# Создание блока управления оборудованием

## Создание следующего проекта автоматики

В предыдущих учебных заданиях была создана комплексная модель, состоящая из модели теплогидравлики и модели автоматики, обменивающихся между собой данными при помощи общей базы данных сигналов.

В данном задании будет продемонстрирована возможность создания типовых блоков управления средствами SimInTech. При моделировании сложных систем часто возникает ситуация, когда необходимо использовать одну и ту же типовую математическую модель многократно. В SimInTech можно использовать одну математическую модель для нескольких однотипных объектов с помощью механизма векторной обработки сигналов.

Для демонстрации возможностей SimInTech мы используем уже существующую базу данных, созданную при выполнении предыдущих заданий. В качестве простейшего примера мы создадим типовой алгоритм управления задвижкой (включая и модель двигателя).

|  |  |
| --- | --- |
| 1. Откройте файл **«Схема автоматики 1.prt»**, созданный при выполнении предыдущих заданий; 2. Удалите существующую **«Субмодель»**; 3. Сохраните файл под новым именем **«Схема автоматики 2.prt»,** используя команду «**Сохранить проект как…**», которая находится в меню «**Файл**» (Рисунок 63). | Рисунок 63. Меню сохранения файла под новым именем |

Поскольку ранее созданный файл был связан с базой данных, файл с новым именем также будет связан с этой базой данных.

В отличие от предыдущего примера, когда в системе автоматики рассчитывалось положение задвижки, и оно непосредственно передавалось в модель теплогидравлики, мы попытаемся создать более приближенную к реальности математическую модель задвижки. Новая модель задвижки будет получать команды на открытие и закрытие и, согласно получаемым сигналам, формировать команду на двигатель и менять положение. Также «блок управления» будет формировать признаки состояния задвижки – открыта она или закрыта.

## Добавление новых сигналов в базу данных

Команды на открытие и закрытие задвижек будут предаваться через базу данных сигналов, поэтому необходимо внести изменения в существующие в базе данных категории. В учебном примере мы используем категорию «Задвижки» для обмена данными с клапанами.

Приступим к редактированию:

1. Войдите в **«Редактор базы данных»**. Пункт меню **«Инструменты → База данных**».
2. В редакторе выберите категорию **«Задвижки»** и нажмите кнопку **«Настроить категорию»** (Рисунок 64), либо выполните двойной щелчок мыши по категории **«Задвижки»**.

Рисунок 64. Редактор базы данных

1. В окне **«Свойства категории»** добавьте два новых сигнала (кнопкой **«Добавить сигнал»** в нижней части окна): **«Команда Открыть»** и **«Команда Закрыть»,** как показано на рисунке ниже(Рисунок 65).

Рисунок 65. Настройка категории

1. Закройте окно **«Свойства категории»** нажатием кнопки **«Ок»**.
2. Если теперь выбрать любую из присутствующих в категории «Задвижки» группу сигналов, то в сводной таблице сигналов мы увидим два новых сигнала, которые мы только что добавили; то есть по модифицированному шаблону добавились сигналы и в группы сигналов.
3. Закройте «**Редактор базы данных**» нажатием кнопки «**Ок**».

## Создание блока управления оборудованием

Под блоками управления оборудованием обычно рассматриваются типовые алгоритмы управления, отвечающие за управление типовым оборудованием, установленным на технологическом объекте. К такому оборудованию как правило относятся задвижки различных типов, запорные и регулирующие клапана, двигатели различного назначения и пр. Данное оборудование, обладая различными физическими характеристиками, обычно имеет один и тот же алгоритм управления, соответствующий ее типу: например, большинство задвижек на таком объекте, как АЭС, будет управляться по алгоритму управления задвижками.

При моделировании зачастую совместно с блоком управления моделируется и сам орган управления (исполнительный механизм), если отсутствует отдельная модель электроэнергетической системы, т.к. требуется предусмотреть типовые отказы на оборудование и на блок управления.

В данном блоке мы создадим простейшую модель управления задвижкой с возможностью расчета ее положения: на вход в модель будут подаваться две управляющих команды «Открыть» и «Закрыть», а на выходе мы будем получать положение задвижки и ее состояние: **«Открыта»** или **«Закрыта»**.

Рассмотрим процедуру создания блока управления оборудованием:

1. Поместите на схему новый блок **«Субмодель»** из закладки **«Субструкутры»**.
2. Создадим подпись для данного блока. Это необходимо для того, чтобы в дальнейшем при работе со схемой ясно понимать в какой субмодели находится та или иная часть алгоритма/модели. Для этого вызовите окно **«Свойства»**, и на вкладке **«Общие»** найдите свойство **«Подпись»**. Введите в поле «**Значение**» текст **«Управление оборудованием»**. Сохраните внесенные изменения нажатием на кнопку **«Ок»**. После этого отображение блока «Субмодель» примет следующий вид (Рисунок 66);
3. Подпись можно ввести и льтернативным способом – двойным нажатием под блоком. Там следует ввести (в появившемся поле редактора) название субмодели **«Управление оборудованием»** (Рисунок 66). **Примечание**: субмодель слева либо уже удалена, либо её можно пока оставить и удалить позже. Также можно воспользоваться режимом исключения из расчета блока субмодели, без удаления из схемы.

Рисунок 66. Создание подписи блока

1. Осуществите двойной клик на блоке **«Управление оборудованием»**.
2. Поместите на схему новый блок **«Субмодель»** из закладки **«Субструкутры»** и подпишите ее как **«БУЗ»**.

В данном блоке мы создадим простейшую модель двигателя и блока управления клапана (задвижки). На вход в модель будут две команды «Открыть» и «Закрыть». На выходе мы будем получать положение задкижки, и сигналы её состояния – открыт он или закрыт.

1. Осуществите двойной клик на блоке **«БУЗ»**.
2. Поместите на схему два блока «**Порт входа**» и три блока «**Порт выхода**» из закладки «**Субструкутры**» (Рисунок 67). Данные блоки работают не с базой даных, а организуют входные и выходные порты для субмодели.
3. Осуществите двойной клик на блоке **«Порт входа»**.

Рисунок 67. Блоки «Порт входа» и «Порт выхода» в палитре блоков

Рисунок 68. Редактирование имени порта субмодели

1. В окне **«Порт субмодели»** введите строку **«Команда Открыть»** для первого порта входа и **«Команда Закрыть»** для второго (Рисунок 68).
2. Измените имена блоков портов субмодели так, чтобы схема приняла следующий вид (Рисунок 69).

Рисунок 69. Схема субмодели БУЗ с переименованными портами

Для осуществления перехода на один уровень вложенности выше выполните двойной клик на пустом месте схемы, либо нажмите на зелёную стрелку вверх в панели инструментов схемного окна. Обратите внимание, что после добавления портов в схему субмодели, её изображение на схеме также дополнилось портами, к которым можно подводить линии связи. При наведении курсора на любой порт всплывает подсказка с именем порта, присвоенным внутри модели (Рисунок 70).

Рисунок 70. Блок субмодели БУЗ после добавления портов

## Векторная обработка сигналов

Создаваемый нами блок управления задвижкой будет являться единым для категории «**Задвижки**». Поэтому сигналы, которые должны быть обработаны данным блоком, должны быть векторными, а каждый порт будет обрабатывать столько сигналов, сколько групп сигналов представлено в категории, с которой работает данный блок.

Продолжим наполнение блока **«Управление оборудованием»** и создадим алгоритм, «умеющий» обрабатывать векторный сигнал:

1. Поместите на схему рядом с блоком **«БУЗ»** два блока **«Чтение из списка сигналов»** и три блока **«Запись в список сигналов»**.
2. Для удобства дальнейшей работы поместите на схему также блоки **«Заметка»**, позволяющие ввести подписи на схему (Рисунок 71).

Рисунок 71. Блок «Заметка» в палитре компонентов

1. Осуществите двойной клик на надписи «Текст» блока «Заметка» и введите в окно редактора текст пояснения для каждого блока. Схема должна принять вид представленный на рисунке (Рисунок 72).

Рисунок 72. Схема блока управления оборудованием

Блоки «Чтение сигнала из списка» и «Запись сигнала в список» в предыдущих учебных заданиях использовались для чтения и записи единичного сигнала в базе данных. В данном учебном задании мы будем получать не один сигнал, а массив сигналов по всем задвижкам в базе данных. Для этого необходимо выполнить следующие действия:

1. Осуществите двойной клик на блоке **«Чтение сигнала из списка»**.
2. В появившемся окне **«Свойства»** прейдите в строку **«Имена сигналов»** и нажмите кнопку вызова текстового редактора, появляющуюся при редактировании строки справа от текста (Рисунок 73).

|  |  |
| --- | --- |
| Рисунок 73. Редактирование свойств блока «Чтение сигналов из списка» | Рисунок 74. Текстовый редактор запроса к базе данных сигналов |

1. В текстовом окне редактора необходимо сформировать запрос к базе данных. Запрос представляет собой текст в виде одной или нескольких строк в фигурных скобках:

{query:

category = "Задвижки";

group = "\*";

name = "yb01"}

где query: – ключевое слово.

Данный запрос состоит из трех полей:

category = "Задвижки" – название категории («Задвижки»), из которой мы хотим получить сигналы.

group = "\*" – название группы сигналов которую необходимо включить в запрос. Знак \* означает, что в запрос необходимо включить все группы сигналов данной категории. Обратите внимание, что база данных позволяет осуществлять выборку групп сигналов по фильтру.

Например: если в проекте есть задвижки, название которых начинаются на букву D, то для получения сигналов только от таких задвижек достаточно записать group="D\*".

name="yb01" – имя сигнала, который требуется получить из базы данных. В данном случае это имя соответствует сигналу «Команда Открыть». Это имя мы задавали, когда формировали свойства категории (Рисунок 65).

1. Закройте текстовый редактор, нажав кнопку **«Применить»** в верхней части окна (Рисунок 74).
2. Закройте окно редактирования свойств блока, нажав кнопку **«Ок»**.
3. Повторите пункты 1–5 для всех блоков чтения и записи сигналов на схеме. При этом следует вводить строки запроса, соответствующие назначению сигналов в базе данных:

«Команда Закрыть»:

{query: category = "Задвижки"; group = "\*"; name = "yb02"}

«Открыта»:

{query: category = "Задвижки"; group = "\*"; name = "xb01"}

«Закрыта»:

{query: category = "Задвижки"; group = "\*"; name = "xb02"}

«Положение»:

{query: category = "Задвижки"; group = "\*"; name = "xq01"}

1. Добавьте на схему блок **«Временной график»** и соедините блоки линиями связи, как показано на рисунке (Рисунок 75):

Рисунок 75. Субмодель управления оборудование после соединения блоков

## Редактирование параметров «нового» блока

Созданный алгоритм позволяет получить массив сигналов (другое название: вектор сигналов) из базы данных по задвижкам, обработать его в блоке **«БУЗ»**, а затем результаты обработки передать обратно в базу данных.

Для корректной работы алгоритма, работающего с вектором сигналов необходимо точно учитывать размерность вектора. Размерность вектора в нашей модели равна количеству групп сигналов, находящихся в категории **«Задвижки»** в базе данных сигналов. В данном учебном задании мы зададим размерность вектора в качестве нового свойства для блока **«БУЗ»**.

SimInTech позволяет создавать новые блоки с произвольным набором свойств и параметров. Созданная субмодель может быть закрыта для редактирования и использована в виде «черного» ящика при разработке более сложных алгоритмов и моделей. Для добавления нового свойства выполните следующие действия:

1. Выделите **«Субмодель «БУЗ»** на схеме, выберите в главном окне программы пункт меню «**Правка**» подпункт «**Изменить блок..**» (Рисунок 76). Если данный пункт недоступен, то необходимо включить «**Режим разработчика**», и повторить пункты сначала. В результате появится окно «**Редактор новых блоков**».
2. В окне **«Редактор новых блоков»** нажмите кнопку **«Добавить свойство»** в нижней части окна. Введите следующие значения полей (Рисунок 77).

|  |  |
| --- | --- |
| Название | Количество задвижек |
| Имя | Z\_Count |
| Тип данных | Целое |
| Значение | {query: category = "Задвижки"; group= "\*"; name= "\*xb01"; what=count} |
| Способ расчета | **Переменная** |

Рисунок 76. Меню изменения блока

В строке «Значение» формируется запрос к базе данных, который возвращает количество сигналов, в имени которых присутствует текст «xb01» из категории «Задвижки». Поскольку во всех задвижках есть только один сигнал «xb01», то возвращаемое значение соответствует количеству задвижек (клапанов) в базе данных сигналов.

1. Закройте диалоговое окно **«Редактор новых блоков»** нажатием кнопки **«Ок»**.

Рисунок 77. Окно редактора новых блоков

## Создание модели управления клапаном

Простейшая модель управления клапаном будет работать по следующему принципу. На вход модели поступают два сигнала логического типа: «Команда Открыть» и «Команда Закрыть». При работе алгоритма возможны несколько вариантов комбинирования данных команд: когда обе команды инициированы, когда обе команды не инициированы, когда инициирована одна из команд. Рассмотрим данные варианты ниже.

В случае, когда обе команды равны «0» (логическая Ложь) или обе команды равны «1» (логическая Истина, но команды противоречивы), задвижка (клапан) не меняет своего положения. Команды не имеют друг перед другом приоритета, и поэтому одновременная инициация обеих команд ведет к их взаимоблокировке.

Если «Команда Открыть» равна «1» и «Команда Закрыть» равна «0», блок управления изменяет положение клапана (задвижки), увеличивая его с постоянной скоростью, до достижения одного из следующих условий:

1 – «Положение» клапана равно «100» (полностью открыта);

2 – «Команда Открыть» становится равной «0».

Если «Команда Закрыть» равна «1» и «Команда Открыть» равна «0», блок управления изменяет положение задвижки до достижения одного из следующих условий:

1 – «Положение» задвижки равно «0» (полностью закрыта);

2 – «Команда Закрыть» равна «0».

При достижении значений «Положение» величины «100» или «0» сигналы на выходе из блока в портах с именами «Открыта» или «Закрыта», соответственно, принимают значения «1» (логическая Истина), в противном случае они равны «0» (логическая Ложь), тем самым сигнализируя о том, что задвижка находится в промежуточном положении.

Войдите в субмодель «БУЗ» и добавьте на схему следующие блоки:

* 1 блок **«Сумматор»** (закладка **«Операторы»**);
* 1 блок **«Интегратор с ограничением»** (закладка **«Динамические»**);
* 2 блока **«Константа»** (закладка **«Источники»**);
* 2 блока **«Логические операции»** (закладка **«Логические»**).

После установки данных блоков на поле схемы, соедините их линиями связи между собой (Рисунок 78).

Затем произведите настройку установленных блоков.

Для блоков **«Константа»** установите следующие значения свойства **«Значение»** на вкладке **«Свойства»** в диалоговом окне: для первого блока установите значение **«100»**, для второго блока установите значение **«0»**.

Для блока «Сумматор» установите значение весовых множителей [1,-1]. Вместо сумматора можно установить и сравнивающее устройство.

Рисунок 78. Структурная схема БУЗ

Таким образом, на выходе из блока «Сумматор» будет значение «1», когда «Команда Открыть» равна «1» и «Команда Закрыть» равна «0». На выходе из блока «Сумматор» будет значение «-1», когда «Команда Открыть» равна «0» и «Команда Закрыть» равна «1». Во всех остальных случаях на выходе сумматора будет нулевое значение «0». Поскольку все сигналы векторные, то в каждой линии связи будет передаваться массив сигналов.

Для формирования сигналов «Открыта» и «Закрыта» используется логический блок «Логические операции» из закладки «Логические» (Рисунок 79)

Рисунок 79. Блок логичесике операции

|  |  |
| --- | --- |
| Рисунок 80. Редактирование блока «Логические операции» | Блок производит логическую операцию над векторным сигналом. В нашем случае мы проводим сравнение векторного сигнала с константой. Блок проверяет каждый из сигналов вектора, входящего в блок, и возвращает вектор, в котором значение «1» (логическая Истина) принимают сигналы, которые равны константе (скалярному сигналу на втором входном порте).  В настройках блока необходимо указать тип второго блока «Скаляр». |

Блок «Интегратор с ограничением» реализует расчет положения задвижек (моделирует двигатель с постоянным быстродействием) на основании поступающих на вход сигналов: если вход равен «-1», то уменьшает положение, если «+1» то увеличивает.

В нашем простейшем примере мы считаем, что положение изменяется линейно с постоянной скоростью, одинаковой для всех задвижек. Скорость изменения пропорциональна коэффициенту усиления интегратора. Для обработки вектора сигналов необходимо задать вектор значений для каждого параметра блока.

Например, если блок рассчитывает 4 сигнала, то необходимо ввести четыре коэффициента усиления для каждого из сигналов в векторе: [1,1,1,1]. Для одинаковых по величине значений можно использовать запись 4#1, которая эквивалентна вектору (массиву) [1,1,1,1].

Войдите в диалоговое окно редактирования свойств блока «Интегратор с ограничением» и установите следующие свойства блока:

|  |  |
| --- | --- |
| Коэффициенты усиления | Z\_Count#1 |
| Минимальное значение | Z\_Count#0 |
| Максимальное значение | Z\_Count#100 |
| Начальные условия | Z\_Count#0 |

Закройте окно редактирования свойств нажатием кнопки «Ок».

Запись Z\_Count#1 позволяет заполнить массив значений для вектора. В зависимости от величины «Z\_Count» формируется различный массив значений. Поскольку «Z\_Count» вычисляется путем обращения к базе данных, то он будет соответствовать количеству задвижек, существующих в базе данных. Таким образом, данный блок будет всегда корректно обрабатывать вектор сигналов при изменении количества задвижек в базе данных.

Если схема набрана правильно, то ее можно запустить на расчет и убедиться , что по всем линиям связи передается вектор сигналов, состоящий из двух элементов, в соответствии с количеством задвижек в базе данных (Рисунок 82).

|  |  |
| --- | --- |
| Рисунок 81. Свойства «Интегратора с ограничением» | А в базе данных для всех задвижек значение сигнала «Закрыта» устанавливается в истину (в соответствии с начальными условиями интегратора). |

Рисунок 82. Проверка работоспособности схемы БУЗ

Сохраните созданную модель.