный образ.

Окно ***Графический редактор*** имеет командное меню, ряд опций которого продублирован командными кнопками в *Панели инструментов* этого окна. В таблицах 12..17 приведено краткое описание опций командного меню, а в таблице 18 - наиболее часто используемые сочетания клавиш.

Таблица 12 – Опции меню **Файл**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Создать** | Очистка содержимого окна редактирования |
| **Открыть…** | Загрузка в окно ***Графический редактор*** из файла |
| **Сохранить как…** | Сохранение содержимого окна ***Графический редактор*** в файл графики с расширением \*.gcn |
| **Заменить группы из файла** | Замена содержимого выделенных групп на новое из файла |
| **Сохранить в файл…** | Сохранение содержимого окна, как набора элементов графического редактора, в файл \*.elt |

Таблица 13 – Опции меню **Правка**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Свойства объекта** | Изменение свойств выделенного объекта или общих свойств (по типу и названию) нескольких выделенных объектов |
| **Отмена действия** | Отмена последнего совершенного действия |
| **Возврат отмены** | Возврат последнего отмененного действия |
| **Вырезать** | Удаление выделенных объектов в буфер обмена |
| **Копировать** | Копирование образов выделенных объектов в буфер обмена |
| **Вставить** | Вставка объектов из буфера обмена в окно ***Графический редактор*** |
| **Удалить** | Удаление выделенных объектов |
| **Собрать в группу** | Создание из выделенных примитивов группы (для объединения примитивов в группу необходимо предварительно их выделить) |
| **Разобрать** | Разрушение выделенной группы (для выполнения данного действия группу необходимо выделить) |
| **Переименовать объекты** | Переименование выделенных объектов |
| **Выделить все** | Выделение всех объектов в окне ***Графический редактор*** |
| **Снять выделение** | Снять выделение со всех объектов, находящихся в окне ***Графический редактор*** |
| **Выделить по типу** | Выделение примитивов заданного типа |
| **Выделить по имени** | Выделение примитивов по выбранному имени |
| **Выдвинуть вперед** | Перемещение выделенного объекты в конец списка объектов, т.е. объект будет перерисован *поверх* всех остальных |
| **Поместить назад** | Перемещение выделенного объекта в начало списка объектов, т.е. объект будет перерисован *позади* всех остальных |
| **Сортировать слева направо** | Сортирует выделенные объекты на схеме проекта с направлением от верхнего левого угла экрана к нижнему правому. |
| **Порядок перерисовки** | Изменение порядка перерисовки объектов |
| **Переместить** | Перемещение выделенных объектов (сначала необходимо выделить нужные объекты, затем выбрать данный пункт меню и далее указать начальную и конечную точки вектора смещения) |

Продолжение таблицы 13

|  |  |
| --- | --- |
| **Повернуть** | Поворот выделенных объектов (для поворота необходимо выделить нужные объекты, затем выбрать данный пункт меню и далее указать «мышью» центр и вторую точку вектора поворота) |
| **Масштабировать** | Пропорциональное изменение размеров выделенных объектов (ля масштабирования необходимо выделить нужные объекты, затем выбрать данный пункт меню и далее указать «мышью» центр масштабирования и вторую точку) |
| **Растянуть-сжать** | Непропорциональное изменение размеров выделенных объектов (последовательность действий аналогична предыдущей команде) |
| **Зеркальное отражение** | Зеркальное отражение изображений выбранных объектов |
| **Выровнять по сетке** | Выравнивание выделенных объектов по координатной сетке |

Таблица 14 – Опции меню **Вид**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Панель примитивов** | Показывает окно ***Примитивы*** (*Панель графических примитивов*) |
| **Визуальные слои** | Открытие окна управления визуальными слоями |
| **Панорамирование по левой кнопки мыши** | Панорамирование окна анимации движением мыши с зажатой левой кнопкой |
| **Цвет фона** | Изменение цвета фона окна анимации |
| **Поверх всех окон** | Включение режима отображения окна анимации поверх всех окон |
| **Обрамление окна** | Обрамление окна анимации системной рамкой |
| **По центру экрана** | Перемещение окна анимации в центр экрана |
| **Прозрачный цвет** | Включает прозрачность цветов окна анимации |
| **Степень прозрачности** | Степень прозрачности окна анимации |
| **Масштабировать толщину линий** | Включение режима масштабирования толщины линий при изменении масштаба кадра |
| **Антиалайзинг** | Включение режима сглаживания при отображения окна анимации |
| **Сетка** | Установка видимости координатной сетки |

Продолжение таблицы 14

|  |  |
| --- | --- |
| **Привязка** | Установка режима привязки координат «мыши» к координатной сетке |
| **Всплывающие подсказки** | Если установить этот флаг, то при наведении «мыши» на объект в окне ***Графический редактор*** будет показана подсказка |
| **Панель инструментов** | Показывает *Панель инструментов* в окне ***Графический редактор*** |
| **Строка состояния** | Установка видимости строки состояния |
| **Полосы прокрутки** | Установка видимости вертикальной и горизонтальной полос прокрутки окна ***Графический редактор*** |

Таблица 15 – Опции меню **Масштаб**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Увеличить в 2 раза** | Увеличение масштаба изображения в 2 раза. |
| **Уменьшить в 2 раза** | Уменьшение масштаба изображения в 2 раза |
| **Увеличить фрагмент** | Увеличение выделенного фрагмента изображения во все окно (для увеличения – сначала выбрать пункт меню, затем выделить фрагмент изображения) |
| **Задать масштаб** | Задание произвольного масштаба в % |
| **Стандартный масштаб** | Установка масштаба изображения 1:1 |
| **Стандартный центр** | Установка экрана на позицию 0,0 координатной сетки |
| **Исходное положени** |  |
| **Исходное положение** | Установка центра координат изображения по центру окна (центр координат в режиме показа с сеткой изображается крестиком) |
| **Исходный размер окна** | Установка исходного размера окна |

Таблица 16 – Опции меню **Сервис**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Глобальные свойства…** | Вызов редактора глобальных переменных *графического контейнера* |
| **Локальные переменные…** | Вызов редактора локальных переменных *графического контейнера* |
| **Генерация сигналов и связей** | Автоматическая генерация сигналов, связей и *скрипта* по шаблонам для выделенных объектов |
| **Связи…** | Вызов редактора связей и дерева проекта данного окна |
| **Скрипт…** | Вызов текстового редактора для редактирования *скрипта* – текстового алгоритма перерисовки |
| **Отладка…** | Вызов ***Окна отладки*** графики текущего редактируемого изображения |
| **Скриншот в буфер** | Помещение скриншота текущего экрана в буфер обмена |

Таблица 17 – Опции меню **Окно**

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| **Скрыть все окна** | Закрывает *все* анимационные окна в активном приложении |
| **Показать все окна** | Открывает *все* анимационные окна в активном приложении |

Таблица 18 – Специальные сочетания клавиш

|  |  |
| --- | --- |
| **Сочетание клавиш** | **Описание** |
| **Ctrl + Левая кнопка «мыши»** | Плавное масштабирование изображения в поле редактора (Zoom) |
| **Ctrl + Правая кнопка «мыши»** | Плавное перемещение изображения в поле редактора (Aerial View) |
| **Shift + Левая кнопка «мыши»** | При щелчке по полю редактирования, если удерживать клавишу Shift, будет появляться окно выбора объектов, в котором можно указать нужный объект, если его заслоняет другой объект |

Во всех анимационных блоках применяется единая структура графических объектов. Графическое изображение состоит из отдельных объектов – примитивов. Каждый примитив имеет набор свойств, которым можно присваивать значения.

Для редактирования свойств примитива необходимо выделить нужный примитив, а затем в меню *Графического редактора* выбрать опцию *Правка - Свойства объекта*. При этом появляется окно редактирования свойств примитива, внешний вид которого представлен на рисунке 14.

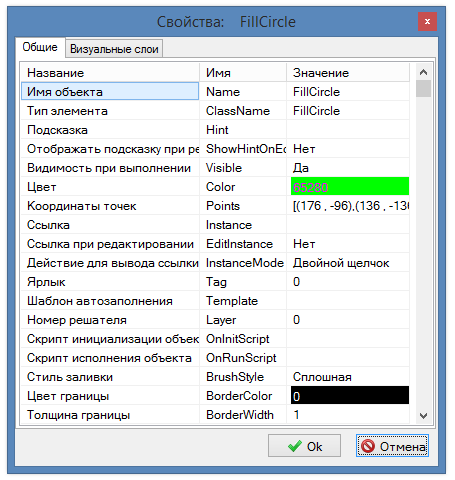


Рисунок 14 - Окно редактирования свойств примитива

Над примитивами можно выполнять ряд операций – перемещение, поворот, масштабирование и изменение пропорций. Все примитивы, которые выводятся на экран, принадлежат специальному объекту – графическому *контейнеру*.

*Контейнер* – объект, объединяющий несколько примитивов одновременно. Он не имеет собственных методов рисования, но имеет метод, позволяющий стирать изображение в заданной прямоугольной области, причем эта область имеет угол поворота. Данный объект содержит список примитивов, которые должны выводиться на экран. *К онтейнер* распределяет команды между примитивами и вызывает метод их перерисовки. Также *контейнер* производит присвоение свойствам примитива каких-либо данных. Данные в *контейнере* содержатся в списке сигналов, глобальных и локальных свойств. Эти списки содержат элементы типа *Сигнал*.

*Сигнал* – объект, содержащий элемент данных определенного типа. Данный объект содержит информацию о типе, имени и названии хранящегося в нем элемента данных, а также методы, позволяющие читать и сохранять элемент данных в поток (файл, буфер).

Сигналы, а также глобальные свойства и локальные переменные (в отличие от примитива) имеют изменяемый тип данных и их можно редактировать в специальном редакторе. Для редактирования списка сигналов необходимо выбрать в меню **Сервис** опцию *Сигналы* (*Глобальные свойства* или *Локальные переменные*). На рисунке 15 представлен внешний вид окна редактирования сигналов контейнера. Для добавления нового сигнала необходимо выполнить щелчок «мышью» по кнопке **Добавить сигнал** (стрелка ↑ в левом нижнем угле рисунка 15).

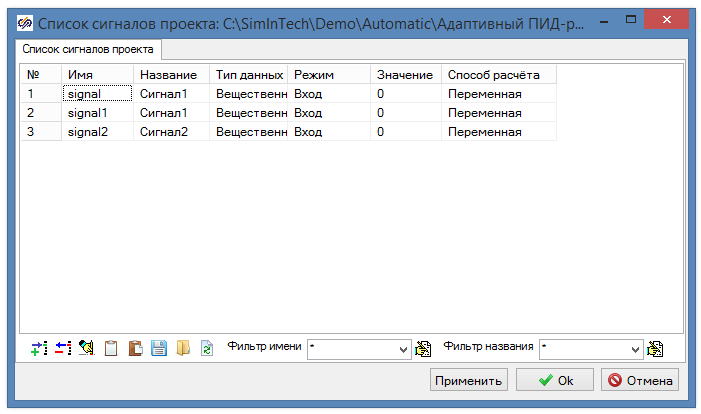


Рисунок 15 - Окно редактирования сигналов контейнера

При редактировании контейнера с объектами происходит сначала присвоение свойств примитивов, а затем их перерисовка.

В *Анимационной Системе* предусмотрено несколько способов передачи данных:

- посредством написанного для *каждого* контейнера *скрипта* на встроенном языке программирования;

путем заполнения *массива* *связей* контейнера.

*Скрипт* позволяет производить над примитивами следующие действия:

- присвоение одному из свойств примитива какого-либо значения;

- перемещение, поворот, масштабирование и растяжение-сжатие объекта-примитива (объектные операции).

Для редактирования *скрипта* необходимо выбрать в меню **Сервис** опцию *Скрипт*, при этом появляется окно редактирования *скрипта*. Пример заполнения окна редактирования скрипта представлен на рисунке 16. Описание языка программирования и его функций находиться в документе «Среда динамического моделирования SimInTech. Описание языка».

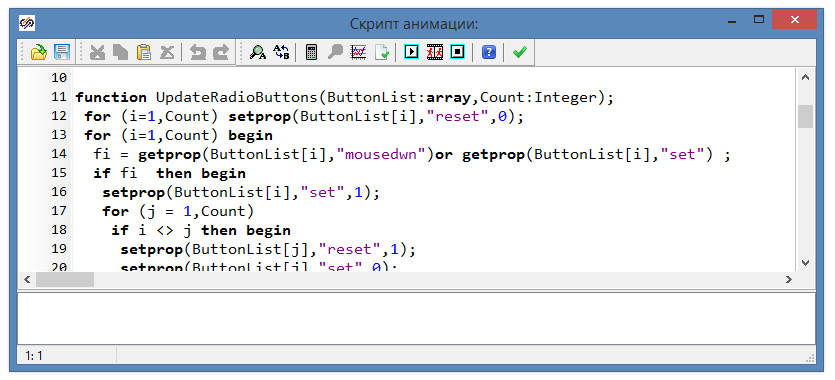


Рисунок 16 - Окно редактирования скрипта

*Массив связей* содержит информацию о списке объектов типа *Связь*. Каждая связь включает информацию об имени источника и имени приемника. При перерисовке в контейнере производится присвоение свойств согласно данной таблице, т.е. данные из списка источников будут перемещены в список приемников. Данный механизм является более быстрым по сравнению со *скриптом* (особенно, если контейнер имеет большое количество портов). Для редактирования списка связей необходимо выбрать в меню **Сервис** опцию *Связи*: появляется диалоговое окно ***Редактор связей***, внешний вид которого представлен на рисунке 17.

Для создания связи необходимо выделить нужный объект на схеме, при этом в закладке *Объект* редактора связей появятся его свойства и список сигналов. Затем «мышью» перетащить нужное свойство в таблицу связей объекта. Аналогично в таблицу перетаскиваются *сигналы*, *глобальные свойства* и *локальные переменные* окна (группы).

После создания примитивов и данных, а также установления связей между ними, можно проверить работу анимации при помощи окна ***Отладка анимации***, которое вызывается посредством меню **Сервис** и далее - опция *Отладка*. Внешний вид этого окна представлен на рисунке 18. В левой части окна ***Отладка анимации*** имеется список доступных сигналов, внизу окна - движок, позволяющий вручную изменять значение выбранного сигнала. Для отладки анимации необходимо щелкнуть «мышью» по кнопке **Отладка** (на рисунке 18 - верхняя справа).

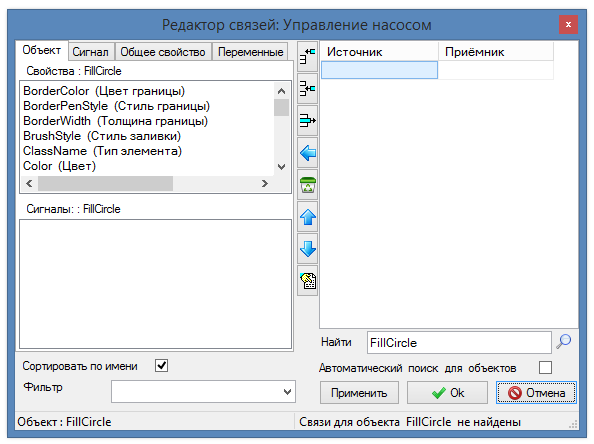


Рисунок 17 - Окно редактирования списка связей контейнера

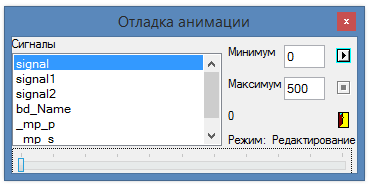


Рисунок 18 - Окно «Отладка анимации»

Содержимое окна ***Графический редактор*** можно записать в файл, для чего необходимо выбрать в меню **Файл** опцию *Сохранить как*… и указать в появившемся диалоговом окне имя файла. Для загрузки изображения из файла необходимо выбрать в меню **Файл** опцию *Открыть*.

Сохраненные графические файлы можно также загрузить и вставить в окно ***Графического редактора*** как единый объект – группу. Для этого нужно выбрать в меню **Файл** опцию *Вставить группу* или нажать кнопку **Вставить группу** на панели примитивов. При загрузке файла как группы, ранее сохраненный список глобальных свойств будет добавлен в свойства группы и станет доступен через редактор свойств, а список сигналов – будет доступен в *скрипте* и в списке *связей*. Файлы с содержимым графического редактора имеют расширение **.gcn**.

Содержимое групп можно изменять (кроме их имен и положения). Для этого необходимо выделить заменяемые группы, в меню **Файл** выбрать опцию *Заменить группы* и указать нужный файл.

*Анимационная Система* позволяем автоматизировать процесс создания сигналов и связей. Для этого предусмотрен *механизм шаблонов*.

*Шаблон* – это текст, согласно которому создаются *сигналы* (*глобальные свойства* или *локальные переменные*), связи и строки *скрипта*. Шаблон может быть задан для любого примитива. Для создания шаблона необходимо в редакторе свойств примитива заполнить поле *Шаблон автозаполнения* (см. рисунок 14).

Чтобы выполнить действия, заданные в шаблоне, необходимо *выделить объект* и выбрать в командном меню **Сервис** опцию*Генерация сигналов и связей*. Шаблон состоит из команд с параметрами. Разделитель между параметрами – запятая. Общий формат записи следующий:

<**Команда**> <Параметр 1>, <Параметр 2> …

Параметры команд шаблона могут задаваться в виде форматированных строк (кроме типа данных и режима данных в команде **SIGNAL**): **$self.PropName$**Текст.

Часть строки **$self.PropName$** - будет заменена на значение свойства с именем «**PropName**», представленном в строковом виде.

В таблице 19 приведены команды, доступные в шаблоне.

Таблица 19 – Команды *шаблона*

|  |  |  |
| --- | --- | --- |
| **Команда** | **Пример написания** | **Результат** |
| **SCRIPT** <Строка> | **SCRIPT** $self.Name$.Width=0 | Если свойство Name объекта присвоено «Line1», то будет создана строка скрипта:  Line1.Width=0 |
| **CONNECT** <Источник>, <Приемник> | **CONNECT** Signal1, $self.Name$.Width | Если имя объекта – Line1, то будет создана связь, где источник будет сигнал с именем Signal1, а приемник – свойство Width объекта Line1. По этой команде создается только связь; новые сигналы не создаются. |

Продолжение таблицы 19

|  |  |  |
| --- | --- | --- |
| **SIGNAL** <Имя>, <Режим – вход или выход>, <Тип данных>, <Название> | **SIGNAL** $self.Name $Out, out, boolean, Состояние | Если имя объекта – Line1, то будет создан сигнал с именем «Line1Out», режим – «выход», тип данных – двоичные, название сигнала – «Состояние». Если уже есть сигнал с таким именем, то команда игнорируется. Если параметр «режим» не указан (пробел), то по умолчанию сигнал будет входом, для того чтобы сигнал был выходом - указать **Out**. Если не указан тип данных, то по умолчанию – вещественное число. Идентификаторы типов данных:  INTEGER – целое  BOOLEAN – вещественное  ARRAY – массив вещественных чисел  COLOR – цвет  Если не указано название, то оно совпадает с именем. В простейшем случае достаточно написать: SIGNAL <Имя сигнала> |

# 6 Описание библиотеки блоков

Библиотека стандартных блоков среды SimInTech реализована в виде динамически загружаемой библиотеки «ClassLib.csl». В палитре элементов главного окна Графической оболочки блоки сгруппированы в отдельные страницы (закладки) по функциональному признаку. На рисунке 19 приведена экранная копия главного окна с активной закладкой «Источники». Ниже приводится описание блоков, входящих в состав каждой из закладок.

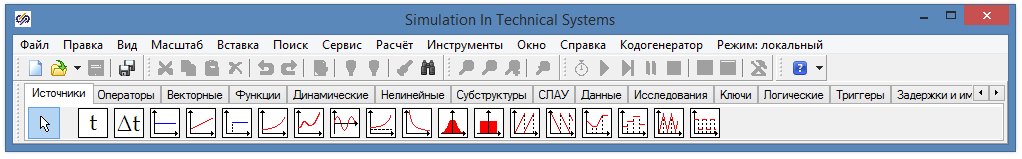


Рисунок 19 – Главное окно с активной закладкой «Источники»

## Закладка «Источники»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Имя блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0000.png | Часы | Блок возвращает текущее модельное время задачи в секундах. |
| D:\3vs\1\automatics\general\images\0208.png | Шаг интегрирования | Блок возвращает текущий шаг интегрирования (расчёта) задачи. |
| D:\3vs\1\automatics\general\images\0002.png | Константа | Параметры:   * Значение – значение на выходе блока, может быть векторным. * Тип данных (для генерации кода) – тип данных выходной переменной при генерации кода для полномасштабной модели. * Название – название величины, для справки. |
| D:\3vs\1\automatics\general\images\0003.png | Линейный источник | Блок возвращает выходное значение вычисленное по формуле:  Y = a + b\*t,  Где a,b – коэффициенты,  t – текущее модельное время в секундах.  Параметры блока:   * Свободный член – коэффициент a * Коэффициент при t – коэффициент b   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0001.png | Ступенька | Блок возвращает выходное значение вычисленное по алгоритму:  ЕСЛИ time > t  TO Y = Yk  ИНАЧЕ Y = Y0  Где t – время срабатывания  Y0 – начальное значение  Yk – конечное значение  time – текущее модельное время в секундах.  Параметры блока:   * Время срабатывания - t * Начальное состояние – Y0 * Конечное состояние - Yk   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0004.png | Парабола | Блок возвращает выходное значение вычисленное по формуле:  Y = a0 + a1\*t + a2\*t2,  Где a0, a1, a2 – коэффициенты,  t – текущее модельное время в секундах.  Параметры блока:   * Свободный член – коэффициент a0 * Коэффициент при t – коэффициент a1 * Коэффициент при t2 - коэффициент a2   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0005.png | Полином n-й степени | Блок возвращает выходное значение вычисленное по формуле:  Yi = a1,i + a2,i\*t + a3,i\*t2 +....an,i\*tn-1,  Где a1 ... an – коэффициенты полинома,  i – номер строки матрицы коэффициентов и элемента выходного вектора.  t – текущее модельное время в секундах.  Параметры блока:   * Коэффициенты полинома – матрица коэффициентов a, где в каждой строке заданы значения коэффициентов полинома для соответствующего элемента выходного вектора. |
| D:\3vs\1\automatics\general\images\0006.png | Синусоида | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*sin(w\*t + f),  Где a, w, f – амплитуда, круговая частота и фаза,  t – текущее модельное время в секундах.  Параметры блока:   * Амплитуда – коэффициент a * Частота – коэффициент w * Фаза – коэффициент f   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0007.png | Экспонента | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*exp(b\*t + c),  Где a, b, c – коэффициенты,  t – текущее модельное время в секундах.  Параметры блока:   * К-т усиления– коэффициент a * Обратная постоянная времени– коэффициент b * Сдвиг - коэффициент c   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0008.png | Гипербола | Блок возвращает выходное значение вычисленное по формуле:  Y = k/(eps + t),  Где k, eps – коэффициенты,  t – текущее модельное время в секундах.  Параметры блока:   * Числитель– коэффициент k * Минимальное значение знаменателя– коэффициент eps.   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0009.png | Нормальный шум | Блок генерирует псевдослучайную последовательность вещественных чисел, нормально распределенных (по Гауссу).  Параметры блока:   * Математическое ожидание - M; * Дисперсия - D; * Период квантования в секундах - qt.   Если период квантования задан равным 0 (нулю), то случайный сигнал на выходе блока будет обновляться после выполнения каждого очередного шага интегрирования.  Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0010.png | Равномерный шум | Блок генерирует псевдослучайную последовательность вещественных чисел, равномерно распределенных в диапазоне от x\_min до x\_max.  Параметры блока:   * Минимальное значение – x\_min; * Максимальное значение – x\_max; * Период квантования в секундах - qt.   Если период квантования задан равным 0 (нулю), то случайный сигнал на выходе блока будет обновляться после выполнения каждого очередного шага интегрирования.  Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0011.png | Пила | Блок генерирует периодически линейно нарастающий пилообразный выходной сигнал.  Параметры блока:   * Размах – y; * Период – t; * Смещение – dy.   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0012.png | Обратная пила | Блок генерирует периодически линейно убывающий пилообразный выходной сигнал.  Параметры блока:   * Размах – y; * Период – t; * Смещение – dy.   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0013.png | Кусочно-линейная | Блок генерирует произвольный кусочно-линейный выходной сигнал, являющийся суперпозицией линейных воздействий.  Параметры блока:   * Время – t – массив временных отметок в сек. * Значение функции – y - массив значений выходного сигнала   Параметры являются матрицами, где в каждой строке задается вектор времён или значений для соответствующего элемента выходного вектора. |
| D:\3vs\1\automatics\general\images\0113.png | Кусочно-постоянная | Блок генерирует кусочно-постоянный (многоступенчатый) выходной сигнал, являющийся суперпозицией ступенчатых воздействий.  Параметры блока:   * Массив временных интервалов – t – в секундах. * Массив значений – y - массив значений выходного сигнала   Параметры являются матрицами, где в каждой строке задается вектор времён или значений для соответствующего элемента выходного вектора. |
| D:\3vs\1\automatics\general\images\0014.png | Треугольный сигнал | Блок генерирует треугольный выходной сигнал.  Параметры блока:   * Размах – y; * Период – t; * Смещение – dy.   Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0015.png | Меандр | Реализует источник прямоугольных импульсов с заданными временными и амплитудными параметрами.  Параметры:   * Длительность 1-го полупериода * Значение 1-го полупериода * Длительность 2-го полупериода * Значение 2-го полупериода |
|  | Циклограмма | Блок реализует функции управляемого функционального генератора и предоставляет возможность задания произвольных периодических функций в виде последовательности опорных точек, которые интерполируются в соответствии с выбранным методом. |
|  | Управляемый синусоидальный генератор | Представляет собой управляемый источник синусоидального напряжения. |

## Закладка «Операторы»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0024.png | Сумматор | Реализует поэлементное суммирование входных величин.  Параметры:   * Весовые множители для каждого из входов – числа на которые умножаются значения входных портов блока. Число портов блока равно размерности этого параметра. |
| D:\3vs\1\automatics\general\images\0038.png | Сравнивающее устройство | То же самое, что и сумматор, но по умолчанию весовые множители равны [1, -1]. |
| D:\3vs\1\automatics\general\images\3003.png | Сложение вектора с числом | Реализует суммирование каждого из элементов векторной величины на первом входе со скалярной величиной на втором входе. Размерность выхода равна размерности первого входа. |
| D:\3vs\1\automatics\general\images\0101.png | Суммирование элементов вектора | Реализует суммирование всех элементов векторов входов. Выход – скалярная величина.  Параметры:   * Весовые коэффициенты входов – числа, на которые умножаются значения входных портов блока. Число портов блока равно размерности этого параметра. |
| D:\3vs\1\automatics\general\images\0025.png | Перемножитель | Реализует поэлементное умножение входных величин блока.  Параметры:   * Количество портов – количество портов блока. |
| D:\3vs\1\automatics\general\images\0102.png | Перемножение элементов вектора | Реализует перемножение всех элементов векторов входов. Выход – скалярная величина.  Параметры:   * Весовые коэффициенты входов – числа, на которые умножаются значения входных портов блока. Число портов блока равно размерности этого параметра. |
| D:\3vs\1\automatics\general\images\3004.png | Умножение на число | Реализует умножение каждого из элементов векторной величины на первом входе со скалярной величиной на втором входе. |
| D:\3vs\1\automatics\general\images\0026.png | Делитель | Реализует поэлементное деление первой входной величины на вторую. |
| D:\3vs\1\automatics\general\images\3005.png | Деление скаляра на вектор | Реализует деление первой скалярной входной величины на каждый из элементов второй величины. Размерность выхода равна размерности второго входа. |
|  | Усилитель | Производит поэлементное умножение входной величины на заданный коэффициент усиления. Размерность выхода равна размерности входа.  Параметры:   * Коэффициент усиления – скалярный множитель для входной величины. |
|  | Векторный усилитель | Производит умножение элементов входной величины на элементы заданного векторного коэффициента усиления. Размерность выхода равна размерности входа и коэффициента усиления.  Параметры:   * Коэффициент усиления. |
| D:\3vs\1\automatics\general\images\0075.png | Абсолютное значение | Реализует вычисление модуля каждого из элементов входной векторной величины. |
| D:\3vs\1\automatics\general\images\0105.png | Размножитель | Реализует умножение элементов входа на векторы коэффициентов усиления.  Yi = ki\*U – для каждого элемента входа. |
| D:\3vs\1\automatics\general\images\0076.png | Знак | Реализует вычисление знака каждого из элементов входной векторной величины.  -1 – число отрицательное.  0 – число равно нулю.  1 – число больше нуля. |
|  | Номер активного элемента | Блок выдает порядковый номер первой единицы в линейном массиве двоичных сигналов, поступающем на вход. |
| D:\3vs\1\automatics\general\images\0266.png | Динамическая выборка | Блок выполняет процедуру выбора элемента из вектора по указанному номеру. |
| D:\3vs\1\automatics\general\images\0269.png | Интеграл от табличной функции | Блок векторизован и реализует вычисление интегралов от таблично заданных функций методом трапеций.    где х – числовой вектор независимой переменной; xmin , xmax - минимальное и максимальное значения независимой переменной, соответственно; u(x) – числовой вектор значений функции.  Параметры:   * Число точек функции – npoint * Число функций – nfun   Блок имеет два векторных входа и один векторный выход.  На первый вход поступает массив значений аргументов таблично заданных функций размерностью npoint.  На второй вход поступает массив значений таблично заданных функций размерностью npoint \* nfun (таким образом, может одновременно вычисляться интеграл от нескольких функций, заданных в одних и тех же точках по независимому аргументу).  На выход блока поступают значения интеграла от функций.  Размерность этого выхода равна nfun. |
| D:\3vs\1\automatics\general\images\0121.png | Компенсация начального значения | Блок реализует компенсатор начальных условий, в соотвествии со следующей формулой:  Y = x – x0  Где x – входное значение;  X0 – входное значение в момент времени t=0;  Блок векторизован. |
| D:\3vs\1\automatics\general\images\1029.png | Нелинейное уравнение y = F(y) | Блок предназначен для вычисления значений в алгебраических петлях и реализует итерационное решение системы НАУ методом, заданным в настройках задачи. Ядро программы производит автоматический подбор выходных значений блока для максимально приближённого выполнения условия: значения вектора выхода блока равны вектору входа.  Параметры:   * Начальное приближение – начальное приближение для выходного вектора блока. |
| D:\3vs\1\automatics\general\images\1028.png | Нелинейное уравнение F(y) = 0 | Блок предназначен для вычисления значений в алгебраических петлях и реализует итерационное решение системы НАУ методом, заданным в настройках задачи. Ядро программы производит автоматический подбор выходных значений блока для максимально приближённого выполнения условия: значения вектора входа равны нулю.  Параметры:   * Начальное приближение – начальное приближение для выходного вектора блока. |
|  | Стоп-расчёт | Блок выполняет процедуру остановки расчёта или временное прерывание расчета при вхождении значения входного сигнала в заданный диапазон.  Параметры:   * Левая граница стоп-интервала - xmin * Правая граница стоп-интервала - xmax * Сообщение при остановке расчёта – Msg |
| D:\3vs\1\automatics\general\images\3007.png | Линейный преобразователь | Реализует линейную интерполяцию входной величины. При этом значение входа равное минимальной входной величине соответствует нулю на выходе, а значение входа равное максимальной входной величине соответствует единице на выходе.  Параметры:   * Минимальное значение входа * Максимальное значение входа |
| D:\3vs\1\automatics\general\images\3008.png | Выборка по активному элементу | Блок выполняет процедуру выбора элемента из вектора по активному элементу управляющего вектора по следующему алгоритму:  ЦИКЛ i ОТ 1 ДО N  ЕСЛИ u1[i] > 0 TO Y=u2[i];ВЫХОД ИЗ ЦИКЛА  КОНЕЦ ЦИКЛА  Где u1 – управляющий вектор;  u2 – вектор выбираемых значений;  n – размерность входов |
| D:\3vs\1\automatics\general\images\0174.png | Целая часть | Блок реализует операцию выделения целой части входного сигнала.  Под целой частью понимается ближайшее целое число в меньшую сторону (по абсолютной величине). |
| D:\3vs\1\automatics\general\images\0175.png | Дробная часть | Блок реализует операцию выделения дробной части входного сигнала.  Под дробной частью понимается разность между сигналом и его целой частью. Дробная часть положительного числа – неотрицательна, а дробная часть отрицательного числа – неположительна |
| D:\3vs\1\automatics\general\images\3051.png | Округление | Выполняет округление (ограничение точности) входного действительного числа с задаваемой пользователем  точностью. Алгоритм работы блока реализует следующую формулу:  ,  где – элемент входного вектора, – параметр, задающий точность округления, – элемент выходного вектора. |

## Закладка «Векторные»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
|  | Мультиплексор | Блок реализует “сжатие” нескольких *скалярных* или *векторных* входных сигналов в **один векторный** выходной сигнал типа “шина” данных. Размерность выходного вектора равна сумме размерностей входов.  Параметры:   * Количество портов |
|  | Демультиплексор | Блок реализует “расщепление” *векторного* входного сигнала в отдельные выходные сигналы. Выходами могут быть как *скалярные*, так и *векторные* (“многожильные”) сигналы. По умолчанию данный блок реализует демультиплексирование 2-х “жильного” входного векторного сигнала в скалярные сигналы.  Параметры:   * Массив размерностей выходов |
|  | Распаковка матрицы | Блок реализует преобразование входного векторного сигнала на несколько векторных выходных сигналов согласно алгоритму:    где *Y(t)* - векторный выходной сигнал;  *X(t)* - векторный входной сигнал.  Входной вектор *X(t)*, интерпретируемый как матрица размера [N\*M] и “запакованная” по строкам или по столбцам, преобразуется в **M** или **N** векторных выходов *Y(t)* (в зависимости от варианта распаковки: по *строкам* или по *столбцам*).  Параметры:   * Число строк в матрице - M * Число столбцов в матрице - N   Матрица запакована по – вариант “запаковки” входной матрицы (по *строкам* или по *столбцам*);  Матрицу распаковать по - вариант “распаковки” входной матрицы (по *строкам* или по *столбцам*); |
|  | Запаковка матрицы | Блок реализует преобразование *нескольких* входных векторных сигналов в *один* векторный выходной сигнал согласно алгоритму:    где *Y(t)* - векторный выходной сигнал;  *X(t)* - векторные входные сигналы.  Входные вектора *X(t)*, интерпретируемые как матрица размера [N\*M] и “распакованная” по *строкам* или по *столбцам*, преобразуется в **один** векторный выход *Y(t)* в зависимости от варианта запаковки: по *строкам* или по *столбцам*.  Параметры:   * Число строк в матрице - M * Число столбцов в матрице - N   Матрица запакована по – вариант “запаковки” входной матрицы (по *строкам* или по *столбцам*);  Матрицу распаковать по - вариант “распаковки” входной матрицы (по *строкам* или по *столбцам*); |
|  | Выборка из вектора | Блок реализует выборку из входного вектора в соответствии с заданным *типом выборки*.  Параметры:   * Тип выборки: * - Ручная; * - Четные элементы; * - Нечетные элементы; * - Первые N элементов; * - Последние N элементов; * - В обратном порядке; * - По возрастанию; * - По убыванию.   Номера элементов (для *Ручной* выборки)  Параметр N (для опций *Первые N* и *Последние N*) |
| D:\3vs\1\automatics\general\images\0146.png | Решение СЛАУ | Блок реализует процедуру решения системы линейных алгебраических уравнений вида:    где ***А*** – матрица коэффициентов; ***х*** – вектор решений; ***u*** – вектор правых частей.  Блок имеет *два векторизованных* входных порта:  1-ый входной порт – для ввода матрицы ***А*** , представляемой в виде *вектора коэффициентов* при распаковке матрицы *по строкам*.  2-ой входной порт – для ввода вектора ***u*** (вектора правых частей).  Блок имеет *один векторный* выходной порт, предназначенный для вывода вектора ***х*** (вектора решений).  Размерность векторного сигнала на 1-ом *входном* порте должна быть равна ***n\*n***, где ***n*** – количество уравнений в системе, а размерности сигналов на 2-ом *входном* порте и на *выходном* порте – одинаковы и равны ***n***.  Параметры:  Число уравнений – размерность матрицы A. |
| D:\3vs\1\automatics\general\images\0147.png | Умножение матрицы на вектор | Блок реализует процедуру умножения **квадратной** матрицы на *вектор* соответствующей размерности.  Блок имеет *два векторизованных* входных порта:  1-ый входной порт – для ввода матрицы ***А*** , представляемой в виде *вектора коэффициентов* при распаковке матрицы *по строкам*.  2-ой входной порт – для ввода вектора ***u*** .  Блок имеет *один векторный* выходной порт, предназначенный для вывода вектора ***y***.  Размерность векторного сигнала на 1-ом *входном* порте должна быть равна ***n\*n***, где ***n*** – количество уравнений в системе, а размерности сигналов на 2-ом *входном* порте и на *выходном* порте – одинаковы и равны ***n***.  Параметры:   * Размерность матрицы. |
| D:\3vs\1\automatics\general\images\0148.png | Транспонирование матрицы | Блок реализует операцию транспонирования квадратной матрицы:  Блок имеет *один векторизованный* входной порт для ввода матрицы ***А***, представляемой в виде *вектора коэффициентов* при распаковке матрицы *по строкам*.  Блок имеет *один векторный* выходной порт, предназначенный для вывода результата транспонирования, представляемого в виде *вектора коэффициентов* при распаковке транспонированной матрицы *по строкам*.  Размерности векторных сигналов на *входном* и на *выходном* портах – *одинаковы*.  Параметры:   * Размерность матрицы. |
| D:\3vs\1\automatics\general\images\0267.png | Интерполяция | Блок предназначен для интерполяции таблично заданных функций. Блок имеет три векторных входа и один векторный выход. На первый вход поступает массив значений аргументов таблично заданных функций размерностью Npnt. На второй вход поступает массив значений таблично заданных функций размерностью Npnt\*Nfun (таким образом, может одновременно проводится интерполяция нескольких функций, заданных в одних и тех же точках по независимому аргументу). На третий вход поступает массив точек аргументов размерностью Narg, в которых необходимо вычислить значение интерполируемых функций. Значение Narg определяется размерностью входного сигнала.  На выход блока поступают вычисленные после проведения интерполяции значения функций в точках, заданных в третьем входном порте. Размер этого выхода равен Narg\*Nfun.  Реализованы методы интерполяции полиномами Лагранжа и кубическими сплайнами. Порядок интерполяции и индекс начала интерполяции имеют значение только для интерполяции полиномами Лагранжа. Значения Order и Idn определяют области интерполяции и экстраполяции. Внутри интервала [ x(Idn) x(Idn+Order) ] функция интерполируется, вне этого интервала – экстраполируется. При интерполяции сплайнами на границах интервала задано условие равенства 0 вторых производных.  Параметры:   * метод интерполяции Met (Полином Лагранжа; * Кубический сплайн; Линейная); * порядок интерполяции Order; * число точек функции Npnt; * индекс начала интерполяции Idn; * число функций Nfun. |
| D:\3vs\1\automatics\general\images\0268.png | МНК-аппроксимация | Блок предназначен для аппроксимации таблично заданных функций методом наименьших квадратов (МНК). Блок имеет три векторных входа и два векторных выхода. На первый вход поступает массив значений аргументов таблично заданных функций размерностью Npnt. На второй вход поступает массив значений таблично заданных функций размерностью Npnt\*Nfun (таким образом, может одновременно проводится аппроксимация нескольких функций, заданных в одних и тех же точках по независимому аргументу). На третий вход поступает массив точек аргументов размерностью Narg, в которых необходимо вычислить значение аппроксимированных функций. Значение Narg определяется размерностью входного сигнала.  На первый выход блока поступают вычисленные после проведения аппроксимации значения функций в точках, заданных в третьем входном порте. Размер этого выхода равен Narg\*Nfun.  На второй выход блока поступают значения коэффициентов аппроксимирующих полиномов (в порядке возрастания степеней). Размер этого выхода равен (Order+1)\*Nfun.  Параметры:   * функция формы fmType (Полином a[m]\*x^m, * a[m]\*sin(m\*pi\*x), a[m]\*(1-x)\*x^m); * порядок аппроксимации Order; * число точек таблично заданной функции Npnt; * число интерполируемых функций Nfun. |

## Закладка «Динамические»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0985.png | Язык программирования | Реализует произвольный алгоритм, написанный на встроенном языке программирования. При генерации кода для блока может быть задан шаблон генерации произвольного кода на языке Си.  Параметры:   * Тип сортировки – флаг, определяющий порядок расчёта блока в общей схеме. * Заголовок модуля – заголовок текста при генерации кода, здесь декларируются специальные комментарии для системы s3. * Секция декларации переменных – текст, вставляемый в начало программы при генерации кода. * Основная секция кода – основной текст программы при генерации кода для блока. * Секция запоминания состояний – текст, вставляемый в конец программы при генерации кода, здесь производиться вычисление переменных состояния. * Декларации переменных DBM – текст, вставляемый в select-файл для описания переменных состояния блока. * Значения переменных по умолчанию – тест для установки начальных значений переменных состояния блока. |
| D:\3vs\1\automatics\general\images\0027.png | Интегратор | Реализует интегрирование элементов вектора входной величины.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Начальные условия – начальное значение выходной величины блока. |
| D:\3vs\1\automatics\general\images\0029.png | Апериодическое звено 1-го порядка | Реализует вычисление выхода апериодического (инерционного) звена первого порядка.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Постоянные времени. * Начальные условия – начальное значение выходной величины блока. |
| D:\3vs\1\automatics\general\images\0037.png | Переменные состояния | Блок реализует описание многомерной линейной динамической системы в матричной форме:  dx/dt = Ax + Bu;  y = Cx + Du,  где A, B, C, D - матрицы: системы, входа, выхода и обхода, соответственно; x -вектор переменных состояния, [Nx]; u - вектор входа, [Nu]; y - вектор выхода, [Ny].  Размерность матрицы А – [Nx\*Nx].  Размерность матрицы B – [Nu\*Nx].  Размерность матрицы C – [Nx\*Ny].  Размерность матрицы D – [Nu\*Ny].  1-е число в квадратных скобках – количество столбцов,  2-е число – количество строк.  Обязательное условие: Nx >= Ny  Вход и выход блока - векторные сигналы, размерностью Nu и Ny, соответственно.  Параметры:   * Число переменных состояния - Nx * Число входных воздействий - Nu * Число выходов - Ny * Матрица A(Nx\*Nx) * Матрица B(Nu\*Nx) * Матрица C(Nx\*Ny) * Матрица D(Nu\*Ny) * Начальные условия (Nx) – вектор начальных условий, размерностью Nx. |
| D:\3vs\1\automatics\general\images\0030.png | Колебательное звено | Блок реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:    где x(t) – входной сигнал в блок; y(t) – выходной сигнал из блока.  При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией:    Обязательное условие: T < > 0.  Параметры:   * Коэффициенты усиления - K * Постоянные времени - T * Коэффициент демпфирования - β * Начальные условия – y(0) * Начальные условия по производной – y`(0)   Блок векторизован, размерности параметров должны быть равны размерности входа. |
| D:\3vs\1\automatics\general\images\0074.png | Идеальное транспортное запаздывание | Блок векторизован и реализует математическую модель звена, динамика которого описывается линейным уравнением с запаздыванием вида:  y ( t ) = x (t - T),  где x(t) – вектор входных сигналов в блок; y(t) – вектор выходных сигналов из блока; Т - вектор времен запаздывания.  При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией (для скалярного варианта сигналов):  W(s) = exp( - T \* s),  Параметры:   * Время запаздывания – T * Начальный размер стека – начальный размер массива, в котором сохраняются входные данные блока. |
| D:\3vs\1\automatics\general\images\0028.png | Передаточная ф-я общего вида | Блок реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:    где    коэффициенты; i = 0…n, j = 0…m; n ≥ m;  x(t) – входной сигнал в блок; y(t) – выходной сигнал из блока.  При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией:    Параметры:   * Коэффициенты числителя – b, вектор размерностью m; * Коэффициенты знаменателя – a, вектор размерностью n; * Начальные условия – y(0); |
| D:\3vs\1\automatics\general\images\0032.png | Инерционно-дифференцирующее звено | Реализует вычисление инерционно-дифференцирующего звена.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Постоянные времени. * Начальные условия – начальное значение выходной величины блока. |
| D:\3vs\1\automatics\general\images\0077.png | Производная | Блок **векторизован** и реализует численное определение производной    на каждом временном шаге как отношение приращений динамической переменной и времени, где **x(t)** – вектор входных сигналов, **y(t)** – вектор выходных сигналов.  Параметры:   * Начальные условия – значение выходного вектора при t = 0; |
| D:\3vs\1\automatics\general\images\0031.png | Инерционно-форсирующее звено | Блок реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:    где x(t) – входной сигнал в блок; y(t) – выходной сигнал из блока.  При нулевых начальных условиях динамика блока может быть представлена следующей передаточной функцией:    Параметры:   * Постоянная времени T1 * Постоянная времени T2 * Начальные условия – y(0) |
| D:\3vs\1\automatics\general\images\0033.png | Инерционно-интегрирующее звено | Реализует вычисление выхода инерционно-интегрирующего звена.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Постоянные времени. * Начальные условия – начальное значение выходной величины блока. * Начальные условия по производной – начальные условия для производной выходной величины. |
| D:\3vs\1\automatics\general\images\0111.png | Функционал квадратичный | Блок **векторизован** и реализует интегральное преобразование:  (квадратичный функционал качества),  где **T = t** - текущее модельное время; **ui(t)** - мгновенное значение вектора входного сигнала; **ai** – вектор весовых коэффициентов; **у(t)** - выходной сигнал.  Параметры:   * Весовые коэффициенты – весовые коэффициенты a для элементов входного вектора. |
|  | Интегратор с ограничением | Реализует интегрирование элементов вектора входной величины с ограничением значения выходной величины.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Начальные условия – начальное значение выходной величины блока. * Минимальное значение – минимальное значение выхода. * Максимальное значение – максимальное значение выхода. |
| D:\3vs\1\automatics\general\images\0034.png | Интегратор с изменяемыми н.у. | Блок **векторизован** и реализует интегрирование входного сигнала:    где x(t) – векторный входной сигнал; y(t) – векторный выходной сигнал; t1, t2 – времена переключения начальных условий.  Блок имеет 3 *входных* порта и при ориентации блока *слева-направо*:  1-ый входной порт (верхний слева) – интегрируемый векторный сигнал;  2-ой входной порт (средний слева) – управляющий сигнал;  3-ий входной порт (нижний слева) – вектор *новых* начальных условий.  При *t* = *t1* происходит установление *новых* начальных условий и при *t1* < *t* < *t2* выполняется интегрирование сигналов входного вектора с *этими новыми* начальными условиями.  При *t* = *t2* происходит ***очередное изменение*** начальных условий и при *t* > *t2* выполняется ***очередной этап*** *интегрирования с новыми* начальными условиями вплоть до **нового очередного**изменения начальных условий.  Переключение начальных условий производится **только** при изменении управляющего сигнала с *одного* значения, например, с **1** на *другое*, например, **2,** или наоборот, с **2** на **1**.  Размерности интегрируемого сигнала и сигнала начальных условий должны быть *одинаковыми*, например, если вектор входа – “трехжильный”, то и сигнал начальных условий тоже “трехжильный”.  Параметры:   * Число переменных состояния – размерность выходного вектора; * 1-е значение управляющего сигнала; * 2-е значение управляющего сигнала; |
| D:\3vs\1\automatics\general\images\0129.png | Интегратор с ограничением и сбросом начальных условий | Блок реализует интегрирование входного сигнала (сигналов) с ограничением на значение интеграла и сбросом состояния:  ,  где xᵢ(t) – элементы вектора входного сигнала, yᵢ(t) – элементы вектора выходного сигнала, t₁ и t₂ – времена переключения начальных условий, kᵢ – элемент вектора коэффициентов усиления.  Блок реализует математическую модель звена, динамика которого описывается линейным обыкновенным дифференциальным уравнением (ОДУ) вида:  ,  при наличии ограничений на значение выходного сигнала:  .  Блок выполняет интегрирование (в пределах ограничений) входного сигнала, поэтому пиктограмма блока имеет вид передаточной функции идеального интегрирующего звена с «добавкой» символа «меньше-равно» и словом reset. Размерности сигналов входа, выхода и вектора коэффициентов должны совпадать. |
| D:\3vs\1\automatics\general\images\0122.png | Переменное транспортное запаздывание | Блок **векторизован** и реализует решение уравнения (системы уравнений) транспортного переноса *распадающейся* скалярной субстанции вида    при допущении о постоянстве линейной скорости переноса скалярной субстанции в пределах участка для каждого момента времени при граничных условиях  и начальных условиях  В уравнении *y(t)* – переносимая скалярная субстанция, *u(t)* – скорость переноса, *L* – длина участка транспортировки (переноса) скалярной субстанции, *x* – пространственная координата.  После ввода безразмерной пространственной координаты *z = x/L* и *мгновенного времени переноса* скалярной субстанции в пределах участка  уравнение записывается как    а начальные условия принимают вид .  Обязательное условие:  Блок имеет 2 входных и 2 выходных порта.  На 1-ый входной порт подается сигнал, соответствующий значению скалярной субстанции на входе в участок транспортировки.  На 2-ой входной порт подается сигнал, соответствующий значению мгновенного времени переноса скалярной субстанции в пределах участка транспортировки.  На 1-ом выходном порте формируется сигнал, соответствующий значению скалярной субстанции на выходе из участка транспортировки.  На 2-ом выходном порте формируется сигнал, соответствующий значению времени пребывания «метки» скалярной субстанции в пределах участка транспортировки.  Параметры:   * Вектор постоянных распада – вектор к-тов λ; * Начальный размер стека – начальный размер массива для временного хранения данных блока; |
| D:\3vs\1\automatics\general\images\0150.png | Аналитическая апериодика 1-го порядка | Реализует вычисление инерционного звена первого порядка по аналитической формуле.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Постоянные времени. * Начальные условия – начальное значение выходной величины блока. * Дополнительные входы – указывает наличие дополнительных входов блока. * Начальные условия берутся из – указатель источника начальных условий блока. |
| D:\3vs\1\automatics\general\images\0151.png | Дискретная апериодика 1-го порядка | Реализует вычисление выхода апериодического (инерционного) звена 1-го порядка по неявной численной схеме.  Параметры:   * Коэффициенты усиления – коэффициент, на который умножается входная величина. * Постоянные времени. * Начальные условия – начальное значение выходной величины блока. * Дополнительные входы – указывает наличие дополнительных входов блока. * Начальные условия берутся из – указатель источника начальных условий блока. |
| D:\3vs\1\automatics\general\images\3052.png | Интегратор на усилителях | Блок выполняет численное интегрирование с ограничением.  Блок векторизован, является субмоделью (сборкой). Для того чтобы посмотреть структуру алгоритма — нажмите правой кнопкой «Действия» → «Войти в субмодель». Алгоритм работы блока реализует численное интегрирование входного сигнала (сигналов). |
| D:\3vs\1\automatics\general\images\3053.png | ДИФ-производная | Выполняет численное дифференцирование, с использованием блока «Дискретная апериодика 1-го порядка». Блок векторизован, представляет собой субмодель (сборку). Для того чтобы посмотреть структуру алгоритма — нажмите правой кнопкой «Действия» → «Войти в субмодель». Алгоритм работы блока реализует численное дифференцирование входного сигнала (сигналов). |
| D:\3vs\1\automatics\general\images\3054.png | Фильтрация сигнала | Выполняет численную фильтрацию, с использованием блока «Дискретная апериодика 1-го порядка». Блок представляет собой субмодель (сборку). Для того чтобы посмотреть структуру алгоритма — нажмите правой кнопкой «Действия» → «Войти в субмодель». Алгоритм работы блока реализует численную фильтрацию входного сигнала. |

## Закладка «Нелинейные»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0069.png | Линейное с насыщением | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* нелинейной статической характеристики типа "насыщение" по следующему алгоритму:    **y ( t ) = K \* x ( t ), если a < x ( t ) < b;**  **y ( t ) = y1, если x ( t ) <= a;**  **y ( t ) = y2, если x ( t ) >= b,**  где **x(t)** – вектор входных сигналов; **y(t)** - вектор выходных сигналов; **К = (y2-y1)/(b - a)**.  Параметры:   * Нижняя граница аргумента - a * Верхняя граница аргумента - b * Нижнее зн-е функции – y1 * Верхнее зн-е функции – y2 |
| D:\3vs\1\automatics\general\images\0070.png | Линейное с зоной нечувствительности | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* нелинейной статической характеристики типа "линейная с зоной нечувствительности" по следующему алгоритму:    **y ( t ) = K \* [x ( t ) - a], если x ( t ) < a;**  **y ( t ) = 0, если a <= x ( t ) <= b;**  **y ( t ) = K \* [x ( t ) - b], если x ( t ) > b.**  где **x(t)** – вектор входных сигналов; **y(t)** - вектор выходных сигналов.  Параметры:   * Нижняя граница аргумента - a * Верхняя граница аргумента – b * К-т усиления - K |
| D:\3vs\1\automatics\general\images\0073.png | Линейное с насыщением и зоной нечувствительности | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* нелинейной статической характеристики типа "линейная с насыщением и зоной нечувствительности" по следующему алгоритму:    **y ( t ) = y1, если x ( t ) <= a1;**  **y ( t ) = K1 \* [x ( t ) - a], если a1 < x ( t ) < a;**  **y ( t ) = 0, если a <= x ( t ) <= b;**  **y ( t ) = K2 \* [x ( t ) - b], если b < x ( t ) < b1;**  **y ( t ) = y2, если x ( t ) >= b1,**  где **x(t)** – вектор входа; **y(t)** - вектор выхода; **К1 = y1/(a1-a); K2 = y2/(b1-b)**.  Параметры:   * Нижняя граница ограничения – a1 * Нижняя граница зоны нечувсвительности – a * Верхняя граница зоны нечувствительности – b * Верхняя граница ограничения – b1 * Нижнее зн-е функции – y1 * Верхнее зн-е функции – y2 |
| D:\3vs\1\automatics\general\images\0071.png | Релейное неоднозначное (гистерезис) | Реализует релейное звено с гистерезисом.  Параметры:   * Нижняя граница переключения – минимальное значение входной величины, при котором значение на выходе равно нижнему значению. * Верхняя граница переключения – максимальное значение входной величины, при котором значение на выходе равно верхнему значению. * Нижнее значение функции – минимальное значение на выходе блока. * Верхнее значение функции – максимальное значение на выходе блока. * Начальные условия (0;1) – начальное состояние реле. 0 – выключено, 1 – включено. |
| D:\3vs\1\automatics\general\images\0072.png | Релейное c зоной нечувствительности | Реализует релейное звено с гистерезисом и зоной нечувствительности.  Параметры:   * Нижняя граница переключения – минимальное значение входной величины, при котором значение на выходе равно нижнему значению. * Нижняя граница зоны нечувствительности. * Верхняя граница зоны нечувствительности. * Верхняя граница переключения – максимальное значение входной величины, при котором значение на выходе равно верхнему значению. * Нижнее значение функции – минимальное значение на выходе блока. * Верхнее значение функции – максимальное значение на выходе блока. * Начальные условия (0;1) – начальное состояние реле. 0 – выключено, 1 – включено. |
| D:\3vs\1\automatics\general\images\0078.png | Производная с ограничением | Блок **векторизован** и реализует численное определение производной с ограничением модуля выходного значения    на каждом временном шаге как отношение приращений динамической переменной и времени, где **x(t)** – вектор входных сигналов, **y(t)** – вектор выходных сигналов.  Параметры:   * Начальные условия – значение выходного вектора при t = 0; * Максимальное значение производной - a |
| D:\3vs\1\automatics\general\images\0079.png | Нижний или верхний предел | Блок **векторизован**, *запоминает* и выдает на выходной порт минимальное значение входного сигнала (или минимальные значения в векторе входных сигналов), зафиксированное (ые) в процессе моделирования на данный момент времени.  Параметры:   * Тип операции – вычисление минимума сигнала или максимума сигнала. |
| D:\3vs\1\automatics\general\images\0125.png | Минимум по входам | Блок производит поэлементное вычисление минимального или максимального из текущих входных значений:  Yi = min(u1i, u2i)  Где i – элемент вектора  u1 – первый входной вектор  u2 – второй входной вектор  Y – выходной вектор  Параметры:   * Количество портов – количество портов блока; * Тип операции – тип операции: минимум или максимум; |
| D:\3vs\1\automatics\general\images\0126.png | Минимум по всем элементам вектора | Блок производит вычисление минимального или максимального значений из элементов входного вектора:  Y= min(un,i)  Где i – элемент вектора  un,i – первый входной вектор  Y – выходной вектор  Параметры:   * Количество портов – количество портов блока; * Тип операции – тип операции: минимум или максимум; |
| D:\3vs\1\automatics\general\images\0081.png | Излом | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* нелинейной статической характеристики типа "Излом" по следующему алгоритму:    **y ( t ) = k1 \* x ( t ), если x ( t ) < x0;**  **y ( t ) = k2 \* x ( t ), если x ( t ) >= x0,**  где **x(t)** – вектор входных сигналов; **y(t)** - вектор выходных сигналов.  Параметры:   * К при X<0 – коэффициент k1 * K при X>0 – коэффициент k2 * Точка излома – x0 |
| D:\3vs\1\automatics\general\images\0120.png | Ломаная статическая характеристика | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* однозначной нелинейной статической характеристики произвольного вида, аппроксимируемой кусочно-ломаной кривой:    **если x ( t ) < x1 , то y ( t ) = y1;**  **если x ( t ) > xN , то y ( t ) = yN.**  где **x(t)** – вектор входных сигналов; **y(t)** - вектор выходных сигналов.  Параметры блока:   * Вектор значений аргумента – t – массив аргументов. * Вектор значений функции – y - массив значений выходного сигнала |
| D:\3vs\1\automatics\general\images\0123.png | Зазор | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* нелинейной статической характеристики типа "Зазор" *при различных начальных условиях* по следующему алгоритму:    y(t) = K\*[x(t) - b], если x(t) > [y(t - dt)/ K + b];  y(t) = K\*[x(t) +b], если x(t) < [y(t - dt)/ K - b];  y(t) = y(t - dt), если [y(t - dt)/ K - b] <= x(t) <= [y(t - dt)/ K + b],  где **x(t)** – вектор входных сигналов; **y(t)** - вектор выходных сигналов, **dt** - предыдущий временной шаг интегрирования.  Параметры:   * Половина ширины зоны зазора – b * Коэффициент усиления – k * Начальные условия – вектор начальных условий y(0) |
| D:\3vs\1\automatics\general\images\0124.png | Люфт | Блок **векторизован** и реализует преобразование вектора входных сигналов посредством *единой* нелинейной статической характеристики типа "Люфт" *при различных начальных условиях* по следующему алгоритму:    **y(t) = y1, если x(t) <= [y1/ K - b];**  **y(t) = y2, если x(t) >= [y2/ K + b];**  **y(t) = K\*[x(t) - b], если [y(t - dt)/ K + b] < x(t) < [y2/ K + b];**  **y(t) = K\*[x(t) +b], если [y1/ K - b] < x(t) < [y(t - dt)/ K - b];**  **y(t) = y(t - dt), если [y(t - dt)/ K - b] <= x(t) <= [y(t - dt)/ K + b],**  где **x(t)** – вектор входных сигналов; **y(t)** - вектор выходных сигналов, **dt** - предыдущий временной шаг интегрирования.  Параметры:   * Половина ширины зоны зазора – b * Коэффициент усиления – k * Начальные условия – вектор начальных условий y(0) * Нижнее зн-е функции – y1 * Верхнее зн-е функции – y2 |
| D:\3vs\1\automatics\general\images\0140.png | Квантователь по уровню | Блок производит квантование сигнала по дискретным уровням с заданным шагом по уровню.  Параметры:   * Шаг квантования – шаг квантования по каждому элементу входного вектора; |
| D:\3vs\1\automatics\general\images\0127.png | Импульсная функция | Блок возвращает на выходе импульс заданной длительности при выполнении указанного условия на входе блока. Амплитуда импульса равна  Y = 1/tau  где tau – длительность импульса  Параметры блока:   * Условие срабатывания – условие, при котором на выходе формируется импульс; * Вектор длительностей импульсов – длительности импульсов для каждого элемента выходного вектора; * Вектор значений нейтрального состояния – значения выходных элементов когда условие не выполнено; |
| D:\3vs\1\automatics\general\images\0152.png | Запоминание значения сигналов | Блок производит запоминание сигнала по следующему алгоритму:  ЕСЛИ u2 > 0 TO y = u1 ИНАЧЕ значение выхода не меняется  u1 – первый вход;  u2 – второй вход;  y – выходное значение  Блок векторизован, размерности входов и выхода должны совпадать. |
| D:\3vs\1\automatics\general\images\0153.png | Запоминание длительности истинного сигнала | Блок производит запоминание сигнала по следующему алгоритму:  ЕСЛИ u > 0 TO y = time – t0  ИНАЧЕ t0 = time;y=0  u – первый вход;  time – модельное время;  t0 – временная отметка начала импульса;  y – выходное значение;  Блок векторизован, размерности входов и выхода должны совпадать. |
| D:\3vs\1\automatics\general\images\3009.png | Ограничитель | Реализует ограничение значения входной величины.  Параметры:   * Минимальное значение входа * Максимальное значение входа |
| D:\3vs\1\automatics\general\images\0200.png | Задержка на шаг интегрирования | Реализует задержку входной величины на один шаг моделирования.  Параметры:   * Развязка алгебраических петель – флаг, показывающий как должен сортироваться блок. Если указано что должен, то блок будет отсортирован в первую очередь. * Начальные условия – начальные условия на выходе блока на первом шаге задачи. |
| D:\3vs\1\automatics\general\images\3010.png | Субмодель релейное с переменными ЗН и ЗВ | Реализует субмодель для реле с гистерезисом и зоной нечувствительности с переменными параметрами. |
| D:\3vs\1\automatics\general\images\3011.png | Субмодель релейное плюс с переменными ЗН и ЗВ | Реализует субмодель для реле с гистерезисом и зоной нечувствительности с переменными параметрами. |
| D:\3vs\1\automatics\general\images\3012.png | Субмодель релейное минус с переменными ЗН и ЗВ | Реализует субмодель для реле с гистерезисом и зоной нечувствительности с переменными параметрами. |
| D:\3vs\1\automatics\general\images\3055.png | Сигнал больше уставки | Блок представляет собой релейное звено с зоной нечувствительности: на выходе блок формирует логическую 1, если входной сигнал превысил значение уставки, и логический 0, если сигнал стал меньше уставки (с учетом зоны возврата, если она задана ненулевой):  ,  где x(t) – входной сигнал; y(t) – выходной сигнал; a, x₀ – величина зоны возврата и уставка. |
| D:\3vs\1\automatics\general\images\3056.png | Сигнал меньше уставки | Блок представляет собой релейное звено с зоной нечувствительности: на выходе блок формирует логическую 1, если входной сигнал меньше значения уставки, и логический 0, если сигнал стал больше уставки (с учетом зоны возврата, если она задана ненулевой):  ,  где x(t) – входной сигнал; y(t) – выходной сигнал; a, x₀ – величина зоны возврата и уставка. |
| D:\3vs\1\automatics\general\images\0035.png | Уточнение разрыва | Блок производит принудительное изменение шага интегрирования при пересечении входа через заданное значение. При этом на выходе блока формируется 1, если условие перехода выполняется. |
| D:\3vs\1\automatics\general\images\3057.png | ПОР ИКП (порог срабатывания) | Блок представляет собой релейное звено с зоной нечувствительности, на выходе формирует логическую 1 или 0 по следующему алгоритму:  На вход блока подаются два сигнала, которые сравниваются между собой. Если их разность превысит порог срабатывания, то блок формирует логическую 1 на выходе. Если разность между первым и вторым входом станет меньше порога срабатывания (с учетом зоны возврата), то на выходе блок формирует логический 0:  ,  где x₁(t), x₂(t) – входные сигналы блока; y(t) – выходной сигнал; a, ε₀ – величина зоны возврата и порог срабатывания. |
| D:\3vs\1\automatics\general\images\3058.png | НОР ИКП | Блок представляет собой релейное звено с зоной нечувствительности и с двумя выходами, на выходах формирует логические 1 или 0 по следующему алгоритму: На входе в блок подаются два сигнала, которые сравниваются между собой. Если их разность превысит верхний предел, то блок формирует логическую 1 на первом выходе. Если разность между первым и вторым входом станет меньше нижнего предела, то на втором выходе блок формирует логическую 1. В других случаях (если разность находится между верхним и нижним пределами, с учетом величины зоны возврата), на выходах формируется логический 0:  ,  где x₁(t), x₂(t) – входные сигналы блока; y₁(t), y₂(t) – выходные сигналы; a, ε₁, ε₂ – величина зоны возврата и пределы изменения разности (пределы срабатывания). |
| D:\3vs\1\automatics\general\images\3066.png | Гистерезис входного сигнала | Блок при помощи релейного неоднозначного звена типа «Гистерезис» формирует на выходе два признака — превысил ли сигнал на первом входе верхний предел, или снизился ли он до уровня менее нижнего предела. При этом учитывается наличие сигнала на втором логическом входе — при отсутствии сигнала на нём блок «не работает» и оба выхода равны логическому нулю. |
| D:\3vs\1\automatics\general\images\3059.png | Мертвая зона (изменяемая) | Блок реализует алгоритм линейного звена с коэффициентом усиления равным 1 и зоной нечувствительности, которая изменяется в зависимости от входного сигнала на втором входном порте. |
| D:\3vs\1\automatics\general\images\3060.png | ФЗН (формирование зоны нечувствительности) | Блок реализует алгоритм линейного звена с коэффициентом усиления равным 1 и зоной нечувствительности, задаваемой в параметрах блока.  Свойства:  Зона нечувствительности — абсолютная величина верхней и нижней границы переключения для блока «Релейное с зоной нечувствительности».  Зона возврата — величина зоны возврата для верхней и нижней границ переключения.  Входные сигналы:  Вход — входной сигнал блока.  Выходной сигнал:  Выход — равен входному сигналу, если он выходит за пределы зоны нечувствительности (с учетом зоны возврата). Равен 0, если входной сигнал лежит в пределах зоны нечувствительности. |
| D:\3vs\1\automatics\general\images\3073.png | Импульсное открытие/закрытие | Блок импульсно выдает логическую единицу на своем выходе, на протяжение всего времени, пока подана единица на вход. Длина импульса (T\_imp) и длина паузы (T\_pause) задаются в параметрах блока. |
| D:\3vs\1\automatics\general\images\3074.png | Запоминание сигнала | Блок передает входной аналоговый сигнал на выход, если разрешена трансляция (x2=1). Если признак трансляции равен 0, то на выходе фиксируется значение, которое было на момент запрещения трансляции. На изображении блока на схеме отображается значение выходного сигнала, которое равно запомненному либо совпадает с входным, если трансляция разрешена. |
| D:\3vs\1\automatics\general\images\3075.png | Линейное с переменным ограничением | Блок реализует алгоритм линейного звена с коэффициентом усиления равным 1 и с ограничением сигнала, которое изменяется в зависимости от входных сигналов на втором и третьем входных портах.  Алгоритм работы блока:  где y₁(t), y₂(t), y₃(t) – сигналы на выходных портах, x(t) – сигнал на первом входном порте (слева), – сигнал на втором и третьем входных портах (сверху). |

## Закладка «Субструктуры»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\1002.png | Субмодель | Блок, содержащий внутри себя вложенную схему. Для доступа к содержимому субмодели необходимо произвести по нему двойной щелчок. |
| D:\3vs\1\automatics\general\images\1003.png | Порт входа | Блок, который вставляется внутрь субмодели и определяющий наличие у субмодели входного порта. Служит для соединения внешней части схемы со схемой находящейся внутри субмодели. Для указания порта соединения необходимо произвести двойной щелчок по блоку. |
| D:\3vs\1\automatics\general\images\1004.png | Порт выхода | Блок, который вставляется внутрь субмодели и определяющий наличие у субмодели выходного порта. Служит для соединения внешней части схемы со схемой находящейся внутри субмодели. Для указания порта соединения необходимо произвести двойной щелчок по блоку. |
| D:\3vs\1\automatics\general\images\1008.png | В память | Блок предназначенный для соединения одной части схемы с другой, находящейся в произвольном месте проекта. Для указания названия точки соединения необходимо произвести двойной щелчок по блоку. |
| D:\3vs\1\automatics\general\images\1009.png | Из памяти | Блок предназначенный для соединения одной части схемы с другой, находящейся в произвольном месте проекта. Для выбора источника, заданного при помощи блока «В память» необходимо произвести двойной щелчок по блоку и выбрать в списке нужную точку соединения. |
| D:\3vs\1\automatics\general\images\3013.png | Двунаправленная шина (вход) | Блок предназначенный для создания двунаправленной линии связи. |
| D:\3vs\1\automatics\general\images\3014.png | Двунаправленная шина (выход) | Блок предназначенный для создания двунаправленной линии связи. |
|  | Комментарий | Блок, в котором можно вставить произвольный комментарий в формате RTF. Для редактирования комментария необходимо произвести двойной щелчок по блоку. |
|  | Заметка | Произвольный текст, который можно вставить на схему. |
| D:\3vs\1\automatics\general\images\3015.png | Уровень не подсоединённых портов | Данный блок позволяет подключить все неподключенные порты блоков к единому источнику. Для этого необходимо вставить на схему данный блок и соединить его с выходом источника. |
| D:\3vs\1\automatics\general\images\3016.png | Блок записи свойств | Блок позволяющий задать свойство блока из заданного источника. Для этого надо вставить данный блок в другой и указать в нём имя присваиваемого свойства блока.  Параметры:   * Имя свойства – имя свойства блока, в который вставлен данный блок, которое должно быть присвоено из источника. |
| D:\3vs\1\automatics\general\images\3061.png | Указатель сортировки | Блок используется для принудительной сортировки пользователем других блоков в требуемом порядке. При инициализации схемы в среде SimInTech происходит сортировка всех блоков расчетной схемы для дальнейшего интегрирования схемы: на каждом шаге интегрирования блоки обсчитываются последовательно в том порядке, в котором они были отсортированы. Иногда требуется, чтобы выполнение (расчёт) одного блока был выполнен строго после другого блока. Блок, подсоединенный к выходу указателя сортировки, выполняется строго после блоков, подсоединённых к входам указателя сортировки. Другими словами, блок является указанием для SimInTech отсортировать следующий блок (подсоединённый к выходу) строго после предыдущих (подсоединённых к входам). |
| D:\3vs\1\automatics\general\images\3062.png | Условие выполнение субмодели | Блок представляет собой механизм для заморозки и разморозки расчета субмодели в зависимости от внешнего условия. Если внутри Субмодели разместить блок Условие выполнения субмодели, то у субмодели появится дополнительный вход, а сама субмодель будет выполняться (расчитываться) только в том случае, если на этот вход подается логическая 1. Если на новый вход субмодели подается логический 0, то вся субмодель и все блоки, находящиеся внутри неё, замораживаются до тех пор, пока на вход субмодели не станет вновь поступать логическая 1. |
| D:\3vs\1\automatics\general\images\3063s.png | Внешняя DLL | Блок является инструментом для подключения к расчетной схеме ранее скомпилированной библиотеки. При этом достигается существенный прирост скорости расчета, а также возможность по одному и тому же алгоритму обсчитывать несколько типовых объектов. Кроме этого, блок позволяет вести расчет в несколько потоков. |
| D:\3vs\1\automatics\general\images\3076.png | Самогенер-ирующийся блок | Блок осуществляет автоматическую генерацию кода СИ |
| D:\3vs\1\automatics\general\images\3077.png | Удалённый прибор | Блок позволяет осуществлять отладку удаленного прибора, подключенного по протоколу TCP |

## Закладка «Данные»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\1001.png | Временной график | Блок, реализующий график величины в зависимости от модельного времени задачи. |
| D:\3vs\1\automatics\general\images\1010.png | Фазовый портрет | Блок, реализующий вывод фазовой диаграммы, т.е. для каждой точки значение координаты X берётся с первого входа, а Y – со второго входа. |
| D:\3vs\1\automatics\general\images\1011.png | График Y от X | Блок реализующий вывод зависимости вектора Y(второй вход) от вектора X (первый вход). |
| D:\3vs\1\automatics\general\images\0016.png | В файл | Производит запись данных в текстовый файл. Каждая строка текстового файла выводиться в формате:  <время> <величина 1> … <величина n> |
| D:\3vs\1\automatics\general\images\0017.png | Из файла | Блок возвращает на выходе значения таблично заданных от времени функций, вычисленных методом линейной интерполяции. Таблица задаются в виде текстового файла, где формат каждой строки должен быть следующим:  <время> <значение ф-ции 1> …. <ф-ции N>  Пример:  0.1 1 2 6 9  0.2 6 3 7 10  0.3 3 5 3 9  Файл считывается один раз при инициализации блока, затем интерполяция производится на основании загруженных данных.  Параметры:   * Число функций в файле – к-во столбцов значений функций в текстовом файле; * Имя файла данных – относительный путь к файлу таблицы;   К-т умножения аргумента – коэффициент на который умножается модельное время при интерполяции; |
|  | Из таблицы | Блок возвращает на выходе значения таблично заданных от агрумента функций, вычисленных методом линейной интерполяции. Таблица задаются в виде текстового файла, где формат каждой строки должен быть следующим:  <аргумент> <значение ф-ции 1> …. <ф-ции N>  Пример:  0.1 1 2 6 9  0.2 6 3 7 10  0.3 3 5 3 9  Файл считывается один раз при инициализации блока, затем интерполяция производится на основании загруженных данных.  Параметры:   * Число функций в файле – к-во столбцов значений функций в текстовом файле; * Имя файла данных – относительный путь к файлу таблицы; * К-т умножения аргумента – коэффициент на который умножается модельное время при интерполяции; |
|  | Двумерная таблица из файла | Блок производит двумерную интерполяцию входных аргументов по заданной таблице.  Первый вход блока – значение аргумента по строкам таблицы.  Второй вход блока – значение аргумента по столбцам таблицы.  Формат таблицы следующий:  <значения аргументов по столбцам>  <аргумент по строке 1> <значения функции>  <аргумент по строке 2> <значения функции>  ….  <аргумент по строке N> <значения функции>  Пример:  1 2 3  1 0.1 0.2 0.3  2 0.6 0.8 0.9  Параметры блока:   * Имя файла с таблицей – относительный путь к файлу таблицы; |
|  | Считывание строк из файла | Блок производит считывание новой строки из текстового файла на каждом шаге расчёта блока.  Таблица задаются в виде текстового файла, где формат каждой строки должен быть следующим:  <значение ф-ции 1> …. <ф-ции N>  Пример:  1 2 6 9  6 3 7 10  3 5 3 9  Параметры блока:   * Имя файла с таблицей – относительный путь к файлу таблицы; * Число столбцов в таблице – соответствует размерности выхода; * Цикл считывания строк – шаг с которым блок считывает новую строку; |
|  | Таблица данных из файла | Блок производит считывание таблицы данных из файла. На каждый выход блока подаётся значение соответствующего столбца таблицы. Блок производит считывание один раз при старте задачи.  Параметры блока:   * Имя файла с таблицей – относительный путь к файлу таблицы; * Число столбцов в таблице – соответствует количеству выходов, размерности выходов равны количеству строк в таблице; |
| D:\3vs\1\automatics\general\images\0116.png | Импорт данных из файла обмена | Блок предназначен для считывания данных из бинарного файла обмена. Считывание производится с заданным шагом. Количество считываемых сигналов равно сумме размерностей выходов.  Формат файла обмена следующий:  [к-во данных N – int32][значение 1 – real 8]….[значение N – real 8]  Параметры блока:   * Массив размерностей выходов – размерности входов блока; * Массив начальных значений выходов – значения выходов при t=0 если свойство «Начальное состояние = Из массива»; * Файл импортируемых данных - относительный путь к файлу обмена данными; * Синхронизировать обмен данными – если флаг стоит «Да», то блок автоматически производит задержку расчёта, если к-во данных в файле обмена = 0. После считывания данный блок запишет в файл обмена количество данных = 0. * Шаг обмена данными – шаг с которым производится считывание данных из файла обмена; * Останов по флагу в файле обмена – если флаг стоит «Да», то блок автоматически производит остановку расчёта, если к-во данных в файле обмена < 0; * Начальное состояние – флаг, указывающий откуда необходимо брать начальные значения выходов блока; * Интервал ожидания – интервал задержки в миллисекундах при синхронизации блока. |
| D:\3vs\1\automatics\general\images\0117.png | Экспорт данных в файл обмена | Блок предназначен для записи данных в бинарный файл обмена данными.  Формат файла обмена следующий:  [к-во данных N – int32][значение 1 – real 8]….[значение N – real 8]  Количество считываемых сигналов равно сумме размерностей выходов.  Параметры блока:   * Количество входов * Файл экспортируемых данных - относительный путь к файлу обмена данными; * Посылать сообщение при останове – при останове расчёта в файл обмена будет записано количество данных = -1 (FFFF); * Шаг обмена данными – шаг с которым производится считывание данных из файла обмена; |
| D:\3vs\1\automatics\general\images\0119.png | Экспорт ненулевых данных в файл обмена | Блок предназначен для записи данных в бинарный файл обмена данными.  Формат файла обмена следующий:  [к-во данных N – int32][индекс значения 1 – real 8][значение 1 – real 8]….[индекс значения N – real 8][значение N – real 8]  Количество считываемых сигналов равно количеству ненулевых данных на входах блока.  Параметры блока:   * Количество входов * Файл экспортируемых данных - относительный путь к файлу обмена данными; * Посылать сообщение при останове – при останове расчёта в файл обмена будет записано количество данных = -1 (FFFF); * Шаг обмена данными – шаг с которым производится считывание данных из файла обмена; |
| D:\3vs\1\automatics\general\images\0128.png | Сервер tcp-ip | Блок производит двунаправленный обмен данными по протоколу TCP-IP.  Параметры:   * Количество входов; * Массив размерностей выходов; * Массив начальных значений выходов; * Шаг обмена данными– шаг с которым производится запись и считывание данных в буферы блока; * Останов по команде клиента – если флаг установлен, то при команде от клиента расчёт будет остановлен; * Интервал ожидания завершения потока – время задержки до завершения потока для клиента; * Порт tcp-ip – номер TCP-порта обмена. |
| D:\3vs\1\automatics\general\images\0130.png | Клиент tcp-ip | Блок производит двунаправленный обмен данными по протоколу TCP-IP.  Параметры:   * Количество входов; * Массив размерностей выходов; * Массив начальных значений выходов; * Шаг обмена данными– шаг с которым производится запись и считывание данных в буферы блока; * Посылать команду остановки серверу – если флаг установлен, то при остановке расчёта будет послана команда остановки серверу; * Синхронизировть модельное время – если флаг установлен, то блок автоматически будет производить задержку расчёта для согласования модельного времени на клиенте и сервере; * Порт tcp-ip – номер TCP-порта обмена. * Адрес сервера – IP-адрес или имя сервера, где находится модель с которой производится обмен данными; |
| D:\3vs\1\automatics\general\images\0131.png | Сервер UDP | Блок производит обмен данными по протоколу UDP |
| D:\3vs\1\automatics\general\images\0132.png | Клиент UDP | Блок производит обмен данными по протоколу UDP |
| D:\3vs\1\automatics\general\images\1020.png | Чтение из списка сигналов | Производит считывание данных из локального списка сигналов проекта (или из списка сигналов подключенной БД). На выходе блока возвращается вектор, размерность которого равна сумме размерностей сигналов считываемых блоком.  Параметры:   * Имена сигналов – список имён считываемых блоком сигналов через точку с запятой. В списке имён допустимо использовать форматирование имён и создание запросов к БД; * Рассчитывать с задержкой на шаг; |
| D:\3vs\1\automatics\general\images\1021.png | Запись в список сигналов | Производит запись данных в локальный список сигналов проекта (или список сигналов подключенной БД).  Параметры:   * Имена сигналов – список имён считываемых блоком сигналов через точку с запятой. В списке имён допустимо использовать форматирование имён и создание запросов к БД; |
| D:\3vs\1\automatics\general\images\3018.png | Выход алгоритма | Производит запись данных в локальный список сигналов проекта (или список сигналов подключенной БД). При двойном щелчке по блоку появляется редактор списка записываемых переменных. Имя переменной задаётся в поле «Имя сигнала». Каждому входу блока соответствует свой сигнал. |
| D:\3vs\1\automatics\general\images\3019.png | Чтение сигналов | Производит считывание данных из локального списка сигналов проекта (или из списка сигналов подключенной БД). При двойном щелчке по блоку появляется редактор списка считываемых переменных. Имя переменной задаётся в поле «Имя сигнала». Каждому входу блока соответствует свой сигнал. |
| D:\3vs\1\automatics\general\images\3021.png | Чтение сигналов (векторный) | Производит считывание данных из локального списка сигналов проекта (или из списка сигналов подключенной БД). При двойном щелчке по блоку появляется редактор списка считываемых переменных. Имя переменной задаётся в поле «Имя сигнала». На выходе блока создаётся вектор, размерность которого равна сумме размерностей считываемых сигналов. |
| D:\3vs\1\automatics\general\images\3020.png | Выход алгоритма (векторный) | Производит запись данных в локальный список сигналов проекта (или список сигналов подключенной БД). При двойном щелчке по блоку появляется редактор списка записываемых переменных. Имя переменной задаётся в поле «Имя сигнала». Размерность входного вектора равна сумме размерностей записываемых сигналов. |
| D:\3vs\1\automatics\general\images\3001.png | Входной контакт | Реализует считывание переменной и передачу её на выход блока. В режиме локальной отладки эквивалентен константе. При удалённой отладке блок считывает переменную из исполнительной системы. |
| D:\3vs\1\automatics\general\images\3002.png | Выходной контакт | Реализует запись переменной исполнительной системы со входа блока |
| D:\3vs\1\automatics\general\images\3049.png | Датчик | Блок аналогичен блоку [«Входной контакт»](../odt/3001.htm#Графический%20объект1%7Cgraphic), с добавлением отображения измеряемой величины. Реализует считывание переменной, передачу её на выход блока, а также отображение значения переменной на схемное окно в заданном пользователем формате. В режиме локальной отладки эквивалентен константе. При удалённой отладке блок считывает переменную из исполнительной системы. |
| D:\3vs\1\automatics\general\images\3022.png | Клиент OPC | Производит считывание и запись сигналов в другие программы с использованием интерфейса обмена данными по протоколу OPC. Размерность входа равна сумме размерностей сигналов, заданных в соответствующем параметре блока. Размерность выхода равна сумме размерностей сигналов, заданных в соответствующем параметре блока.  Параметры:   * Шаг обмена данными, сек; * Имя OPC-сервера – имя COM-сервера, с которым производится обмен данными; * Задержка при старте, мсек – задержка в мсек при подсоединении к COM-серверу; * Имена считываемых параметров – список имён сигналов в OPC-сервере, которые блок читает (разделитель ; или перевод каретки); * Имена записываемых параметров – список имён сигналов в OPC-сервере, которые блок записывает (разделитель ; или перевод каретки); * Размерности считываемых параметров; * Размерности записываемых параметров; |
| D:\3vs\1\automatics\general\images\0133.png | Связь с расчётным кодом РАСНАР | Блок предназначен для организации обмена данными текущего проекта с расчетным кодом РАСНАР, выполняющим моделирование объекта управления. |
| D:\3vs\1\automatics\general\images\0134.png | Связь с расчётным кодом TPP | Блок предназначен для организации обмена данными текущего проекта с расчетным кодом TPP, выполняющим удалённо моделирование объекта управления. |
| D:\3vs\1\automatics\general\images\0135.png | Контроль версии модели | Блок осуществляет функции автоматического контроля версий схемы проекта и сгенерированного кода |

## Закладка «Ключи»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0082.png | Ключ-0 | Блок векторизован и реализует функцию управляемого ключа по алгоритму:  y ( t ) = Y0, если x ( t ) < К;  y ( t ) = x ( t ), если x ( t ) >= К,  где x(t) - вектор входных сигналов;  y(t) - вектор выходных сигналов;  К - вектор "порогов" (параметров уставок);  Y0 - вектор выходных сигналов блока при входных сигналах, меньших пороговых значений. |
| D:\3vs\1\automatics\general\images\0083.png | Ключ-1 | Блок векторизован и реализует функцию управляемого ключа по алгоритму:  y1 ( t ) = x ( t ), если x ( t ) >= К;  y1 ( t ) = Y0, если x ( t ) < К;  y2 ( t ) = x ( t ), если x ( t ) < К;  y2 ( t ) = Y0, если x ( t ) >= К,  где x(t) - вектор входных сигналов;  y1(t) - вектор сигналов на 1-ом выходном порте;  y2(t) - вектор сигналов на 2-ом выходном порте;  К - вектор пороговых значений (уставок), определяющих условия перекоммутации сигналов на выходных портах блока;  Y0 - вектор сигналов на 1-ом выходном порте блока, при входных сигналах, меньших пороговых значений К или вектор сигналов на 2-ом выходном порте блока, при входных сигналах, больших пороговых значений К. |
| D:\3vs\1\automatics\general\images\0084.png | Ключ-2 | Блок векторизован и реализует функцию управляемого ключа по алгоритму:  y1 ( t ) = x ( t ), если x ( t ) <= К1;  y1 ( t ) = Y10, если x ( t ) > К1;  y2 ( t ) = x ( t ), если x ( t ) > К2;  y2 ( t ) = Y20, если x ( t ) <= К2,  где x(t) - вектор входных сигналов;  y1(t) - вектор сигналов на 1-ом выходном порте;  y2(t) - вектор сигналов на 2-ом выходном порте;  К1, K2 - векторы пороговых значений (уставок), определяющих условия перекоммутации сигналов на выходных портах блока;  Y10 - вектор сигналов на 1-ом выходном порте блока, при входных сигналах, больших пороговых значений К1;  Y20 - вектор сигналов на 2-ом выходном порте блока, при входных сигналах, меньших пороговых значений К2. |
| D:\3vs\1\automatics\general\images\0085.png | Ключ-3 | Блок векторизован и реализует функцию управляемого ключа по алгоритму:  y ( t ) = x1 ( t ), если x2 ( t ) < К;  y ( t ) = x3 ( t ), если x2 ( t ) >= К,  где x1(t) - вектор входных сигналов на 1-ом входном порте;  x2(t) - вектор управляющих сигналов на 2-ом входном порте;  x3(t) - вектор входных сигналов на 3-ом входном порте;  y(t) - вектор сигналов на выходном порте;  К - вектор пороговых значений управляющих сигналов (обычно логических уставок), определяющих условия перекоммутации входных сигналов на выход блока; |
| D:\3vs\1\automatics\general\images\0086.PNG | Ключ-4(t) | Блок **векторизован** и реализует функцию управляемого ключа по алгоритму:  y ( t ) = Y0, если t < Т0;  y ( t ) = x ( t ), если t >=Т0,  где **x(t)** - вектор входных сигналов;  **y(t)** - вектор выходных сигналов;  **Т0** - вектор времен “замыкания” соответствующих входов и выходов;  **Y0 -** вектор выходных сигналов блока при временах, меньших **Т0**.  Параметры:   * Уставка по времени – T0 * Подпороговое зн-е на выходе – Y0 |
| D:\3vs\1\automatics\general\images\0087.PNG | Ключ-5(t) | Блок **векторизован** и реализует функцию управляемого ключа по алгоритму:  **y1 ( t ) = x ( t ), если t < x ( t );**  **y1 ( t ) = Y0, если t >= x ( t );**  **y2 ( t ) = Y0, если t < x ( t );**  **y2 ( t ) = x ( t ), если t >= x ( t ).**  где **x(t)** - вектор входных сигналов;  **y1(t)** - вектор сигналов на 1-ом выходном порте;  **y2(t)** - вектор сигналов на 2-ом выходном порте;  **Y0 -** вектор сигналов на **1**-ом выходном порте блока при **t >= x(t)** и, соответственно, вектор сигналов на **2**-ом выходном порте блока при **t < x(t).**  Параметры:   * Подпороговое зн-е на выходе – Y0 |
| D:\3vs\1\automatics\general\images\0088.PNG | Ключ-6(t) | Блок **векторизован** и реализует функцию управляемого ключа по алгоритму:  **y1 ( t ) = x ( t ), если t < Т10;**  **y1 ( t ) = Y10, если t >= Т10;**  **y2 ( t ) = Y20, если t < Т20;**  **y2 ( t ) = x ( t ), если t >= Т20;**  где **x(t)** - вектор входных сигналов;  **y1(t)** - вектор сигналов на 1-ом выходном порте;  **y2(t)** - вектор сигналов на 2-ом выходном порте;  **Y10 -** вектор сигналов на **1**-ом выходном порте блока при **t >= Т10;**  **Y20 -** вектор сигналов на **2**-ом выходном порте блока при **t =< Т20.**  Параметры:   * Уставка первого выхода – T10 * Уставка второго выхода – T20 * Начальное зн-е 1 выхода – Y10 * Начальное зн-е 2-го выхода – Y20 |
| D:\3vs\1\automatics\general\images\0089.PNG | Ключ-7(t) | Блок **векторизован** и реализует функцию управляемого ключа по алгоритму:  **y ( t ) = x1 ( t ), если t < Т0;**  **y ( t ) = x2 ( t ), если t >= Т0,**  где **x1(t)** - вектор сигналов на 1-ом входном порте;  **х2(t)** - вектор сигналов на 1-ом входном порте;  **y(t)** - вектор сигналов на выходном порте;  **Т0 -** вектор времен перекоммутации входов с выходом блока.  Параметры:   * Подпороговое зн-е на выходе – Y0 |
| D:\3vs\1\automatics\general\images\0090.png | Ключ интегратора | Блок реализует функцию управляемого ключа для интегрирующего привода (типа интегратора или инерционно-интегрирующего звена) по следующему алгоритму:  y ( t ) = x1 ( t ), если a1 < x2 ( t ) < a2;  y ( t ) = 0, если x2 ( t ) <= a1 и x1 ( t ) <= 0,  или x2 ( t ) >= a2 и x1 ( t ) >= 0;  y ( t ) = x1 ( t ), если x2 ( t ) <= a1 и x1 ( t ) > 0,  или x2 ( t ) >= a2 и x1 ( t ) < 0. |
| D:\3vs\1\automatics\general\images\0091.png | Ключ управляемый нормально разомкнутый тип 1 | Блок векторизован и реализует функцию управляемого ключа по алгоритму:  где x(t) — входной сигнал блока (или вектор сигналов), y(t) — выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок сверху. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ разомкнут, при подаче управляющего сигнала ключ замкнут |
| D:\3vs\1\automatics\general\images\0092.png | Ключ управляемый нормально разомкнутый тип 2 | Блок реализует функцию управляемого ключа по алгоритму:  где x(t) — входной сигнал блока (или вектор сигналов), y(t) — выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок снизу. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ разомкнут, при подаче управляющего сигнала ключ замкнут |
| D:\3vs\1\automatics\general\images\0093.png | Ключ управляемый нормально замкнутый тип 1 | Блок реализует функцию управляемого ключа по алгоритму:  где x(t) — входной сигнал блока (или вектор сигналов), y(t) — выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок сверху. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ замкнут, при подаче управляющего сигнала ключ разомкнут |
| D:\3vs\1\automatics\general\images\0094.png | Ключ управляемый нормально замкнутый тип 2 | Блок реализует функцию управляемого ключа по алгоритму:  где x(t) — входной сигнал блока (или вектор сигналов), y(t) — выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок снизу. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ замкнут, при подаче управляющего сигнала ключ разомкнут |
| D:\3vs\1\automatics\general\images\0095.png | Ключ управляемый нормально разомкнутый тип 3 | Блок реализует функцию управляемого ключа по алгоритму:  где x₁(t), x₂(t) — входные сигналы блока (или векторы сигналов), y₁(t), y₂(t) — выходные сигналы блока (или векторы сигналов). Управляющий сигнал входит в блок сверху. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ разомкнут, при подаче управляющего сигнала ключ замкнут |
| D:\3vs\1\automatics\general\images\0096.png | Ключ управляемый нормально разомкнутый тип 4 | Блок реализует функцию управляемого ключа по алгоритму:  где x₁(t), x₂(t) — входные сигналы блока (или векторы сигналов), y₁(t), y₂(t) — выходные сигналы блока (или векторы сигналов). Управляющий сигнал входит в блок снизу. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ разомкнут, при подаче управляющего сигнала ключ замкнут |
| D:\3vs\1\automatics\general\images\0097.png | Ключ управляемый нормально замкнутый тип 3 | Блок реализует функцию управляемого ключа по алгоритму:  где x₁(t), x₂(t) — входные сигналы блока (или векторы сигналов), y₁(t), y₂(t) — выходные сигналы блока (или векторы сигналов). Управляющий сигнал входит в блок сверху. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ замкнут, при подаче управляющего сигнала ключ разомкнут |
| D:\3vs\1\automatics\general\images\0098.png | Ключ управляемый нормально замкнутый тип 4 | Блок реализует функцию управляемого ключа по алгоритму:  где x₁(t), x₂(t) — входные сигналы блока (или векторы сигналов), y₁(t), y₂(t) — выходные сигналы блока (или векторы сигналов).  Управляющий сигнал входит в блок снизу. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ замкнут, при подаче управляющего сигнала ключ разомкнут |
| D:\3vs\1\automatics\general\images\0099.png | Ключ управляемый перекидной по входам тип 1 | Блок векторизован и реализует функцию управляемого ключа по алгоритму:  где x₁(t), x₂(t) — входные сигналы блока (или векторы сигналов), y(t) — выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок сверху. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ перекинут на первый вход, при подаче управляющего сигнала ключ перекидывается на второй вход |
| D:\3vs\1\automatics\general\images\0100.png | Ключ управляемый перекидной по входам тип 2 | Блок реализует функцию управляемого ключа по алгоритму:  где x₁(t), x₂(t) — входные сигналы блока (или векторы сигналов), y(t)— выходной сигнал блока (или вектор сигналов). Управляющий сигнал входит в блок снизу. В процессе расчета изображение блока меняется, в зависимости от наличия управляющего сигнала: в нормальном положении ключ перекинут на первый вход, при подаче управляющего сигнала ключ перекидывается на второй вход |
| D:\3vs\1\automatics\general\images\0103.png | Коммутатор входных сигналов | Блок представляет собой коммутатор, подключающий к выходу один из аналоговых входных сигналов (X1 или X2). Управление коммутацией происходит с помощью двух дискретных входов (X3 и X4) |
| D:\3vs\1\automatics\general\images\3050.png | Запись по условию | На выход блока передаются входные значения (со входа X), если выполняется условие (если есть логический сигнал на входе IF), иначе на выход передаются сигналы, заданные в свойстве блока |
| D:\3vs\1\automatics\general\images\0104.png | Кнопка | Блок формирует на выходе логический сигнал, равный 0 или 1, в зависимости от своего «состояния». Состояние кнопки управляется однократным нажатием на нее левой кнопкой мыши в процессе расчета |
| D:\3vs\1\automatics\general\images\0107.png | Выделение минимума с формированием признака | Блок пропускает на аналоговый выход минимальный из двух аналоговых входных сигналов либо повторяет значение сигнала, если оно одинаково для обоих входов. Дискретный выход признака становится равен 1, если минимальным является значение на втором входе. В противном случае признак равен 0 |
| D:\3vs\1\automatics\general\images\0108.png | Выделение максимума с формированием признака | Блок пропускает на аналоговый выход максимальный из двух аналоговых входных сигналов либо повторяет значение сигнала, если оно одинаково для обоих входов. Дискретный выход признака становится равен 1, если максимальным является значение на втором входе. В противном случае признак равен 0. Пример работы блока в разных случаях представлен на рисунке |
| D:\3vs\1\automatics\general\images\0109.png | Задатчик с варьируемым значением | Блок передает на аналоговый выход значение первого входа (x1), если разрешена трансляция (если вход x2 имеет значение 1) и если значение сигнала x1 входит в диапазон между максимальным и минимальным значением, задаваемым соответственно сигналами на входах x3 и x4. Если величина сигнала не входит в диапазон x4÷x3, то сигнал на выходе приобретает соответственно значение либо верхнего предела (x3), либо нижнего (x4). В случае, если трансляция запрещена (x2=0), то на выходе блока фиксируется то значение, которое было на момент запрещения трансляции |

## Закладка «Логические»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0060.png | Оператор И | Реализует операцию логического И.  Параметры:   * Количество портов |
| D:\3vs\1\automatics\general\images\0061.png | Оператор ИЛИ | Реализует операцию логического ИЛИ.  Параметры:   * Количество портов |
| D:\3vs\1\automatics\general\images\0062.png | Оператор НЕ | Реализует операцию логического НЕ. |
| D:\3vs\1\automatics\general\images\0163.png | Логические операции | Реализует произвольную логическую операцию.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. * Тип операнда – тип логической операции. |
| D:\3vs\1\automatics\general\images\0063.png | Операция БОЛЬШЕ | Возвращает логическую единицу, если первый вход больше второго.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. |
| D:\3vs\1\automatics\general\images\0064.png | Операция МЕНЬШЕ | Возвращает логическую единицу, если первый вход меньше второго.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. |
| D:\3vs\1\automatics\general\images\0065.png | Операция РАВНО | Возвращает логическую единицу, если первый вход равен второму.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. |
| D:\3vs\1\automatics\general\images\0066.png | Операция НЕ РАВНО | Возвращает логическую единицу, если первый вход не равен второму.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. |
| D:\3vs\1\automatics\general\images\0067.png | БОЛЬШЕ ИЛИ РАВНО | Возвращает логическую единицу если первый вход больше или равен второму.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. |
| D:\3vs\1\automatics\general\images\0068.png | МЕНЬШЕ ИЛИ РАВНО | Возвращает логическую единицу если первый вход меньше или равен второму.  Параметры:   * Тип второго операнда – указывает тип второго операнда. Если вектор – то делается поэлементная операция. Скаляр – производиться операция с каждым элементов первого входа и значением второго входа. |
| D:\3vs\1\automatics\general\images\0164.png | XOR | Производит операцию поэлементного XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ) для входного вектора. |
| D:\3vs\1\automatics\general\images\0165.png | NOT XOR | Производит операцию поэлементного NOT XOR (ИСКЛЮЧАЮЩЕЕ НЕ-ИЛИ) для входного вектора. |
| D:\3vs\1\automatics\general\images\0160.png | M из N | Возвращает логическую единицу, если количество входов, на которых значение равно логической единице больше заданного количества.  Параметры:   * Количество портов * Число входных True для срабатывания |
| D:\3vs\1\automatics\general\images\0161.png | M из N поэлементное | Блок является аналогом логического блока M из N, векторизован и выполняет операцию мажорирования по следующему алгоритму:  Таким образом, на выходе по каждому из i (по каждому из сигналов входных векторов) формируется логическая 1, если C ≥ M (количество ненулевых i-ых сигналов во входных векторах превышает параметр M). Если C < M то на выходе формируется логический 0 (по каждому из i-ых элементов). |
| D:\3vs\1\automatics\general\images\0166.png | Один из многих | Блок векторизован и реализует специальный алгоритм формирования логических сигналов в векторе выхода в зависимости от изменения типа логического сигнала в векторе входа.  Алгоритм работы блока:  До начала процесса моделирования в диалоговом окне блока только один сигнал (например, N-ый) в векторе выхода может быть задан как сигнал типа TRUE (активный), независимо от типа соответствующего (N-го) сигнала в векторе входа. Все остальные выходные сигналы считаются сигналами типа FALSE (неактивный), независимо от типа соответствующих сигналов в векторе входа.  Если в процессе моделирования какой-то логический сигнал (любой, кроме N-го) в векторе входа изменил свой тип с FALSE на TRUE, то соответствующий сигнал в векторе выхода изменяет свой тип на TRUE (активный) и одновременно N-ый сигнал в векторе выхода становится типа FALSE (неактивный). |
| D:\3vs\1\automatics\general\images\0167.png | Один из многих по выбору | Блок реализует специальный алгоритм формирования вектора выходных логических сигналов в зависимости от значения входного скалярного сигнала, округленного до целого.  Алгоритм работы блока:  Если в процессе моделирования скалярный входной сигнал при округлению до целого равен М (1 =< M =< N, где N – размерность вектора выхода), то в векторе выхода тип TRUE (активный) имеет только М-ый сигнал, а все остальные сигналы в векторе выхода соответствуют типу FALSE (неактивный).  Если значение входного скалярного сигнала, округленное до целого, меньше 1 или больше N, то все сигналы в векторе выхода соответствуют типу FALSE. |
| D:\3vs\1\automatics\general\images\0168.png | Счётчик | Производит подсчёт входных импульсов на входе.  Параметры:   * Нижний предел нечувствительности * Верхний предел нечувствительности * Счётчик если вход (Внутри диапазона, Вне диапазона) |
| D:\3vs\1\automatics\general\images\3065.png | Векторное И | Блок является аналогом логического блока «И», векторизован и работает по следующему алгоритму:  ,  где y(t) – выходной скалярный сигнал блока, – i-ый элемент j-ого входного векторного сигнала. |
| D:\3vs\1\automatics\general\images\3064.png | Векторное ИЛИ | Блок является аналогом логического блока «ИЛИ», векторизован и работает по следующему алгоритму:  ,  где x(t) – выходной скалярный сигнал блока, – i-ый элемент j-ого входного векторного сигнала. |
|  | Логическое И с пассивным входом | Блок является аналогом логического блока «И», с выделением активного (первого) и т. н. «пассивного» (второго) порта. Результат работы блока зависит от последовательности во времени моментов появления логических единиц на входах, а именно:  если первой появилась единица на первом (активном) входе, а единица на втором (пассивном) входе появилась после первой, то выход блока становится равен 1.  если первой появилась единица на втором (пассивном) входе, а единица на первом входе «пришла» после второй, то выход блока не устанавливается в 1. |

## Закладка «Триггеры»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\1201.png | RS-триггер с приоритетом по сбросу | Реализует RS-триггер с приоритетом по сбросу, т.е. если уровень логической единицы присутствует на входах R и S, то выход принимает значение логического нуля.  Параметры:   * Начальные условия – начальные значение выхода. * Добавить инверсный выход – указывает наличие инверсного выхода. |
| D:\3vs\1\automatics\general\images\1202.png | RS-триггер с приоритетом по установке | Реализует RS-триггер с приоритетом по сбросу, т.е. если уровень логической единицы присутствует на входах R и S, то выход принимает значение логической единицы.  Параметры:   * Начальные условия – начальные значение выхода. * Добавить инверсный выход – указывает наличие инверсного выхода. |
| D:\3vs\1\automatics\general\images\1203.png | Триггер T | Реализует счётный триггер, изменяющий своё состояние при изменении значение счётного входа.  Параметры:   * Начальные условия – начальные значение выхода. * Добавить инверсный выход – указывает наличие инверсного выхода. * Тип счётного входа (по фронту, по спаду) – условие срабатывания счётного входа. |
| D:\3vs\1\automatics\general\images\1204.png | Триггер TR | Реализует RS-триггер с приоритетом по сбросу, т.е. если уровень логической единицы присутствует на входах R и S, то выход принимает значение логического нуля, и счётным входом.  Параметры:   * Начальные условия – начальные значение выхода. * Добавить инверсный выход – указывает наличие инверсного выхода. * Тип счётного входа (по фронту, по спаду) – условие срабатывания счётного входа. |
| D:\3vs\1\automatics\general\images\1205.png | Триггер TS | Реализует RS-триггер с приоритетом по сбросу, т.е. если уровень логической единицы присутствует на входах R и S, то выход принимает значение логической единицы, и счётным входом.  Параметры:   * Начальные условия – начальные значение выхода. * Добавить инверсный выход – указывает наличие инверсного выхода. * Тип счётного входа (по фронту, по спаду) – условие срабатывания счётного входа. |
| D:\3vs\1\automatics\general\images\1206.png | Определение первого события | Является вариантом реализации D-триггера. Запоминает какое из событий (логическая единица), поданное на D-входы, возникло первым. После возникновения события, на соответствующем выходе формируется логическая единица и состояния всех выходов фиксируется до возникновения сигнала сброса R, установка которого сбрасывает все выходы в 0, а снятие возвращает блок в работу. |

## Закладка «Задержки и импульсы»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
|  | Задержка по включению | Реализует задержку входного сигнала на заданное время при изменении входного сигнала от логического нуля до единицы.  Параметры:   * Массив времён подтверждения – время задержки включения. * Способ задания времени подтверждения – если указан способ задания «Вход», то время задержки берётся со второго входа. |
|  | Задержка по включению и выключению | Реализует задержку входного сигнала на заданное время при любом изменении входного сигнала.  Параметры:   * Массив времён подтверждения – время задержки включения. * Способ задания времени подтверждения – если указан способ задания «Вход», то время задержки берётся со второго входа. |
|  | Задержка по выключению | Реализует задержку входного сигнала на заданное время при изменении входного сигнала от логической единицы до нуля.  Параметры:   * Массив времён подтверждения – время задержки включения. * Способ задания времени подтверждения – если указан способ задания «Вход», то время задержки берётся со второго входа. |
|  | Импульс по фронту | Формирует импульс величиной 1, длительностью в один шаг интегрирования, при изменении входной величины от нуля до единицы. |
|  | Импульс по срезу | Формирует импульс величиной 1, длительностью в один шаг интегрирования, при изменении входной величины от единицы до нуля. |
|  | Импульс по фронту или срезу | Формирует импульс величиной 1, длительностью в один шаг интегрирования, при изменении входной величины. |
|  | Импульс | Формирует импульс заданной длительности при изменении входа от нуля до единицы.  Параметры:   * Длительность импульса * Длительность задается через – способ задания длительности импульса (через вход или через параметр блока) |
|  | Импульс длительностью не более заданной | Формирует импульс заданной длительности не более заданной при изменении входа от нуля до единицы.  Параметры:   * Длительность импульса * Длительность задается через – способ задания длительности импульса (через вход или через параметр блока) |
|  | Импульс с пролонгированием | Формирует импульс заданной длительности не менее заданной при изменении входа от нуля до единицы.  Параметры:   * Длительность импульса * Длительность задается через – способ задания длительности импульса (через вход или через параметр блока) |
|  | Временное подтверждение | Изменяет значение на выходе при выполнении условия более заданного времени.  Параметры:   * Вектор времён подтверждения. * Подтверждение требуется – условие, при котором формируется импульс. * Тип выхода – тип выхода блока. Обычный или инверсный. |
|  | Одновибратор | Формирует импульс заданной длительности при изменении входной величины до заданного порога.  Параметры:   * Длительность импульса * Порог срабатывания. * Тип выхода – тип выхода блока. Обычный или инверсный. |

## Закладка «Релейные»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\3038.png | Обмотка реле | Субмодель, реализующая передачу сигнала от входа данного блока к контактам реле.  Параметры:   * Имя реле – имя, по которому контакты ассоциируются с данным блоком. |
| D:\3vs\1\automatics\general\images\3039.png | Двухпозиционное реле (SET) | Субмодель, реализующая установку состояния логической единицы двухпозиционного реле.  Параметры:   * Имя реле – имя, по которому контакты ассоциируются с данным блоком. |
| D:\3vs\1\automatics\general\images\3040.png | Двухпозиционное реле (RESET) | Субмодель, реализующая установку состояние логического нуля двухпозиционного реле.  Параметры:   * Имя реле – имя, по которому контакты ассоциируются с данным блоком. |
| D:\3vs\1\automatics\general\images\3068.png | Двухпозиционное реле (RS) | Блок представляет собой субмодель, содержащую RS-триггер с приоритетом по сбросу |
| D:\3vs\1\automatics\general\images\3041.png | Замыкающий контакт реле | Реализует операцию логического И для входа блока и значения, поданного на соответствующую управляющую обмотку реле.  Параметры:   * Имя реле – имя, по которому определяется управляющее воздействие. |
| D:\3vs\1\automatics\general\images\3042.png | Размыкающий контакт реле | Реализует операцию логического И-НЕ для входа блока и значения, поданного на соответствующую управляющую обмотку реле.  Параметры:   * Имя реле – имя, по которому определяется управляющее воздействие. |
| D:\3vs\1\automatics\general\images\3043.png | Переключающий контакт реле | Реализует операцию переключения входных контактов в зависимости от значения на управляющей обмотке.  Параметры:  Имя реле – имя, по которому определяется управляющее воздействие. |
| D:\3vs\1\automatics\general\images\3044.png | Замыкающий контакт реле с задержкой по замыканию | Реализует операцию логического И с задержкой по включению для входа блока и значения, поданного на соответствующую управляющую обмотку реле.  Параметры:   * Имя реле – имя, по которому определяется управляющее воздействие. * Время задержки – время задержки при включении контакта; |
| D:\3vs\1\automatics\general\images\3045.png | Замыкающий контакт реле с задержкой по размыканию | Реализует операцию логического И с задержкой по выключению для входа блока и значения, поданного на соответствующую управляющую обмотку реле.  Параметры:   * Имя реле – имя, по которому определяется управляющее воздействие. * Время задержки – время задержки при выключении контакта; |
| D:\3vs\1\automatics\general\images\3046.png | Размыкающий контакт реле с задержкой по замыканию | Реализует операцию логического И-НЕ с задержкой по выключению обмотки (замыканию) для входа блока и значения, поданного на соответствующую управляющую обмотку реле.  Параметры:   * Имя реле – имя, по которому определяется управляющее воздействие. * Время задержки – время задержки при замыкании контакта; |
| D:\3vs\1\automatics\general\images\3047.png | Размыкающий контакт реле с задержкой по размыканию | Реализует операцию логического И-НЕ с задержкой по включению обмотки (размыканию) для входа блока и значения, поданного на соответствующую управляющую обмотку реле.  Параметры:   * Имя реле – имя, по которому определяется управляющее воздействие. * Время задержки – время задержки при размыкании контакта; |
| D:\3vs\1\automatics\general\images\3067.png | Внешний контакт реле | Реализует операцию логического И для входа блока и значения, указанного в параметрах блока (аналог управляющей обмотки реле для блока Замыкающий контакт реле). Таким образом, имя контакта указывается непосредственно внутри субмодели |
| D:\3vs\1\automatics\general\images\3069.png | Внешний размыкающий контакт реле | Реализует операцию логического И-НЕ для входа блока и значения, указанного в параметрах блока (аналог управляющей обмотки реле для блока Размыкающий контакт реле). Таким образом, имя контакта указывается непосредственно внутри субмодели. |

## Закладка «Дискретные»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0202.png | Запаздывание на период квантования | Блок реализует дискретное запаздывание входного сигнала на один период квантования:  **y(t) = u(t[k-1]),**  где **y(t)** - значение выходного сигнала в текущий момент модельного времени, **u(t[k-1])** - значение входного сигнала в предпоследний момент дискретизации **t[k-1].**  **Примечание:** Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока; * Начальные условия – значение выхода при t=0; |
| D:\3vs\1\automatics\general\images\0201.png | Экстраполятор | Блок **векторизован** и реализует экстраполяцию нулевого порядка:  **y(t) = u(t[k]),**  где **y(t)** - значение выходного сигнала в текущий момент модельного времени, **u(t[k])** - значение входного сигнала в последний момент дискретизации **t[k].**  **Примечания:**  Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока; |
| D:\3vs\1\automatics\general\images\0203.png | Дискретная передаточная ф-я общего вида | Блок реализует дискретную передаточную функцию:    **Обязательные условия: m <= n и an <> 0.**  **Примечания:**  1. Коэффициенты **bj** и **ai** вводятся по *возрастающим* степеням **z.**  2. Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока; * Коэффициенты числителя – вектор b * Коэффициенты знаменателя – вектор a * Начальные условия – значения выходного вектора при t=0; |
| D:\3vs\1\automatics\general\images\0204.png | Дискретная передаточная ф-я обратного аргумента | Блок реализует дискретную передаточную функцию:    **Обязательное условие: a0 <> 0.**  **Примечания:**  1. Коэффициенты **bj** и **ai** вводятся по *убывающим* степеням **z**.  2. Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока; * Коэффициенты числителя – вектор b * Коэффициенты знаменателя – вектор a * Начальные условия – значения выходного вектора при t=0; |
| D:\3vs\1\automatics\general\images\0206.png | Дискретные переменные состояния | Блок реализует описание многомерной линейной дискретной системы в матричной форме:  **x[k+1] = Ax[k] + Bu[k];**  **y[k] = Cx[k] + Du[k],**  где **A**, **B**, **C**, **D** - матрицы: собственная, входа, выхода и обхода, соответственно;  **x** -вектор переменных состояния, [Nx]; **u** - вектор входа, [Nu]; **y** - вектор выхода, [Ny].  Размерность матрицы **А** – [Nx\*Nx].  Размерность матрицы **B** – [Nu\*Nx].  Размерность матрицы **C** – [Nx\*Ny].  Размерность матрицы **D** – [Nu\*Ny].  1-е число в квадратных скобках – количество столбцов,  2-е число – количество строк.  **Примечания:**  1. Если блок имеет 1 вход (Nu = 1) и 1 выход (Ny = 1), D = 0 и, соответственно, 7-я строка – (0).  2. В “сборе” матрицы **A**, **В**, **C** и **D** образуют “прямоугольник”, размерностью:  - строк - (Nx+Ny);  - столбцов - (Nx+Nu).  3. Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока; * Число переменных состояния - Nx * Число входных воздействий - Nu * Число выходов - Ny * Матрица A(Nx\*Nx) * Матрица B(Nu\*Nx) * Матрица C(Nx\*Ny) * Матрица D(Nu\*Ny) * Начальные условия (Nx) – вектор начальных условий, размерностью Nx. |
| D:\3vs\1\automatics\general\images\0207.png | Дискретная производная | Блок **векторизован** и реализует численное определение производной по формуле    где **y(t)** - значение выходного сигнала в текущий момент модельного времени, **u(t[k])** и **u(t[k-1])** - значения входного сигнала в последний и предпоследний моменты дискретизации **t[k]** и **t[k-1].**  **Примечания:**  Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока; * Начальные условия – значение выхода при t=0; |
| D:\3vs\1\automatics\general\images\0209.png | Разность нулевого порядка | Блок **векторизован** и реализует вычисление разности 1-го порядка по формуле  y(t) = u(t) - u(t - dt) ,  где **y(t)** - значение выходного сигнала в текущий момент модельного времени; **u(t-dt)** - значения входного сигнала на предыдущем шаге интегрирования; **dt** – текущий шаг интегрирования.  Параметры:   * Начальные условия – значение выхода при t=0; * Развязка алгебраических петель – флаг указывающий должен ли данный блок производить развязку петель; |
| D:\3vs\1\automatics\general\images\0205.png | Дискретный ПИД-регулятор | Блок реализует передаточную функцию дискретного ПИД-регулятора:    где **Kp**, **Ki**, **Kd** - пропорциональная, интегральная и дифференциальная составляющие; **T** - период квантования.  **Примечание:** Для правильной работы данного блока необходимо задать максимальный шаг интегрирования не больше, чем **T/2**. При интегрировании с постоянным шагом рекомендуется задать шаг таким, чтобы период квантования был кратен шагу интегрирования (**T = k\*h**).  Параметры:   * Период квантования – период квантования расчёта блока T; * Начальные условия – значение выхода при t=0; * Пропорциональная составляющая - Kp * Интегральная составляющая - Ki * Дифференциальная составляющая - Kd |

## Закладка «Функции»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0041.png | Линейная функция | Блок возвращает выходное значение вычисленное по формуле:  Y = a + b\*x,  где a,b – коэффициенты,  x – входное значение.  Параметры блока:   * Свободный член – коэффициент a * Коэффициент при t – коэффициент b   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0042.png | Параболическая функция | Блок возвращает выходное значение вычисленное по формуле:  Y = a0 + a1\*x + a2\*x2,  Где a0, a1, a2 – коэффициенты,  x – входное значение.  Параметры блока:   * Свободный член – коэффициент a0 * Коэффициент при t – коэффициент a1   Коэффициент при t2 - коэффициент a2  Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0043.png | Полиноминальная функция | Блок возвращает выходное значение вычисленное по формуле:  Yi = a1,i + a2,i\*xi + a3,i\*xi2 +....an,i\*xin-1,  где a1 ... an – коэффициенты полинома,  xi – входное значение.  Параметры блока:  i – номер строки матрицы коэффициентов и элемента выходного вектора.  Коэффициенты полинома – матрица коэффициентов a, где в каждой строке заданы значения коэффициентов полинома для соответствующего элемента выходного вектора. |
| D:\3vs\1\automatics\general\images\0044.png | Синусоидальная функция | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*sin(w\*x + f),  где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0045.png | Экспоненциальная функция | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*exp(b\*x + c),  Где a, b, c – коэффициенты,  x – входное значение.  Параметры блока:   * К-т усиления– коэффициент a * Обратная постоянная времени– коэффициент b * Сдвиг - коэффициент c   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0046.png | Гиперболическая функция | Блок возвращает выходное значение вычисленное по формуле:  Y = k/(eps + x),  Где k, eps – коэффициенты,  x – входное значение.  Параметры блока:   * Числитель– коэффициент k * Минимальное значение знаменателя– коэффициент eps.   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0047.png | Арксинус | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*arcsin(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0048.png | Арккосинус | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*arccos(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0049.png | Арктангенс | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*arctg(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0050.png | Арккотангенс | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*arcctg(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0051.png | Гиперболический синус | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*sh(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0052.png | Гиперболический косинус | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*ch(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0053.png | Гиперболический тангенс | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*th(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0054.png | Гиперболический котангенс | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*cth(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0055.png | Степенная функция | Блок возвращает выходное значение вычисленное по формуле:  Y = xa,  Где a – показатель степени,  x – входное значение.  Параметры блока:   * Показатель степени – коэффициент a   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0056.png | Показательная функция | Блок возвращает выходное значение вычисленное по формуле:  Y = Ab\*x,  Где A – основание степени,  B – коэффициент при степени,  x – входное значение.  Параметры блока:   * Основание степени в функции a^(b\*u)– коэффициент a; * Коэффициент в степени – коэффициент b;   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0057.png | Показательная функция с переменной амплитудой | Блок возвращает выходное значение вычисленное по формуле:  Y = A\*(u1b\*u2),  Где A – множитель,  B – коэффициент при степени,  u1 – входное значение на первом входе блока.  U2 – входное значение на втором входе блока.  Параметры блока:   * Множитель в функции a\*(u1^(b\*u2))– коэффициент a; * Коэффициент в степени – коэффициент b;   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0059.png | Логарифм натуральный | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*ln(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0058.png | Логарифм десятичный | Блок возвращает выходное значение вычисленное по формуле:  Y = a\*lg(w\*x + f),  Где a, w, f – амплитуда, круговая частота и фаза,  x – входное значение.  Параметры блока:   * Амплитуда– коэффициент a * Частота– коэффициент w * Фаза - коэффициент f   Параметры могут быть векторами, размерности которых должны быть равны размерности входа, в этом случае на выходе будет также вектор. |
| D:\3vs\1\automatics\general\images\0176.png | Логарифм натуральный с защитой 0 | Блок векторизован, возвращает выходное значение, вычисленное по алгоритму:  где – амплитуда, круговая частота и фаза,  – входной и выходной сигналы,  – константа, определяющая ограничение «снизу».  Таким образом, блок работоспособен при любых изменениях входного сигнала, а расчет не останавливается при пересечении входным сигналом нуля. |
| D:\3vs\1\automatics\general\images\0177.png | Логарифм десятичный с защитой 0 | Блок векторизован, возвращает выходное значение, вычисленное по алгоритму:  где – амплитуда, круговая частота и фаза,  – входной и выходной сигналы,  – константа, определяющая ограничение «снизу».  Таким образом, блок работоспособен при любых изменениях входного сигнала, а расчет не останавливается при пересечении входным сигналом нуля. |
| D:\3vs\1\automatics\general\images\3048.png | Корень квадратный | Блок реализует вычисление квадратного корня числа (для каждого элемента входного вектора):    Где x – входное значение.  Размерность выходного вектора равна размерности входного. |

## ЗАКЛАДКА «ИССЛЕДОВАНИЯ»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
|  | Оптимизация параметров модели | Блок оптимизации параметров модели предназначен для подбора таких параметров оптимизации, которые бы удовлетворяли необходимым значениям критериев оптимизации. |
|  | IOSO оптимизация | Блок IOSO оптимизации параметров модели предназначен для подбора таких параметров оптимизации, которые бы удовлетворяли необходимым значениям критериев оптимизации. |
|  | Построение передаточных функций | Блок позволяет получить математическое описание динамического объекта (блока/системы). |
|  | Построение частотных характеристик | Блок позволяет строить на графике выбранные амплитудно-фазовые частотные характеристики динамического объекта. |
|  | Гармонический анализатор | Блок представляет собой готовую модель, реализующую алгоритм прямого расчёта частотной характеристики произвольной системы при подаче на её вход тестового синусоидального сигнала с дискретно изменяющейся частотой и заданной амплитудой колебаний. |

## Закладка «СТАТИСТИКА»

|  |  |  |
| --- | --- | --- |
| **Пиктограмма** | **Наименование блока** | **Описание и свойства блока** |
| D:\3vs\1\automatics\general\images\0212.png | Среднее арифметическое | Блок реализует вычисление среднего арифметического значения входного сигнала по формуле:  где N – объем выборки (количество значений),  xᵢ – значения i-го элемента выборки. |
| D:\3vs\1\automatics\general\images\0213.png | Среднеквадратическое отклонение | Блок реализует вычисление среднеквадратического отклонения значения входного сигнала по следующей формуле:  где N – объем выборки (количество значений),  xᵢ – значения i-го элемента выборки,  M – математическое ожидание случайной величины.  Эта величина характеризует мощность пульсационной составляющей сигнала. |
| D:\3vs\1\automatics\general\images\0218.png | Коэффициент эксцесса | Блок реализует вычисление коэффициента эксцесса (третьего момента) входного сигнала по следующей формуле:  где N – объем выборки (количество значений),  xᵢ – значения i-го элемента выборки,  M – математическое ожидание случайной величины.  Коэффициент эксцесса характеризует асимметрию плотности вероятности. Для симметричной плотности вероятности, например для нормального закона распределения, его значение равно 0. |
| D:\3vs\1\automatics\general\images\0219.png | Фактор сплющиваемости | Блок реализует вычисление фактора сплющиваемости (четвертого момента) входного сигнала по следующей формуле:  где N – объем выборки (количество значений),  xᵢ – значения i-го элемента выборки,  M – математическое ожидание случайной величины.  Фактор сплющиваемости характеризует отклонение формы кривой плотности вероятности от нормального закона распределения. Для нормального закона эта величина равна 0. |
| D:\3vs\1\automatics\general\images\0217.png | Коэффициент корреляции | Блок реализует вычисление коэффициента корреляции двух входных сигналов по следующей формуле:  где n – объем выборки (количество значений),  xᵢ – величина i-го значения первого входного сигнала,  yᵢ – величина i-го значения второго входного сигнала,  M(x) – математическое ожидание первого сигнала,  M(y) – математическое ожидание второго сигнала.  Коэффициент корреляции показывает, насколько сильна зависимость между двумя сигналами. Если сигналы являются линейно зависимыми, то коэффициент корреляции по модулю равен 1. |
| D:\3vs\1\automatics\general\images\0211.png | Гистограмма распределения | Блок предназначен для построения плотности вероятности распределения сигнала (гистограммы) по следующему алгоритму:  Сначала вычисляется минимальное и максимальное значения случайного сигнала.  Весь интервалразбивается на заданное число интервалов m, причём границы каждого интервала рассчитываются по формулам: где L – левая граница i-го интервала; R – правая граница i-го интервала. Ширина интервала:  Затем рассчитывается число попаданий величины в i-й интервал Nᵢ.  Плотность вероятности в i-м интервале оценивается по формуле:  pᵢ = Nᵢ/(N·h),  где N – размер выборки. |
| D:\3vs\1\automatics\general\images\0214.png | Спектральная плотность | Блок реализует вычисление спектральной плотности скалярного входного сигнала по следующему алгоритму:  Реализация делится на отдельные серии, причём размер серии должен быть целой степенью числа 2.  Выполняется предварительная подготовка данных, состоящая: в удалении из исходных данных среднего значения или линейного тренда; в умножении данных на весовые коэффициенты (временное окно). Умножение данных на временное окно позволяет уменьшить погрешности, связанные с вычислением дискретного преобразования Фурье, например, уменьшить просачивание через боковые максимумы.  Методом быстрого преобразования Фурье (БПФ) вычисляется дискретное преобразование Фурье последовательности определяемое по формуле:  Абсолютная спектральная плотность определяется по формуле:  , где– период дискретизации сигнала; – коэффициент, зависящий от способа взвешивания входной величины; – значение дискретного преобразования Фурье на k-й гармонике.  Нормированная спектральная плотность вычисляется путем деления абсолютной на дисперсию сигнала. |
| D:\3vs\1\automatics\general\images\0215.png | Взаимная спектральная плотность | Блок реализует вычисление взаимной спектральной плотности двух скалярных входных сигналов по следующему алгоритму:  Реализации двух сигналов делятся на отдельные серии, причём размер серии должен быть целой степенью числа 2.  Выполняется предварительная подготовка данных, состоящая: в удалении из исходных данных среднего значения или линейного тренда; в умножении данных на весовые коэффициенты (временное окно). Умножение данных на временное окно позволяет уменьшить погрешности, связанные с вычислением дискретного преобразования Фурье, например, уменьшить просачивание через боковые максимумы.  Методом быстрого преобразования Фурье (БПФ) вычисляются дискретные преобразования Фурье двух действительных последовательностей:  Абсолютная взаимная спектральная плотность вычисляется по следующей формуле:  , где – период дискретизации сигнала; – коэффициент, зависящий от способа взвешивания входной величины.  Нормирование взаимной спектральной плотности осуществляется путём деления абсолютной на величину: , где – дисперсия величины x,– дисперсия величины у. |
| D:\3vs\1\automatics\general\images\0216.png | Функция взаимной корреляции | Блок реализует вычисление взаимной корреляции двух скалярных входных сигналов.  Корреляционная функция величин х и у определяется как:  ,  где – ковариационная функция, – автоковариационные функции величин x и у; N – размер выборки; r=1...m, m < N – временной сдвиг; – период дискретизации. |