1. Структура и принципы функционирования
   1. Функциональное наполнение

SimInTech имеет следующее функциональное наполнение, позволяющее отнести его к программно-инструментальным средствам автоматизации динамических расчетов технических систем, описываемых во входо-выходных отношениях:

* интерактивные и графические средства формирования образа исследуемого технического устройства (или его общепринятого упрощенного изображения);
* интерактивные средства формирования математической модели рабочего процесса в исследуемой технической системе (устройстве);
* базы данных математических моделей (электронные библиотеки) для описания рабочих процессов в элементах исследуемых технических устройств;
* открытость базы данных математических моделей;
* динамический обмен данными с внешними программами (устройствами) в синхронном и асинхронном режимах;
* численное моделирование рабочего процесса в режиме реального времени или в режиме масштабирования модельного времени;
* встроенные средства анализа, синтеза и параметрической оптимизации систем автоматического управления;
* развитые средства графического отображения текущих результатов расчета;
* архивация расчетных данных, воспроизведение процесса моделирования;
* встроенные расчетные модули для постобработки результатов численного моделирования, обобщения и для анализа полученных результатов, включая модули, обеспечивающие статистический анализ расчетных данных;
* интерактивные, графические и расчетные средства, обеспечивающие формирование виртуальных пультов управления с аналогами измерительных приборов и устройств, включая поддержку мультимедийных свойств (звук, анимация, видео и др.);
* встроенную систему диагностики ошибок на различных этапах работы;
* встроенную контекстную справочную систему, подробную техническую документацию (Инструкцию Пользователя).
  1. Схема функционирования

Для автоматизации процесса создания и исследования математической модели сложных технических систем на макроуровне в среде SimInTech используется метод структурного моделирования [18...20], позволяющий Пользователю в режиме человек – машина формировать и исследовать поведение модели технической системы.

На рисунке 2.1 представлена блок-схема метода структурного моделирования, реализованного в SimInTech. Методика работы в среде SimInTech состоит из нескольких этапов.

На первом этапе проводится анализ математической модели объекта исследования с целью ее структуризации на отдельные, функционально самостоятельные подсистемы, выявляется характер взаимосвязей подсистем. Используя библиотеку типовых блоков или создавая новые типы блоков, в графическом редакторе создается структурная схема модели объекта.

На втором этапе в редакторе параметров задаются численные характеристики модели. При этом используются механизмы, позволяющие структурировать параметры модели по сфере их действия в пределах структурной схемы.

На третьем этапе выбираются тип исследования (расчета), задаются параметры выбранного численного метода, формируются средства отображения текущих результатов расчета и вид сохранения данных моделирования для последующего анализа.

При запуске на расчет *автоматически* (на основании анализа топологии структурной схемы) формируется математическая модель объекта исследования в виде системы нелинейных дифференциально–алгебраических уравнений (ДАУ):

|  |  |
| --- | --- |
|  | (2.1) |
|  | (2.2) |

где система уравнений (2.1) описывает непрерывные, а система (2.2) – дискретные блоки.



Рисунок 2.1 – Блок-схема структурного моделирования в среде SimInTech

* 1. Описание блока структурной схемы
     1. Математическая модель блока

Блок является фундаментальным понятием структурного моделирования. Каждый типовой блок представляет собой реализованную (программным образом) математическую модель того или иного явления, процесса или устройства, открытую для обмена информацией с другими элементами структурной схемы. При этом блок имеет своё уникальное графическое изображение, позволяющее однозначно идентифицировать его на структурной схеме. Многие графические изображения анимированы, для того чтобы визуально в процессе расчёта отображать те или иные расчётные величины или состояния блока (положения ключей, опорные расчетные величины блока и т.п.).

В общем случае математическая модель блока может включать в себя следующие типы уравнений, соотношений и операций:

1. Систему обыкновенных дифференциальных уравнений (ОДУ):

|  |  |
| --- | --- |
|  | (2.3) |

где*x, u, y*– векторысостояний, входов и выходов, соответственно; *f(x, u)*, *g(x, u, y)* – известные нелинейные функции.

1. Систему линейных алгебраических уравнений (ЛАУ):

|  |  |
| --- | --- |
|  | (2.4) |

где *A* ­– матрица коэффициентов; *x* ­– вектор решений; *f* – вектор правых частей.

1. Систему нелинейных алгебраических уравнений (НАУ):

|  |  |
| --- | --- |
|  | (2.5) |

1. Систему разностных уравнений

|  |  |
| --- | --- |
|  | (2.6) |

где *k* – индекс такта квантования по времени дискретной системы.

1. Внешние программы, описывающие поведение того или иного блока в форме входо–выходных соотношений. Как правило, внешние программы обмениваются значениями переменных с шагом синхронизации.
2. Логические операции и операции отношения.
3. Различные нелинейные функции (в том числе разрывные и типовые нелинейности), ключи, звено переменного транспортного запаздывания и т.п.
4. Динамические звенья, описываемые обыкновенными дифференциальными уравнениями, которые при нулевых начальных условиях можно представить в виде передаточных функций

|  |  |
| --- | --- |
|  | (2.7) |

где *W(s)* – передаточная функция; *N(s)*, *L(s)* – полиномы степени *m* и *n*, соответственно (*m* ≤ *n*).

Примечание: пользователь может создать своё собственное звено, в общем случае динамическое, на встроенном языке программирования, которое может реализовывать любую математическую модель из перечисленных, или их комбинацию.

* + 1. Конструкция блока

Общая теория систем управления позволяет описать все перечисленные формы в виде одной конструкции из шести атрибутов:

|  |  |
| --- | --- |
|  | (2.8) |

Операторами *F* и *G* описывается математическая модель блока, при этом оператор *F* отвечает за поведение переменных состояния блока *X* (например, в виде системы уравнений для производных переменных состояния), а при помощи оператора *G* формируются выходы блока *Y* (рисунок 2.2). Параметры блока *P* обычно определяют коэффициенты в уравнениях математической модели блока. Как видно из рисунка 2.2, параметры *P*, состояния *X*, операторы *F* и *G* являются внутренними атрибутами блока, а входы *U* и выходы *Y* служат для обмена информационными потоками с другими частями структурной схемы. Из структуры уравнений для выходов *Y* следует (см. рисунок 2.2), что внутри блока происходит преобразование входных сигналов в выходные.

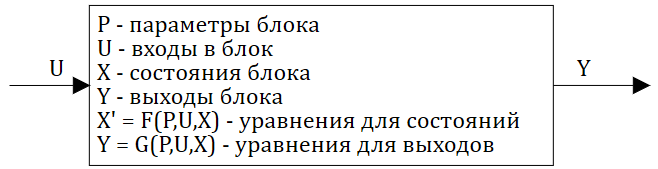


Рисунок 2.2 - Общая конструкция блока

Таким образом, математическая модель объекта исследования формируется в виде структурной схемы, блоки которой описываются в форме входо-выходных соотношений. Если представить схему в форме ориентированного графа (орграфа), то блоки служат вершинами графа, а линии связи – дугами графа. Последующий анализ орграфа позволяет автоматически сформировать математическую модель объекта в виде системы нелинейных дифференциально–алгебраических уравнений, причем систему дифференциальных уравнений практически всегда удается привести к форме Коши.

* + 1. RUN-функция блока

Математическая модель блока программно реализована в специальной функции, называемой RUN-функцией блока.

Каждый тип блока имеет свою RUN-функцию, причем эта функция имеет универсальный вид и не привязана к ядру среды моделирования. Это позволяет расширять элементную базу, в том числе со стороны пользователя, создавая новые библиотеки моделей в виде dll-приложений. При этом динамически подгружаемая библиотека представляет собой совокупность RUN-функций блоков, входящих в эту библиотеку, а также специальной функции GetLibInfo, в которой описываются общие характеристики блоков библиотеки (число блоков, их универсальные индексы, число, тип и описание параметров блоков и т.п.).

Графическое изображение блока на схеме, информация о входных и выходных портах, списки свойств и параметров блоков с их описанием и указанием типа данных содержатся в отдельном файле, с расширением \*.csl (от английского ClassLib). Файл библиотеки блоков содержит набор блоков, предназначенный для использования в той или иной расчетной схеме для конкретного математического решателя (схема автоматики, схема теплогидравлики, схема электрики и т.д.).

На разных этапах работы со структурно заданной моделью расчетному ядру требуется получать информацию о тех или иных характеристиках математической модели блока в необходимом виде. В среде SimInTech это требование реализуется путем вызова RUN-функции блока и/или INFO-функции блока с различными флагами. В таблице 2.1 приводится перечень основных флагов вызова для INFO-функции. В таблице 2.2 приводится перечень основных флагов вызова для RUN-функции. Информационная функция вызывается на этапе инициализации (как правило, однократно) и предназначена для сообщения расчетному ядру информации о блоке – о его портах, количестве переменных состояния и т.п. RUN-функция вызывается на каждом шаге расчёта.

Таблица 2.1 – Перечень основных флагов вызова INFO-функции блока

|  |  |
| --- | --- |
| **Обозначение** | **Действие** |
| i\_GetBlockType | Размерность входов/выходов блока |
| i\_GetDifCount | Число непрерывных состояний блока |
| i\_GetAlgCount | Получить число алгебраических переменных |
| i\_GetCount | Получить размерности входов\выходов |
| i\_GetInit | Получить флаг зависимости выходов от входов |
| i\_GetPropErr | Проверка правильности задания параметров блока (перед сортировкой) |
| i\_HaveSpetialEditor | RUN-объект имеет специализированный редактор блока |
| i\_GetPostSection | Блоку нужна пост-секция для выполнения run-функции |
| i\_ReconnectPorts | Действия, выполняемые до сортировки для переназначения портов блока |
| i\_GetSyncPostSection | Блоку нужна секция, выполняемая последовательно при синхронном получении данных |
| i\_GetDisCount | Получить к-во переменных для дискретных блоков |

Таблица 2.2 – Перечень основных флагов вызова RUN-функции блока

|  |  |
| --- | --- |
| **Обозначение** | **Действие** |
| f\_InitState | Запись (инициализация) переменных состояния начальными значениями, перед началом расчёта |
| f\_UpdateOuts | Обновить выходы на предварительном шаге |
| f\_GoodStep | Обновить выходы на "хорошем" шаге |
| f\_GetDeri | Вычислить значения правых частей дифференциальных уравнений |
| f\_GetAlgFun | Вычислить значения правых частей алгебраических уравнений |
| f\_SetState | Вычислить значения дискретных переменных состояния (после шага интегрирования) |
| f\_UpdateProps | Обновить список параметров (с учётом флага изменяемости) |
| f\_UpdateJacoby | Обновить выходы блока при расчёте матрицы Якоби (эквивалент f\_UpdateOuts с дополни-тельным вызовом обновления дискретных состояний) |
| f\_RestoreOuts | Обновить выходы после рестарта (только если очень надо, т.к. выходы всё равно будут за-поминаться) |
| f\_SetAlgOut | Выставить выходы блока, содержащих алгебраические переменные |
| f\_InitAlgState | Выставить начальное приближение для алгебраические переменных |
| f\_Stop | Вызывается при остановке расчёта (конец моделирования) |
| f\_InitObjects | Инициализация объектов, массивов и т.д. (сразу после сортировки) (начало моделирования) |
| f\_EndTimeTask | Вызывается по окончании выполнения задачи (для проверки оптимизации и т.п.) |
| Примечание: флаги ниже используются для частотного анализа и расчётов Якобиана | |
| f\_GetDisState | Получить значения дискретных переменных состояния блока |
| f\_SetDisState | Присвоить возмущение для дискретной переменной |
| f\_GetDelayTime | Получить время задержки блока (для частотного анализа) |

Все блоки, задействованные в расчете, сгруппированы по основным типам (для сортировки, частотного анализа, синтеза) и каждый из блоков относится к одному из 8-ми типов. Перечень приведён в таблице 2.3

Таблица 2.3 Типы блоков

|  |  |
| --- | --- |
| **Тип блока** | **Описание** |
| t\_none = 0 | Сервисный блок, в расчете не участвует |
| t\_src = 1 | Блок-источник сигнала |
| t\_fun = 2 | Функциональный блок |
| t\_dst = 3 | Блок-приемник информации |
| t\_del = 4 | Блоки запаздывания |
| t\_ext = 5 | Блоки-экстраполяторы |
| t\_der = 6 | Блоки-производные |
| t\_imp = 7 | Блоки-импортеры данных |
| t\_exp = 8 | Блоки-экспортеры данных |

* 1. Библиотеки математических моделей

Для успешного функционирования системы автоматизации динамических расчетов (САДР) необходимо создать достаточное количество надежно работающих расчетных блоков, которые можно уверенно использовать в работе. В настоящее время разработано более 500 типовых блоков, которые входят в состав SimInTech в виде ряда библиотек, сгруппированных, в основном, по функциональному признаку и/или по математическому решателю. При этом имеется базовый набор библиотек типовых блоков, входящих в стандартный комплект и образующих *Общетехническую* библиотеку типовых блоков, предназначенных для использования в расчетном слое «Автоматика».

Другие библиотеки носят специализированный характер (например, библиотеки **СПТ**, **Статистика**, **Кинетика** нейтронов, **Гидроавтоматика**, **Свойства** и др.).

*Общетехническая* библиотека, в основном, описывает процессы в типовых элементах систем автоматического управления (САУ), а также предоставляет Пользователю ряд сервисных функций общего назначения (субмодель, блоки отображения информации, обработки сигналов, создания именованных глобальных переменных и др.). В версии SimInTech 1.6 она состоит из 15 отдельных библиотек и включает в себя более 260 типовых блоков. В таблице В.1 Приложения В приведен полный перечень типовых блоков, имеющихся в *Общетехнической* библиотеке SimInTech.

Учитывая, что невозможно сформировать абсолютно полную библиотеку моделей, в среде SimInTech разработаны встроенные средства, которые позволяют Пользователю самому расширить состав библиотеки за счет создания новых типов блоков:

* наличие блока-интерпретатора математических функций позволяет прямо в процессе работы создавать при помощи встроенного языка программирования экземпляры блоков со своими оригинальными математическими моделями;
* dll-интерфейс позволяет подключать внешние библиотеки в виде типовых блоков на структурной схеме, при этом обмен данных между средой разработки и внешней библиотекой происходит через формальные параметры специально оговоренных подпрограмм;
* реализация принципа вложенности субструктур позволяет сохранять на диске в виде макроблоков (подпрограмм) отдельные фрагменты структурной схемы, что дает возможность Пользователю создавать библиотеки унифицированных узлов (подсистем) установки, математическая модель которых определяется их внутренней структурной схемой.
  1. Интеграция с другим ПО

Для обеспечения интеграции SimInTech с другими программными средами и/или устройствами разработан ряд блоков библиотек **Данные** и **Субструктуры**. Также, ряд общетехнических библиотек и блоков приведён с открытым исходным кодом.

Блоки, входящие в состав этих библиотек, обеспечивают различные способы динамического обмена данными с внешними программами (устройствами). Состав библиотеки позволяет выбирать следующие параметры процесса обмена данными:

* шаг обмена данными;
* режим обмена данными (синхронный или асинхронный);
* протокол обмена данными (файловый, TCP/IP, UDP);
* структуру пакета данных;
* тип данных в пакете.
  1. Способы исследования динамических систем

Среда динамического моделирования SimInTech предоставляет следующие возможности исследования динамических систем:

* МОДЕЛИРОВАНИЕ;
* ОПТИМИЗАЦИЯ;
* АНАЛИЗ;
* КОНТРОЛЬ И УПРАВЛЕНИЕ;
* ГЕНЕРАЦИЯ КОДА;
* УДАЛЕННАЯ ОТЛАДКА.

Функции моделирования обеспечивают:

* моделирование нестационарных процессов в непрерывных, дискретных и гибридных технических системах, в том числе и при наличии обмена данными (синхронный или асинхронный) с внешними программами и устройствами;
* редактирование параметров структурной схемы и расчета в режиме “on-line”;
* расчет в реальном времени или в режиме масштабирования времени;
* рестарт, архивацию и воспроизведение результатов моделирования;
* выполнение статистической обработки сигналов (в том числе и внешних), основанную на быстром преобразовании Фурье (БПФ).

Функции оптимизации позволяет решать задачи:

* параметрической оптимизации САУ и идентификации опытных данных;
* синтеза оптимальных регуляторов и оптимального управления в многокритериальной постановке при наличии ограничений на значения динамических переменных, управляющих воздействий, параметров элементов системы автоматического управления, функционалов качества.

Блоки анализа обеспечивают:

* расчет амплитудно-фазовых частотных характеристик для любой линейной и большинства нелинейных систем (ЛАХ, ФЧХ, различные годографы и др.), методом расчёта матрицы Якоби системы;
* реализацию метода D-разбиений на плоскости 1-го комплексного параметра;
* расчет коэффициентов, полюсов и нулей передаточных функций.
* частотный анализ любых (линейных и нелинейных) систем методом гармонического анализа, прямым моделированием при подаче на вход системы синусоидального испытательного сигнала.

Инструменты контроля и управления позволяют создавать:

* виртуальные аналоги пультов управления с измерительными приборами и управляющими устройствами применительно к задачам оперативного контроля и управления рабочими процессами в технических системах;
* виртуальные аналоги мнемосхем с мультимедийными эффектами (техническая анимация, звуковое сопровождение, интерактивные диалоговые окна и т.п.) для задач оперативного контроля и управления технологическими процессами.

Модуль генерации кода предоставляет возможность автоматического создания программного кода на языке Си на основе математической модели объекта, сформированной в виде структурной функциональной блок-схемы.

Удаленная отладка обеспечивает функции контроля за выполнением программы на удаленной исполнительной системе. При этом отображение значений соответствующих сигналов происходит в среде SimInTech на структурной схеме, по которой был сгенерирован удаленно-исполняемый код. Запуск кода на борту удаленного контроллера происходит в исполнительной среде NordWind, работающей под управлением операционной системы реального времени QNX, Unix/Linux или любой другой POSIX-совместимой системы. Для доступа к рабочей станции по сети Ethernet, получения значений переменных и управления выполнением программы используются модули NordWind GbdServer и Libnet.

* 1. Формирование расчетной модели
     1. Анализ топологии структурной схемы

Метод структурного моделирования является весьма эффективным с точки зрения гибкости, простоты и удобства задания и редактирования математической модели объекта исследования. В то же время такая форма представления математической модели неприемлема с точки зрения реализации численного решения уравнений модели. Связано это с тем, что общая модель объекта включает в себя не только модели отдельных блоков структурной схемы, но и дополнительные уравнения, которые неявным образом задаются посредством линий связи между выходными и входными портами отдельных блоков. Эти уравнения отражают взаимосвязи отдельных блоков в структурной схеме и могут быть учтены в общей модели путем изменения порядка расчета отдельных блоков. Для этого перед началом процесса моделирования проводится анализ топологии структурной схемы.

Анализируя информационные потоки в структурной схеме, можно определить сам факт взаимосвязи входа одного блока с выходом другого блока, но нельзя ничего сказать о характере этой связи. Характер связи определяется особенностями модели самого блока, а именно тем, как влияет тот или иной вход блока на его выход.

Если выход блока явно зависит от его входов, то на момент расчета этого блока все его входы должны быть определены. В математической модели таких блоков уравнения для выходов имеют следующий вид:

|  |  |
| --- | --- |
|  | (2.9) |

Примером блоков такого типа является усилитель сигнала, передаточная функция общего вида, где порядок числителя равен порядку знаменателя, блок *Переменные состояния* с ненулевой матрицей D и т.д.

Если в математической модели блока уравнение для выхода имеет вид

|  |  |
| --- | --- |
|  | (2.10) |

то такой блок называется приоритетным. Выходы приоритетных блоков являются функцией только их внутренних атрибутов (параметров и состояний блока). Примерами приоритетных блоков являются источники сигнала, динамические блоки и т.д.

Объединяя уравнения вида (2.9), (2.10) всех блоков структурной схемы и, учитывая, что входным сигналом блока является выход другого блока или переменная, зависящая от модельного времени *t*, получим уравнения всей системы в виде:

|  |  |
| --- | --- |
|  | (2.11) |
|  | (2.12) |

где *х* - вектор всех переменных состояния, *y* - вектор выходных переменных всех блоков модели.

При реализации методов численного интегрирования необходимо получить уравнения моделируемой системы в нормальной форме Коши, то есть в виде

|  |  |
| --- | --- |
|  | (2.13) |
|  | (2.14) |

Опишем процедуру приведения уравнений к форме Коши. Отметим, что не обязательно получать явные выражения в виде (2.13-2.14), достаточно сформировать процедуру, которая позволяет по заданным значениям модельного времени *t* и вектора переменных состояния *x* вычислить вектор производных *x*’ и вектор выходов *у*. Идея этой процедуры – упорядочение (сортировка) всех блоков таким образом, чтобы выход очередного блока мог быть рассчитан по уже имеющейся к этому моменту информации. Практически сортировка позволяет получить уравнения выходов блоков в виде (2.14). Затем производные могут быть рассчитаны с использованием (2.11).

Перед началом сортировки будем считать, что нам известны *t* и *x*. В процессе сортировки будет сформирован упорядоченный список, содержащий в общем случае **5 групп блоков** и соответствующих им выходных переменных.

* 1. Сначала выделим и упорядочим все блоки, выходы которых могут быть рассчитаны по известному модельному времени *t*:

|  |  |
| --- | --- |
|  | (2.15) |

* 1. Далее выделим и упорядочим все блоки, выходы которых могут быть рассчитаны по известным . Это соответствует записи

|  |  |
| --- | --- |
|  | (2.16) |

Этот этап завершает первичную сортировку. Если система не имеет алгебраических контуров, то все блоки структурной схемы к этому моменту станут упорядоченными, и процедура будет завершена, если нет – следует продолжить упорядочение.

* 1. Выделим все блоки, непосредственно входящие в алгебраические контуры. Из них выделим определяющие (). Выходы остальных входящих в алгебраические контуры блоков () можно рассчитать по известным выходам уже упорядоченных и определяющих блоков:

|  |  |
| --- | --- |
|  | (2.17) |

Метод выделения определяющих переменных рассматривается ниже.

* 1. Поместим в список определяющие блоки, выходы которых запишутся в виде

|  |  |
| --- | --- |
|  | (2.18) |

Можно исключить из (2.18), однако приведенная запись позволяет лучше понять рассмотренные ниже схемы расчета алгебраических контуров.

* 1. В последнюю очередь рассчитываются блоки, не входящие в алгебраические контуры, но выходы которых зависят от блоков, входящих в алгебраические контуры:

|  |  |
| --- | --- |
|  | (2.19) |

В результате сортировки будет получен такой порядок расчета блоков, который соответствует использованию формул (2.15)…(2.19). Если в модели нет алгебраических контуров, можно по известным найти выходы всех блоков, в противном случае следует решать алгебраические уравнения относительно определяющих переменных .

Расчет алгебраических контуров можно выполнить по следующим итерационным схемам.

Метод простых итераций:

|  |  |  |
| --- | --- | --- |
|  |  | (2.20) |

Метод Ньютона:

|  |  |  |
| --- | --- | --- |
|  |  | (2.21) |

В общем случае матрица Якоби может не обновляться в течение нескольких итераций (модифицированный метод Ньютона).

Алгоритм расчета выходов блоков и производных переменных состояния запишется следующим образом:

1. По известным *t, x*, используя (2.15), (2.16), рассчитать .
2. Используя итерационную схему (2.20) или (2.21), найти .
3. Используя (2.19), вычислить .
4. Рассчитать производные переменных состояния всех динамических блоков.
   * 1. Развязка алгебраических контуров

Наличие в структурной схеме алгебраических контуров означает, что входы блоков неявным образом (через другие блоки и линии связи) зависят от их выходов. В математической форме уравнение алгебраического контура выражается в виде неявной функции от выходов:

|  |  |
| --- | --- |
|  | (2.22) |

На рисунке 2.3 приведен пример структурной схемы без алгебраического контура, а на рисунке 2.4 – при его наличии. На рисунке 2.3 обратная связь идет с выхода интегратора 4 на вход сумматора 2, который зависит только от его состояния:

|  |  |
| --- | --- |
|  | (2.23) |

поэтому при порядке расчета блоков 1, 4, 2, 3 входы сумматора 2 и усилителя 3 на момент их расчета определены. На рисунке 2.4 обратная связь идет с выхода усилителя 3 на вход сумматора 2, и при любом порядке расчета сумматора и усилителя их входы не могут быть однозначно определены.

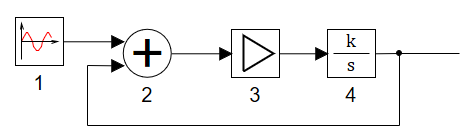


Рисунок 2.3 – Пример структурной схемы без алгебраического контура

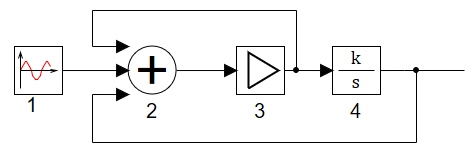


Рисунок 2.4 – Пример структурной схемы с алгебраическим контуром

При наличии в схеме алгебраических контуров необходимо определить их число и выделить выходы блоков, относительно которых будет решаться система нелинейных уравнений. Для этого используется метод определяющих переменных [21]. Суть метода заключается в том, что в алгебраических контурах определяется минимальное число выходных сигналов блоков, при удалении которых из схемы размыкаются все обратные связи в контурах. Рассмотрим этот метод на примере системы НАУ вида:

|  |  |
| --- | --- |
|  | (2.24) |

Если бы и были известны, то из 3-го, 4-го, 5-го уравнений можно было бы последовательно определить , , . Неизвестные и называют определяющими, поскольку по ним легко могут быть определены все остальные неизвестные. Подставляя последовательно 5-е, 4-е, 3-е уравнения в 1-е и 2-е, получим систему уравнений для определяющих неизвестных:

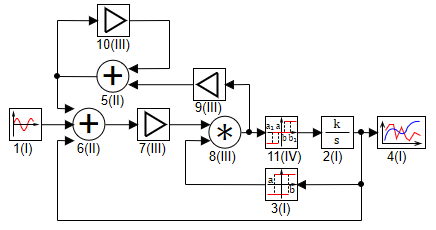
|  |  |
| --- | --- |
|  | (2.25) |

решая которую, находим и . Остальные неизвестные находим из (2.25) прямой подстановкой. В рассмотренном примере решение системы алгебраических уравнений с пятью неизвестными свелось к решению системы с двумя неизвестными. Таким образом, метод определяющих неизвестных позволяет уменьшить размер решаемой системы, используя топологию структурной схемы.

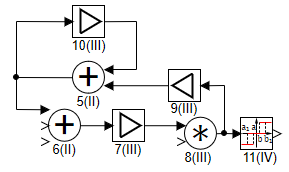
Алгоритм выделения минимального числа определяющих блоков рассмотрен на примере структурной схемы, приведенной на рисунке 2.5.

Структурная схема представляется в виде ориентированного графа, где вершинами графа являются блоки, а дугами графа – линии связи. Алгоритм заключается в последовательном исключении вершин графа.

В процессе первичной сортировки исключаются (вместе со всеми принадлежащими им дугами) вершины, соответствующие приоритетным (на рисунке обозначены номерами 1 и 2) и первично отсортированным блокам (3 и 4). В результате исходный граф приводится к виду, представленному на рисунке 2.5(б).



а) Исходная структурная схема



б) Структурная схема после первичной сортировки

|  |  |
| --- | --- |
|  |  |
| в) Образования петли при исключении вершины 10 | г) Определяющие блоки |

Рисунок 2.5 – Выделение определяющих переменных

Затем начинается собственно процесс выявления определяющих блоков. В качестве очередной исключаемой вершины выбирается вершина, имеющая наименьшее значение из произведения числа входящих и выходящих дуг. На рисунке 2.5(б) – это блок с номером 10. Если после исключения очередной вершины образовалась петля, то вершина с петлей и принадлежащими ей дугами удаляется из графа, соответствующая этой вершине переменная включается в список определяющих, а блок считается отсортированным. На рисунке 2.5(в) таким блоком является сумматор 5. Результат процесса выявления определяющих блоков показан на рисунке 2.5(г). К этому моменту все определяющие блоки отсортированы, и дальнейший процесс упорядочения блоков в структурной схеме аналогичен первичной сортировке.

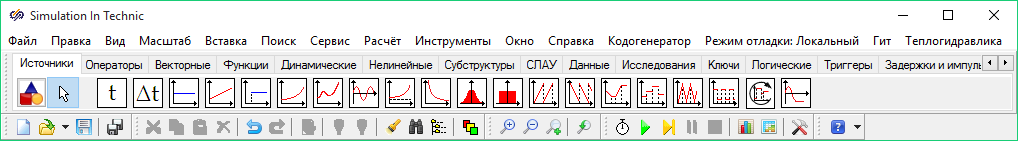
Итоговый порядок расчета блоков структурной схемы, представленной на рисунке 2.5, соответствует их нумерации на схеме (примечание: отображение нумерации включается опцией Главного Окна, пункт меню Вид – Отображать номера блоков на схеме).

* 1. Графический редактор структурных схем

Практика моделирования реальных технических систем показывает, что их структурные схемы включают в себя даже тысячи блоков с многочисленными связями между ними. Поэтому большое значение имеет простота, удобство, наглядность и широкие функциональные возможности режима взаимодействия и диалога человека с программой при создании и редактировании структурной схемы модели объекта. Эти возможности обеспечивает графический редактор структурных схем. Пример работы в среде SimInTech представлен на рисунке 2.6. В окне графического редактора в виде структурной схемы набрана математическая модель следящего привода с двухступенчатым редуктором.

Командные кнопки, команды меню, функции работы с мышью и клавиатурой создают среду (интерфейс) для общения Пользователя со средой моделирования. Графический редактор поддерживает более сорока функций редактирования. В окне редактора Пользователь может перемещать, копировать, удалять, вставлять отдельные фрагменты структурной схемы. Возможно изменение размеров, ориентации, цветового, текстового оформления отдельных блоков. Графический редактор позволяет масштабировать структурную схему, привязывать блоки и линии к сетке. Часть схемы можно исключить из процесса моделирования, не удаляя из самой схемы, что резко упрощает отладку больших проектов. SimInTech позволяет сохранять структурные схемы и результаты расчетов в файлах, выводить на печать в виде рисунков схемы, графики или таблицы результатов расчетов.

Главная Панель Управления

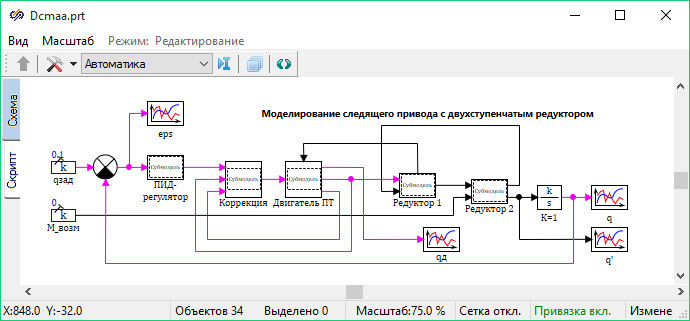


# Командное меню

Библиотека типовых блоков

Командные кнопки

Главное Схемное Окно (Окно графического редактора)



Структурная схема *Двигатель ПТ* *Графическое окно*

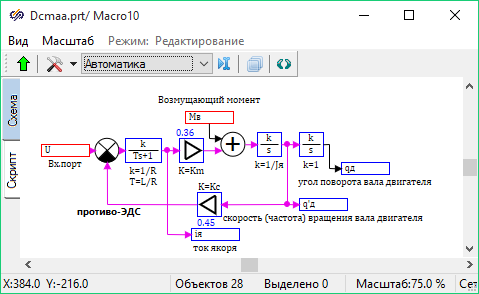
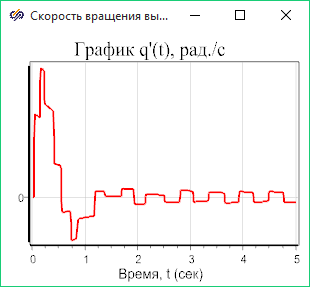
 

Рисунок 2.6 - Пример работы в среде моделирования SimInTech

Графический редактор позволяет формировать многоуровневые структурные схемы. При этом на структурной схеме уровня N, подструктуры уровня N-1 представлены в виде блоков, называемых субмоделями. В отличие от обычных блоков, математическая модель блока субмодели записана не в виде уравнений, а представлена внутренней структурной схемой субмодели. Переход с уровня на уровень позволяет редактировать каждую подструктуру в отдельности. На рисунке 2.6 показан пример двухуровневой схемы, где некоторые блоки представлены в виде субмоделей (субструктур, макроблоков). Например, *Двигатель ПТ* в Главном Схемном окне – подструктура нижнего уровня. В субмодельном схемном окне показана внутренняя структурная схема этого блока. Принцип вложенности позволяет представить структурную схему сложной системы в удобном, легко читаемом виде, при этом уровень вложенности субструктур неограничен. В Главном меню (Поиск – Структура) доступен инструмент, позволяющий в древовидном виде осуществлять навигацию по структурной схеме как текущего проекта, так и любого другого проекта, открытого в текущем экземпляре программы.

Создание математической модели объекта предполагает задание значений внутренних свойств *P* и начальных значений переменных состояния каждого из блоков структурной схемы. Для сложных технических объектов общее число этих параметров исчисляется тысячами, при этом очень часто параметры функционально зависят друг от друга, от некоторых глобальных для модели параметров или являются переменными во времени.

Для решения этой проблемы все свойства (параметры, сигналы, переменные) модели делятся по области действия на *локальные* и *глобальные* свойства и сигналы. *Локальные* свойства являются характеристиками элементарного блока. Сфера их действия ограничена математической моделью блока и, возможно, его панелью управления.

*Глобальные* сигналы и параметры являются параметрами всего проекта (задачи) или конкретной субмодели (макроблока), причем они разделяются на константы и переменные (так же и свойства блока могут быть неизменными величинами, а могут меняться в процессе расчета). Константы задаются или вычисляются один раз перед началом моделирования и не изменяют своих значений при моделировании. Переменные вычисляются в процессе моделирования, и их значения в общем случае зависят от независимой переменной – модельного времени. Константы задаются своими численными значениями (первичные константы) или в виде выражений от определенных выше констант (вторичные константы) и используются для задания локальных параметров элементарных блоков модели. Переменные определяются как функции времени или как произвольные интерпретируемые выражения, в том числе зависящие от других свойств и параметров данного блока, либо от других сигналов, определённых на уровнях выше.

Каждое свойство субмодели может быть однозначно идентифицировано с помощью уникального (в пределах данной субмодели) имени. Область действия именованного свойства строго определена – свойство «видно» только в блоках, находящихся на внутренних (по отношению к субмодели, где задано данное глобальное свойство) уровнях структурной схемы. Если свойство переопределено во внутренней субмодели, он заслоняется этим новым свойством. На рисунке 2.7 приведен пример использования глобальных свойств субмодели.

Необходимость использования глобальных свойств субмодели диктуется следующими соображениями:

1. При выполнении вариантных расчетов необходимо часто изменять некоторые параметры модели. Это удобнее делать, когда все такие параметры поименованы и их описания сосредоточены в одном месте.
2. Один и тот же параметр (например, период квантования дискретной системы) может присутствовать в нескольких блоках модели. В этом случае изменение этого параметра потребовало бы внесения изменений во все эти блоки. Если задать этот параметр глобальным, то все изменения сводятся к изменению одного параметра.
3. При создании модели (субмодели) какого-либо устройства можно разделить все параметры модели на изменяемые, зависящие от конкретной модификации данного устройства, и неизменяемые, одинаковые для всех устройств такого типа. В этом случае целесообразно задать изменяемые параметры как глобальные.
4. С помощью глобальных сигналов удобно задавать оптимизируемые параметры в задачах параметрической оптимизации и оптимального управления.



Рисунок 2.7 - Состав, структура и параметры субмодели

Наряду с глобальными параметрами модели в среде моделирования SimInTech введено понятие *глобальных переменных* модели. Глобальные переменные формируются путем использования типового блока *В память*, входной сигнал которого Пользователь может сделать именованным. По сути блок *В память* задаёт имя для подключенной к нему линии связи.

Кроме того, глобальными переменными проекта являются элементы списка сигналов проекта и/или элементы подключенной к проекту базы данных сигналов (например, базы сигналов SDB), запись и чтение значений которых производится с помощью блоков *Чтение из списка сигналов*, *Запись в список сигналов*. Переменные, записанные с помощью блока *В память*, и глобальные сигналы проекта не пересекаются друг с другом и представляют из себя два разных списка, потому что предназначены они для разных задач, а именно: глобальные переменные, записываемые с помощью блока *В память*, передаются без запаздывания на один расчетный шаг, которое возникает при записи и чтении сигналов проекта. Однако использование сигналов проекта более предпочтительно в некоторых случаях, так как позволяет получать доступ к сигналам из любого места проекта (из скрипта, графического контейнера и т. д.), либо из любого места комплексной модели, в случае использования пакета проектов или сетевого расчета. В обоих случаях, конструкция блока записи глобальной переменной описывается в виде:

*K* = {*U*, *PU*}; *PU* ≡ *U*,

где *PU* - именованный глобальный параметр или сигнал модели, являющийся внутренним параметром типового блока *В память* или блока *Запись в список сигналов.*

*Глобальные переменные* и сигналы модели задаются, а затем используются в любой части структурной схемы проекта. Сигналы базы сигналов могут быть общими между несколькими проектами. Типовой блок *Из памяти* позволяет просмотреть список всех имеющихся в структурной схеме глобальных переменных. Перечень доступных глобальных сигналов можно увидеть в перечне сигналов проекта (пункт Главного меню Сервис – Сигналы), а также в подключенной базе сигналов при наличии такого подключения.

Выбранную Пользователем глобальную переменную либо сигнал проекта можно назначить выходом блока *Из памяти* или *Чтение из списка сигналов* соответственно. Конструкция данных блоков похожа и имеет описание:

*K* = {*Y*, *PY*}; *PY* ≡ *PU*; *Y* ≡ *PY*,

где *PY* – внутренний параметр блоков *Из памяти* или *Чтение из списка сигналов,* обозначающий имя считываемой переменной.

Именованные глобальные переменные широко используются в различных режимах работы. Такой механизм является удобным средством для проведения «невидимых» линий связи между любыми уровнями структурной схемы в любом направлении (с верхних уровней на нижние и наоборот), что позволяет улучшить читаемость сложных многосвязных моделей. Также механизм передачи сигналов и переменных используется при реализации функций КОНТРОЛЯ и УПРАВЛЕНИЯ при помощи построения динамических видеокадров и щитов управления.

Каждый блок имеет свой уникальный графический образ (векторный рисунок и пиктограмму в палитре блоков), позволяющий однозначно идентифицировать его на структурной схеме, а также набор стандартных диалоговых форм для взаимодействия со свойствами и параметрами блока.

На рисунке 2.8 представлена экранная копия закладки ***Свойства*** типового диалогового окна редактирования свойств, на примере блока *Инерционное звено 1-го порядка*, которое на рисунке 2.6 в субмодели *Двигатель ПТ* использовано для описания динамики основной составляющей тока якоря. Свойства математической модели, реализуемой типовым блоком, могут быть заданы как в численном виде, так и с использованием интерпретируемого выражения. Так, на рисунке 2.8 свойство *Начальные условия* задано в численном виде (**[0]**), а свойства *Коэффициенты усиления* и *Постоянные времени* введены с использованием глобальных сигналов (**k\_i** и **t\_i**, соответственно).

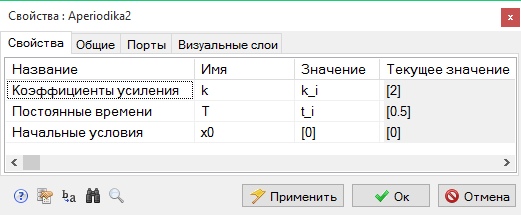


Рисунок 2.8 - Копия диалогового окна (закладка ***Свойства***)

Примечание: запись вектора единичной длины **[0]** в данном случае идентична записи действительного числа **0**. Но, если бы глобальные переменные **k\_i** и **t\_i** были бы векторами из 2-х и более элементов (например, векторами с кол-вом элементов, равным 4), то и начальные условия необходимо было бы сформировать вектором (из 4-х элементов, и записать его как **[0,0,0,0]** или **4#0**, при нулевых начальных условиях). Таким образом, в колонке «значение» в действительности всегда записывается строковое начальное интерпретируемое выражение, которое в момент инициализации блока вычисляется, а вычисленное значение присваивается текущему значению свойства.

Закладка ***Общие*** (рисунок 2.9) того же окна позволяет производить редактирование свойств, являющихся общими для всех блоков, например:

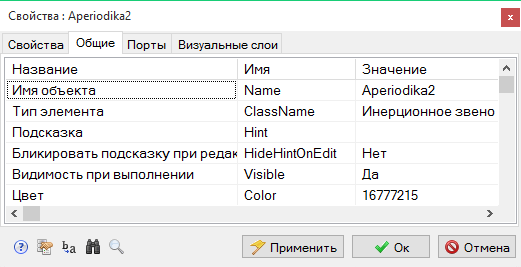


Рисунок 2.9 – Копия диалогового окна (закладка ***Общие***)

* изменять имя блока (уникальный идентификатор в пределах проекта);
* добавлять скрипты инициализации и исполнения объекта;
* изменять поясняющую подпись, ее положение и шрифт;
* изменять цвет фона блока, его прозрачность и положение;
* редактировать изображение блока и др.

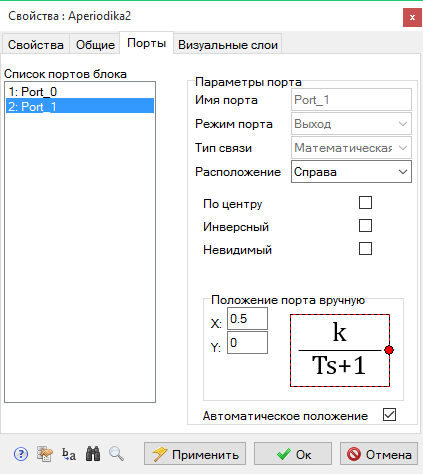


Рисунок 2.10 - Копия диалогового окна (закладка ***Входы***)

Закладка ***Порты*** – содержит инструменты управления расположением входных и выходных портов (рисунок 2.10).

Закладка ***Визуальные слои*** позволяет размещать блок на различных визуальных слоях. Отображением и активностью визуальных слоев расчетной схемы можно управлять, включая или выключая видимость соответствующих наборов блоков. Расчет математической модели блока происходит одинаково, независимо от того, на каком визуальном слое расположен блок. Управление видимостью и активностью слоёв происходит в панели инструментов проекта. Там же возможно переименование слоёв.

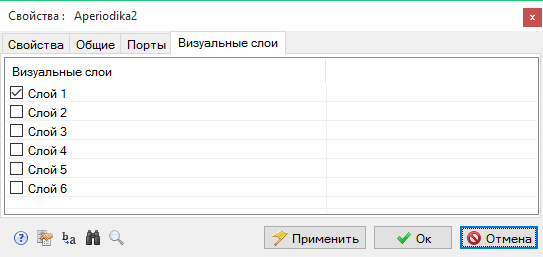
******

Рисунок 2.11 – Копия диалогового окна (закладка ***Визуальные слои***)

* 1. Диагностика ошибок

Среда SimInTech включает в себя систему диагностики ошибок. В таблице 2.4 приведен перечень основных типовых ошибок, определяемых системой диагностики на различных этапах работы среды SimInTech (применительно к расчетному слою «Автоматика» и общетехническим блокам).

Таблица 2.4 – Основные ошибки, формируемые системой диагностики SimInTech

|  |  |
| --- | --- |
| **Сообщение системы диагностики** | **Этап возникновения ошибки** |
| T и k должны быть больше нуля | Проектирование |
| Аргумент арксинуса выходит за границы интервала [-1,1] | Проектирование |
| Аргумент выходит за пределы применимости | Инициализация |
| Аргумент гиперболического котангенса не может быть равен 1 | Проектирование |
| Аргумент логарифма c защитой нуля должен быть больше или равен нулю | Расчет |
| Аргумент логарифма должен быть больше нуля | Расчет |
| Блок должен иметь 2 входа | Инициализация |
| Блок должен иметь два входа | Инициализация |
| Блок должен иметь не менее двух входов | Инициализация |
| Блок должен иметь не менее одного входа | Инициализация |
| Блок должен иметь один входной порт | Инициализация |
| Блок должен иметь хотя бы один вход | Инициализация |
| Блок задания свойства с выхода другого блока не соединён | Инициализация |
| Блок задания уровня свободных портов не подсоединён | Инициализация |
| Блок имеет повторяющийся UID, возможны проблемы генерации кода. Переименуйте его! | Генерация кода |
| Блок подключен к служебному блоку или блоку другого слоя | Инициализация |
| В конце выражения стоит неверный символ | Проектирование |
| Векторная переменная задана неверно | Проектирование |
| Возможно не присвоен выход | Проектирование |
| Возможно не присвоена производная | Инициализация |
| Время запаздывания должно быть БОЛЬШЕ НУЛЯ, т.к. скорость не бесконечна | Инициализация |
| Время запаздывания не может быть отрицательным | Инициализация |
| Вход блока не подсоединён+ | Инициализация |
| Вход блока связан с выключенным из расчета блоком | Инициализация |
| Вход блока соединён с блоком для которого код не может быть сгенерирован | Генерация кода |
| Входная матрица не является квадратной | Инициализация |
| Входная переменная не найдена | Инициализация |
| Выражение задано не полностью | Проектирование |
| Деление на ноль | Расчет |
| Делитель должен иметь два входа | Инициализация |
| Для данного блока программа не может быть сгенерирована | Генерация кода |
| Доступ к элементу невозможен | Проектирование |
| Задан решатель системы другого типа | Инициализация |
| Заданная точность не достигается! | Расчет |
| Запись задана неверно | Проектирование |
| К-во столбцов в файле меньше заданного в параметрах блока | Проектирование |
| К-во функций в файле меньше заданного в параметрах блока | Проектирование |
| Ключевое слово задано неверно | Проектирование |
| Количество аргументов не совпадает | Проектирование |
| Количество шагов должно быть ненулевым | Инициализация |
| Комплексное число не может быть так задано | Проектирование |
| Константу нельзя присваивать | Проектирование |
| Коэффициент в знаменателе равен нулю | Проектирование |
| Массив не может быть создан с данными операндами | Инициализация |
| Массив не существует | Проектирование |
| Матрица вырождена или линейно зависима | Инициализация |
| Матрица должна быть квадратной | Инициализация |
| Матричная переменная задана неверно | Проектирование |
| Метка не найдена | Проектирование |
| Модуль генерации кода не загружен | Генерация кода |
| Модуль генерации кода не инициализирован | Генерация кода |
| Найден неподключенный входной порт | Инициализация |
| Найдена алгебраическая петля | Инициализация |
| Не загружена run-функция | Инициализация |
| Не задано имя линейной системы | Инициализация |
| Не инициализирована run-функция блока | Инициализация |
| Не создать файл для записи данных | Инициализация |
| Не удалось открыть файл обмена | Инициализация |
| Не удалось открыть файл с данными | Проектирование |
| Не удалось получить данные из таблицы | Проектирование |
| Не удалось прочитать строку таблицы | Проектирование |
| Неверный разделитель | Проектирование |
| Невозможно вычислить производные | Расчет |
| Невозможно получить доступ к полю переменной | Проектирование |
| Невозможно привести типы выходов за заданное число итераций | Расчет |
| Недопустимое имя переменной | Проектирование |
| Неизвестная ошибка выполнения | Инициализация |
| Несовпадение с исполняемой системой, свойство: | Удаленная отладка |
| Несоответствие размерностей входов/выходов | Инициализация |
| Номер решателя блока задан неверно! | Инициализация |
| Операнд не найден или не может быть присвоен | Проектирование |
| Оператор не может быть создан | Проектирование |
| Операция не применима к данному операнду | Проектирование |
| Отрицательный аргумент в действительном корне | Расчет |
| Ошибка в файле | Проектирование |
| Ошибка времени выполнения | Инициализация |
| Ошибка выделения памяти под переменную | Инициализация |
| Ошибка выполнения библиотечной функции | Инициализация |
| Ошибка выполнения функции | Инициализация |
| Ошибка вычисления LU-декомпозиции матрицы | Расчет |
| Ошибка генерации кода | Генерация кода |
| Ошибка генерации кода в секции присвоения переменных состояния | Генерация кода |
| Ошибка доступа к массиву или матрице | Инициализация |
| Ошибка доступа к переменной по ссылке | Инициализация |
| Ошибка доступа к файлу | Инициализация |
| Ошибка доступа к файлу обмена | Инициализация |
| Ошибка запроса списка имён генерируемого текста | Инициализация |
| Ошибка именование выходов блоков | Инициализация |
| Ошибка именования выходных переменных блока | Инициализация |
| Ошибка нахождения решения СЛАУ | Расчет |
| Ошибка при возведении числа в заданную степень | Расчет |
| Ошибка при вызове флага f\_InitObjects | Расчет |
| Ошибка при вызове флага f\_SetAlgCount | Расчет |
| Ошибка при вызове флага f\_SetState | Расчет |
| Ошибка при вызове флага f\_Stop | Расчет |
| Ошибка при итерации алгебраической петли | Расчет |
| Ошибка при расчёте алгебраических функций | Расчет |
| Ошибка при расчёте возмущений по переменным | Расчет |
| Ошибка при расчёте правых частей ДАУ | Расчет |
| Ошибка при расчёте производных блоков | Расчет |
| Ошибка при расчёте якобианов для блоков | Расчет |
| Ошибка присвоения типов данных для выходов блока | Инициализация |
| Ошибка удаления памяти по ссылке | Инициализация |
| Ошибка форматирования текста | Проектирование |
| Переменная не найдена | Проектирование |
| Переменные состояния для блока не могут быть декларированы | Инициализация |
| Перемножитель должен иметь два входа | Инициализация |
| Перемножитель должен иметь хотя бы один вход | Инициализация |
| Переполнение в правых частях системы | Расчет |
| Период сигнала должен быть больше нуля | Инициализация |
| Полином задан неверно | Инициализация |
| Порядок знаменателя меньше 2 | Проектирование |
| Порядок знаменателя меньше чем числителя | Проектирование |
| Порядок числителя меньше 2 | Проектирование |
| Постоянная времени T2 блока равна или меньше нуля | Проектирование |
| Постоянная времени блока равна или меньше нуля | Проектирование |
| Превышено ограничение по количеству блоков | Инициализация |
| Превышено ограничение по количеству динамических переменных | Инициализация |
| Произошло деление на ноль | Расчет |
| Произошло деление на ноль - введите ненулевое значение eps | Инициализация |
| Работа модуля генерации кода завершена некорректно | Генерация кода |
| Развязка петли рекомендуется для этого блока | Инициализация |
| Размер вектора должен быть степенью 2 | Инициализация |
| Размерности массивов tau\_on и tau\_of должны быть одинаковыми | Инициализация |
| Размерности массивов параметров не совпадают | Инициализация |
| Размерности матриц не подходят | Инициализация |
| Размерность tau меньше чем y0 | Инициализация |
| Размерность вектора задана неверно | Проектирование |
| Размерность векторов времён и значений не совпадают | Инициализация |
| Размерность массива a меньше чем у массива x0 | Инициализация |
| Размерность массива a1 или массива a2 меньше чем у массива a0 | Проектирование |
| Размерность массива b или массива c меньше чем у массива a | Проектирование |
| Размерность массива b меньше чем у массива a | Проектирование |
| Размерность массива b меньше чем у массива a | Инициализация |
| Размерность массива d или qt меньше чем у массива m | Инициализация |
| Размерность массива eps меньше чем у массива k | Проектирование |
| Размерность массива k или массива T меньше чем у массива x0 | Проектирование |
| Размерность массива k или массива T меньше чем у массива y0 | Проектирование |
| Размерность массива k или массивов T1, T2 меньше чем у массива x0 | Проектирование |
| Размерность массива k меньше чем у массива x0 | Проектирование |
| Размерность массива k меньше чем у массива y0 | Проектирование |
| Размерность массива t или массива dy меньше чем у массива y | Инициализация |
| Размерность массива w или массива f меньше чем у массива a | Инициализация |
| Размерность массива w или массива f меньше чем у массива a | Инициализация |
| Размерность массива xmax или qt меньше чем у массива xmin | Инициализация |
| Размерность массива y0 или массива yk меньше чем у массива t | Инициализация |
| Размерность массива ymin меньше чем у массива ymax | Проектирование |
| Размерность матриц не соответствует указанным количествам переменных | Проектирование |
| Размерность одного из массивов меньше чем у массива x0 | Проектирование |
| Размерность одного из массивов меньше чем у массива y01 | Проектирование |
| Размерность одного из массивов меньше чем у массива y1 | Инициализация |
| Размножитель должен иметь один вход | Инициализация |
| Результат выражения не может быть найден | Проектирование |
| Решение ДАУ не сходится | Расчет |
| Символ не может быть использован | Проектирование |
| Символ не может быть использован в выражении | Проектирование |
| Скобка не закрыта или содержит недопустимое выражение | Проектирование |
| Списка имён данных или списка сигналов не найдено | Инициализация |
| Сумматор должен иметь два входа | Инициализация |
| Сумматор должен иметь хотя бы один вход | Инициализация |
| Табличная функция задана неоднозначно | Инициализация |
| Тип данных и тип ссылки несовместимы | Инициализация |
| Тип переменной задан неверно | Проектирование |
| Типы данных несовместимы | Проектирование |
| У данного блока должно быть чётное число входов | Инициализация |
| У этого блока должно быть не менее двух портов | Инициализация |
| Уровень свободного порты задан дважды | Инициализация |
| Файл не найден | Проектирование |
| Файл обмена не создан | Инициализация |
| Функция задана неверно | Инициализация |
| Функция не может быть переопределена | Проектирование |
| Функция не найдена | Проектирование |
| Элемент массива задан неверно | Проектирование |