



Модуль UM Control

Модуль UM Control включает в себя набор инструментов, которые позволяют связывать модели, созданные в «Универсальном механизме», с расчётными схемами в Matlab/Simulink, SimInTech, встроенном редакторе схем и собственными расчётными кодами пользователей.

Оглавление

19. МОДУЛЬ UM CONTROL	19-3
19.1. ИНСТРУМЕНТ COSIMULATION	19-3
19.1.1. Экспорт моделей	19-4
19.1.1.1. Мастер экспорта в Matlab/Simulink	19-4
19.1.1.2. Включение нескольких UM моделей в модель Matlab/Simulink	19-5
19.1.1.3. Настройки синтеза m-файлов	19-6
19.1.2. Структура m-файла	19-8
19.1.2.1. Инициализация	19-8
19.1.2.2. Расчет выходных значений	19-10
19.1.2.3. Завершение моделирования	19-11
19.1.3. Особенности работы в Matlab/Simulink	19-12
19.1.3.1. Настройка параметров Matlab/Simulink	19-12
19.1.3.2. Инициализация параметров S-функции	19-14
19.1.3.3. S-функция с несколькими входами и выходами	19-14
19.1.4. Настройки модели UM	19-17
19.1.5. Переносимость моделей	19-18
19.2. ИНСТРУМЕНТ UDP BRIDGE.....	19-19
19.2.1. Введение	19-19
19.2.2. IP-адрес и порт	19-19
19.2.3. Синхронизация запуска приложений	19-21
19.2.4. Обмен данными в процессе работы	19-22
19.2.5. Моделирование в реальном времени	19-23
19.2.6. Формат данных.....	19-24
19.2.7. Чтение и запись нескольких сигналов	19-25
19.2.8. Особенности UDP интерфейса в SimInTech.....	19-25

19. Модуль UM Control

19.1. Инструмент CoSimulation

Инструмент **CoSimulation** из модуля **UM Control** позволяет экспортировать модели, созданные в «Универсальном механизме», для последующей их интеграции в модели Matlab/Simulink. Для импорта UM моделей в Matlab/Simulink необходимо использовать так называемые S-функции, которые и будут осуществлять связь между Matlab/Simulink и ПК «Универсальный механизм».

S-функция имеет список входных и выходных сигналов, а также список параметров. Входные сигналы S-функции связываются с параметрами UM модели, которые обычно описывают силы и моменты (управляющее воздействие). Таким образом в UM модель передаются силы и моменты, сформированные в модели Matlab/Simulink. Выходные величины S-функции являются переменными, созданными в **Мастере переменных**, которые обычно описывают кинематику механической системы. В качестве параметров S-функции устанавливаются те параметры модели «Универсального механизма», которые инициализируются перед началом процесса моделирования и в течение моделирования остаются неизменными. К таким параметрам может относиться, например, жесткость пружины, масса тела и т.д.

Для корректного функционирования S-функции необходим специальный программный код, который будет управлять этой S-функцией. Управляющий код представляет собой файл с расширением *.m, который написан на языке Matlab, так называемый m-файл.

19.1.1. Экспорт моделей

Управляющий m-файл для S-функции генерируется автоматически при подготовке UM модели к экспорту с помощью **Мастера экспорта** в **Matlab/Simulink**, программа **UM Simulation**. Вместе с тем пользователь может создать самостоятельно в среде Matlab, придерживаясь той структуры, которая описана выше.

19.1.1.1. Мастер экспорта в Matlab/Simulink

С помощью **Мастера экспорта** в **Matlab/Simulink** происходит создание файлов настроек и управляющего m-файла, необходимых для подключения UM моделей к моделям систем управления в Matlab/Simulink. Подробнее об особенностях работы с **Мастером экспорта** в **Matlab/Simulink** см. раздел «Начинаем работать: интерфейс с Matlab/Simulink» руководства пользователя UM.

В общем случае указание необходимых величин для модели механической системы и сохранение связанных с этим настроек выполняется в программе **UM Simulation**, пункт меню **Инструменты | Мастер экспорта в Matlab/Simulink**, см. рис. 19.1.

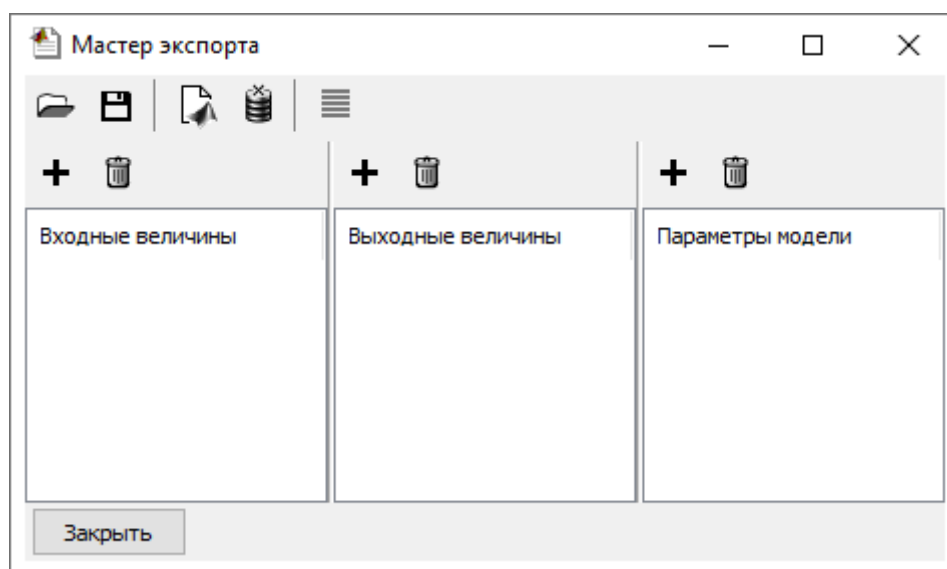


Рис. 19.1. Мастер экспорта в Matlab/Simulink

Входные величины и параметры S-функции связываются с параметрами (идентификаторами) модели.

Поэтому, несмотря на то, что назначение параметров модели в качестве входных величин или параметров S-функции производится только в программе моделирования динамики (**UM Simulation**), создать и должным образом параметризовать соответствующие силовые элементы нужно еще на этапе подготовки модели в программе **UM Input**.

Для выбора параметра UM модели, которому будет присваиваться значение, получаемое из S-функции, сделайте двойной щелчок на нужном элементе из списка **Входные величины**, см. рис. 19.1. Откроется диалоговое окно выбора параметров, см. рис. 19.2.

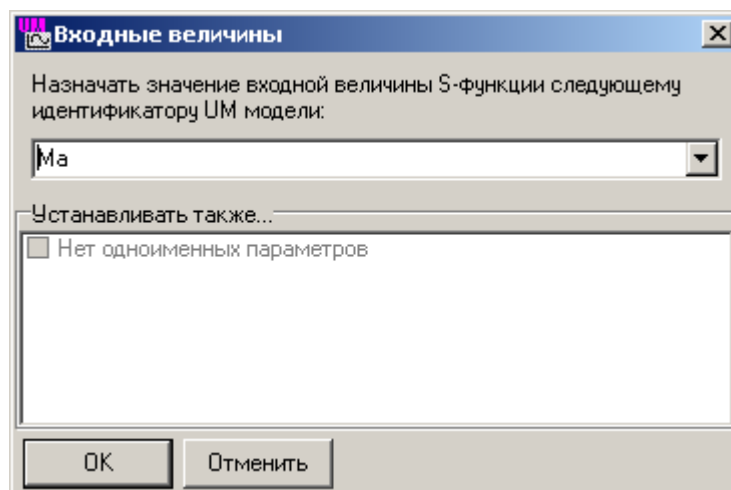



Рис. 19.2. Назначение параметров на вход S-функции

На выход S-функции назначаются переменные, см. список **Выходные величины** на рис. 19.1. Для назначения переменной на некоторый выход S-функции требуемую переменную нужно просто перетащить мышкой на этот выход, пользуясь обычной технологией *Drag-and-Drop* из **Мастера переменных** или **Списка переменных**. Таким образом, любая переменная, которую можно создать в **Мастере переменных**, доступна для назначения на выход S-функции.

19.1.1.2. Включение нескольких UM моделей в модель Matlab/Simulink

В тех случаях, когда модель системы управления **Matlab/Simulink** предусматривает возможность включения в нее нескольких моделей механических систем, созданных в UM, то необходимо для каждой такой модели механической части создать свою S-функцию. С помощью **Мастера экспорта в Matlab/Simulink** необходимо для каждой модели создать и сохранить файлы настроек и управляющий m-файл.

Сформировав списки входных, выходных величин и параметров модели, нужно открыть **Окно параметров моделирования**. Для этого нажмите на кнопку . Появится окно, представленное на рис. 19.3.

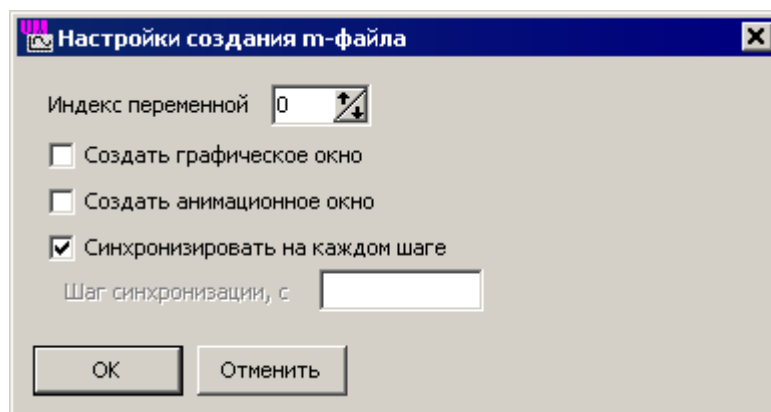


Рис. 19.3. Параметры синтеза m-файла

В поле **Индекс переменной** необходимо для каждой вновь создаваемой модели механической части указать уникальное число. Например, если система управления содержит две модели механической системы, то в **Мастере экспорта в Matlab/Simulink** для первой модели можно указать **Индекс переменной** равный 1, а для второй модели – равный 2. Это делается для того, чтобы для каждой модели механической системы создавался свой экземпляр COM-сервера со своим уникальным именем. Указав значение **Индекса переменной**, нужно сохранить файлы настроек и сгенерировать m-файл.

Помимо этого в m-файл можно вставлять команды для создания анимационных и графических окон UM, которые будут появляться во время работы в Matlab/Simulink.

19.1.1.3. Настройки синтеза m-файлов

Окно задания настроек m-файла показано на рис. 19.3. Рассмотрим подробнее смысл данных настроек.

Индекс переменной

Данная настройка используется при загрузке нескольких моделей UM в одну модель Matlab/Simulink, особенности работы с данной настройкой подробно рассмотрены в п. 19.1.1.2. *"Включение нескольких UM моделей в модель Matlab/Simulink"*, с. 19-5.

Создать анимационное (графическое) окно

При включении данных флажков в код m-файла записываются команды создания анимационного и/или графического окна UM, соответственно. Таким образом, в процессе моделирования динамики системы в Matlab/Simulink одновременно появятся анимационное и/или графическое окна UM. Это бывает очень полезно, особенно на этапе отладки моделей и их взаимодействия.

Синхронизировать на каждом шаге

Включенный флажок **Синхронизировать на каждом шаге** означает, что передача данных в UM, выполнение там шага интегрирования и получение данных обратно в Matlab/Simulink будет выполняться на каждом шаге численного метода Matlab/Simulink. Если снять флажок **Синхронизировать на каждом**, то в поле **Шаг синхронизации** можно установить время в секундах, через которое будет происходить обмен данными между Matlab/Simulink и UM. Если Шаг синхронизации окажется меньше шага численного метода Matlab/Simulink, то синхронизация будет выполняться на каждом шаге численного метода независимо от установленного **Шага синхронизации**.

В некоторых случаях, когда частота процессов в модели Matlab/Simulink гораздо выше частот механической части, целесообразно выполнять обмен данными с шагом большим, чем шаг интегрирования, установленный (или выбранный автоматически) в Matlab/Simulink. В таких случаях имеет смысл высокочастотные (например, электрические) и, как правило, низкочастотные механические процессы интегрировать с разным шагом. Это позволит существенно уменьшить вычислительные затраты и, соответственно, ускорить процесс численного моделирования.

Типичным примером такой системы является модель системы управления асинхронным двигателем, выполненная в Matlab/Simulink и связанная с механической системой, смоделированной в UM. Частоты электрической части (ШИМ) в таком случае будут на несколько порядков выше частот механической части. В частности, это означает, что шаг моделирования электрической и механической части будут отличаться примерно в той же пропорции.

Применение такого приема на практике позволяет ускорить моделирование подобных задач на несколько порядков практически без внесения заметных погрешностей в результаты. Для большинства практических задач можно рекомендовать устанавливать **Шаг синхронизации** в интервале от $1E-2$ до $1E-4$ секунд. Вместе с тем, при использовании **Шага синхронизации** строго рекомендуется проверить его влияние на получаемые результаты моделирования. Для этого необходимо провести несколько тестовых расчетов с разным **шагом синхронизации**, начиная с шага, близкого к шагу модели в Matlab/Simulink, с постепенным увеличением шага, и, таким образом, выбрать оптимальный **шаг синхронизации** по соотношению точность/время моделирования.

19.1.2. Структура m-файла

Синтаксис функции m-файла имеет следующий вид:

$[sys, x0, str, ts] = f(t, x, u, flag, p1, p2, \dots)$, где

f – название S-функции;

t – текущее время;

x – вектор состояний соответствующего блока S-функции;

u – вектор входных значений блока S-функции;

$flag$ – флаг, который указывает, какая задача будет выполняться на очередном этапе моделирования (инициализация, расчет выходных значений, завершение моделирования);

p_1, p_2 – параметры блока S-функции.

В процессе моделирования Simulink повторно вызывает функцию f , указывая с помощью флага $flag$, какая именно задача будет выполняться для конкретного вызова. Ниже рассмотрим основные задачи, которые выполняются при вызове функции:

инициализация (`mdlInitializeSizes`);

расчет выходных значений (`mdlOutputs`);

завершение моделирования (`mdlTerminate`).

19.1.2.1. Инициализация

Когда значение $flag$ равно 0, выполняется этап инициализации, за который отвечает подфункция `mdlInitializeSizes`. На этом этапе определяются основные характеристики блока S-функции, такие как количество входов, выходов и параметров, начальные условия непрерывных и дискретных состояний и некоторые другие. Все характеристики блока S-функции сохраняются в специальной структуре `sizes`. Ниже приведен листинг подфункции `mdlInitializeSizes` для модели системы управления перевернутым маятником.

```
function [sys,x0,str,ts] = mdlInitializeSizes()
sizes = simsizes;
sizes.NumContStates= 0;
sizes.NumDiscStates= 0;
sizes.NumOutputs=1;
sizes.NumInputs=1;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
sys = simsizes(sizes);
x0 = []; % No continuous states
str = []; % No state ordering
ts = [0]; % Inherited sample time
global h1
if (iscom(h1))==0
h1=actxserver('UMCosimulation.UMMatlab');
h1.LoadObjectFromFile('C:\UM\samples\Cosimulation\inv_pend_cosim \input.dat');
h1.LoadMatlabSettings('C:\UM\samples\Cosimulation\inv_pend_cosim
\inv_pend_cosim.cosim');
h1.ReadTotalConfiguration('C:\UM\samples\Cosimulation\inv_pend_cosim\inv_pend
_cosim');
h1.PrepareIntegration();
end
% End of mdlInitializeSizes.
```

Следует отметить, что моделирование динамики системы происходит с помощью СОМ-сервера, который включает в себя все математическое ядро УМ. На этапе инициализации происходит создание экземпляра этого сервера и вызов некоторых его методов.

Создание экземпляра сервера

```
h1=actxserver('UMCosimulation.UMLMatlab');
```

Загрузка модели перевернутого маятника

```
h1.LoadObjectFromFile('C:\UM\samples\Cosimulation\inv_pend_cosim\input.dat');
```

Загрузка настроек модели

Помимо самой модели механической части, необходимо загрузить файл настроек, созданный с помощью **Мастера экспорта в Matlab/Simulink**.

```
h1.LoadMatlabSettings('C:\UM\samples\Cosimulation\inv_pend_cosim\nv_pend_cosim.cosim');
```

Загрузка других необходимых файлов

Файлы с расширением .par, .sim, .xv, .icf, которые сохраняются одновременно с файлом .cosim разгружаются следующим образом:

```
h1.ReadTotalConfiguration('C:\UM\samples\Cosimulation\inv_pend_cosim\inv_pend_cosim');
```

В этом методе в качестве входного аргумента служит путь с названием файла без расширения, поскольку имя файла является общим для вышеописанных настроек.

Подготовка к интегрированию

```
h1.PrepareIntegration();
```

На этом этапе инициализации завершается.

19.1.2.2. Расчет выходных значений

Значению `flag` равному 3 соответствует подфункция `mdlOutputs`, которая служит для вычисления выходных значений S-функции. Для модели перевернутого маятника она имеет следующий вид:

```
function sys = mdlOutputs(t,x,u)
global h1
value=[u(1) 0.0];
h1.SetValues(value);
h1.DoIntegrationInterval(t);
sys=h1.GetValues();
% End of mdlOutputs.
```

Для модели перевернутого маятника на очередном шаге выполнения передается значение усилия, подаваемого на тележку. Это выполняется с помощью метода

```
h1.SetValues(value);
```

Входным значением этого метода является массив. Поскольку вектор `u` содержит только одно значение (величину силы), то мы должны создать массив `value`, в котором первым элементом будет являться значение силы, а второй элемент со значением 0.0 будет просто указывать на то, что в метод передается массив, но на процесс моделирования он никак влиять не будет.

Установив значение силы, необходимо выполнить шаг интегрирования

```
h1.DoIntegrationInterval(t);
```

Выполнив интегрирование, в вектор `sys` заносятся выходные величины

```
sys=h1.GetValues();
```

В нашем случае величина одна, это угол отклонения маятника от вертикали.

19.1.2.3. Завершение моделирования

Значению flag равному 9 соответствует подфункция mdlTerminate, которая служит для завершения процесса моделирования. Эта подфункция вызывается либо когда время моделирования истекло, либо когда процесс моделирования принудительно остановлен. Ниже приведен листинг подфункции mdlTerminate

```
function sys = mdlTerminate(t,x,u)
global h1
h1.FinishIntegration();
h1.delete;
sys=[];
% End of mdlTerminate.
```

Завершение процесса интегрирования

```
h1.FinishIntegration();
```

Освобождение экземпляра сервера

```
h1.delete;
```

В вектор sys помещается нулевое значение

```
sys=[];
```

19.1.3. Особенности работы в Matlab/Simulink

19.1.3.1. Настройка параметров Matlab/Simulink

Для предотвращения проблем с типами данных настоятельно рекомендуется в настройках модели выключить флажок **Preferences | Simulink | Launch Simulink Preferences | Optimization | Implement logic signals as boolean data (vs. double)**, см. рис. 19.4. Кроме того, это позволит избежать проблем совместимости для моделей, созданных в более ранних версиях **Simulink**.

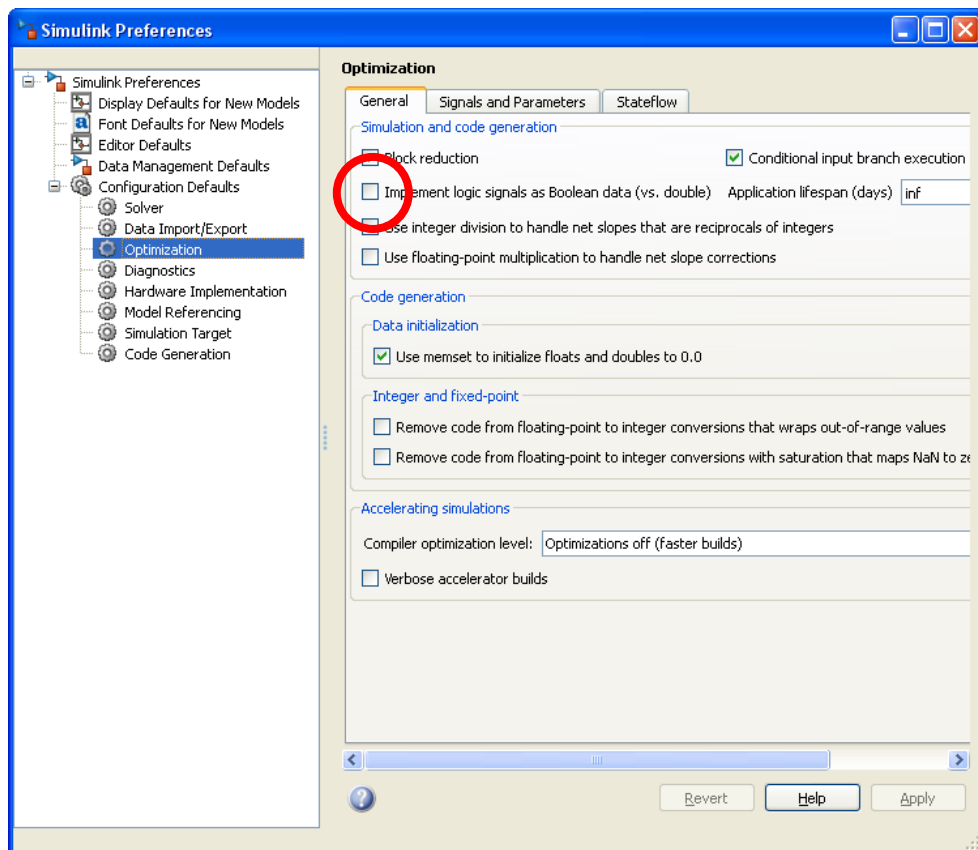


Рис. 19.4. Окно настройки параметров

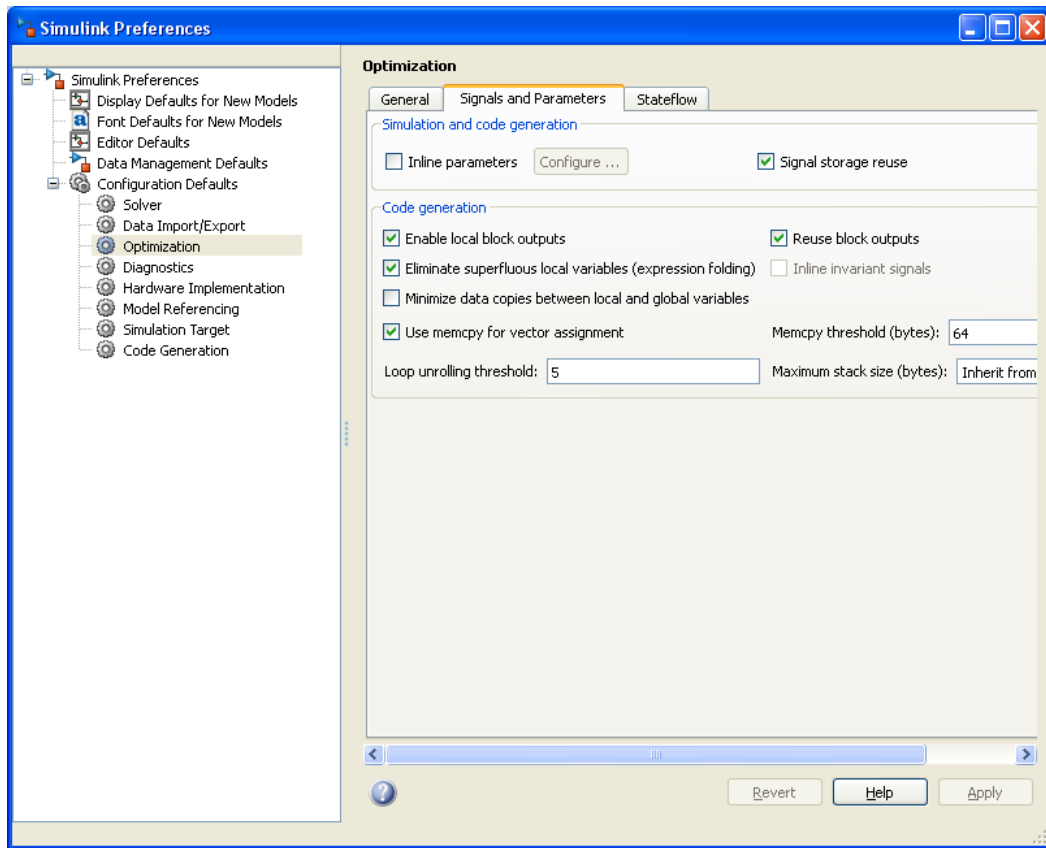


Рис. 19.5. Окно настройки параметров

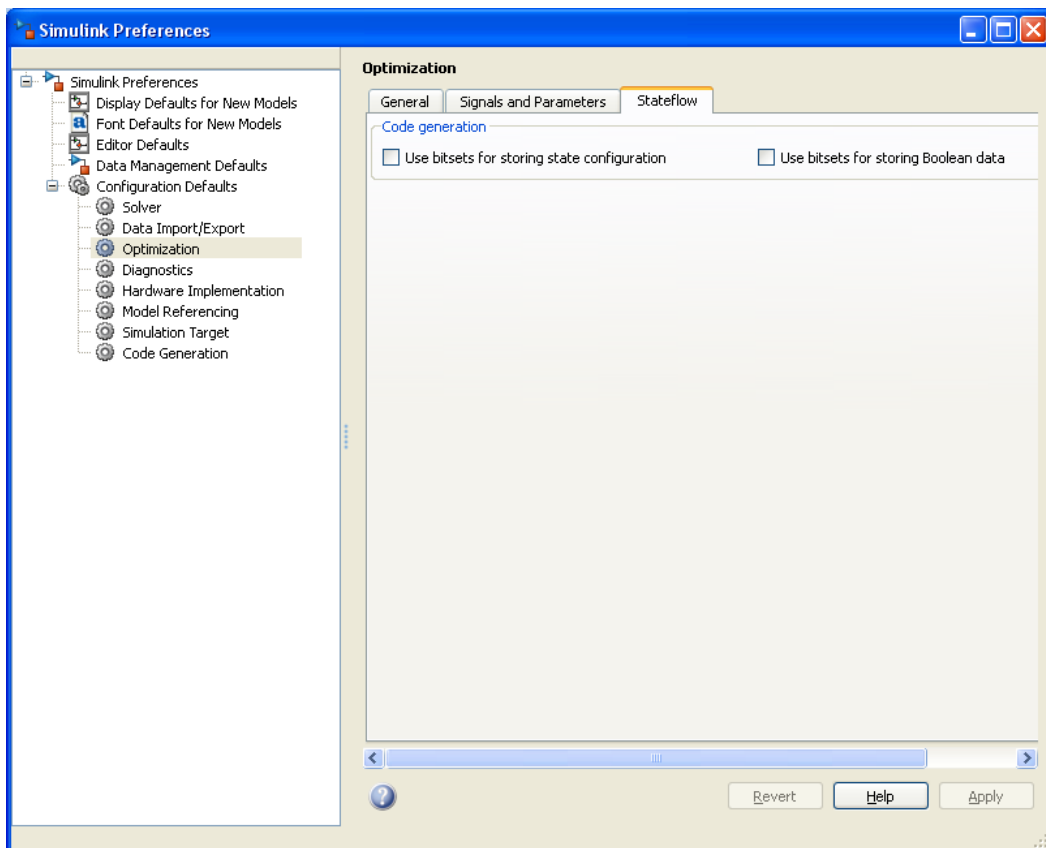


Рис. 19.6. Окно настройки параметров

19.1.3.2. Инициализация параметров S-функции

Как уже было сказано выше, те величины модели механической системы, которые инициализируются перед началом процесса моделирования и в течение моделирования остаются неизменными, называются параметрами модели. Они добавляются также как и входные величины модели в **Мастере экспорта в Matlab/Simulink**. После генерации m-файла список добавленных параметров появляется в заголовке функции.

$[sys,x0,str,ts]=f(t,x,u,flag,p1,p2,...)$ где

$p1, p2$ – параметры блока S-функции.

Инициализация параметров блока S-функции происходит в диалоге задания параметров S-функции вместе с заданием имени S-функции, см. рис. 19.7. Для этого в поле **S-function parameters** необходимо указать значения параметров S-функции в том порядке, в каком они добавлялись в **Мастере экспорта в Matlab/Simulink** (в таком же порядке они перечисляются в m-файле). В качестве значений параметров нужно указать либо константы, либо имена переменных, которые содержатся в модели системы управления Matlab/Simulink. Пример инициализации двух параметров S-функции приведен на рис. 19.7.

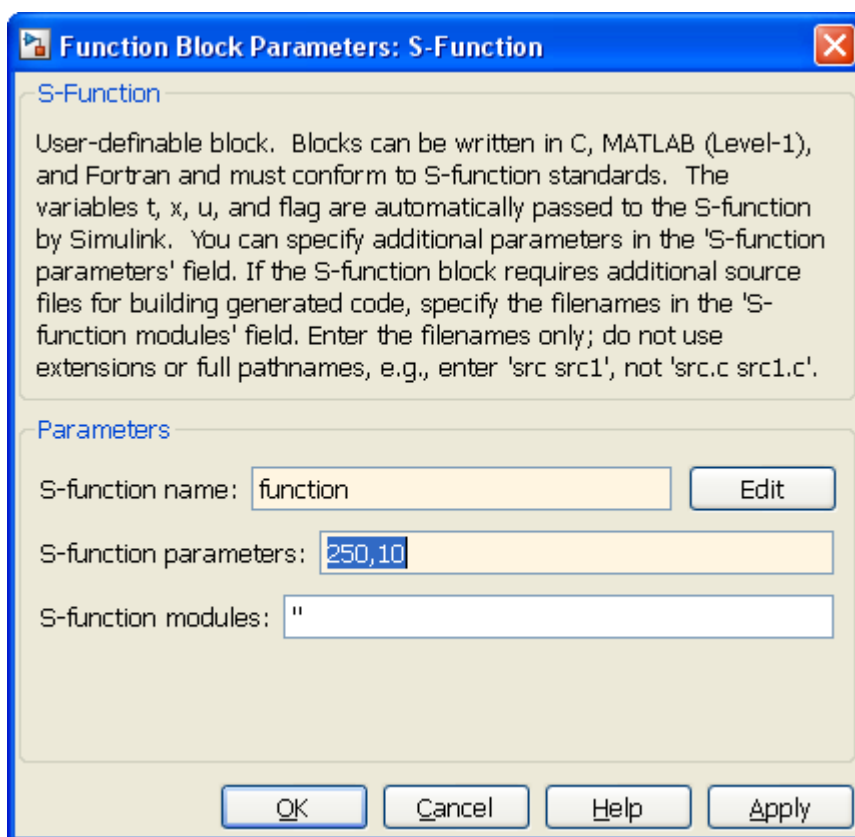


Рис. 19.7. Параметры S-функции

19.1.3.3. S-функция с несколькими входами и выходами

Блок S-функции может иметь только один входной и один выходной порты. Если предполагается, что в качестве входных величин S-функции будет передаваться несколько

сигналов, то все они должны приходиться на этот входной порт. Для реализации этого в модели Matlab/Simulink добавляется компонент **Mux**. Аналогично, если необходимо работать с несколькими выходными величинами, то необходимо воспользоваться компонентом **Demux**. Эти элементы можно найти в библиотеке **Simulink Commonly Used Blocks**. Пример S-функции с двумя входными и тремя выходными сигналами приведен на рис. 19.8.

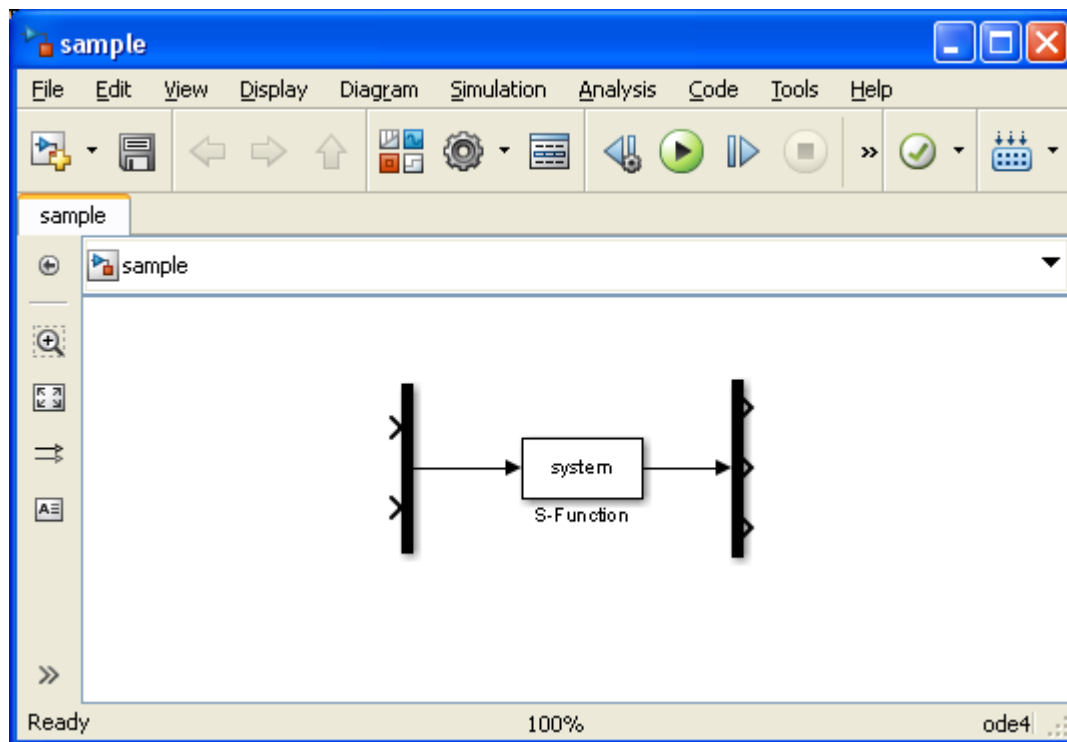


Рис. 19.8. Компоненты Mux и Demux

Элементы **Mux** и **Demux** связывается соответственно с входом и выходом S-функции. Количество сигналов, которые будут подаваться на **Mux** или забираться с **Demux**, нужно указать в диалоге настройки параметров этих элементов, которое вызывается двойным щелчком мыши по каждому из них, см. рис. 19.9 и рис. 19.10.

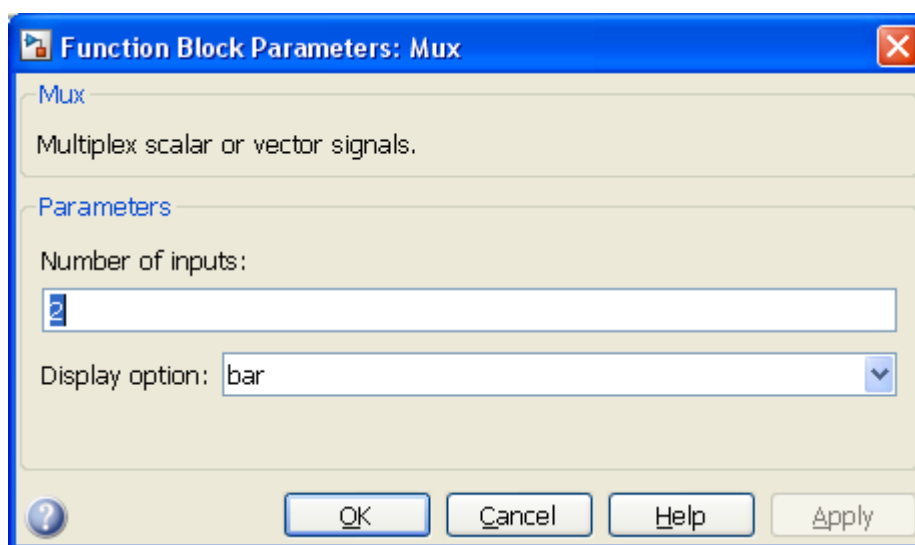


Рис. 19.9. Параметры компонента Mux

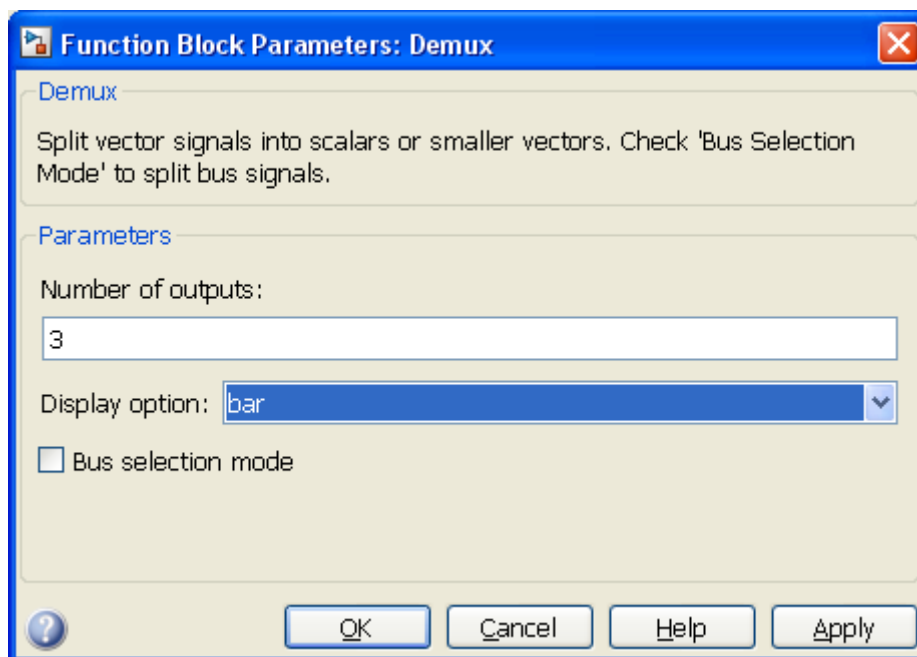


Рис. 19.10. Параметры компонента Demux

19.1.4. Настройки модели UM

Сохранение всех настроек модели UM (начальные условия, параметры интегрирования, значения идентификаторов, конфигурация железнодорожных экипажей, автомобилей, гусеничных машин) происходит при генерации m-файла в файлы .par, .icf, .car и т.д. с именем m-файла. Таким образом, перед генерацией m-файла необходимо установить те значения параметров, с которыми модель в дальнейшем будет рассчитываться в Matlab/Simulink.


Загрузка настроек UM модели в среде Matlab/Simulink выполняется на этапе инициализации модели с помощью метода **ReadTotalConfiguration**, подробнее про инициализацию UM модели см. п. 19.1.2.1. *"Инициализация"*, с. 19-8.

Рассмотрим более подробно порядок присвоения значений идентификаторам UM модели при ее выполнении в среде Matlab/Simulink.

1. На инициализации выполняется загрузка значений идентификаторов UM модели из par-файла, который был сохранен при синтезе управляющего m-файла, см. п. 19.1.2.1. *"Инициализация"*, с. 19-8.
2. После чего также на инициализации назначаются актуальные значения идентификаторам, которые связаны с параметрами S-функции, см. п. 19.1.3.1. *"Настройка параметров Matlab/Simulink"*, с. 19-12. Таким образом, значения, присвоенные идентификаторам в п. 1 переопределяются.
3. Далее, в процедуре на каждом шаге численного метода текущие значения входов S-функции назначаются связанным с ними параметрам, см. п. 19.1.2.2. *"Расчет выходных значений"*, с. 19-10.

19.1.5. Переносимость моделей

Обратите внимание, что при автоматической генерации m-файла в него записываются прямые пути к UM-модели и к файлам настроек, см. п. 19.1.2.1. *"Инициализация"*, с. 19-8. Поэтому при перемещении или копировании модели UM в другой каталог или на другой компьютер, ранее созданный m-файл в общем случае будет содержать уже некорректные пути и его необходимо сгенерировать заново.

Для этого необходимо загрузить модель в программу **UM Simulation**, открыть **Мастер экспорта в Matlab/Simulink**, загрузить нужный .cosim файл и щелкнуть кнопку  для вызова диалога сохранения m-файла. После этого m-файл будет сгенерирован с учетом текущего пути к модели UM.

Так как при создании m-файла будут переписаны все файлы настроек, то перед генерацией m-файла необходимо загрузить ранее подготовленные файлы настроек, которые включают начальные условия, параметры интегрирования, значения идентификаторов, конфигурацию железнодорожных экипажей, автомобилей, гусеничных машин. При загрузке .cosim файла появляется приглашение загрузить также все настройки из одноименных файлов. Подробнее про настройки UM-модели см. п. 19.1.4. *"Настройки модели"*, с. 19-17.

19.2. Инструмент UDP Bridge

19.2.1. Введение

Инструмент **UDP Bridge** из модуля **UM Control** позволяет связывать модели, созданные в «Универсальном механизме», с другими программами по сетевому протоколу UDP. Этот инструмент позволяет отправлять результаты моделирования в "Универсальном механизме" на удалённый компьютер, а также принимать сигналы с удалённого компьютера. Из "Универсального механизма" можно отправлять любую переменную, созданную в **Мастере переменных**. Обычно отправляются кинематические величины: координаты, скорости, ускорения. Принимаемые сигналы присваиваются идентификаторам модели и обычно это идентификаторы, описывающие силовое воздействие.

Можно описать несколько типовых сценариев использования инструмента **UDP Bridge**. Первый сценарий подразумевает подключение программных или аппаратных систем управления к модели в "Универсальном механизме". Из "Универсального механизма" отправляются показания виртуальных датчиков. А в "Универсальный механизм" поступает управляющее силовое воздействие. Второй сценарий подразумевает использование УМ как ядро расчёта динамики в рамках некоторого интерактивного комплекса, например, симулятора водителя автомобиля или машиниста поезда. При таком сценарии из УМ во внешние программы по UDP протоколу отправляется положение и ориентация кабины для формирования корректной картинке на каждом шаге моделирования и другие сигналы, например, для формирования виртуальной приборной панели. А на вход в "Универсальный механизм" принимаются сигналы, характеризующие действия водителя, например, положение педали газа или тормоза.

Ниже рассмотрим некоторые ключевые особенности связывания УМ с другими программами по UDP протоколу.

19.2.2. IP-адрес и порт

Номер порта должен принадлежать диапазону 0–65535. Все порты разделены на три диапазона – общеизвестные (или системные, 0–1023), зарегистрированные (или пользовательские, 1024–49151) и динамические (или частные, 49152–65535). **Для собственных программ рекомендуется использовать динамические (частные) порты в диапазоне 49152–65535.** Этот диапазон зарезервирован для временных соединений, не конфликтует с известными службами (0–1023) и зарегистрированными приложениями (1024–49151), обеспечивая стабильность и отсутствие конфликтов при работе приложений.

Чтение данных выполняется из порта на вашем компьютере. Поэтому при настройке получения сигналов опускается указание IP-адреса, это всегда IP-адрес вашего компьютера. Отправка данных выполняется в порт удалённого компьютера, поэтому задаётся пара IP-адрес получателя и порт получателя. Для отправки данных в порт своём компьютере укажите IP-адрес **127.0.0.1**.

Если вы организуете обмен данными в пределах одного своего компьютера, то нельзя указать один и тот же порт и для получения и для отправки сигналов. В таком случае нужно указывать разные порты.

Интерфейс **Мастера связи по UDP протоколу** для отправки и получения данных в УМ приведен на рис. 19.11 и рис. 19.12.

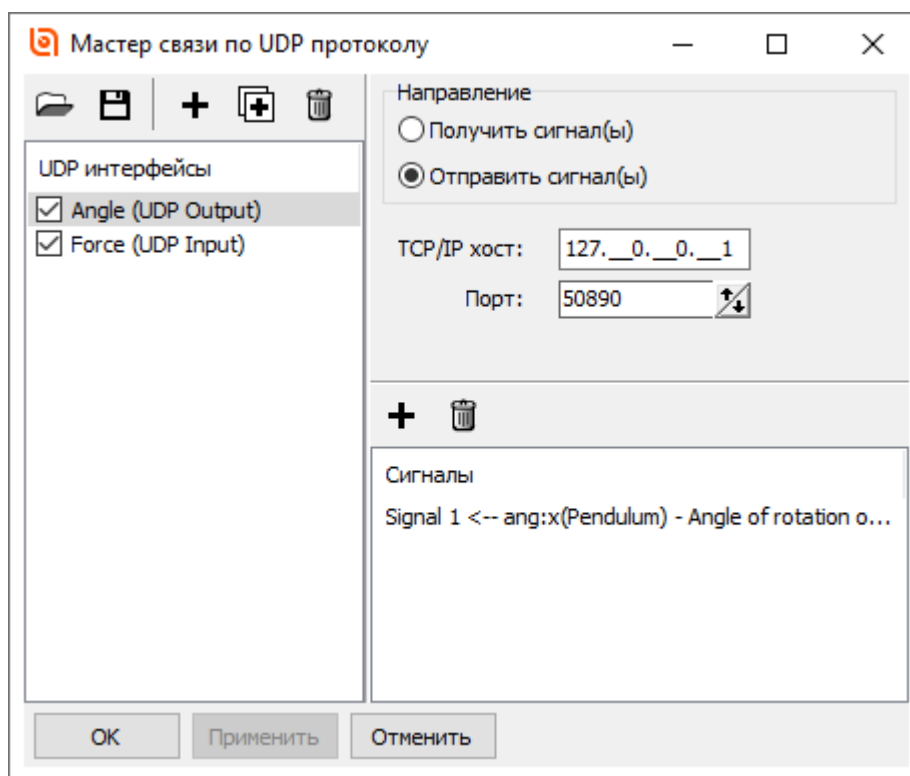


Рис. 19.11. Настройка отправки сигналов

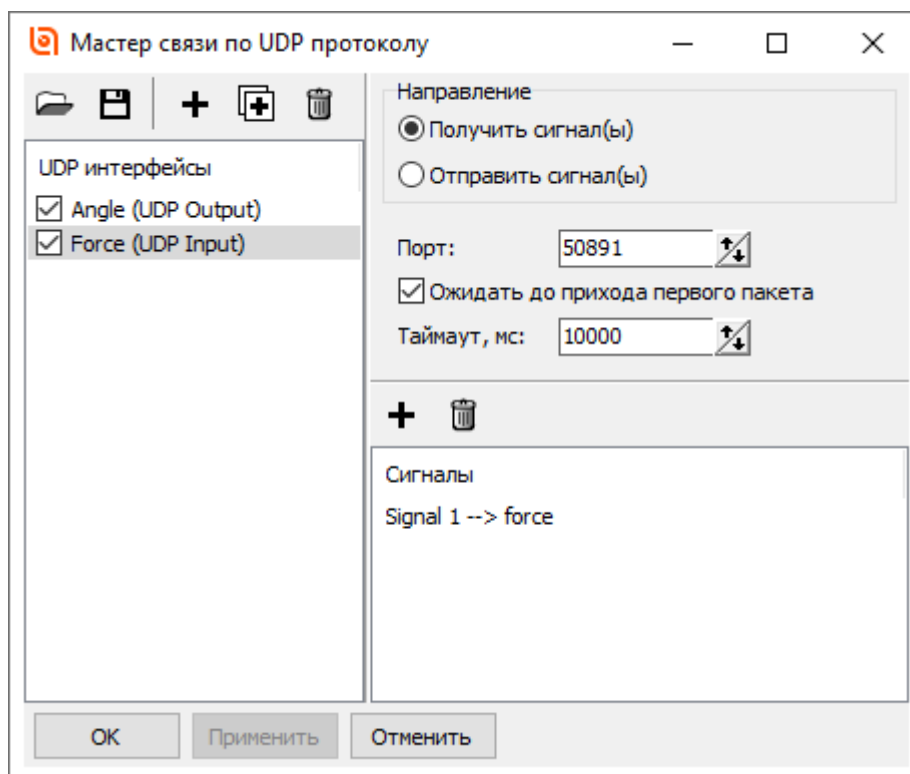


Рис. 19.12. Настройка получения сигналов

19.2.3. Синхронизация запуска приложений

Важной особенностью связывания приложений по UDP протоколу является необходимость синхронизации запуска приложений. Действительно, если запустить расчёт модели перевёрнутого маятника в "Универсальном механизме" (см. рис. 19.13) без работающей на момент запуска расчётов системы управления, то это приведёт к тому, что вместо стабилизации в перевёрнутом вертикальном положении маятник просто будет совершать свободные колебания вокруг нижнего устойчивого положения равновесия.

Рассмотрим ситуацию на примере подключения системы управления из программы SimInTech. Чтобы корректно синхронизировать Универсальный механизм и SimInTech в этом случае, нужно в свойствах **UDP сервера** в **SinInTech** включить флажок **Ожидать до прихода пакета** (см. рис. 19.14) и сначала запустить модель системы управления в SimInTech, а потом уже модель перевёрнутого маятника в Универсальном механизме. Таким образом модель в SimInTech будет ждать до прихода первого сигнала из УМ и далее две части этой системы совместно и корректно заработают.

Для этой же цели служит флажок **Ожидать до прихода первого пакета** в свойствах интерфейса для получения данных **Мастера связи по UDP протоколу**, см. рис. 19.12. Время ожидания первого пакета указывается ниже в поле **Таймаут**. Таймаут указывается в миллисекундах.

Вместе с тем, если модели в Универсальном механизме и связанные модели в других программах находятся в безразличном положении при отсутствии сигналов по UDP, то флажки **Ожидать до прихода первого пакета** в УМ и аналогичные в связанных программах можно не включать. Примером механической модели в безразличном состоянии является, например, модель автомобиля на ровной площадке или на стояночном тормозе. С такой механической системой ничего не произойдёт до того, как человек или автоматизированная система не начнут подачу команд.

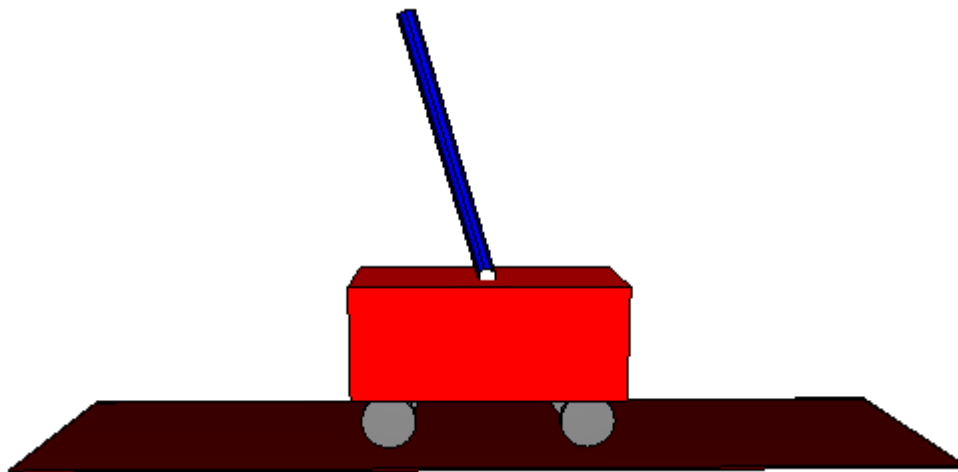


Рис. 19.13. Модель перевёрнутого маятника в УМ

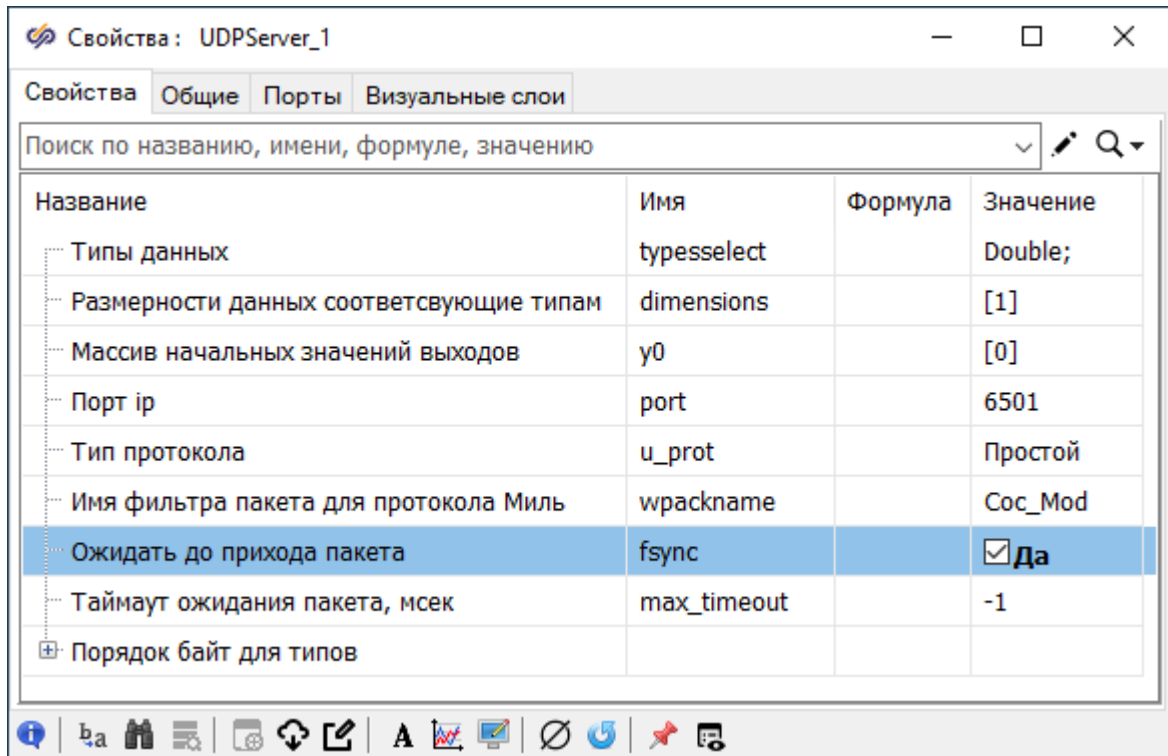


Рис. 19.14. Свойства UDP сервера в SinInTech

19.2.4. Обмен данными в процессе работы

Обратите внимание, что Универсальный механизм получает и отправляет сигналы по UDP на каждом **Шаге расчета переменных**, см. рис. 19.15. Для большинства случаев можно рекомендовать интервал шага расчёта переменных от 0,001 до 0,0001 с. Для ускорения общей производительности моделирования в Универсальном механизме также можно рекомендовать увеличение параметра **Фактор шага анимации**. Для большинства задач и пользователей разумным компромиссом между плавностью анимации и производительностью модели будет обновление анимационного окна с частотой 25-50 кадров в секунду. Таким образом при назначении **Фактора шага анимации** ориентируйтесь на то, чтобы произведение **Шага расчета переменных** на **Фактор шага анимации** было в интервале 0,02-0,04 с.

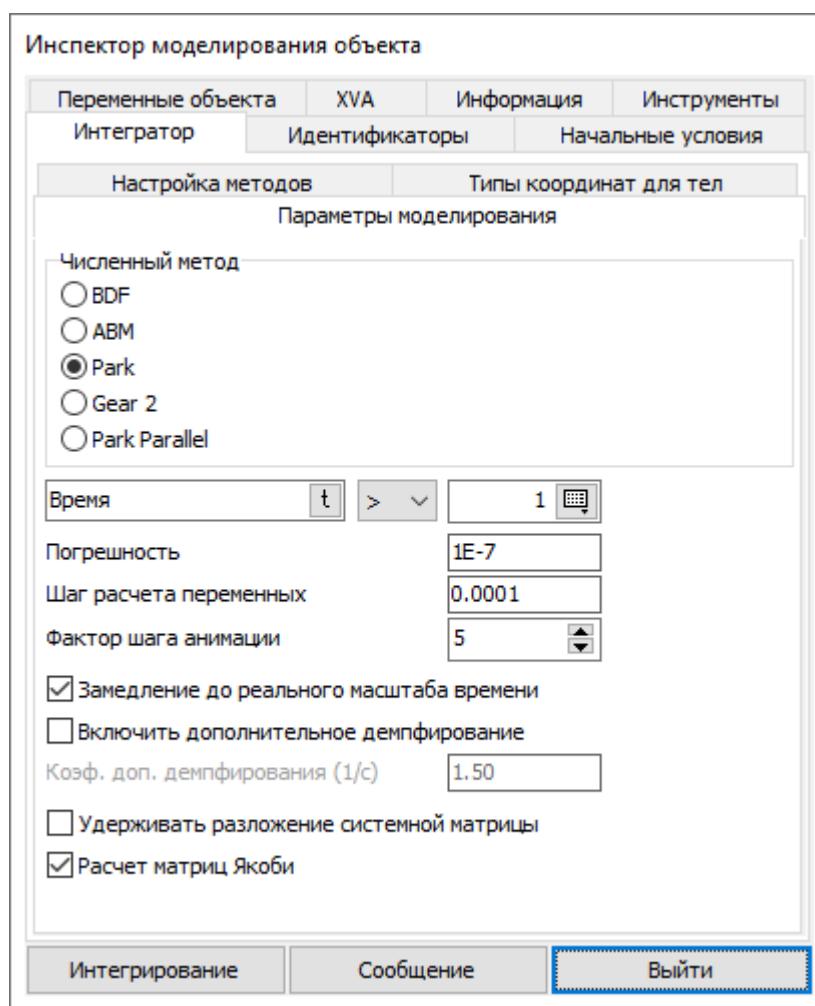


Рис. 19.15. Параметры моделирования на стороне УМ

19.2.5. Моделирование в реальном времени

Отсутствие механизмов синхронизации времени или задания модельного времени при обмене данными по протоколу UDP делает общепринятым рабочим стандартом работу всех связанных приложений в реальном времени. Включение моделирования в реальном времени в УМ выполняется с помощью флажка **Замедление до реального масштаба времени**, см. рис. 19.15.

Замечание Обратите внимание, что Универсальный механизм не является системой реального времени и не гарантирует моделирование в реальном времени. Производительность программы зависит в первую очередь от размерности модели, от наличия жёстких сил, от заданной погрешности, от установленного шага расчёта переменных, от выбранного численного метода и его настроек и многих других параметров. В пограничных случаях программа может считать то быстрее, то медленнее реального времени. Реальную производительность модели в УМ можно оценить с помощью переменной **Доля от реального времени** с вкладки **Переменные интегратора Мастера переменных**, см. рис. 19.16.

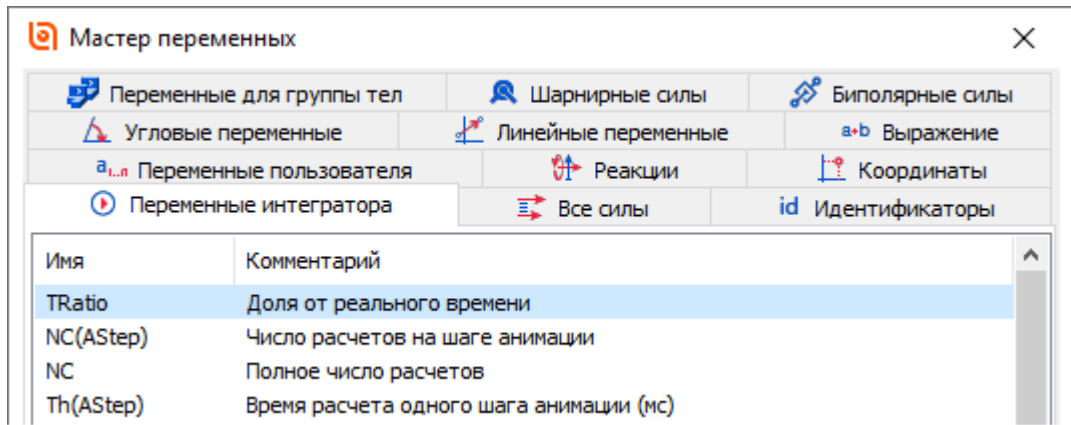


Рис. 19.16. Переменная Доля от реального времени в Мастере переменных

Работа в реальном времени в SimInTech включается кнопкой **Синхронизация с реальным временем** на инструментальной панели окна модели, см. рис. 19.17.

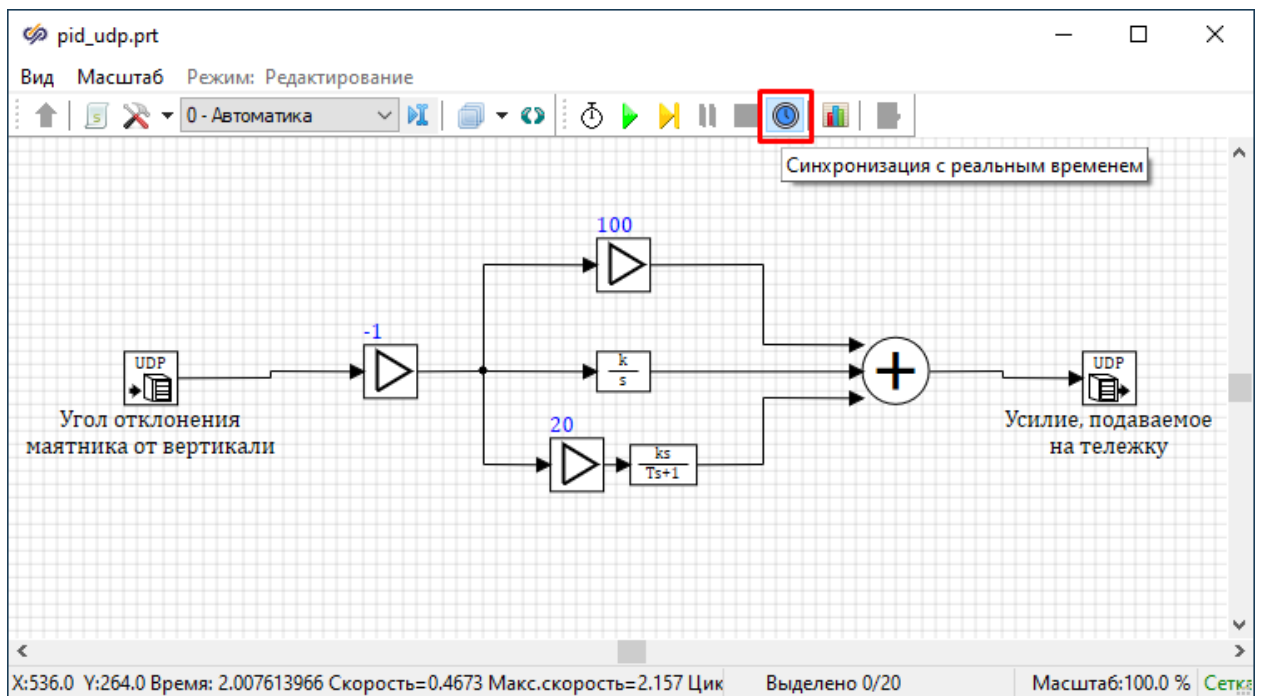


Рис. 19.17. Окно модели в SimInTech

19.2.6. Формат данных

Универсальный механизм принимает и отправляет данных по протоколу UDP только в формате *Double* (в соответствии со стандартом IEEE 754) – это число с плавающей запятой двойной точности. Размер: 8 байт (64 бита). Порядок байт: младший байт хранится по младшему адресу (Endianness: Little-endian). В настоящей реализации обмена данных на стороне УМ нет возможности настраивать/менять порядок байт в приходящих или отправляемых данных.

19.2.7. Чтение и запись нескольких сигналов

Универсальный механизм поддерживает чтение и запись нескольких сигналов в порт. Для этого добавьте необходимое вам количество сигналов в список **Сигналы** при описании конкретного UDP интерфейса, см. рис. 19.11 и рис. 19.12. Сигналы отправляются и принимаются в том же порядке, в котором они приведены в списке **Сигналы**.

На стороне связанных приложений это также нужно несколько сигналов обработать соответствующим образом. Так, например, в SimInTech для этого нужно использовать блоки **Мультиплексор** и **Демультимплексор** с вкладки **Векторные**, см. рис. 19.18.

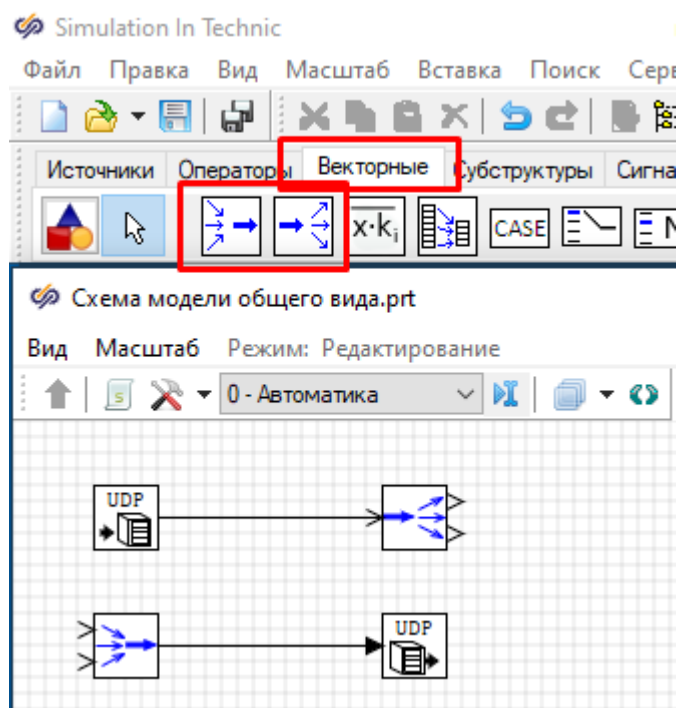


Рис. 19.18. Блоки Мультиплексор и Демультимплексор в SimInTech

19.2.8. Особенности UDP интерфейса в SimInTech

Обмен данными по протоколу UDP в SimInTech поддерживается блоками **Сервер UDP** и **Клиент UDP** с вкладки **Обмен данными**, см. рис. 19.19.

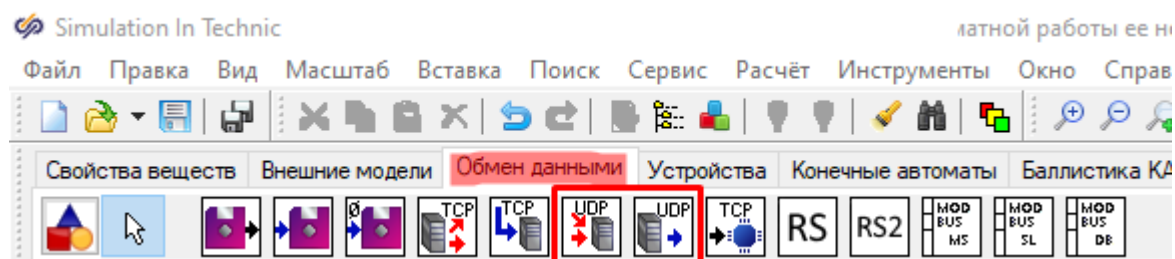


Рис. 19.19. Блоки работы с UDP протоколом в SimInTech

При работе с блоками **Сервер UDP** и **Клиент UDP** убедитесь, что свойство **Тип протокола** установлено в значение **Простой**, см. рис. 19.20. По умолчанию это свойство может иметь другое значение.

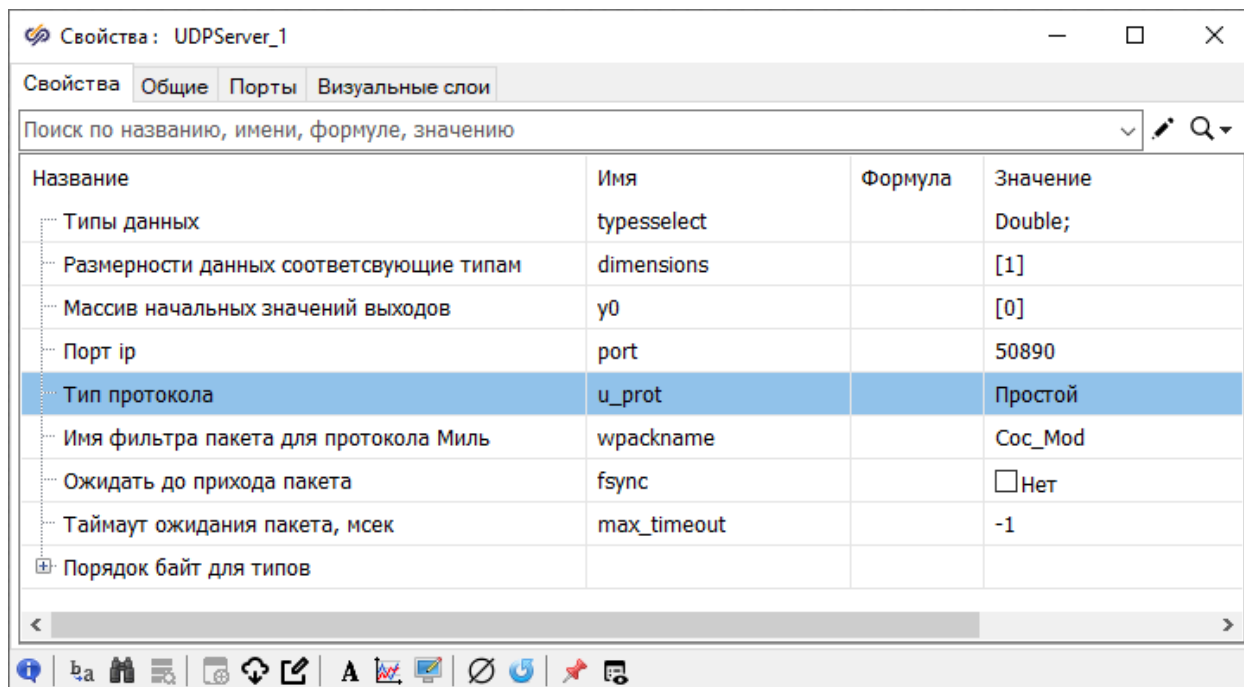


Рис. 19.20. Свойства клиента и сервера UDP в SimInTech

При работе УМ и SimInTech в связке посредством обмена данными по протоколу UDP в общем случае рекомендуется включать синхронизацию с реальным временем и на стороне УМ и на стороне SimInTech. В SimInTech это можно сделать либо нажав кнопку **Синхронизация с реальным временем** на панели инструментов модели, либо включив одноимённый флажок в окне параметров проекта, см. рис. 19.21.

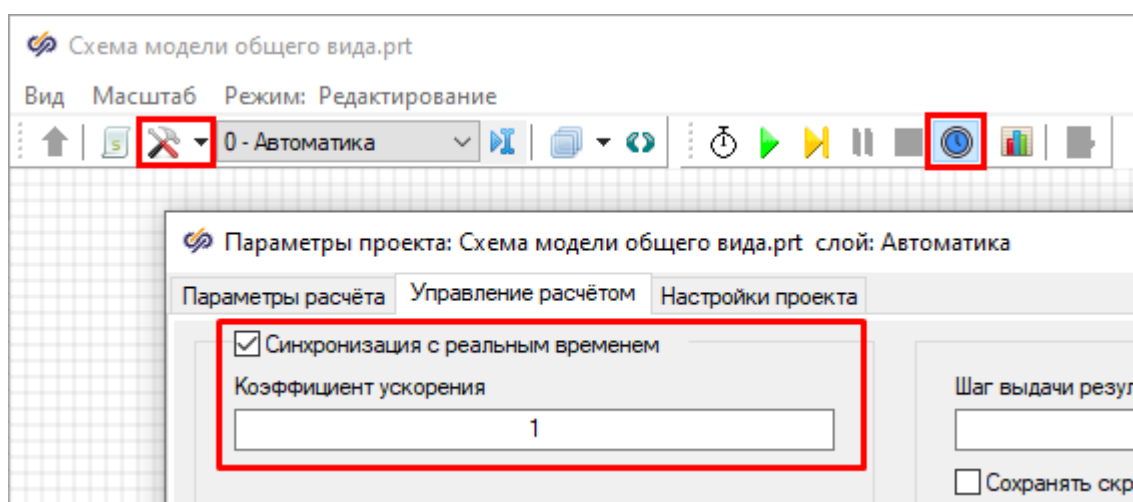


Рис. 19.21. Синхронизация с реальным временем в SimInTech

Также настоятельно рекомендуется установить одинаковый интервал обмена данными в УМ и SimInTech. На стороне SimInTech это регулируется параметрами **Минимальный**

шаг и **Максимальный шаг** в **Параметрах расчёта**. см. рис. 19.22. На стороне Универсального механизма это **Шаг расчета переменных** в окне **Инспектора моделирования объекта**, см. рис. 19.15.

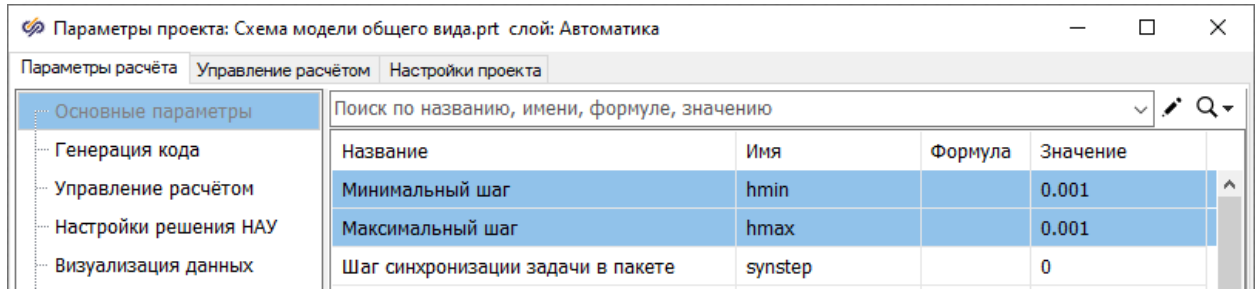


Рис. 19.22. Параметры расчёта проекта в SimInTech