

УНИВЕРСАЛЬНЫЙ МЕХАНИЗМ 6.0



Интерфейс с Matlab/Simulink

Начинаем работать

2010

Это руководство поможет вам изучить как особенности интеграции моделей, созданных в среде Matlab/Simulink, в модели программного комплекса «Универсальный механизм», так и наоборот, интеграцию моделей программного комплекса «Универсальный механизм» в модели, созданные в среде Matlab/Simulink.

Начинаем работать в программном комплексе «Универсальный механизм»: интерфейс с Matlab/Simulink

Это руководство поможет вам изучить как особенности интеграции моделей, созданных в среде Matlab/Simulink, в модели программного комплекса (ПК) «Универсальный механизм», так и наоборот, интеграцию моделей программного комплекса «Универсальный механизм» в модели, созданные в среде Matlab/Simulink.

Предполагается, что вы уже изучили раздел, посвященный введению в моделирование в ПК УМ, который находится в файле **gs_UM.pdf**¹, и умеете выполнять в УМ простые действия: создать новую модель, добавлять тела и шарниры, синтезировать уравнения движения (UM Input) и работать в программе моделирования движения (UM Simulation).

В настоящем руководстве мы шаг за шагом рассмотрим два примера. Первый пример продемонстрирует моделирование перевернутого маятника с системой управления, которая удерживает его в вертикальном перевернутом положении. Механическая часть системы – это обычная модель ПК «Универсальный механизм», модель системы управления создается в среде Matlab/Simulink. Затем в зависимости от выбранного подхода либо (1) модель Matlab/Simulink импортируется в ПК «Универсальный механизм» и связывается с моделью механической части с помощью инструмента Matlab Import, входящего в модуль UM Control, либо (2) модель механической части импортируется и включается в модель системы управления Matlab/Simulink с помощью инструмент CoSimulation, входящего в тот же модуль. Оба способа позволяют проанализировать динамику управляемого движения.

Далее мы рассмотрим пример, который иллюстрирует подключение двигателя постоянного тока к валу. Модель двигателя создана в Matlab/Simulink. Этот пример иллюстрирует разгон вала двигателем до номинальной угловой скорости с последующим приложением нагрузки. В конце этого примера мы построим графики силы тока, напряжения и электромагнитного момента на двигателе, а также график угловой скорости вала.

Предполагается, что Вы изучаете примеры данного руководства последовательно, один за другим. Информация, приведенная в одном примере, далее

¹ http://www.umlab.ru/download/60/rus/gs_um.pdf

может быть дана ссылкой на предыдущий пример, изложена в более сжатом виде или вовсе опущена.

Совместимость

UM Control / Matlab Import

Импорт моделей Matlab/Simulink в UM доступен только в поставке с инструментом UM Control / Matlab Import. Для проверки наличия этого инструмента запустите программу UM Simulation, выберите пункт меню **Помощь/О программе**. В появившемся окне в разделе **Конфигурация** вы увидите список доступных модулей и инструментов.

В настоящей версии программы поддерживается интерфейс с Matlab 6.X и Matlab 7.X (релизы R12, R13, R14, R2007).

Для компиляции моделей Matlab/Simulink необходимо наличие компилятора Microsoft Visual C/C++ version 5.0 или более позднего.

Модели, описанные в Matlab/Simulink для успешного импорта в UM должны содержать только непрерывные элементы, в настоящей версии интерфейса не допускается использование в импортируемой модели дискретных элементов.

UM Control / CoSimulation

Экспорт моделей UM в Matlab/Simulink доступен только в поставке с инструментом UM Control / CoSimulation. Для поддержки расчета динамики UM-модели из Matlab/Simulink необходимо, чтобы на компьютере было установлено динамическое ядро UM в виде COM-сервера.

Ответственность и авторские права

Данное руководство может изменяться время от времени. Авторы не несут никакой ответственности за любые ошибки и несоответствия, которые могут иметь место в данном документе.

Лаборатория вычислительной механики, Брянский государственный технический университет. Все права защищены ©, 2010.

Все товарные знаки принадлежат их законным владельцам.

Оглавление

НАЧИНАЕМ РАБОТАТЬ В ПРОГРАММНОМ КОМПЛЕКСЕ «УНИВЕРСАЛЬНЫЙ МЕХАНИЗМ»: ИНТЕРФЕЙС С MATLAB/SIMULINK	2
1. ВВЕДЕНИЕ	6
2. ЗНАКОМСТВО С МОДЕЛЯМИ	7
2.1. Перевернутый маятник.....	7
2.1.1. Описание модели	7
2.1.2. Подготовка модели на этапе создания.....	9
2.2. Двигатель постоянного тока.....	10
2.2.1. Описание модели	10
2.2.2. Особенности описания механической части	12
2.2.3. Особенности модели двигателя	13
2.3. Особенности и ограничения.....	14
3. МОДЕЛИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ MATLAB IMPORT	15
3.1. Порядок работы.....	15
3.2. Перевернутый маятник.....	16
3.2.1. Экспорт системы управления из Matlab/Simulink	17
3.2.2. Подключение библиотеки Matlab/Simulink в UM	25
3.2.3. Моделирование движения.....	32
3.3. Двигатель постоянного тока.....	35
3.3.1. Описание системы управления	36
3.3.2. Экспорт двигателя из Matlab/Simulink.....	37
3.3.3. Подключение библиотеки Matlab/Simulink в UM	38
3.3.4. Моделирование движения.....	43
4. МОДЕЛИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ COSIMULATION	46
4.1. Порядок работы.....	47
4.2. Перевернутый маятник.....	48
4.2.1. Подготовка системы управления в Matlab/Simulink	49
4.2.2. Экспорт модели механической системы из UM.....	50
4.2.3. Создание связи между UM-моделью и Matlab/Simulink	55
4.2.4. Моделирование движения.....	56
4.3. Двигатель постоянного тока.....	58
4.3.1. Подготовка системы управления в Matlab/Simulink	59
4.3.1. Экспорт модели механической системы из UM.....	60
4.3.2. Создание связи между UM-моделью и Matlab/Simulink	63
4.3.3. Моделирование движения.....	64

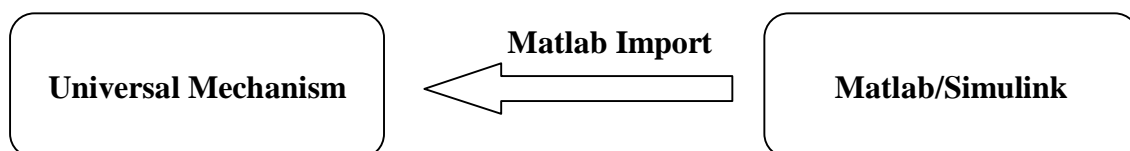
4.4. Антиблокировочная система автомобиля (ABS)	65
4.4.1. Подготовка системы управления в Matlab/Simulink	66
4.4.1. Экспорт модели механической системы из UM.....	67
4.4.2. Моделирование движения.....	68
4.4.2.1. Загрузка модели	68
4.4.2.2. Результаты моделирования движения с ABS	68
4.4.2.1. Результаты моделирования движения без ABS.....	71

1. Введение

В данном руководстве будут рассмотрены два подхода, позволяющие связать механическую систему, созданную в ПК «Универсальный механизм», с моделью системы управления, построенной в среде Matlab/Simulink. Остановимся подробнее на особенностях каждого из предлагаемых вариантов моделирования.

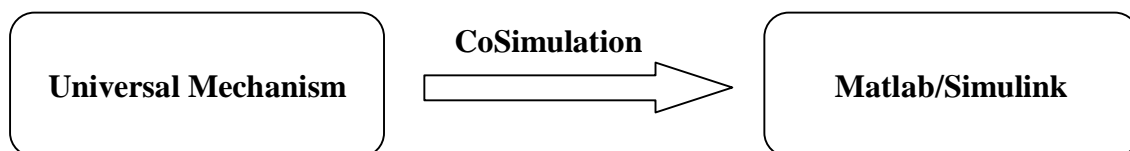
1. Инструмент Matlab Import

Этот подход предполагает, что модель системы управления, созданная в среде Matlab/Simulink, сначала компилируется в динамическую библиотеку (DLL), а затем уже эта библиотека средствами ПК «Универсальный механизм» подключается к модели механической части. Процедура подключения динамической библиотеки производится с помощью **Мастера связи с внешними библиотеками** в ПК «Универсальный механизм». И само моделирование динамики управляемого движения происходит в ПК «Универсальный механизм».



2. Инструмент CoSimulation

Другой вариант моделирования предполагает, что в модель системы управления Matlab/Simulink включается отдельный блок (в терминах Matlab это S-функция), который представляет собой модель механической системы ПК «Универсальный механизм». **Мастера экспорта в Matlab/Simulink** формирует m-файл и все необходимые вспомогательные файлы для работы UM-модели.



На рисунках выше показаны пути экспорта/импорта моделей и приложение, в котором будет выполняться моделирование сложной модели, включающей UM-модель и модель Matlab/Simulink. Поток данных в процессе численного моделирования уравнений движения в любом случае идет двусторонне.

2. Знакомство с моделями

В этой части руководства будут рассмотрены модели перевернутого маятника и двигателя постоянного тока. Мы подробно рассмотрим физическую составляющую предложенных моделей, дадим характеристику основным параметрам моделей, остановимся на особенностях подготовки этих моделей для дальнейшей связи с системами управления Matlab/Simulink.

2.1. Перевернутый маятник

2.1.1. Описание модели

Модель, представленная на рис. 2.1, состоит из тележки, массой M и перевернутого маятника массой m , момент инерции которого относительно оси вращения равен I . Параметры модели даны в табл. 1. На вход системы управления подается угол отклонения маятника от вертикали ψ , с выхода системы управления снимается величина управляющей силы F , которую нужно приложить к тележке для балансировки перевернутого маятника.

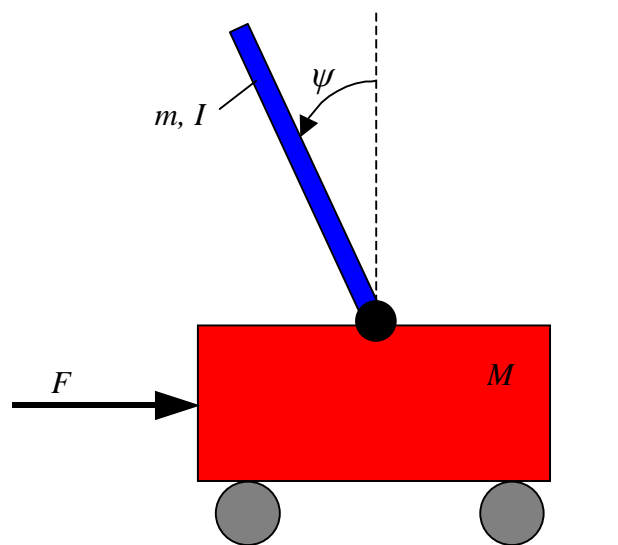


Рис. 2.1. Расчетная схема модели

Таблица 1

Параметры модели

Параметр на рисунке	Параметр модели	Комментарий	Значение
F	force	Сила, приложенная к тележке	
M	mass_cart	Масса тележки	0,5 кг
m	mass_pend	Масса маятника	0,2 кг
I	ix	Момент инерции маятника относительно оси вращения	0,006 кг·м ²
	l	Длина от оси вращения до центра масс маятника	0,3 м
	b	Коэффициент вязкого трения для тележки	0,1 Н/м/с
ψ		Угол отклонения маятника от вертикали	рад

2.1.2. Подготовка модели на этапе создания

Следует отметить, что предусмотреть возможность дальнейшей связи UM-модели механической системы с системой управления необходимо уже на этапе описания модели в программе UM Input.

Если система управления возвращает силы или моменты, которые нужно приложить к механической системе, то на этапе описания модели такие силы или моменты нужно в ней предусмотреть.

В нашей модели перевернутого маятника в модель механической части добавлена *общая сила Control force*, действующая вдоль оси Y. Величина силы задается параметром модели **force**. В дальнейшем мы свяжем выходную величину модели Matlab/Simulink с этим параметром, и, таким образом, приложим к системе управляющее усилие.

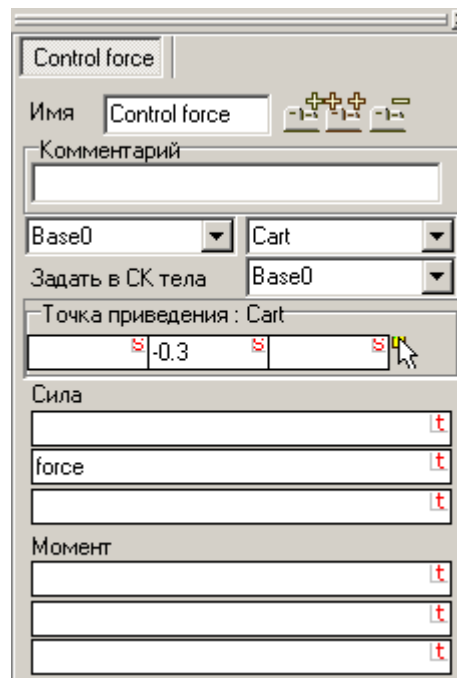


Рис. 2.2. Описание общей силы

2.2. Двигатель постоянного тока

2.2.1. Описание модели

В качестве другого примера рассмотрим модель вала, на который действует активный момент M_a и момент сил сопротивления M_c . Активный момент действует со стороны двигателя постоянного тока. Общий вид модели приведен на рис. 2.3.

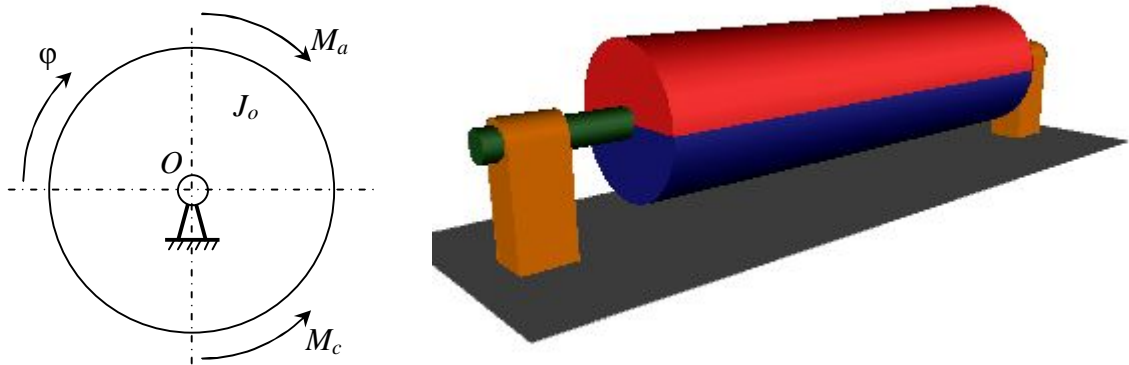


Рис. 2.3. Расчетная схема и общий вид модели

Электромеханическая модель, представленная на рис. 2.3, состоит из вала, момент инерции которого относительно оси вращения равен J_o . Вал соединен вращательным шарниром с базой (инерциальной системой координат) и, соответственно, имеет одну вращательную степень свободы.

Модель двигателя постоянного тока описана в Matlab/Simulink. На вход модели двигателя подается угловая скорость вала – предполагается, что ротор двигателя и вал жестко связаны между собой. С выхода модели двигателя снимается три величины: активный электромагнитный момент M_a , ток и напряжение цепи якоря.

Электромагнитный момент M_a описан как активный момент. Момент сил сопротивления описан как шарнирный момент. Момент сил сопротивления вводится как функция времени для иллюстрации реакции двигателя на ступенчатую нагрузку, см. рис. 2.4.

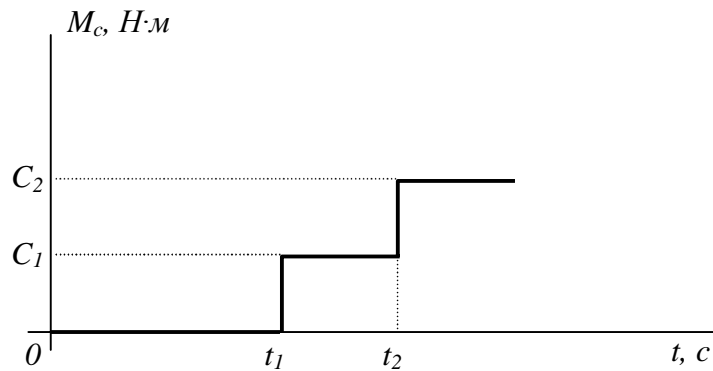


Рис. 2.4. График момента сил сопротивления (ступенчатая нагрузка)

Таблица 2

Параметры модели			
Параметр на рисунке	Параметр модели	Комментарий	Значение
M_a	driving_torque	Активный электромагнитный момент	
M_c		Момент сил сопротивления	F(t)
J_o	Iy	Момент инерции вала относительно оси вращения	0,1 кг·м ²
t_1	t1	Момент времени t_1 , см. рис. 2.4	5 с
t_2	t2	Момент времени t_2 , см. рис. 2.4	8 с
C_1	C1	Величина момента сил сопротивления, начиная с $t=t_1$, см. рис. 2.4	5 Н·м
C_2	C2	Величина момента сил сопротивления, начиная с $t=t_2$, см. рис. 2.4	10 Н·м

2.2.2. Особенности описания механической части

Как уже было сказано выше, предусмотреть возможность дальнейшей связи UM-модели механической системы с системой управления необходимо еще на этапе описания модели в программе **UM Input**. Если система управления по некоторым входным параметрам возвращает силы или моменты, которые нужно приложить к механической системе, то на этапе описания модели такие силы или моменты нужно в ней предусмотреть.

В рассматриваемой модели добавлена *общая сила Driving torque* (движущий момент), см. рис. 2.5, слева. Эта сила имеет один ненулевой компонент – момент относительно оси Y, заданный идентификатором **Ma**, который в дальнейшем мы и свяжем с электромагнитным моментом, вычисляемым в модели Matlab/Simulink.

Момент сил сопротивления в данной модели вводится как шарнирный момент, заданный функцией времени. Значение момента задается выражением

$$M_r = c1*heavi(t-t1)+(c2-c1)*heavi(t-t2), \text{ где}$$

t1, t2, c1 и c2 – параметры модели, см. рис. 2.4, причем

$$heavi(t) = \begin{cases} 1, t > 0 \\ 0, t \leq 0 \end{cases}.$$

Для того чтобы впоследствии при моделировании вывести величины тока и напряжения на двигателе, в модель формально введены два параметра: **I** (ток якоря) и **U** (напряжение цепи якоря).

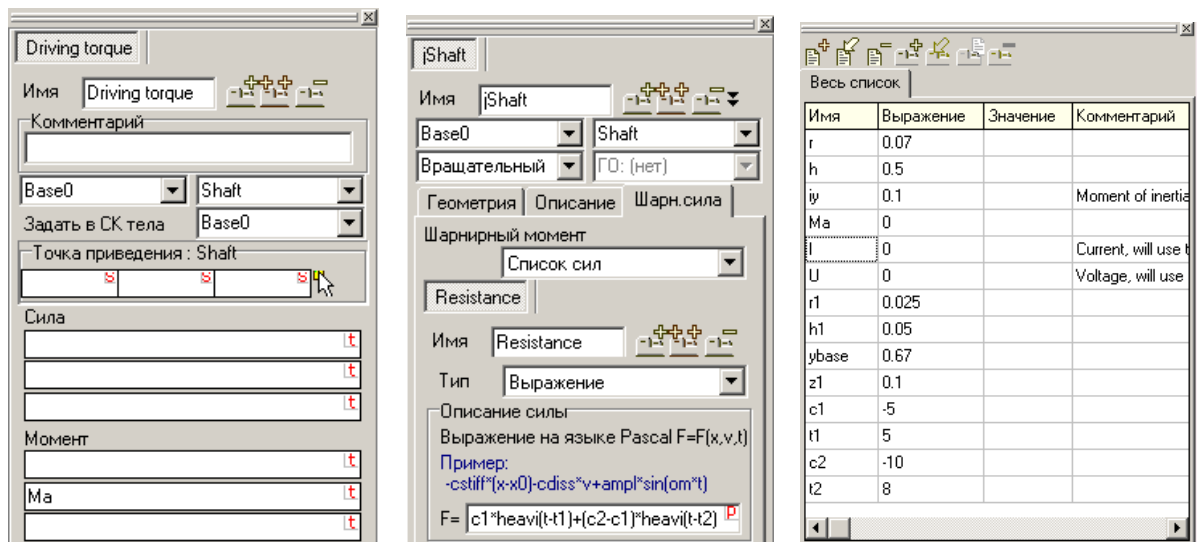


Рис.2.5. Описание активного момента, момента сил сопротивления и параметров модели

2.2.3. Особенности модели двигателя

Для данного примера выберем двигатель типа 2ПН112МУХЛ4 – двигатель постоянного тока с независимым возбуждением и постоянным магнитным потоком. Справочные данные для данного типа двигателя приведены в табл. 3.

Таблица 3

Параметры двигателя	
Параметр	Значение
Мощность, кВт	3,6
Напряжение, В	220
Номинальная частота вращения, об/мин	3000
Максимальная частота вращения, об/мин	4000
КПД, %	79
Сопротивление обмотки при 15°C, Ом	
	якоря
	0,42
	добавочных полюсов
	0,356
	возбуждения
	129
Индуктивность цепи якоря, мГн	4,8

Двигатель включается при помощи двухступенчатого пускового реостата. Первая ступень сопротивлением 2,5 Ом выключается через 1,2 с после включения двигателя. Вторая ступень, сопротивлением 2,035 Ом выключается через 2,5 с.

2.3. Особенности и ограничения

Существует ограничение на связь идентификаторов UM-модели с выходными величинами Matlab/Simulink модели, которое заключается в следующем. С выходными величинами Matlab/Simulink модели можно связывать только те идентификаторы, которые параметризуют силовые элементы: координаты точек прикрепления/действия сил, параметры жесткости, диссипации и прочее.

Согласно общим принципам УМ запрещается назначать выходные величины Matlab/Simulink модели идентификаторам UM-модели, параметризующим инерционные параметры тел (массу и моменты инерции) и кинематику шарниров (положение шарнирных точек и прочее). Это приведет к получению неправильных результатов. Значения таких параметров можно менять между отдельными численными экспериментами, но не в процессе проведения одного численного эксперимента.

3. Моделирование с использованием Matlab Import

В этой части руководства мы подробно рассмотрим особенности подключения систем управления Matlab/Simulink к механическим моделям «Универсального механизма» посредством экспорта систем управления из Matlab/Simulink как динамических библиотек (DLL). Вначале остановимся на основных этапах, которые необходимо выполнить, чтобы правильно подключить систему управления к модели механической системы и выполнить моделирование динамики управляемого движения.

3.1. Порядок работы

В общем случае моделирования динамики механических систем с подключением библиотек Matlab/Simulink предполагает выполнение следующих этапов.

- Описание модели системы управления в среде Matlab/Simulink.
- Экспорт созданной модели из Matlab/Simulink в виде DLL-библиотеки.
- Создание модели механической системы в программе UM Input.
- Загрузка подготовленной модели механической части в программу UM Simulation. Загрузка и подключение DLL-библиотеки Matlab/Simulink с помощью **Мастера связи с внешними библиотеками**.
- Моделирование динамики управляемого движения.

Модель системы управления из Matlab/Simulink, подключаемая в UM, рассматривается как черный ящик, который по некоторому закону преобразует входные величины в выходные. При интеграции системы управления в модель механической системы на входы системы управления назначаются *переменные*, которые создаются при помощи **Мастера переменных**. Выходные величины связываются с параметрами UM-модели.

Для реализации управляющих усилий со стороны системы управления в модель механической системы вводятся силы/моменты, значения или характеристики (например, коэффициент жесткости, коэффициент диссипации) которых задаются параметрами модели. Затем с помощью Мастера связи с внешними библиотеками эти параметры связываются с выходами системы управления.

3.2. Перевернутый маятник

Модель перевернутого маятника без подключенной системы управления находится в каталоге `{um_root}\samples\tutorial\inv_pend`. Перед началом урока, проверьте ее наличие в указанном каталоге. Если такой модели нет, то скачайте ее из интернета по адресу:

http://www.umlabor.ru/download/60/inv_pend.zip. В данном уроке мы не будем подробно разбирать этапы создания модели механической части, ограничимся только описанием связи механической части с системой управления Matlab/Simulink.

Готовая модель перевернутого маятника с подключенной системой управления находится в каталоге `{um_root}\samples\simulink\inv_pend_ctrl`¹.

¹ http://www.umlabor.ru/download/60/inv_pend_ctrl.zip

3.2.1. Экспорт системы управления из Matlab/Simulink

Экспорт системы управления из Matlab/Simulink заключается в компиляции подготовленной модели в динамическую библиотеку функций (DLL), которая в последствии подключается к программному комплексу «Универсальный механизм».

Замечание. Здесь предполагается, что вы знакомы с основами моделирования в среде Matlab/Simulink. В любом случае сейчас вы можете пропустить данный параграф и вернуться к нему позднее. Результатом экспорта системы управления из Matlab/Simulink является библиотека `\inv_pend\pendpid_cnr.dll`. Вы можете пропустить этап создания dll-библиотеки для системы управления и воспользоваться уже подготовленной.

Процесс экспорта модели системы управления состоит из следующих этапов.

1. Подготовка модели системы управления к экспорту.
2. Копирование необходимых для компиляции модели файлов в каталог с моделью системы управления.
3. Настройка процесса компиляции и собственно компиляция модели.

Подготовка модели системы управления к экспорту

Для того чтобы модель системы управления было возможно экспортировать в dll-библиотеку, необходимо исключить из нее компоненты производящие какие-либо операции ввода/вывода, также компоненты, для которых в системе Matlab/Simulink отсутствуют исходные коды или компиляция которых невозможна.

Помимо этого в модель системы управления следует включить компоненты «In» и «Out», которые будут обеспечивать связь модели системы управления с моделью ПК «Универсальный механизм». В нашей модели перевернутого маятника входной переменной для системы управления является угол отклонения маятника от вертикали. Соответственно нужно предусмотреть один вход в модель системы управления. В свою очередь для организации управления перевернутым маятником в нашей модели системы управления предусмотрен один выход – усилие, подаваемое на тележку.

Подготовленная модель системы управления перевернутым маятником в Matlab/Simulink представлена на рис. 3.1.

Замечание. Вы можете самостоятельно собрать модель, представленную на рис. 3.1 или же загрузить готовую модель Matlab/Simulink из файла `samples\tutorial\inv_pend\pendpid_cntr.mdl`.

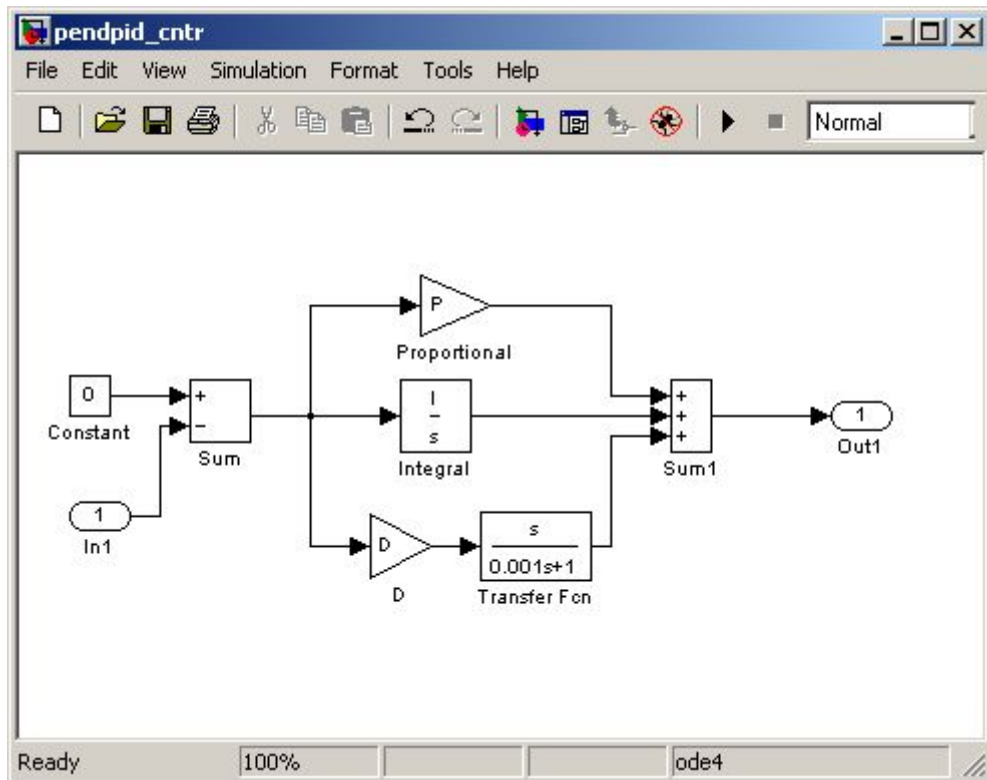


Рис. 3.1. Модель системы стабилизации перевернутого маятника

Копирование файлов и настройка путей

1. Для компиляции модели системы управления необходимо скопировать некоторые вспомогательные файлы.

Matlab 6.0, 6.1 / Release 12

Из каталога `{um_root}\simulink\R12` в каталог, содержащий файл модели, скопируйте следующие файлы: `um.tlc`, `deriv.tlc`, `grt.tlc` и `grt_vc.tmf`.

Matlab 6.5 / Release 13

Из каталога `{um_root}\simulink\R13` в каталог, содержащий файл модели, скопируйте следующие файлы: `um.tlc`, `deriv.tlc`, `grt.tlc` и `grt_vc.tmf`.

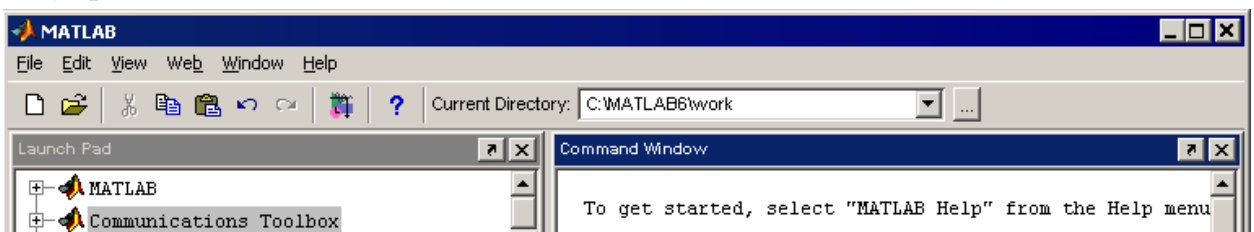
Matlab 7.X / Release 14

Из каталога `{um_root}\simulink\R14` в каталог, содержащий файл модели, скопируйте следующие файлы: `rsim.tlc`, `rsim_vc.tmf` и `um.tls`.

Matlab 7.X / Release 2007, 2008

Из каталога `{um_root}\simulink\R2007_2008` в каталог, содержащий файл модели, скопируйте следующие файлы: `rsim.tlc`, `rsim_vc.tmf` и `um.tls`.

2. Помимо этого необходимо, чтобы поле **Current Directory** имело значение каталога, в котором находится файл экспортируемой модели системы управления.



Настройка процесса компиляции и компиляция модели

1. Перейдите в командное окно системы Matlab. С помощью команды **'mex -setup'** выберите используемый Вами компилятор (Microsoft Visual C/C++ version 6.0/5.0).
2. Перейдите в окно модели. В меню **Tools** выберите пункт **Real-Time Workshop**, после чего выберите пункт **Options**.

Дальнейшие действия по компиляции модели зависят от версии Matlab, которую вы используете.

Matlab 6.X

1. В появившемся диалоговом окне в поле **System target file** задайте значение **grt.tlc**.
2. В поле **Template makefile** введите **grt_vc.tmf**.
3. Убедитесь, что поле **Make command** содержит значение **make_rtw**.
4. Перейдите на закладку **Solver** и в поле **Solver options / Type** выберите значение **Fixed Step**.
5. В поле **Solver options** выберите наиболее подходящий, по вашему мнению, численный метод решения, например **ode4 (Runge-Kutta)**.
6. Для запуска процесса компиляции и сборки dll-библиотеки перейдите на закладку **Real-Time Workshop** и нажмите кнопку **Build**.

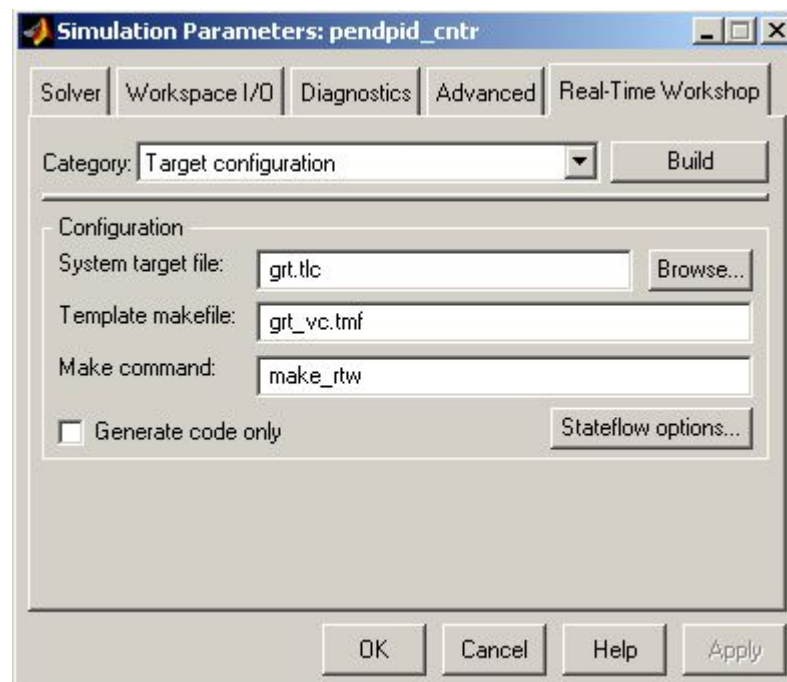


Рис.3.2. Настройка процесса компиляции в Matlab 6.X

Matlab 7.X

1. В появившемся диалоговом окне в поле **RTW system target file** задайте значение **rsim.tlc**.
2. Убедитесь, что в поле **Make command** задано **make_rtw**.
3. В поле **Template makefile** выберите **rsim_vc.tmf**.
4. В дереве объектов конфигурирования (диалоговое окно, слева) выберите **Solver** и в поле **Solver options / Type** выберите значение **Fixed Step**.
5. В поле **Solver options / Solver** выберите наиболее подходящий, по вашему мнению, численный метод решения, например **ode4 (Runge-Kutta)**.
6. Для запуска процесса компиляции и сборки dll-библиотеки вернитесь в поле **Real-Time Workshop** и нажмите кнопку **Build**.

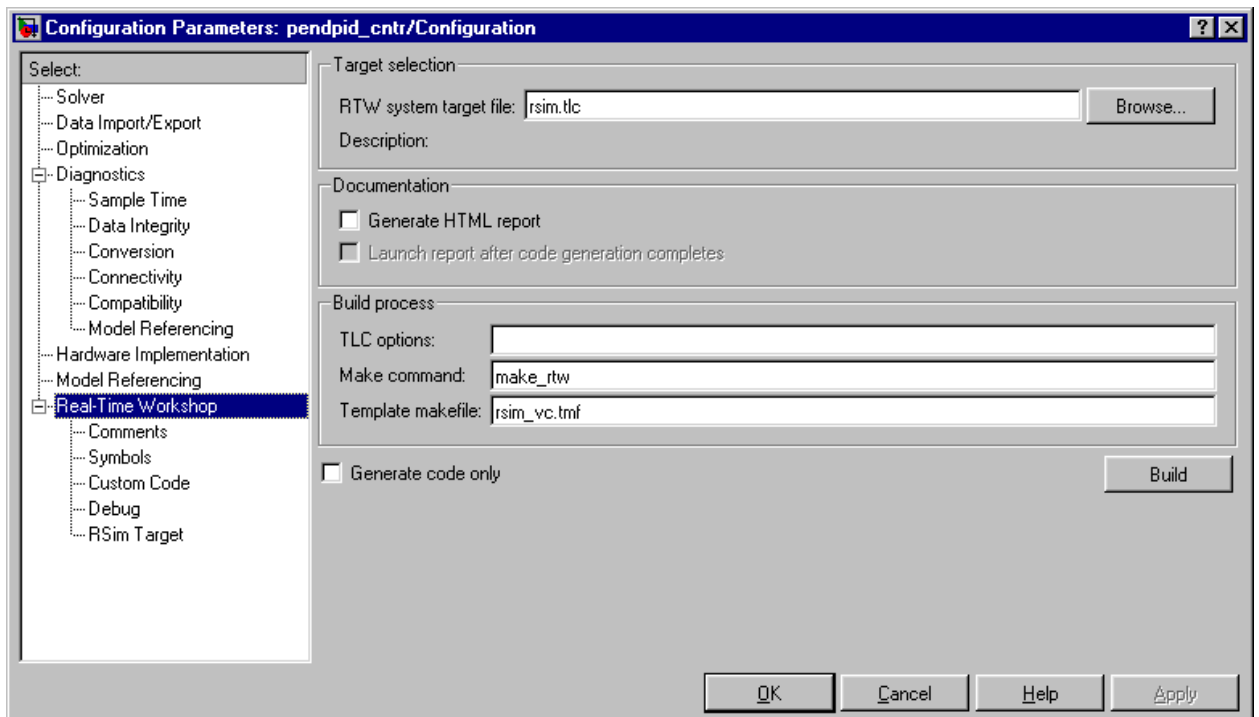


Рис.3.3. Настройка процесса компиляции в Matlab 7.X

Matlab R2007, R2008

1. В появившемся диалоговом окне в дереве слева выберите элемент **Real-Time Workshop**.
2. поле **RTW system target file** задайте значение **rsim.tlc**.
3. В поле **Language** установите значение **C**.
4. Убедитесь, что в поле **Make command** задано **make_rtw**.
5. В поле **Template makefile** выберите **rsim_vc.tmf**.
6. В дереве объектов конфигурирования (диалоговое окно, слева) выберите **Solver** и в поле **Solver options / Type** выберите значение **Fixed Step**.
7. В поле **Solver options / Solver** выберите наиболее подходящий, по вашему мнению, численный метод решения, например **ode4 (Runge-Kutta)**.
8. Для запуска процесса компиляции и сборки dll-библиотеки вернитесь в поле **Real-Time Workshop** и нажмите кнопку **Build**.

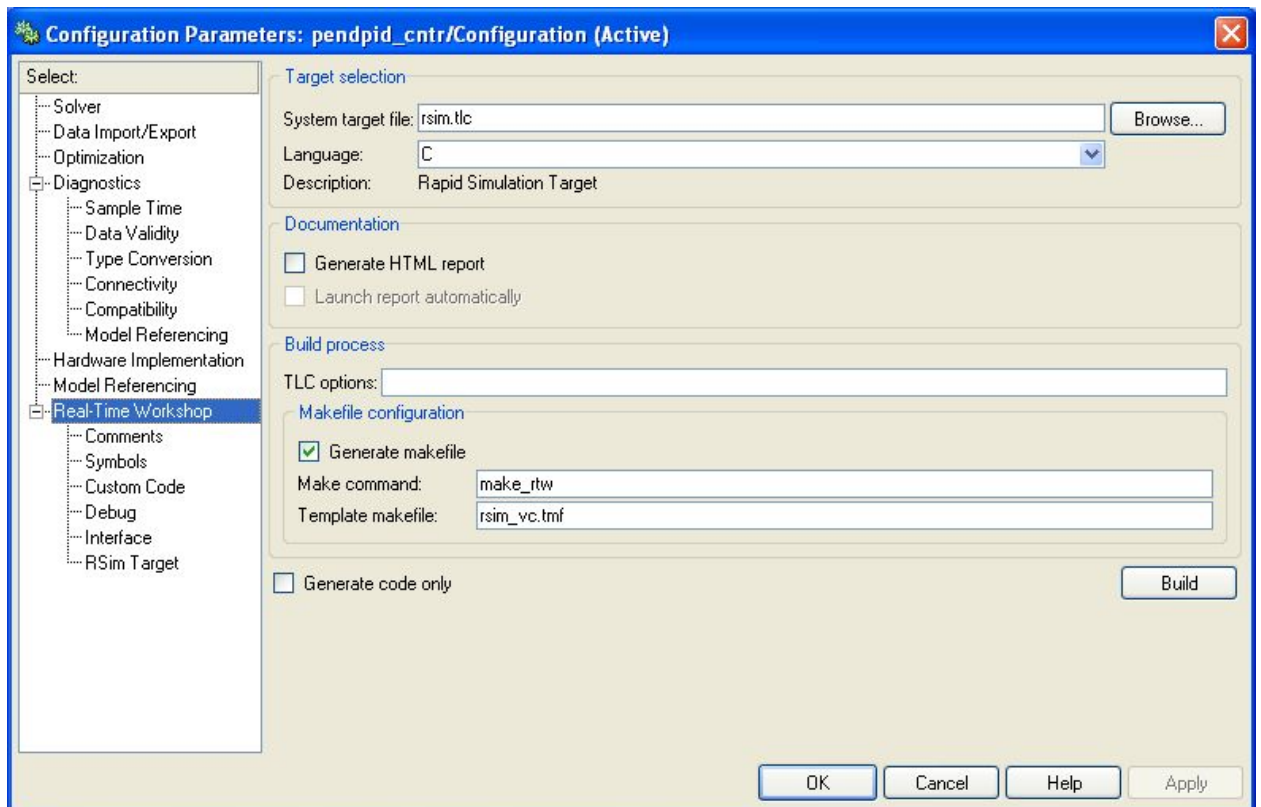


Рис.3.4. Настройка процесса компиляции в Matlab R2007, R2008

Процесс компиляции сопровождается выводом сообщений о ходе процесса в командное окно системы Matlab/Simulink. Успешная компиляция

библиотеки в частности для нашей модели системы управления завершается сообщением «Creating library ..\pendpid_cntr.lib and object ..\pendpid_cntr.exp».

Помимо этого по окончанию компиляции в релизах Matlab R12, R13, R14 на экране может появляться диалоговое окно с сообщением

«Error executing build command: Error using ==> make_rtw

Error using ==> rtw_c

Error(s) encountered while building model “pendpid_cntr”»,

которое можно игнорировать.

Замечание. Для работы динамической библиотеки созданной в релизах Matlab R2007/R2008 в программном комплексе UM необходимо, чтобы на компьютере была установлена соответствующая версия Matlab.

В результате выполнения компиляции в рабочем каталоге появятся файлы библиотеки. Для нашего примера это - **pendpid_cntr.dll, pendpid_cntr.lib, pendpid_cntr.exp**. А также подкаталог, содержащий исходные коды скомпилированной библиотеки. Для нашей модели он будет называться **pendpid_cntr_grt_rtw** для Matlab 6.X и **pendpid_cntr_rsim_rtw** для Matlab 7.X.

3.2.2. Подключение библиотеки Matlab/Simulink в UM

Загрузка модели механической части

1. Запустите программу моделирования **UM Simulation**.
2. Загрузите модель `{um_root}\samples\tutorial\inv_pend`.
3. Откройте новое анимационное окно.

Загрузка модели Matlab/Simulink

4. Выберите пункт меню **Инструменты / Интерфейс с внешними библиотеками**.

Появится окно **Мастера связи с внешними библиотеками**.

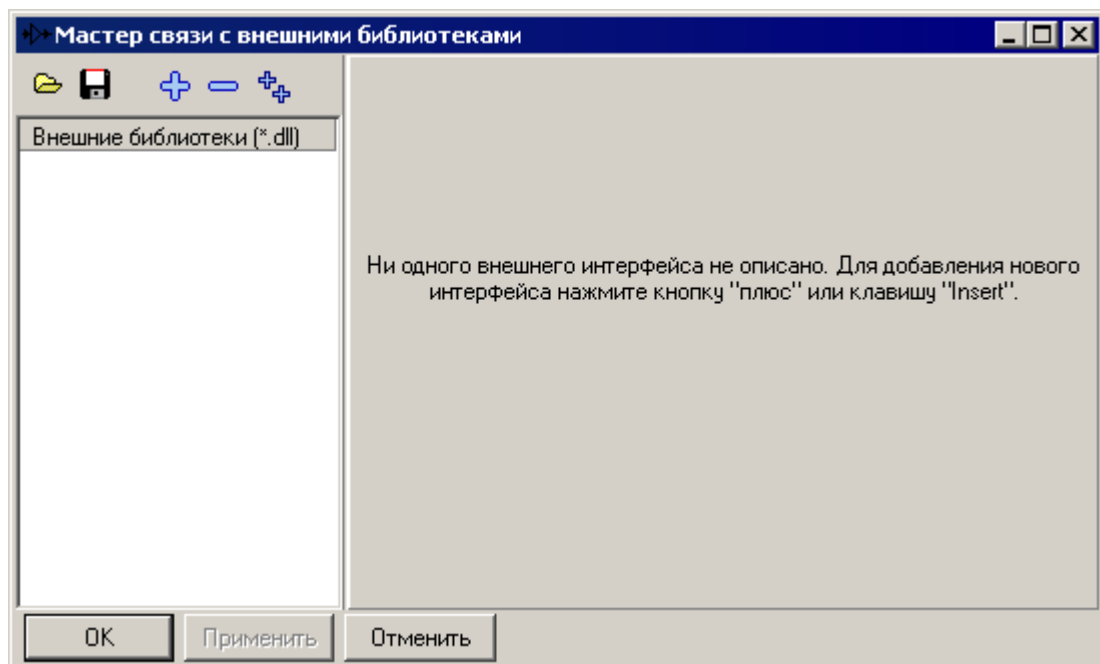

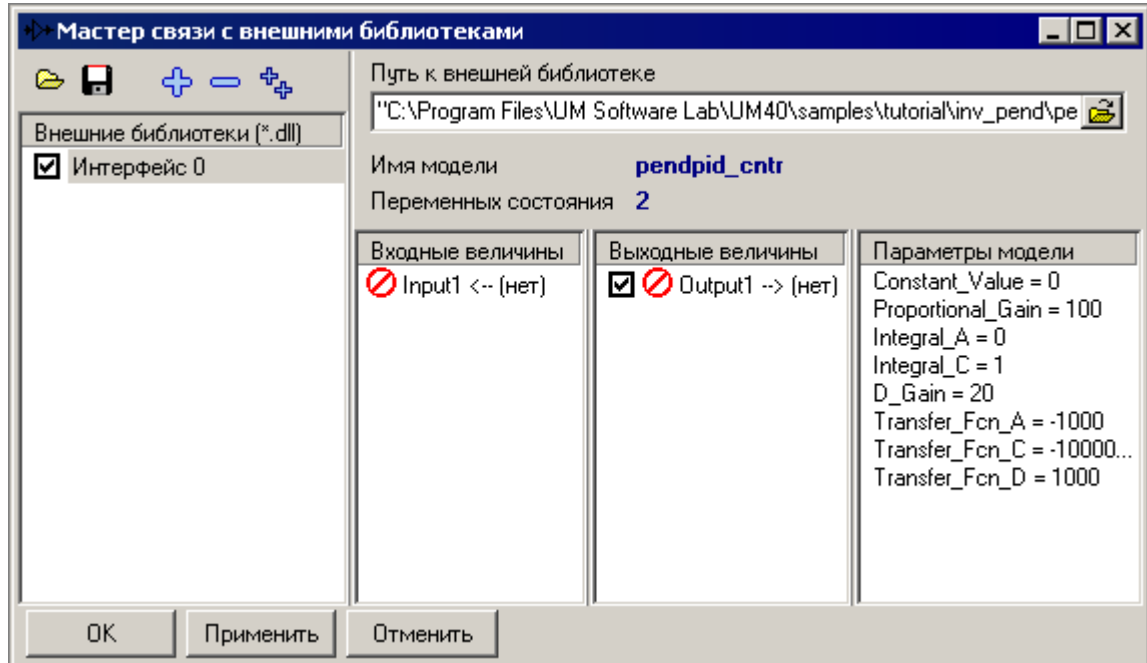


Рис.3.5. Мастер связи с внешними библиотеками

5. Нажмите кнопку  для добавления новой библиотеки Matlab/Simulink.
6. В поле **Путь к внешней библиотеке** выберите файл **pendpid_cntr.dll**. Если вы создали выходную dll самостоятельно, то выберите ее, иначе выберите готовую dll системы управления из каталога `..\inv_pend`.

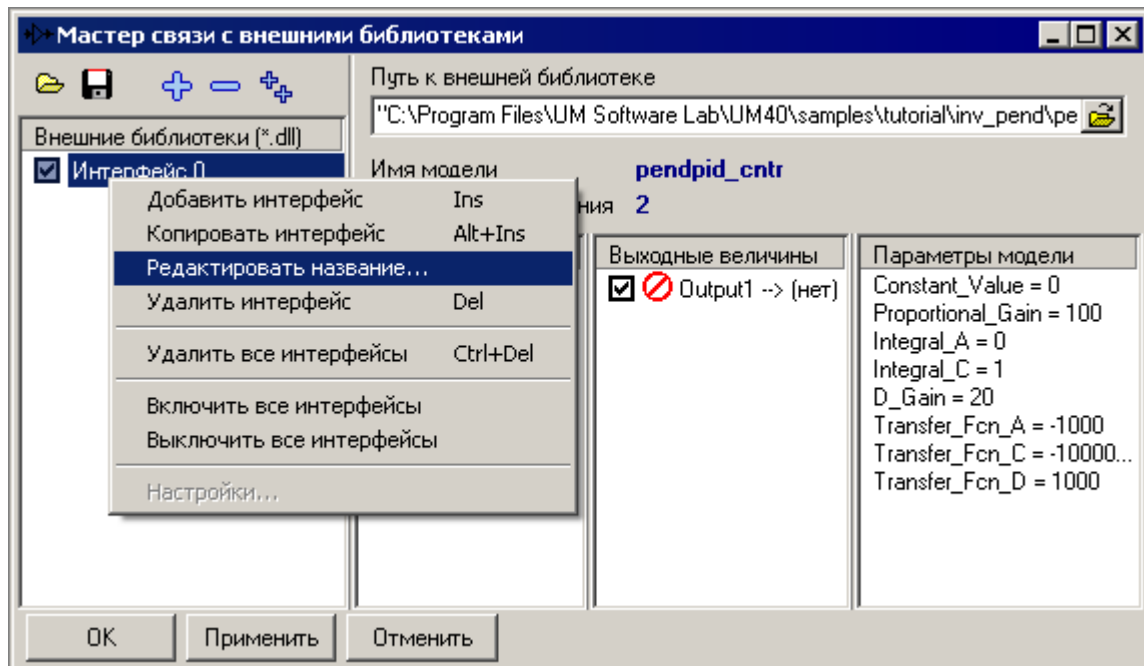
Мастер связи загрузит указанную модель и установит число входных и выходных величин модели системы управления, а также число параметров этой модели и переменных состояния.

В данном случае в окне появится одна входная и одна выходная величина, см. рис. ниже. На вход системы управления мы подадим угол отклонения маятника от вертикали, а выходную величину подадим в качестве усилия на тележку.



Переименование интерфейса

7. Выберите в списке **Внешние библиотеки (*.dll)** текущий интерфейс **Интерфейс 0**, правой кнопкой мыши вызовите контекстное меню и в нем выберите пункт **Редактировать название...**
8. В поле редактирования названия введите **Control force** и нажмите клавишу **Enter**.



Назначение входных величин модели Matlab/Simulink

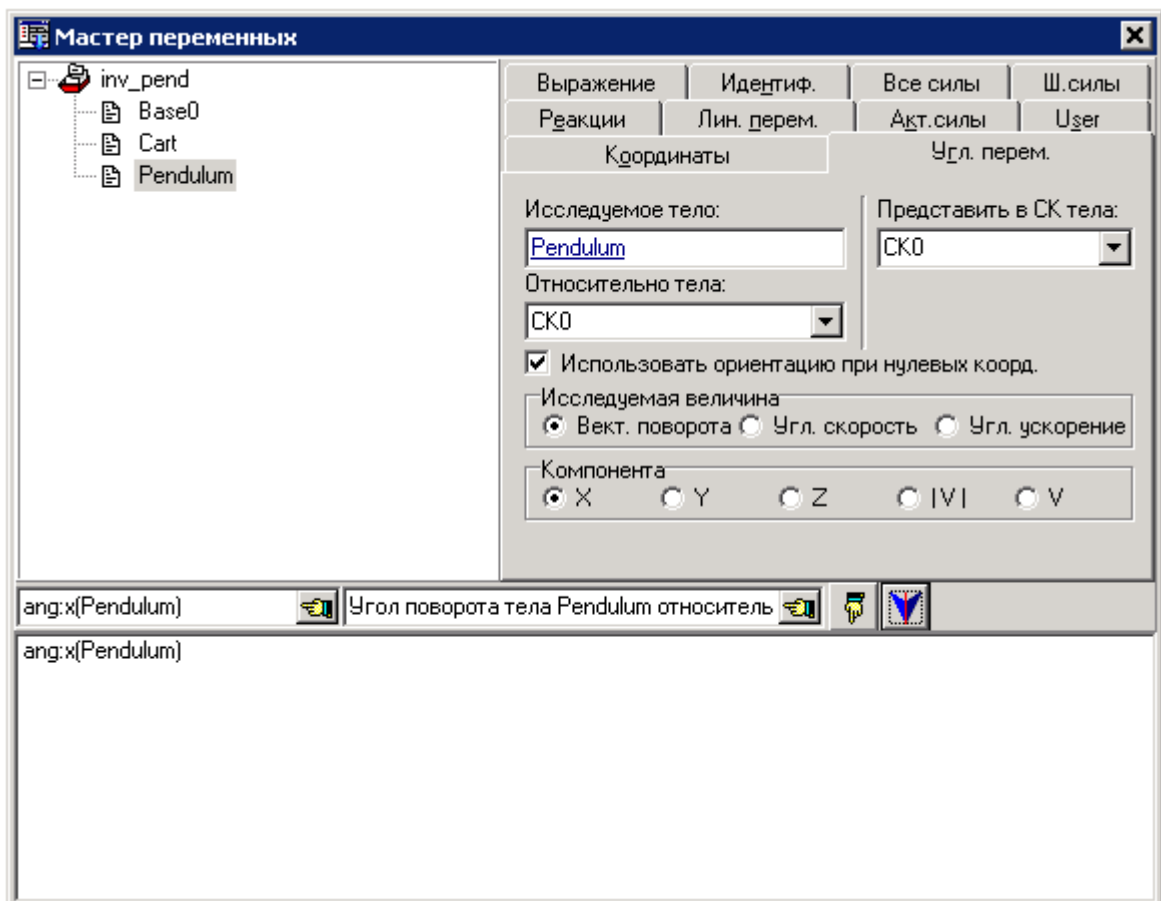
С помощью мастера переменных сформируем переменную «угол отклонения маятника от вертикали» и назначим ее в качестве входной величины для системы управления.

9. Запустите **Мастер переменных**.

10. Перейдите на закладку **Угл. перем.** (угловые переменные).

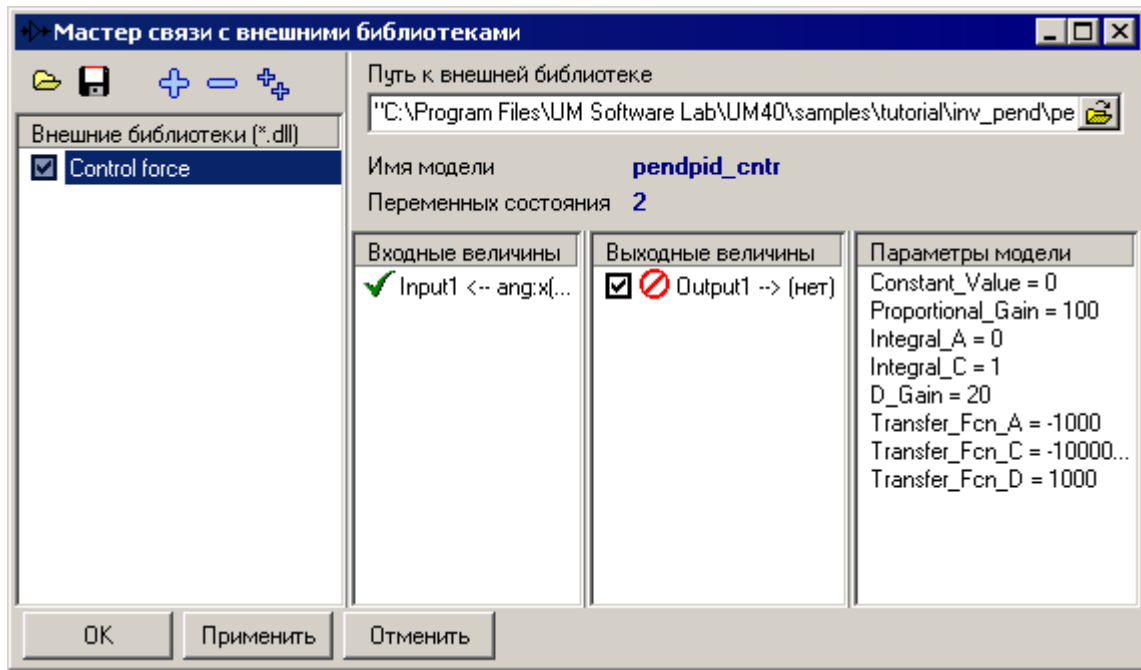
11. В списке тел слева выберите **Pendulum**, в поле **Исследуемая величина** выберите **Вект.(ор) поворота**, а в поле **Компонента** выберите **X**.

12. Создайте переменную кнопкой . Переменная **ang:x(Pendulum)** появится в контейнере переменных.



13. Перетащите мышкой созданную переменную **ang:x(Pendulum)** в окно **Мастера связи с внешними библиотеками** на вход **Input1**.

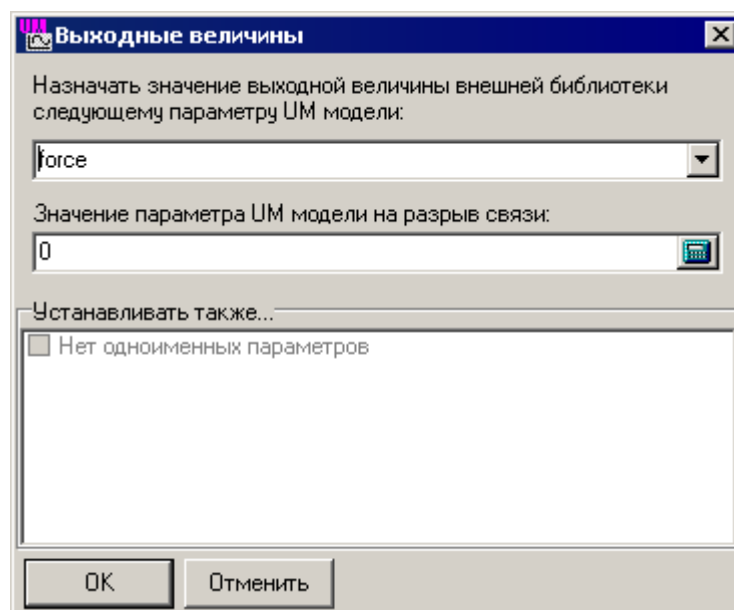
Вход модели **Input1** станет помечен зеленой галочкой, что означает, что на данный вход модели Matlab/Simulink назначена переменная.



Назначение выходных величин модели Matlab/Simulink

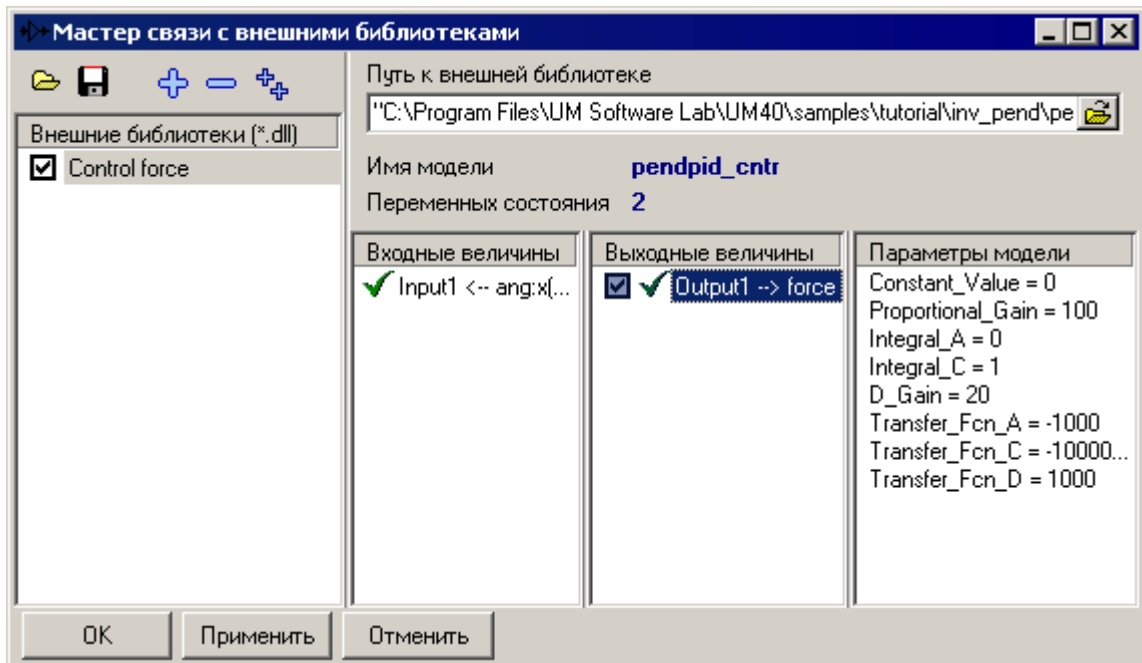
Выходные величины модели Matlab/Simulink связываются с параметрами UM-модели.

1. Дважды щелкните на элементе **Выходные величины / Output1**. Появится диалоговое окно назначения параметров UM-модели на выход модели Matlab/Simulink.
2. В поле **Параметр UM модели** выберите **force**.
3. Закройте окно кнопкой **ОК**.



Модель Matlab/Simulink полностью связана с моделью механической системы.

4. Нажмите **Применить** для актуализации настроек без закрытия окна **Мастера связи с внешними библиотеками** или **ОК** для сохранения данных с закрытием окна.

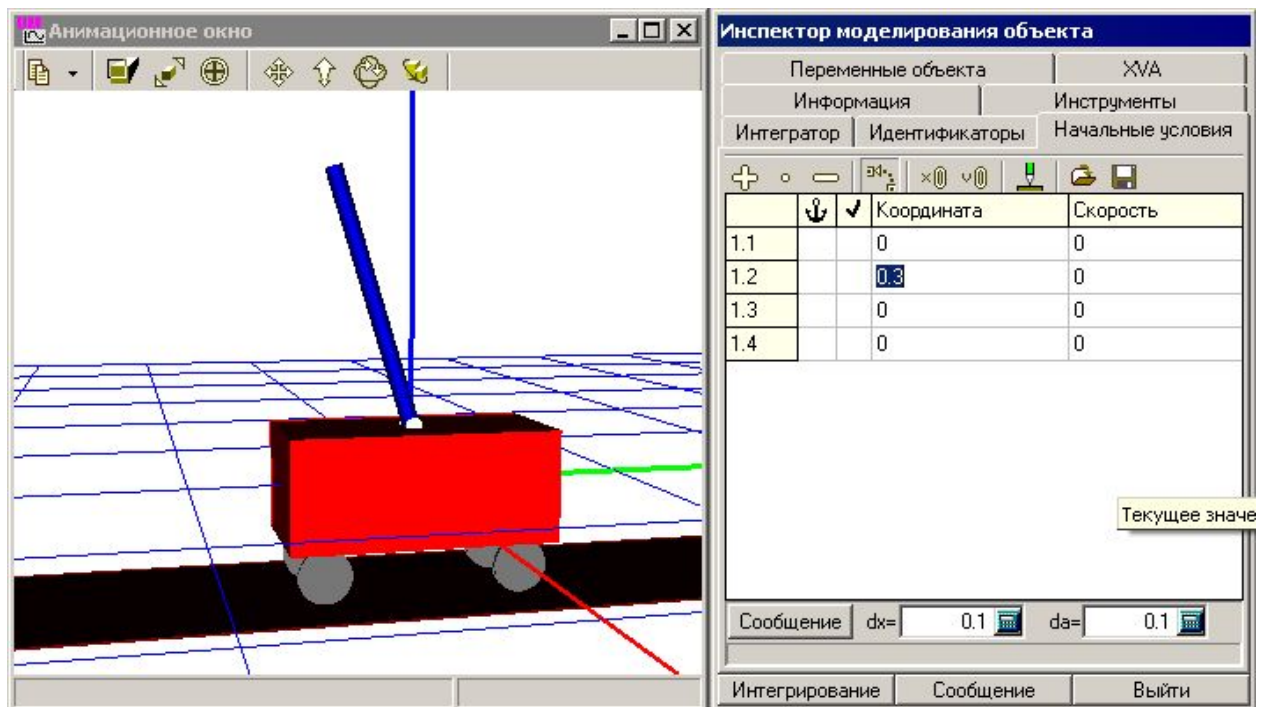


3.2.3. Моделирование движения

Теперь наша модель перевернутого маятника полностью описана. Сейчас мы отклоним маятник из положения равновесия и проверим, действительно ли система управления работает.

1. Выберите пункт меню **Анализ / Моделирование**. Откроется **Мастер моделирования объекта**.
2. Перейдите на закладку **Начальные условия**.
3. Отклоним маятник на 0,3 радиана от положения равновесия. В поле второй координаты (Координата/1.2) введите **0.3**.

Замечание. Первая координата – положение тележки относительно оси Y. Третья и четвертая координаты – переменные состояния модели Matlab/Simulink.





4. Перейдите на закладку **Интегратор**, в поле **Численный метод** выберите **Метод Парка**, в поле **Время моделирования** установите **0.1**, в поле **Шаг представления результатов** установите значение **0.0002**, а в поле **Погрешность** установите **1E-7**.
5. Нажмите кнопку **Интегрирование**.

Действительно, как вы можете наблюдать, результаты моделирования показывают, что система управления стабилизировала перевернутый маятник.

Построение вектора управляющей силы и графиков переменных

Выведем в анимационное окно график управляющей силы, действующей на тележку, а в графические окна – величину этой силы и угол поворота маятника.

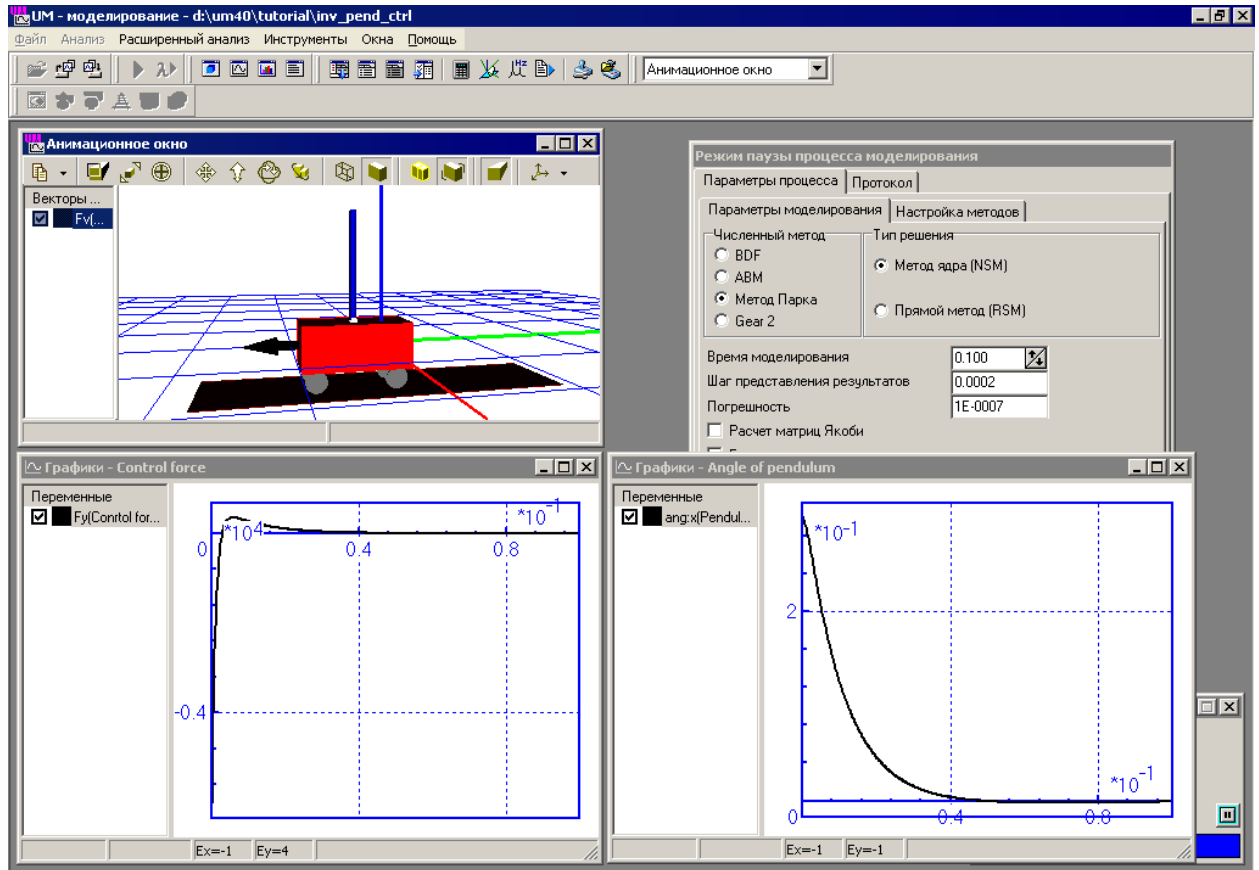
Сформируем переменную – вектор силы, для отрисовки в анимационном окне.

1. Откройте **Мастер переменных**.
2. Перейдите на закладку **Акт. силы** (Активные силы), в поле **Исследуемая величина** выберите **Сила**, в поле **Компонента** – **V** (вектор), в поле **Действует на тело** – **Cart**.
3. Нажав кнопку  поместите сформированную переменную в контейнер переменных и перетащите ее анимационное окно.
4. Кнопкой  переключите анимационное окно в режим проволочной графики. Это позволит наблюдать вектор силы даже когда он будет изображаться внутри тележки.

А теперь сформируем две переменные для отрисовки в графических окнах: величину управляющей силы и угол поворота маятника.

5. Откройте новое графическое окно. В нем будем строить график управляющей силы.
6. Перейдите обратно в **Мастер переменных**, на закладку **Акт. силы**. В поле **Компонента** выберите **Y**. Поместите переменную в контейнер и перетащите ее в графическое окно.
7. Откройте еще одно графическое окно. В этом окне мы будем строить зависимость угла поворота маятника от времени.
8. Перейдите в окно **Мастера переменных** на закладку **Угл. перем.** (угловые переменные).
9. В списке тел выберите **Pendulum**, в поле **Исследуемая величина** выберите **Вект.(ор) поворота**, в поле **Компонента** – **X**.

10. Поместите переменную в контейнер переменных, а затем перетащите ее в графическое окно.
11. Закройте окно **Мастера переменных**.
12. Запустите процесс моделирования.



3.3. Двигатель постоянного тока

Модель механической части системы без подключенного электрического двигателя находится в каталоге `{um_root}\samples\tutorial\dc_motor`. Перед началом урока, проверьте ее наличие в указанном каталоге. Если такой модели нет, то скачайте ее из интернета по адресу: http://www.umlabor.ru/download/60/dc_motor.zip.

Как и в предыдущем уроке, мы не будем подробно разбирать этапы создания моделей механической и электрической частей. Предполагается, что читатель может самостоятельно открыть и изучить модель механической части в Универсальном механизме, и модель двигателя постоянного тока – в Matlab/Simulink. В данном примере мы ограничимся только описанием связи механической части с двигателем, импортированным из Matlab/Simulink.

Готовая модель электромеханической системы находится в каталоге `{um_root}\samples\simulink\dc_motor_fin`¹.

¹ http://www.umlabor.ru/download/60/dc_motor_fin.zip

3.3.1. Описание системы управления

Модель двигателя имеет один вход – угловую скорость ротора (рад/с) и три выхода: (1) электромагнитный момент, (2) ток якоря и (3) напряжения цепи якоря. Копии экрана Simulink-модели рассматриваемого двигателя приведены на рис. 3.6 и 3.7.

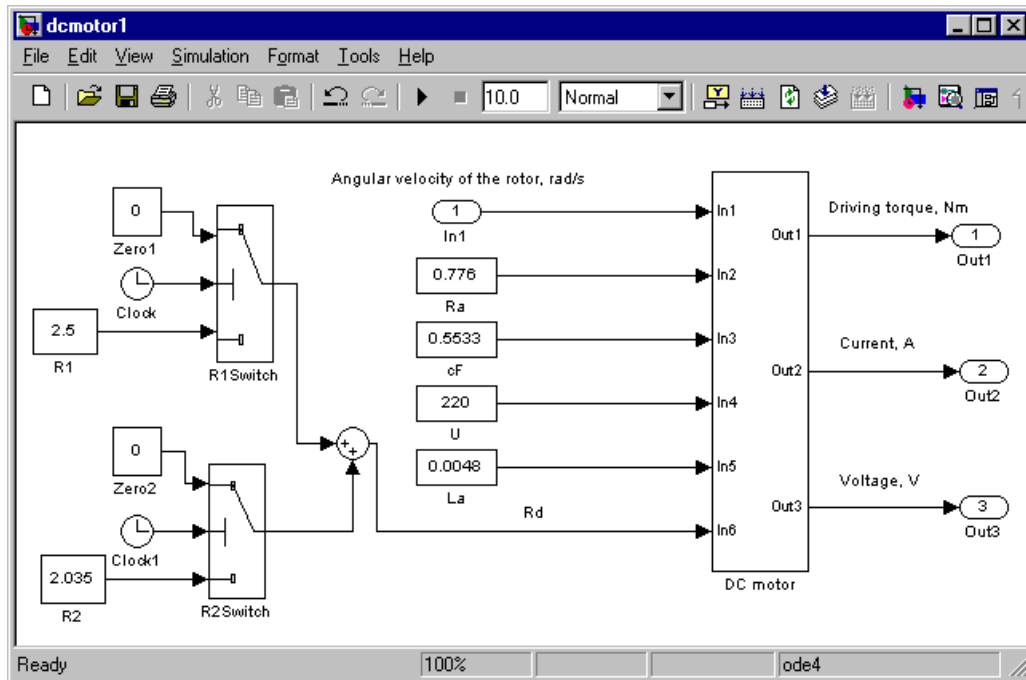


Рис.3.6. Модель двигателя в Matlab/Simulink

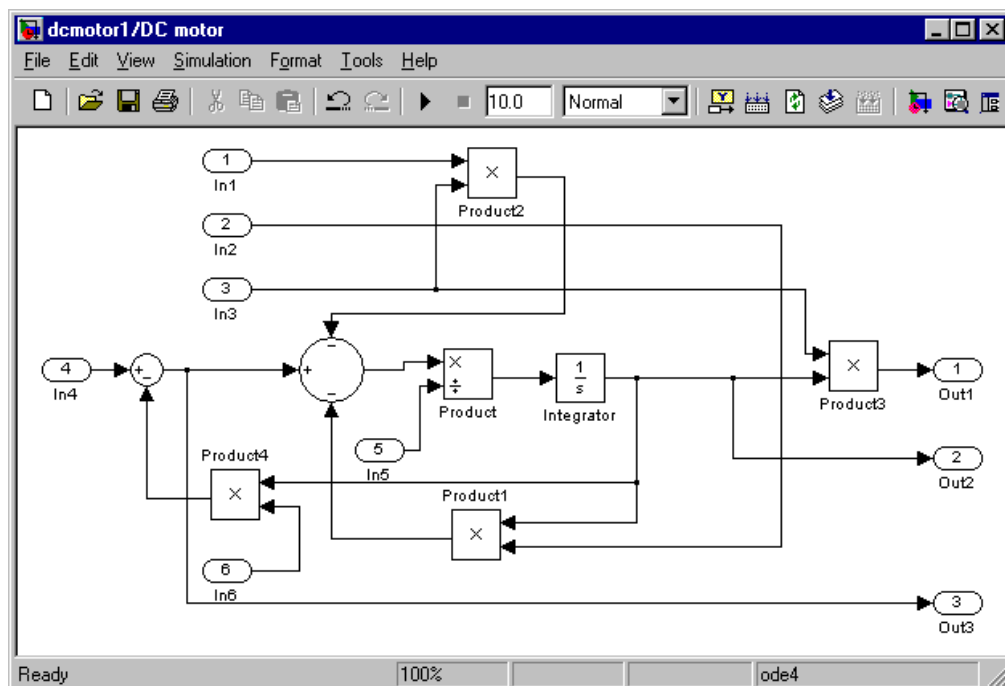


Рис.3.7. Подсистема DC motor

3.3.2. Экспорт двигателя из Matlab/Simulink

Сейчас мы должны из *.mdl файла, который содержит нашу Simulink-модель двигателя, получить файл *.dll.

Matlab/Simulink модель двигателя постоянного тока находится в файле `..\dc_motor\dcmotor1.mdl`. Вы можете либо откомпилировать данный *.mdl файл пользуясь инструкциями, изложенным в п. 3.2.1, стр. 17, либо пропустить этап компиляции файла *.mdl и воспользоваться уже готовым файлом **dcmotor1.dll**.

3.3.3. Подключение библиотеки Matlab/Simulink в UM


Загрузка модели механической части

1. Запустите программу моделирования UM Simulation.
2. Откройте модель `{um_root}\samples\tutorial\dc_motor`.
3. Откройте новое анимационное окно.

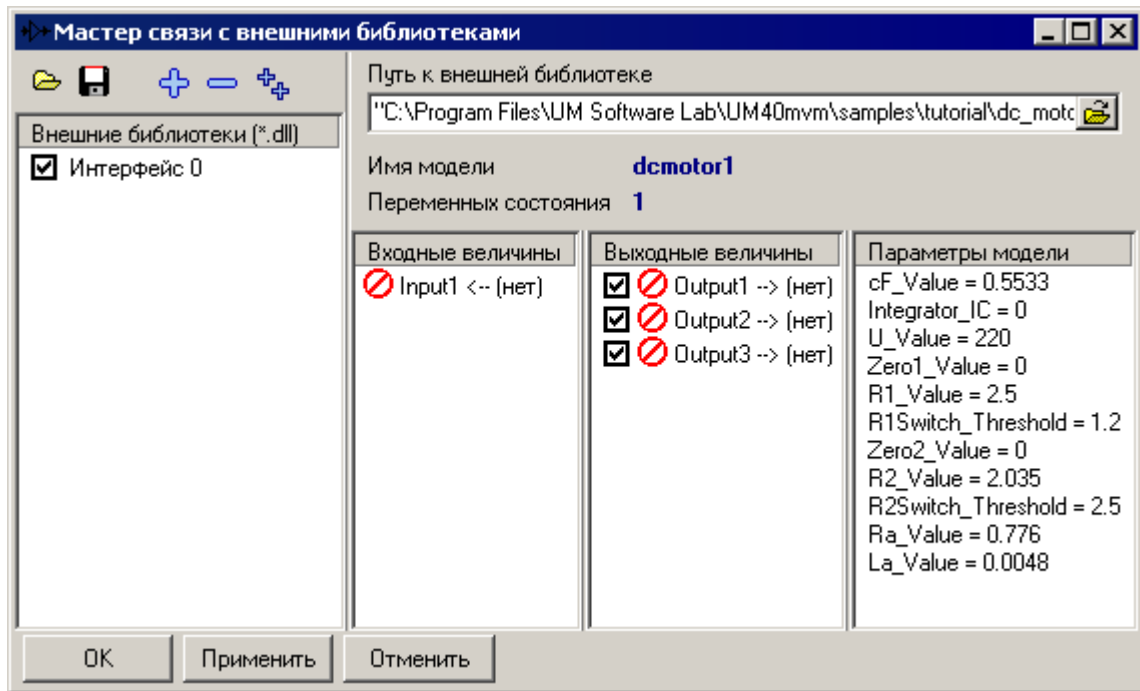
Загрузка модели Matlab/Simulink

1. Выберите пункт меню **Инструменты / Интерфейс с внешними библиотеками**.

Появится окно **Мастера связи с внешними библиотеками**.

2. Нажмите кнопку  для добавления новой внешней библиотеки.
3. В поле **Путь к внешней библиотеке** выберите файл `dcmotor1.dll`. Если вы создали выходную библиотеку dll самостоятельно, то выберите ее, иначе выберите готовую dll системы управления из каталога `..\dc_model`.

Мастер связи загрузит указанную модель, установит число входных и выходных величин модели двигателя, а также число параметров модели и переменных состояния. В данном случае в окне появится одна входная и три выходных величины, см. рис. ниже. На вход системы управления мы подадим угол поворота вала, первый выход назначим параметру **Ma**, второй и третий – параметрам **I** и **U**, соответственно.




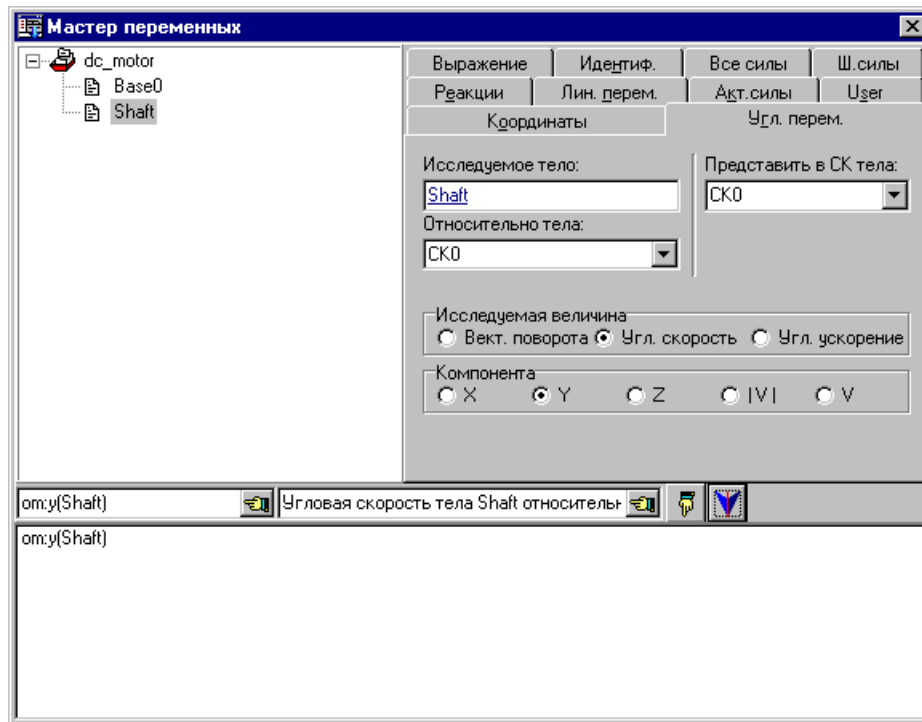
Переименование интерфейса

1. Выберите в списке **Модели Simulink (*.dll)** текущий интерфейс **Интерфейс 0**, правой кнопкой мыши вызовите контекстное меню и в нем выберите пункт **Редактировать название...**
2. В поле редактирования названия введите **DC motor** и нажмите клавишу **Enter**.

Назначение входных величин модели Matlab/Simulink

С помощью мастера переменных сформируем переменную «угловая скорость вала» и назначим ее в качестве входной величины для системы управления.

1. Запустите **Мастер переменных** (пункт меню **Инструменты/Мастер переменных**).
2. Перейдите на закладку **Угл. перем.** (угловые переменные).
3. В списке тел слева выберите **Shaft**, в поле **Исследуемая величина** выберите **Угл. скорость**, а в поле **Компонента** выберите **Y**.
4. Создайте переменную кнопкой . Переменная **om:y(Shaft)** появится в контейнере переменных.



5. Перетащите мышкой созданную переменную **om:y(Shaft)** в окно **Мастера связи с Matlab/Simulink** на вход **Input1**.

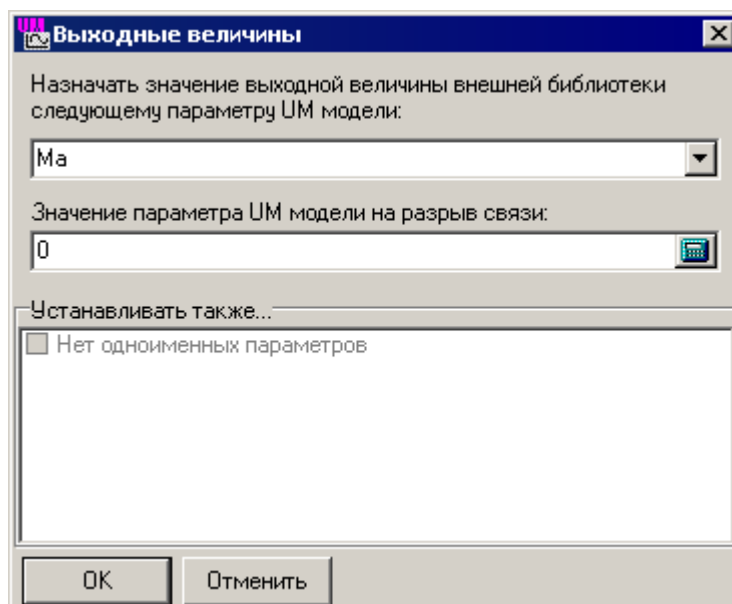
Вход модели **Input1** станет помечен зеленой галочкой, что означает, что на данный вход модели Matlab/Simulink назначена переменная.

6. Закройте окно **Мастера переменных**.

Назначение выходных величин модели Matlab/Simulink

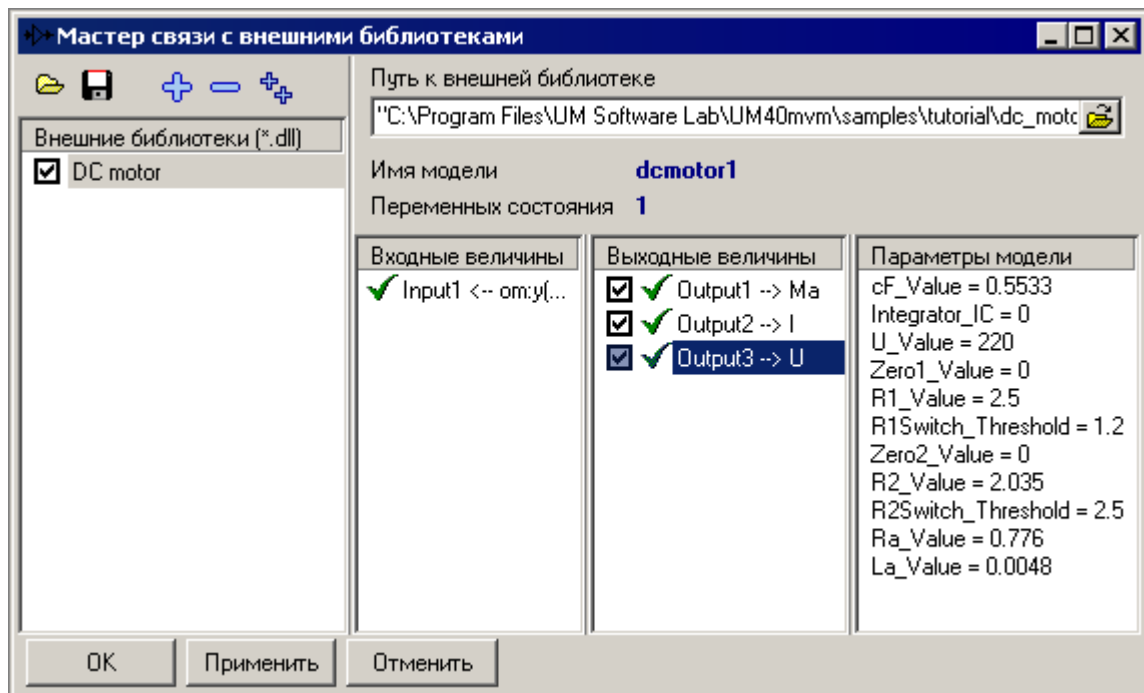
Выходные величины модели Matlab/Simulink связываются с параметрами UM-модели.

1. Дважды щелкните на элементе **Выходные величины / Output1**. Появится диалоговое окно связывания параметров UM-модели с выходами модели Matlab/Simulink.
2. В поле **Параметр UM модели** выберите **Ma**.
3. Закройте окно кнопкой **ОК**.



4. Выход **Output2** назначьте параметру **I**.
5. Выход **Output3** назначьте параметру **U**.

6. Нажмите кнопку **ОК** для сохранения данных с закрытием окна **Мастера связи с Matlab/Simulink**.

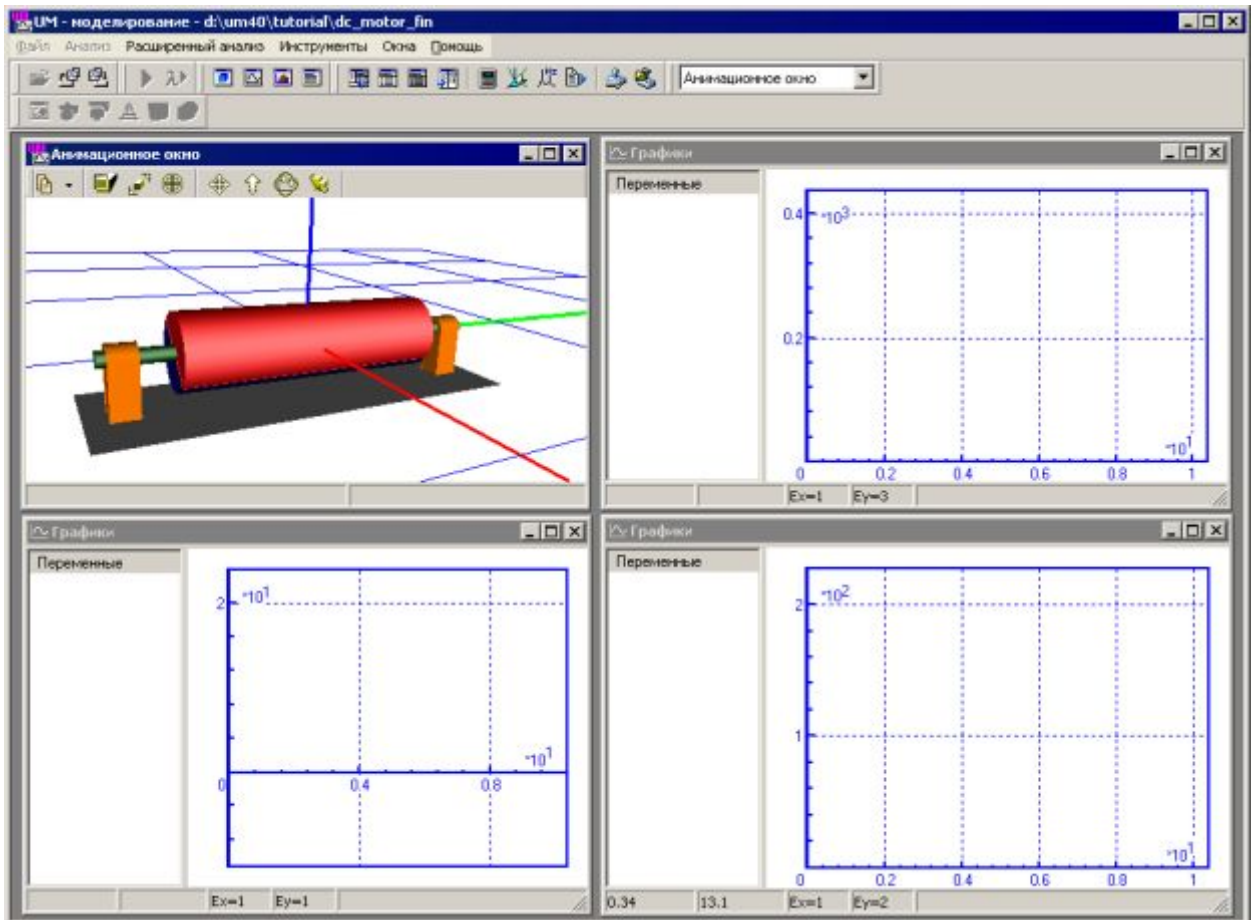


3.3.4. Моделирование движения

Модель подготовлена для дальнейшего анализа. Сейчас мы откроем одно анимационное окно и несколько графических окон. В графических окнах будем строить графики следующих переменных: активного момента и момента сил сопротивления, угловой скорости вала, а также тока и напряжения на двигателе.



Подготовка моделирования

1. Для создания нового анимационного окна выберите пункт меню **Инструменты / Анимационное окно**.
2. При помощи пункта меню **Инструменты / Графическое окно** создайте три новых графических окна. Разместите окна так, как показано на рисунке ниже.



3. Откройте окно **Мастера переменных** (пункт меню **Инструменты / Мастер переменных**).

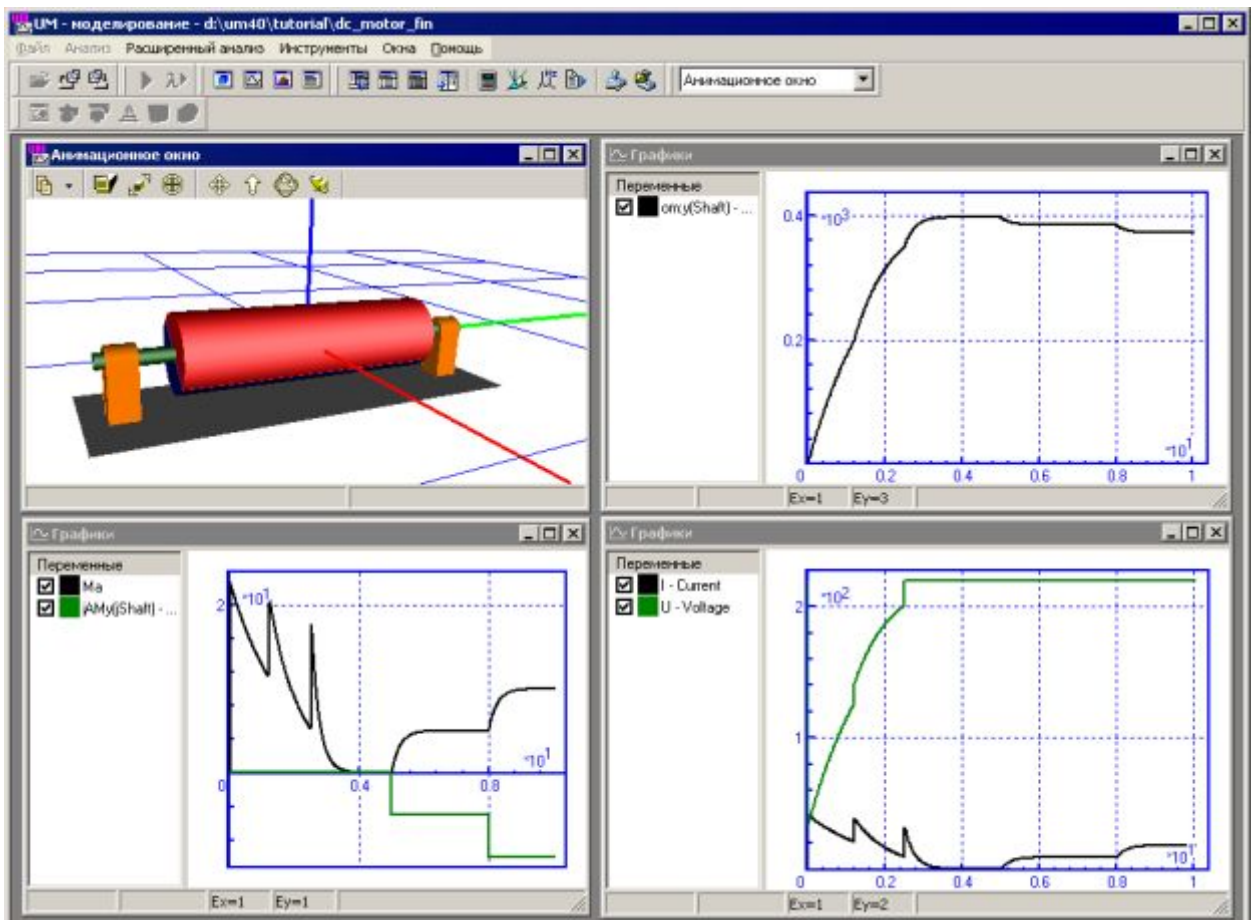
Выведем на график значение электромагнитного момента, действующего на вал со стороны электродвигателя, а также момент сил сопротивления.

4. В **Мастере переменных** перейдите на закладку **Идентиф.** (Идентификаторы). В списке идентификаторов модели слева выберите **Ma**. Создайте переменную кнопкой . Переменная **Ma** появится в контейнере переменных снизу.
5. Перейдите на закладку **Ш. силы** (Шарнирные силы). В поле **Исследуемая величина** выберите **Момент**, в поле **Компонента** выберите **Y**. В поле **Действует на тело** выберите **Shaft**. Создайте переменную кнопкой .
6. Контейнер переменных **Мастера переменных** сейчас содержит две переменные: активный момент и момент сил сопротивления. Выделите эти переменные и мышкой перетащите их в левое нижнее графическое окно.
7. Создайте и перетащите в правое верхнее окно переменную – угловую скорость вала (закладка **Угл. перем.**, **Исследуемая величина** = **Угл. скорость**, **Компонента** = **Y**).
8. Создайте и перетащите в правое нижнее окно две переменные – ток и напряжение на двигателе (закладка **Идентиф.**, идентификаторы **I** и **U**).
9. Закройте **Мастер переменных**.

Моделирование динамики системы

1. Выберите пункт меню **Анализ / Моделирование**. Откроется **Мастер моделирования объекта**.
2. Перейдите на закладку **Интегратор/Параметры моделирования** и проверьте установки. В поле **Численный метод** должно быть выбран **BDF**, в поле **Время моделирования** – 10 с, **Шаг представления результатов** – **0,02**, **Погрешность** – **0,001**.
3. Нажмите кнопку **Интегрирование**.
4. В процессе расчета будут построены графики, см. рис. ниже.

Проанализируем полученные результаты. На графике электромагнитного момента и тока четко видны скачки, соответствующие выключению ступеней пускового реостата. До номинальной угловой скорости вал разгоняется примерно за 4с, после чего электромагнитный момент становится равным нулю. Далее, в момент времени 5с происходит ступенчатое увеличение момента сил сопротивления. Вслед за внешним моментом сил сопротивления электромагнитный момент примерно за одну секунду выходит на новое установившееся значение. Увеличение нагрузки отрабатывается увеличением электромагнитного момента, тока якоря и снижением угловой скорости вала.



4. Моделирование с использованием CoSimulation

В этой части руководства будут рассмотрены примеры моделей систем управления с использованием инструмента **CoSimulation** из модуля UM Control. Это подразумевает, что модель механической части будет экспортирована из UM и импортирована в Matlab/Simulink, где будет представляться в системе управления в виде одного блока, имеющего некоторое количество входов и выходов. Остановимся на основных моментах, поясняющих суть предлагаемой технологии моделирования.

4.1. Порядок работы

В общем случае моделирования динамики механических систем с использованием **CoSimulation** предполагает выполнение следующих этапов.

- Описание модели системы управления в среде Matlab/Simulink.
- Включение в эту модель блока S-функции, который представляет собой модель механической системы.
- Создание модели механической системы в программе UM Input.
- Загрузка подготовленной модели механической части в программу UM Simulation. Формирование файлов настроек и управляющего m-файла с помощью **Мастера экспорта в Matlab/Simulink** для подключения этой модели в систему управления Matlab/Simulink.
- Создание связи между UM-моделью и системой управления Matlab/Simulink.
- Моделирование динамики управляемого движения в среде Matlab/Simulink.

Модель механической системы UM, подключаемая в систему управления Matlab/Simulink, рассматривается как черный ящик, который по некоторому закону преобразует входные величины в выходные. При включении механической системы в модель системы управления на выходы механической системы назначаются *переменные*, которые создаются при помощи **Мастера переменных**. Входные величины и параметры S-функции связываются с параметрами UM-модели.

Для реализации управляющих усилий со стороны системы управления в модель механической системы вводятся силы/моменты, значения или характеристики (например, коэффициент жесткости, коэффициент диссипации) которых задаются параметрами модели. Затем с помощью **Мастера экспорта в Matlab/Simulink** эти параметры связываются с величинами, которые являются входами S-функции.

4.2. Перевернутый маятник

Модель перевернутого маятника с готовыми файлами настроек и управляющим m-файлом находится в каталоге `{um_root}\samples\cosimulation\inv_pend_cosim`. Перед началом урока, проверьте ее наличие в указанном каталоге. Если такой модели нет, то скачайте ее из интернета по адресу: http://www.umlabor.ru/download/60/inv_pend_cosim.zip.

В данном уроке мы не будем подробно разбирать этапы создания модели механической части, ограничимся только описанием связи механической части с системой управления Matlab/Simulink.

4.2.1. Подготовка системы управления в Matlab/Simulink

Модель системы управления Matlab/Simulink в случае CoSimulation, см. рис. 4.1, очень похожа на ту модель, которая рассматривалась ранее применительно к инструменту Matlab Import, рис. 3.1, но имеет одно существенное отличие: модель механической системы представляет собой отдельный блок – S-функцию. Этот компонент можно найти в библиотеке Simulink **User-Defined Functions**. При первом добавлении S-функции в модель системы управления она будет иметь название **System**.

Именно S-функция обеспечивает связь модели системы управления с моделью ПК «Универсальный механизм». В нашей модели для организации управления перевернутым маятником входной переменной для S-функции будет величина силы, подаваемой на тележку, а выходной величиной S-функции будет угол отклонения маятника от вертикали.

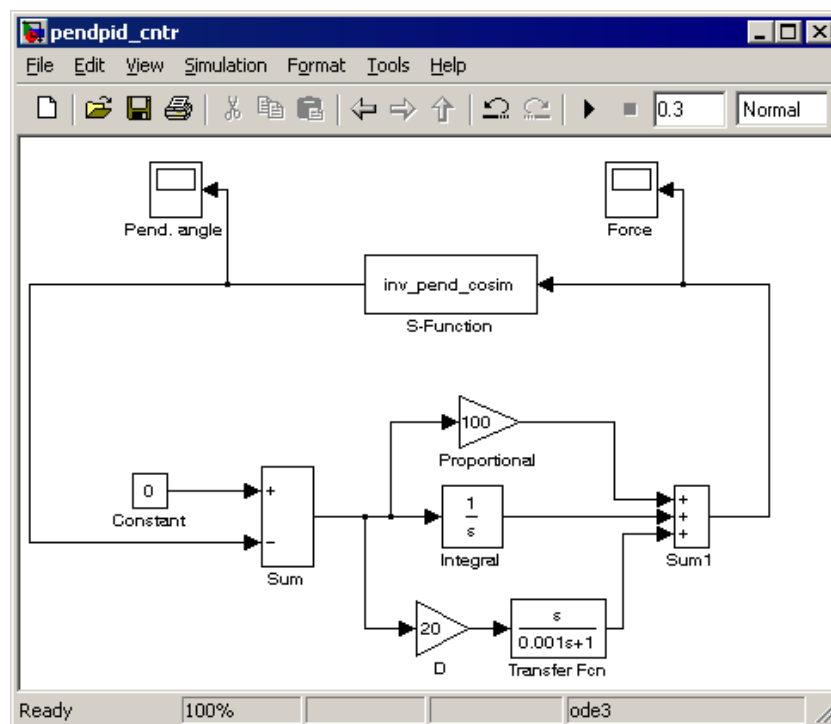


Рис. 4.1. Система стабилизации перевернутого маятника с механической системой как S-функцией

Подготовленную модель системы управления перевернутым маятником в Matlab/Simulink вы можете найти в файле `{um_root}\samples\cosimulation\inv_pend_cosim\pendpid_cntr.mdl`, рис. 4.1. К модели в Matlab/Simulink мы обратимся в п. 4.2.3, а пока рассмотрим особенности экспорта механической части из UM.

4.2.2. Экспорт модели механической системы из UM

На следующем шаге нам необходимо указать, какие именно входные и выходные величины S -функции потребуются для моделирования, сохранить связанные с этим настройки и сгенерировать специальный m -файл, который будет управлять S -функцией.

Загрузка модели механической части

1. Запустите программу моделирования **UM Simulation**.
2. Загрузите модель `{um_root}\samples\cosimulation\inv_pend_cosim`.

Мастер экспорта в Matlab/Simulink

1. Выберите пункт меню **Инструменты / Мастер экспорта в Matlab/Simulink**.

Появится окно **Мастера экспорта в Matlab/Simulink**.

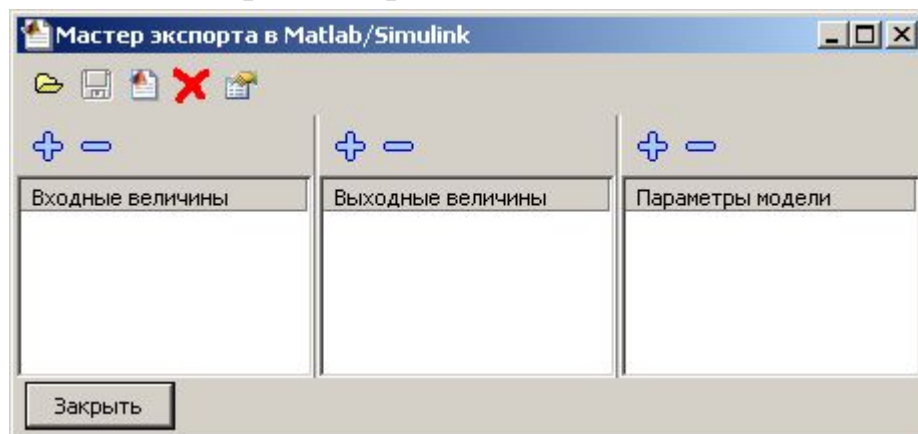



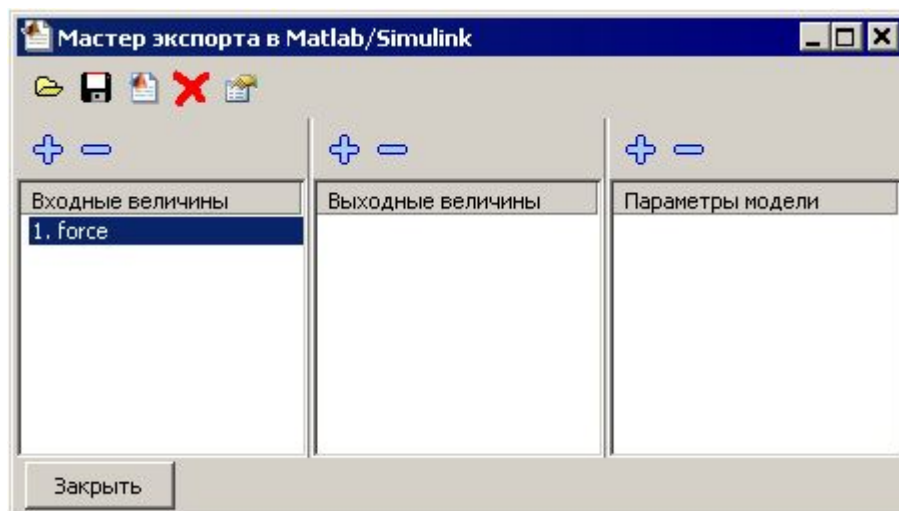
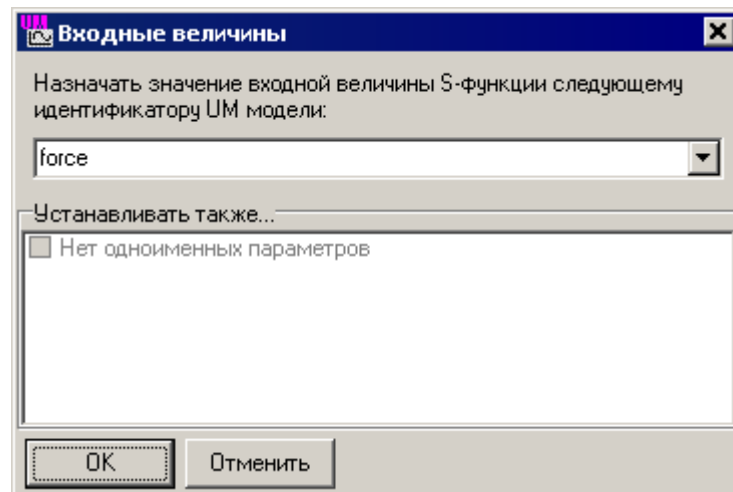
Рис. 4.2. Мастер экспорта в Matlab/Simulink

В данном случае нам необходимо назначить одну входную и одну выходную величины. На вход модели перевернутого маятника мы подадим величину усилия на тележку, а в качестве выходной величины назначим угол отклонения маятника от вертикали.

Назначение входных величин


Для того чтобы добавить новую входную величину для UM-модели необходимо выполнить следующие действия.

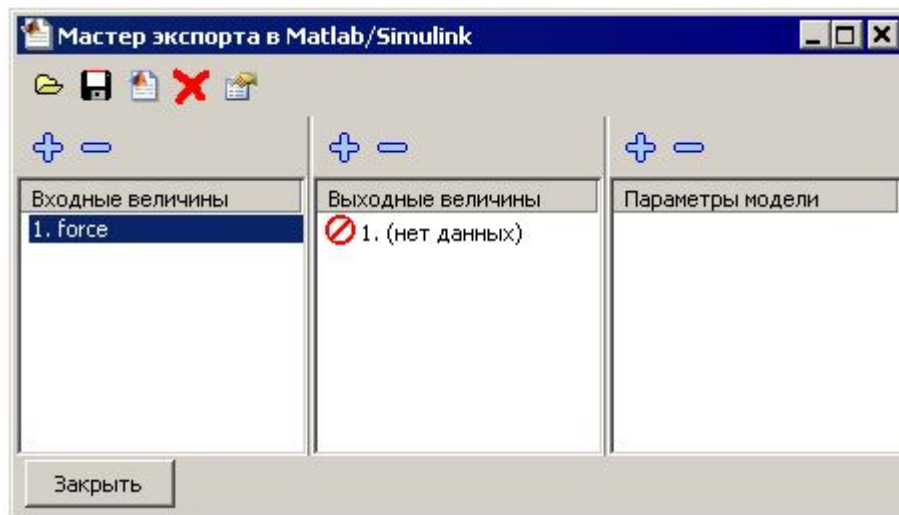
1. Нажмите кнопку  над списком **Входные величины**. Появится диалоговое окно назначения входных величин S-функции идентификаторам UM-модели.
2. В поле **Назначать значение входной величины S-функции...** выберите идентификатор модели **force**.
3. Закройте окно кнопкой **ОК**.




Назначение выходных величин для модели перевернутого маятника

С помощью **Мастера переменных** сформируем переменную **Угол отклонения маятника от вертикали** и назначим ее в качестве выходной величины S-функции.

1. Нажмите кнопку  над списком **Выходные величины**. В списке выходных величин появится одна новая величина.



2. Запустите **Мастер переменных** (меню **Инструменты/Мастер переменных**).
3. Перейдите на закладку **Угл. перем.** (угловые переменные), см. рис. 4.3.
4. В списке тел слева выберите **Pendulum**, выключите флажок **Использовать ориентацию при нулевых координатах**, в поле **Исследуемая величина** выберите **Вект.(ор) поворота**, а в поле **Компонента** выберите **X**.
5. Создайте переменную кнопкой . Переменная **ang:x(Pendulum)** появится в контейнере переменных.

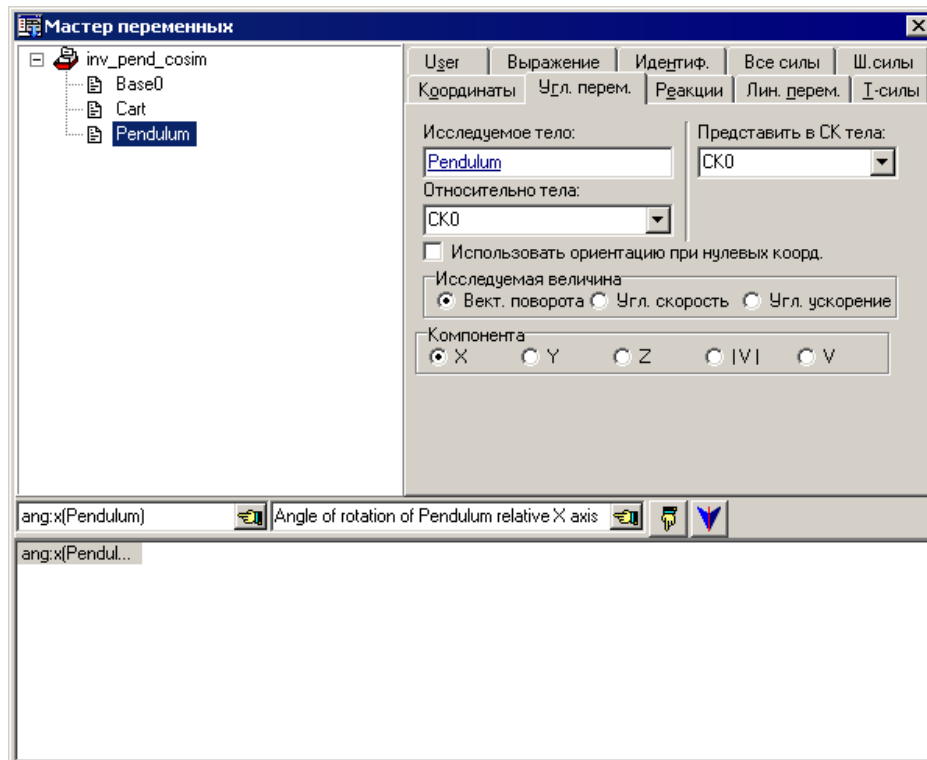
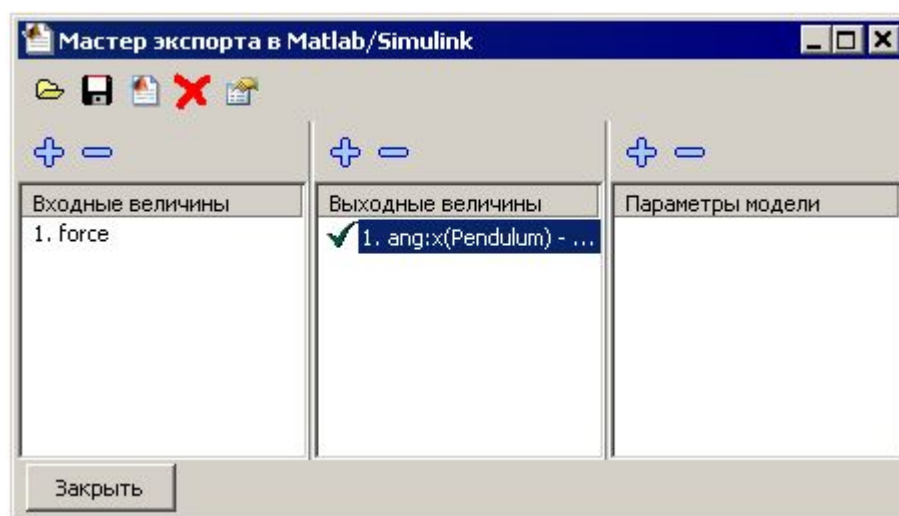


Рис. 4.3. Создание переменной

6. Перетащите мышкой созданную переменную **ang:x(Pendulum)** в окно **Мастера экспорта в Matlab/Simulink** и назначьте ее в качестве выходной величины.

Выход модели станет помечен зеленой галочкой, что означает, что на данный выход S-функции назначена переменная.



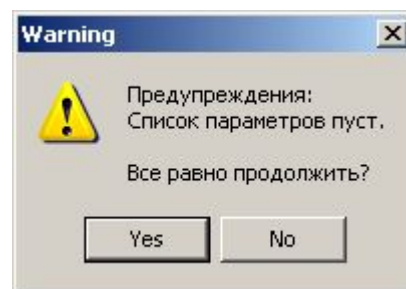
Все необходимые настройки для экспорта UM-модели в Matlab/Simulink закончены. Далее сгенерируем m-файл, который обеспечит работу UM-модели, как S-функции, одновременно сохраним файл настроек *.cosim.

Сохранение управляющего m-файла

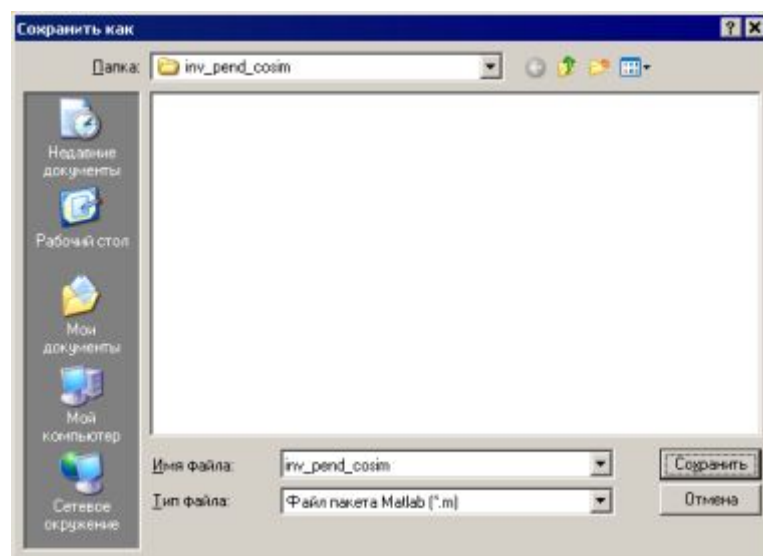
Теперь, когда мы описали количество входов и выходов, а также связали *идентификаторы* UM модели с входами, а *переменные* – с выходами, нужно сформировать m-файл, который необходим для связи UM-модели и системы управления Matlab/Simulink. Для этого выполните следующие действия.

1. Нажмите кнопку  для вызова диалога сохранения m-файла.

При этом появится предупреждение о том, что список параметров модели пуст. Это сообщение можно проигнорировать и продолжить работу, нажав на кнопку **Yes**.



2. В появившемся окне диалога сохранения файла необходимо указать в названии файла **inv_pend_cosim** и сохранить его. Одновременно с созданием m-файла будет сохранен одноименный файл **inv_pend_cosim.cosim**, который будет использоваться «Универсальным механизмом» при работе под управлением Matlab/Simulink.



На этом действия на стороне UM закончены. Переходим к описанию связи на стороне Matlab/Simulink.

4.2.3. Создание связи между UM-моделью и Matlab/Simulink

Подготовленный на предыдущем этапе m-файл, который по умолчанию сохранен в каталоге UM-модели, в общем случае необходимо скопировать в каталог, где находится система управления Matlab/Simulink. В нашем случае файл системы управления (.mdl) также находится в каталоге модели, поэтому копировать m-файл в данном случае никуда не нужно. Для создания связи между системой управления Matlab/Simulink и механической моделью выполните следующие действия.

1. Запустите **Matlab**

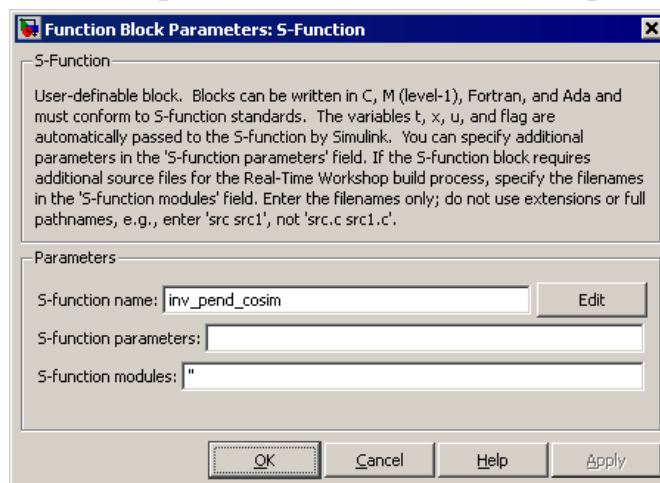
2. В поле **Current Directory** выберите каталог **{um_root}\samples\cosimulation\inv_pend_cosim**.



3. Откройте файл **pendpid_cntr.mdl**, в котором хранится модель системы управления перевернутым маятником.

4. Двойным щелчком мыши на **S-функции** вызовите окно параметров этого блока.

5. Убедитесь, что в поле **S-function name** установлено **inv_pend_cosim**, что соответствует названию ранее подготовленного m-файла.



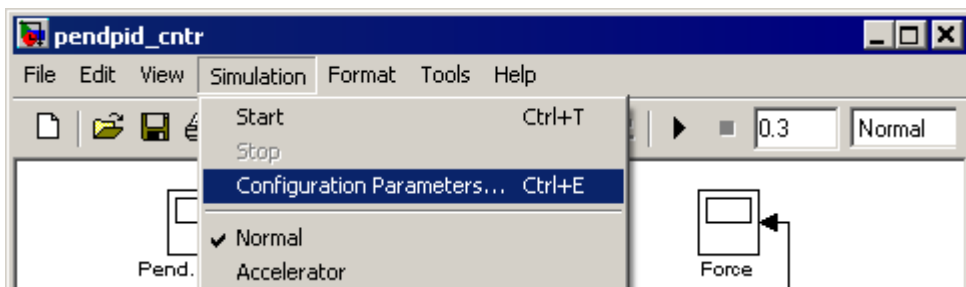
6. Нажмите **OK**.

4.2.4. Моделирование движения

Модель подготовлена для дальнейшего анализа. Сейчас мы установим параметры моделирования, а затем запустим модель на выполнение.

Установка параметров моделирования

1. Выберите пункт меню **Simulation/Configuration Parameters**.



2. Перейдите на вкладку **Solver**.
3. В поле **Start time** установите **0.0**, в поле **Stop time** – значение **0.3**, в поле **Type** выберите **Fixed-step**, в поле **Fixed-step size** установите **1E-4**, в поле **Solver** выберите метод **ode3(Bogacki-Shampine)**. Если вы все сделали верно, то ваши настройки должны совпадать с теми, которые приведены на рис. 4.4 (этот пример приведен для версии Matlab R2006a).

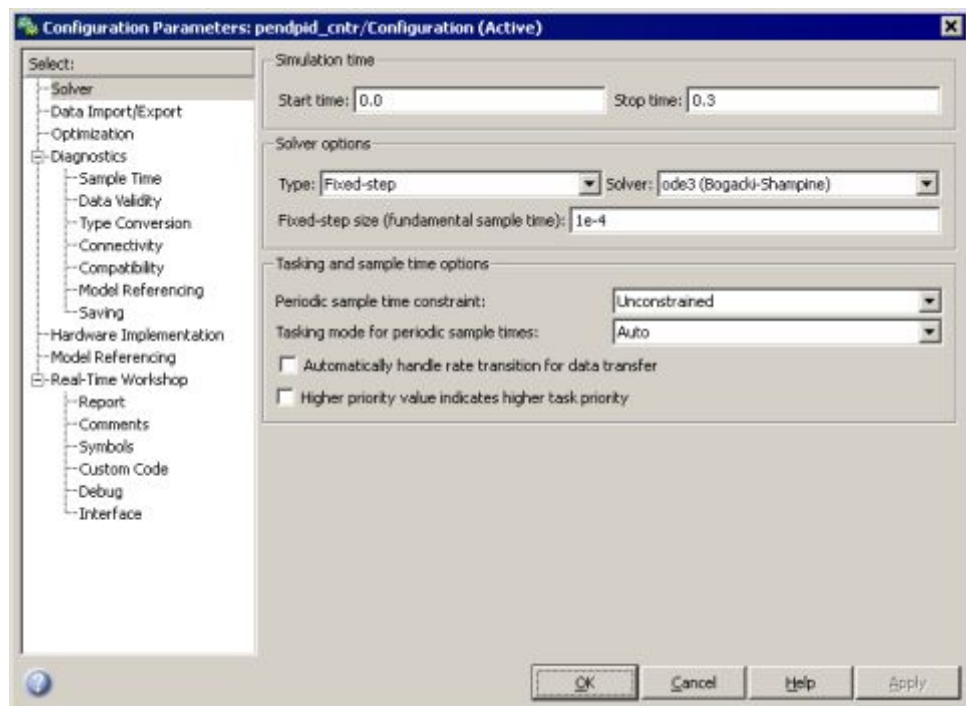



Рис. 4.4. Параметры конфигурации модели

4. Нажмите **ОК**.

Моделирование динамики системы

1. Двойным щелчком мыши откройте окна **Scope**, в которых будут отображаться угол отклонения маятника от вертикали и величина усилия, подаваемого на тележку.
2. Кнопкой  запустите модель на выполнение.
3. По истечении времени моделирования вы увидите результаты, приведенные на рис. 4.5 и 4.6.

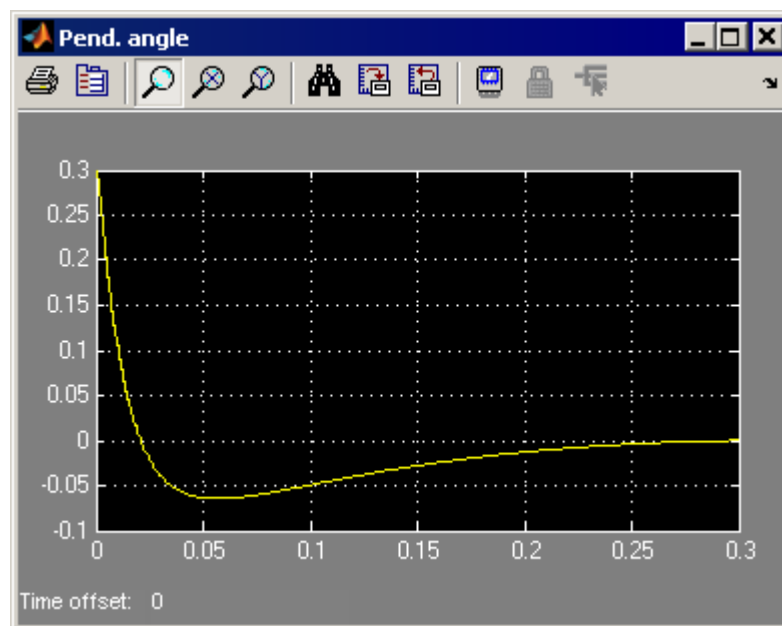


Рис. 4.5. График изменения угла отклонения маятника от вертикали

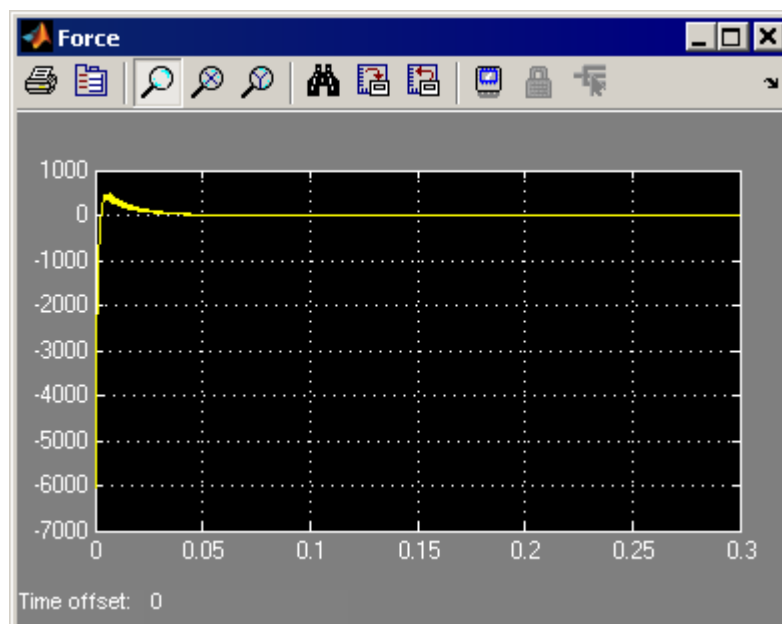


Рис. 4.6. График изменения величины усилия, подаваемого на тележку

4.3. Двигатель постоянного тока

UM и Matlab/Simulink модели двигателя постоянного тока с готовыми файлами настроек и управляющим m-файлом находится в каталоге `{um_root}\samples\cosimulation\dcmotor_cosim`. Перед началом урока, проверьте наличие этого каталога. Если такого каталога нет, то скачайте его из интернета по адресу: http://www.umlabor.ru/download/60/dcmotor_cosim.zip. В данном уроке мы не будем подробно разбирать этапы создания модели механической части, ограничимся только описанием связи механической части с системой управления Matlab/Simulink.

4.3.1. Подготовка системы управления в Matlab/Simulink

Подготовленная модель системы управления двигателя постоянного тока в Matlab/Simulink представлена на рис. 4.7. Подсистема DC motor показана на рис. 3.7, стр. 36.

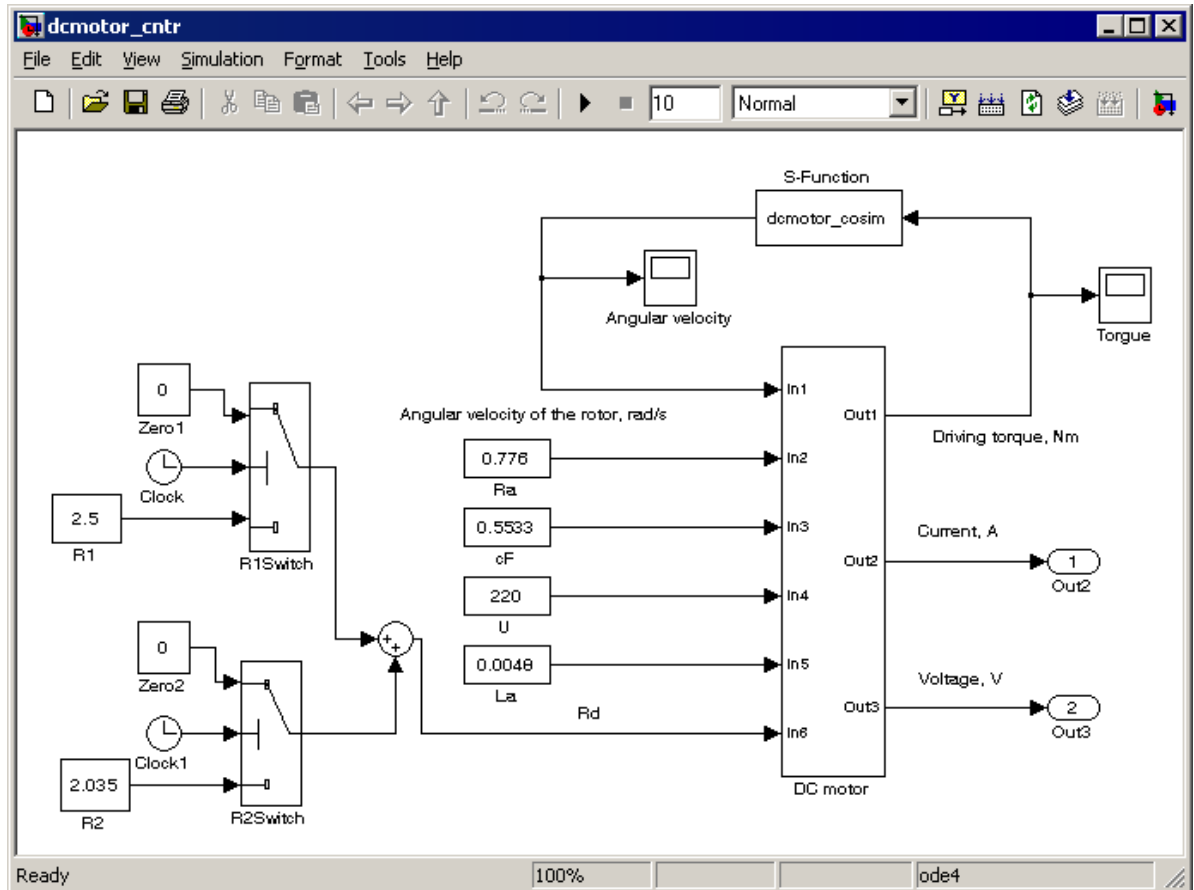


Рис. 4.7. Модель двигателя в Matlab/Simulink

В нашей модели для организации управления двигателем постоянного тока входной переменной для S-функции будет величина вращательного момента, подаваемого на вал со стороны электродвигателя. В свою очередь, выходной величиной S-функции будет являться угловая скорость вала.

Подготовленную модель системы управления перевернутым маятником в Matlab/Simulink вы можете найти в файле `{um_root}\samples\cosimulation\dcmotor_cosim\dcmotor_cntr.mdl`, рис. 4.7. К модели в Matlab/Simulink мы обратимся в п. 4.3.2, а пока рассмотрим особенности экспорта механической части из UM.

4.3.1. Экспорт модели механической системы из UM

Загрузка модели механической части

1. Запустите программу моделирования **UM Simulation**.
2. Загрузите модель `{um_root}\samples\cosimulation\dcmotor_cosim`.


Начало экспорта в Matlab/Simulink

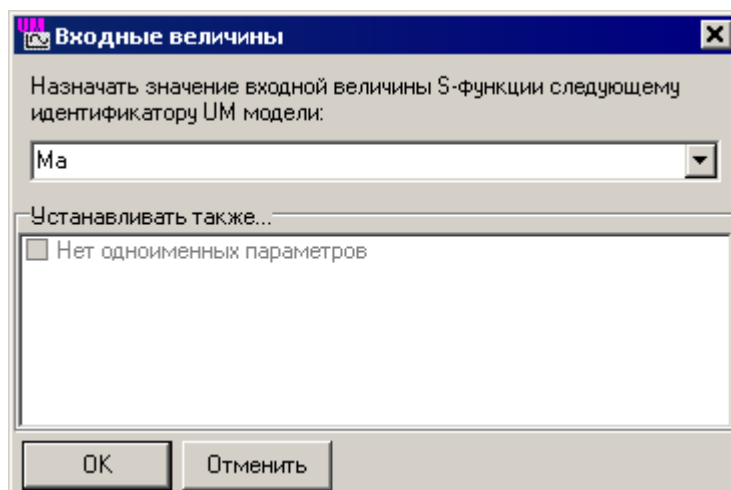
1. Выберите пункт меню **Инструменты / Мастер экспорта в Matlab/Simulink**. Появится окно **Мастера экспорта в Matlab/Simulink**.

В данном случае нам необходимо назначить одну входную и одну выходную величины. На вход модели двигателя постоянного тока мы подадим величину вращательного момента, а в качестве выходной величины назначим угловую скорость вала.

Назначение входных величин



Добавим новую входную величину.

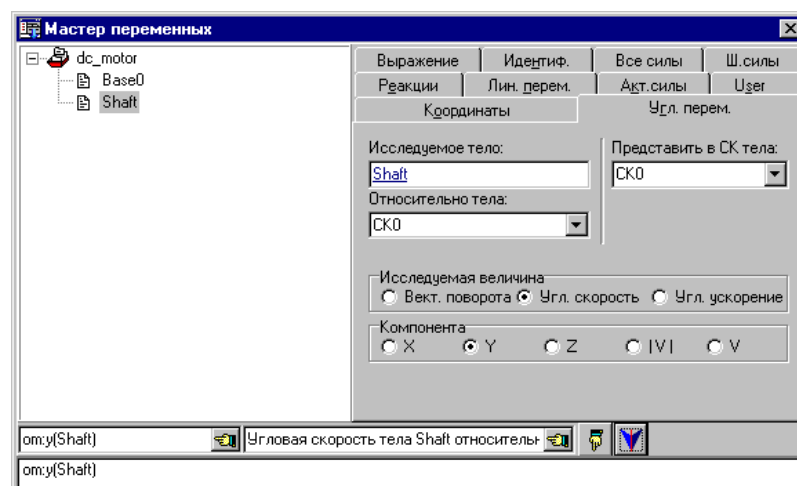
1. Нажмите кнопку  над списком входных величин. Появится диалоговое окно назначения входных величин S-функции идентификаторам UM-модели.
2. В поле **Назначать значение входной величины S-функции...** выберите **Ma**.
3. Закройте окно кнопкой **ОК**.



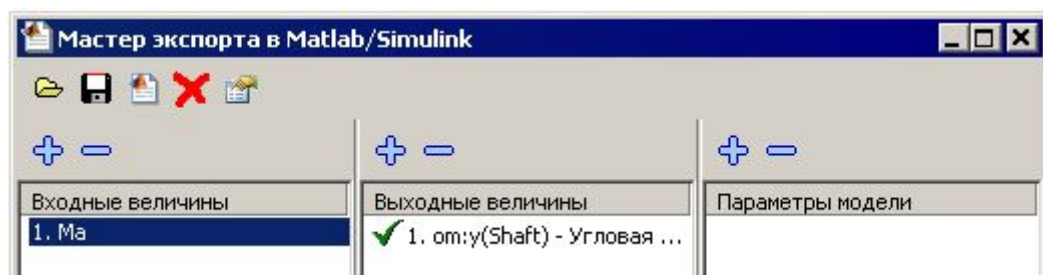
Назначение выходных величин

С помощью **Мастера переменных** сформируем переменную «**угловая скорость вала**» и назначим ее в качестве выходной величины S-функции.


1. Создайте новую выходную величину нажав кнопку  над списком **Выходные величины**. В списке выходных величин появится одна новая величина.
2. Запустите **Мастер переменных** (пункт меню **Инструменты/Мастер переменных**).
3. Перейдите на закладку **Угл. перем.** (угловые переменные).
4. В списке тел слева выберите **Shaft**, в поле **Исследуемая величина** выберите **Угл. скорость**, а в поле **Компонента** выберите **Y**. Создайте переменную кнопкой . Переменная **om:y(Shaft)** появится в контейнере переменных.

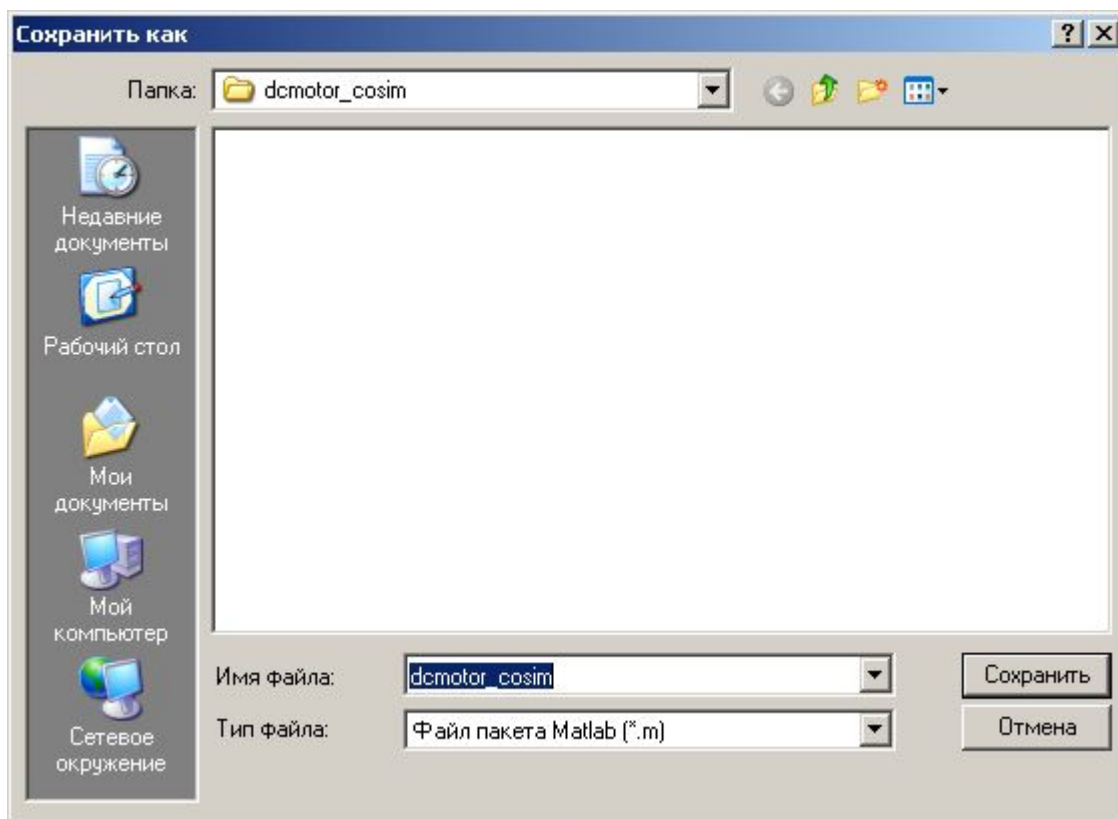


5. Перетащите мышкой созданную переменную **om:y(Shaft)** в окно **Мастера экспорта в Matlab/Simulink** и назначьте ее в качестве выходной величины. Выход модели станет помечен зеленой галочкой, что означает, что на данный выход S-функции назначена переменная.



Сохранение управляющего m-файла

1. Нажмите кнопку  для вызова диалога сохранения m-файла. При этом появится предупреждение о том, что список параметров модели пуст. Это сообщение нужно проигнорировать и продолжить работу, нажав на кнопку **Yes**.
2. В появившемся окне диалога сохранения файла необходимо указать в названии файла **dcmotor_cosim** и сохранить его.

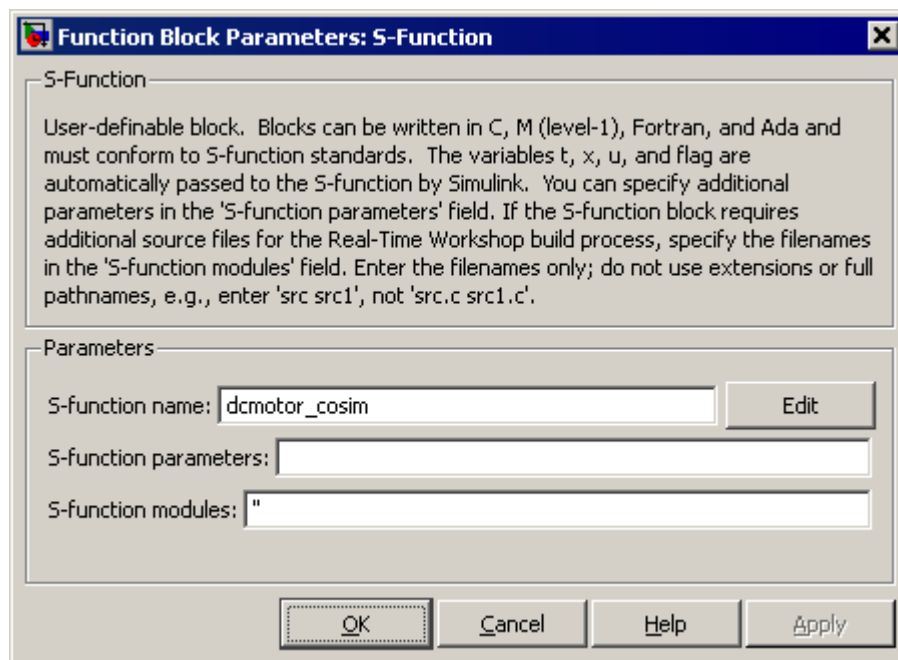


На этом действия на стороне UM закончены. Переходим к описанию связи на стороне Matlab/Simulink.

4.3.2. Создание связи между UM-моделью и Matlab/Simulink

Для создания связи между системой управления Matlab/Simulink и механической моделью выполните следующие действия.

1. Запустите **Matlab**
2. В поле **Current Directory** выберите каталог **{um_root}\samples\cosimulation\ dcmotor_cosim**.
3. Откройте файл **dcmotor_cntr.mdl**, в котором хранится модель системы управления двигателем постоянного тока.
4. Двойным щелчком мыши вызовите окно параметров S-функции.
5. Убедитесь, что в поле **S-function name** установлено **dcmotor_cosim**, что соответствует названию m-файла.



6. Нажмите кнопку **ОК**.

Модель готова для дальнейшего анализа.

4.3.3. Моделирование движения

Моделирование динамики системы

1. Двойным щелчком мыши откройте окна **Scope**, в которых будут отображаться угловая скорость вала и величина вращательного момента.
2. Запустите модель на выполнение.
3. По истечении времени моделирования вы увидите следующие результаты, см. рис. ниже.

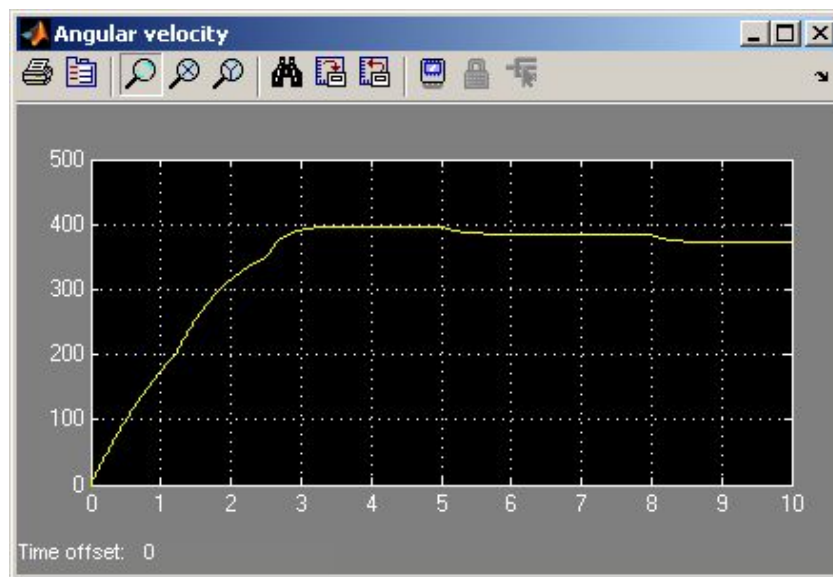


Рис.4.8. Изменение угловой скорости вала

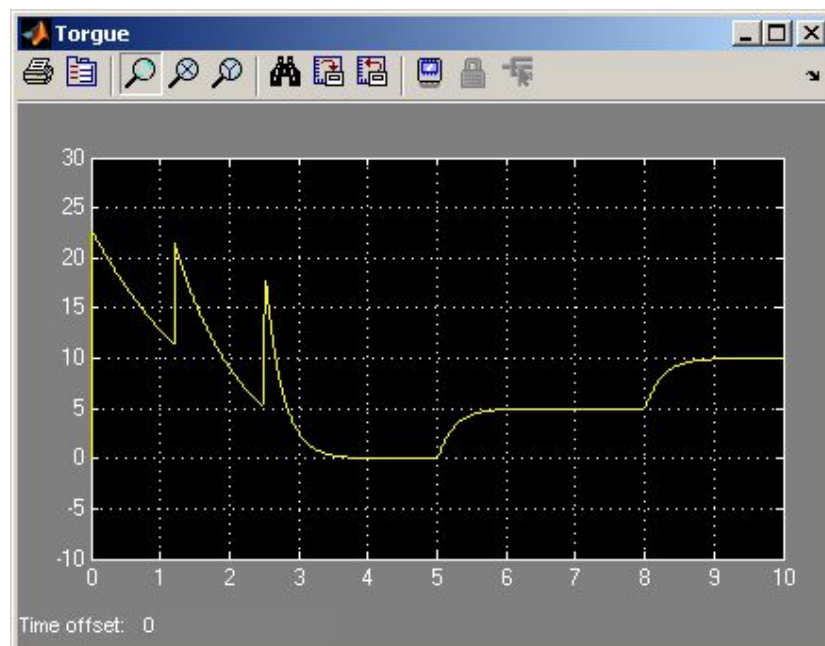


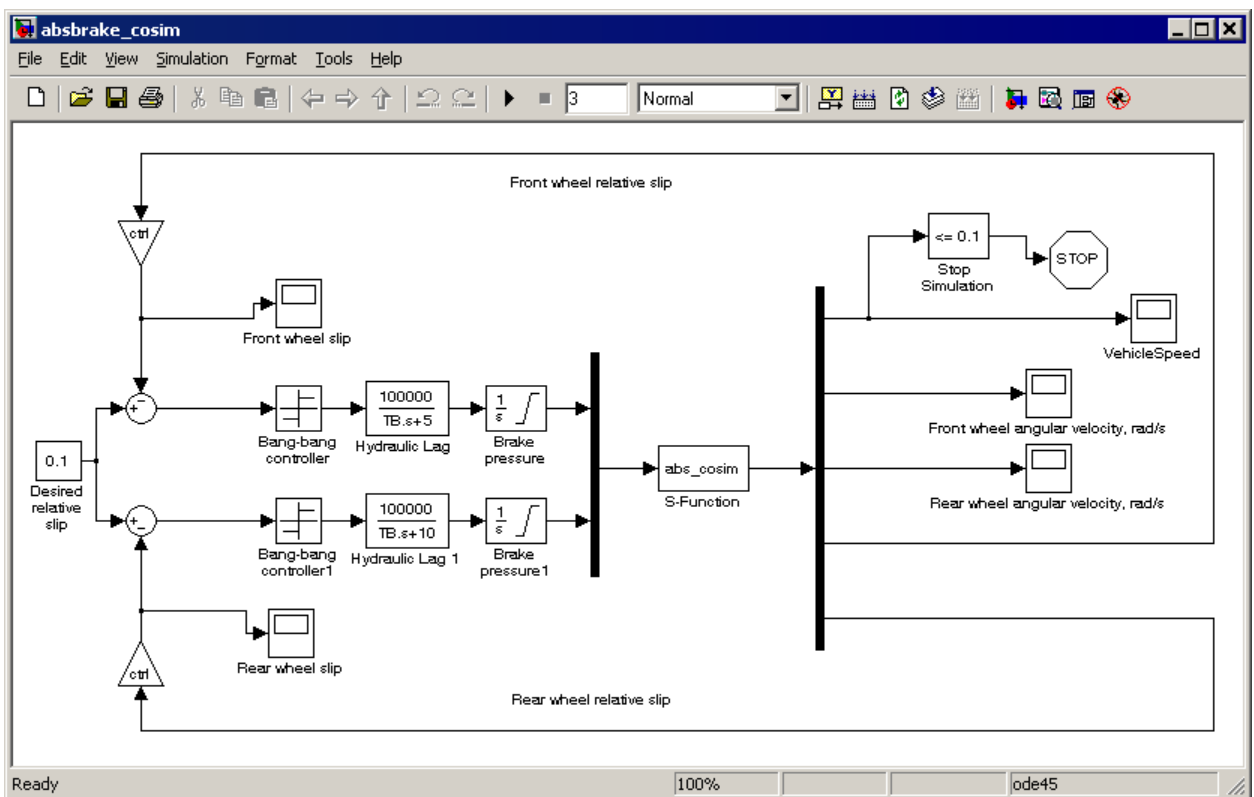
Рис.4.9. Изменение активного электромагнитного момента на валу

4.4. Антиблокировочная система автомобиля (ABS)

Модель автомобиля с готовыми файлами настроек, управляющим m-файлом и моделью Matlab/Simulink находится в каталоге {um_root}\samples\cosimulation\vaz21_09abs. Перед началом урока, проверьте ее наличие в указанном каталоге. Если такой модели нет, то скачайте ее из интернета по адресу: http://www.umlabor.ru/download/60/vaz21_09abs.zip. В данном уроке мы не будем подробно разбирать этапы создания модели механической части, ограничимся только описанием связи механической части с системой управления Matlab/Simulink.

4.4.1. Подготовка системы управления в Matlab/Simulink

Электронный контроль системы ABS позволяет достигать гораздо лучшего торможения в аварийных ситуациях. Главной задачей системы ABS является предотвращение полной блокировки колес и тем самым обеспечение максимально возможного сцепления колес с дорогой и сохранения управляемости, в том числе на скользкой и мокрой дороге и гравии. За счет этого сохраняется управляемость автомобиля при любых состояниях дорожного покрытия. Простая модель ABS с поосным регулированием приведена на рисунке ниже.




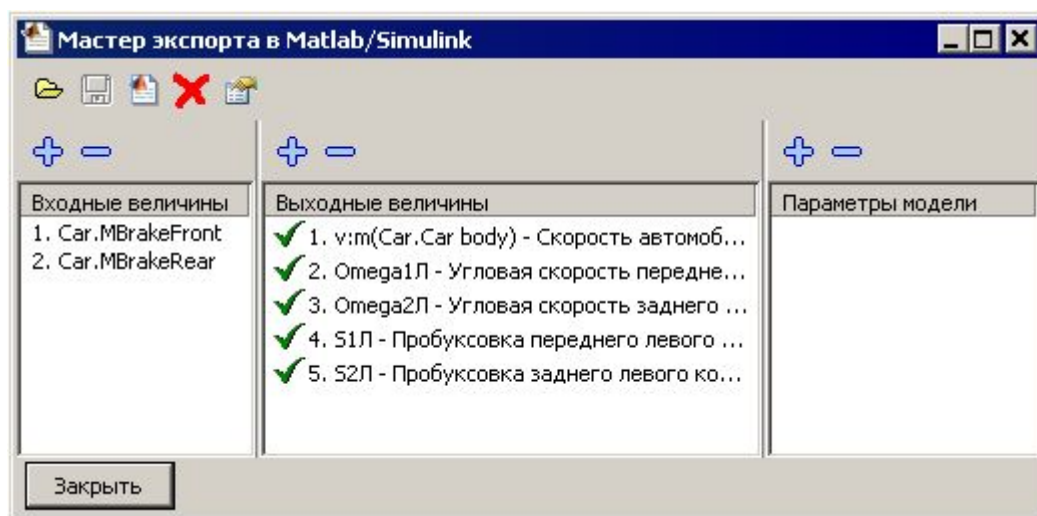
В нашей модели для организации управления автомобилем входными переменными для S-функции будут тормозные усилия, подаваемые на переднюю и заднюю оси автомобиля. В свою очередь, выходными величинами S-функции будут являться скорость движения автомобиля, угловые скорости передних и задних колес, пробуксовка на передних и задних колесах и давление, подаваемое на передние колеса.

4.4.1. Экспорт модели механической системы из UM

Не будем подробно описывать создание и сохранением настроек модели автомобиля, ограничимся лишь копиями экранов для иллюстрации проделанной работы.

Окно **Мастера экспорта в Matlab/Simulink** со списками входных и выходных величин представлено ниже. Для моделирования динамики в среде Matlab/Simulink необходимо заново сгенерировать управляющий m-файл.

1. Запустите **UM Simulation**.
2. Загрузите модель {um_root}\samples\cosimulation\vaz21_09abs.
3. Откройте окно **Мастера экспорта в Matlab/Simulink** (меню **Инструменты/Мастер экспорта в Matlab/Simulink**).
4. В **Мастере экспорта в Matlab/Simulink** загрузите файл **abs_cosim.cosim**.
5. Ответьте «Да» на приглашение загрузить дополнительные файлы настроек.
6. Создайте заново m-файл кликнув по кнопке .



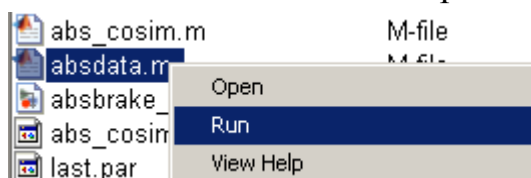
Сохранив управляющий m-файл, а вместе с ним и файлы, можно переходить к моделированию работы ABS.

4.4.2. Моделирование движения

Рассмотрим торможение автомобиля сначала с включенной, а затем с выключенной ABS. Ниже приведены зависимости скорости движения автомобиля, угловых скоростей передних и задних колес, пробуксовки передних и задних колес от времени, от времени.

4.4.2.1. Загрузка модели

1. Запустите **Matlab**
2. В поле **Current Directory** выберите каталог **{um_root}\samples\cosimulation\vaz21_09abs**.
3. Выполните инициализацию констант системы ABS. В списке файлов выделите **absdata.m** и в контекстном меню выберите команду **Run**.



4. Откройте файл **absbrake_cosim.mdl**, в котором хранится модель системы управления антиблокировочной системой.
5. Запустите моделирование.

4.4.2.2. Результаты моделирования движения с ABS

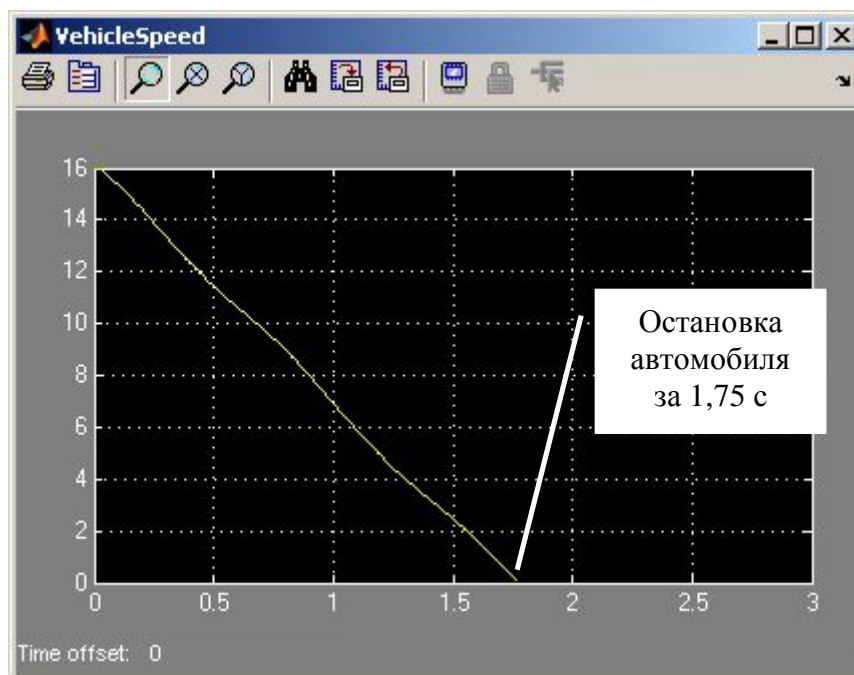


Рис.4.10. Скорость автомобиля с включенной ABS

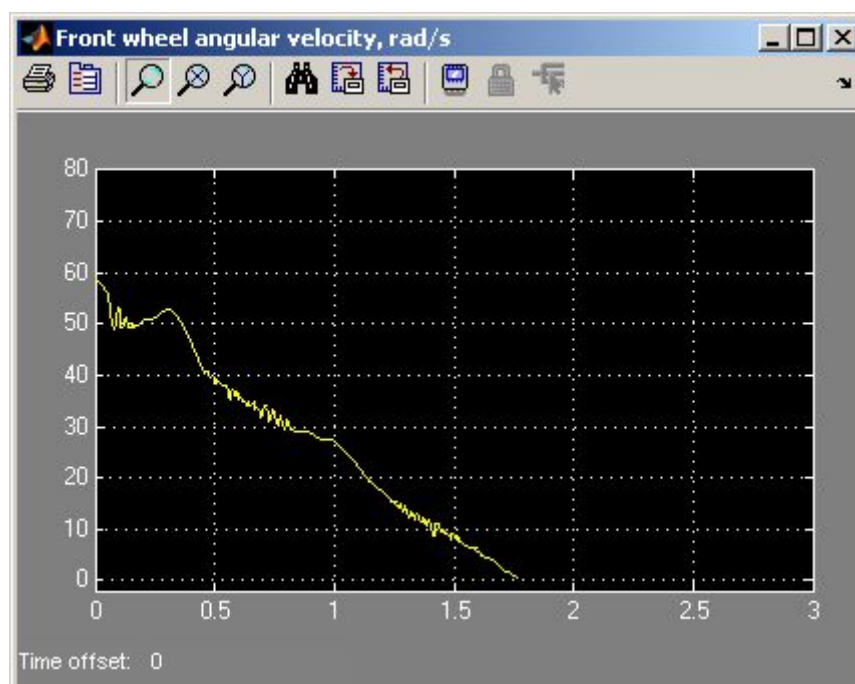


Рис.4.11. Угловая скорость переднего колеса

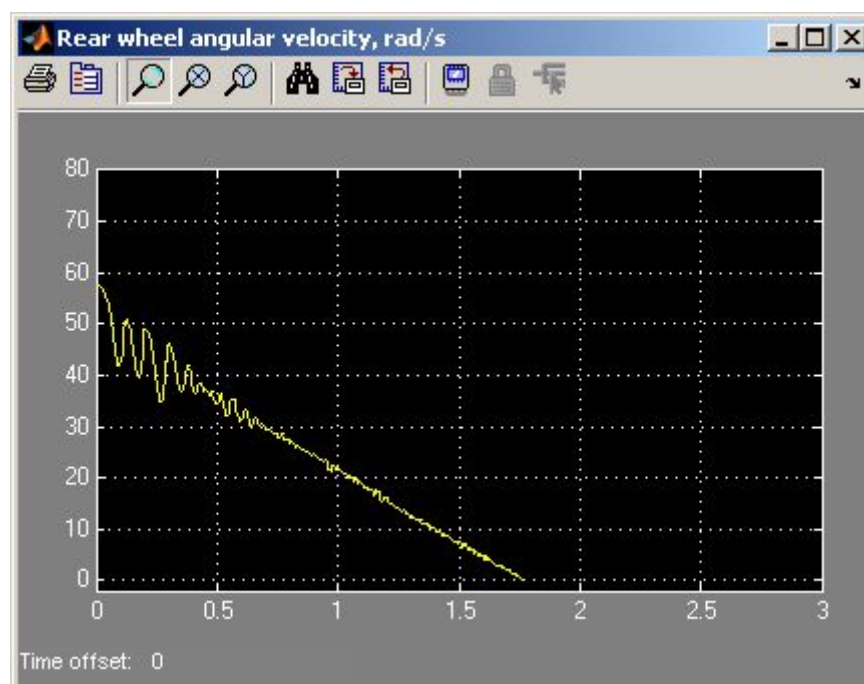


Рис.4.12. Угловая скорость заднего колеса

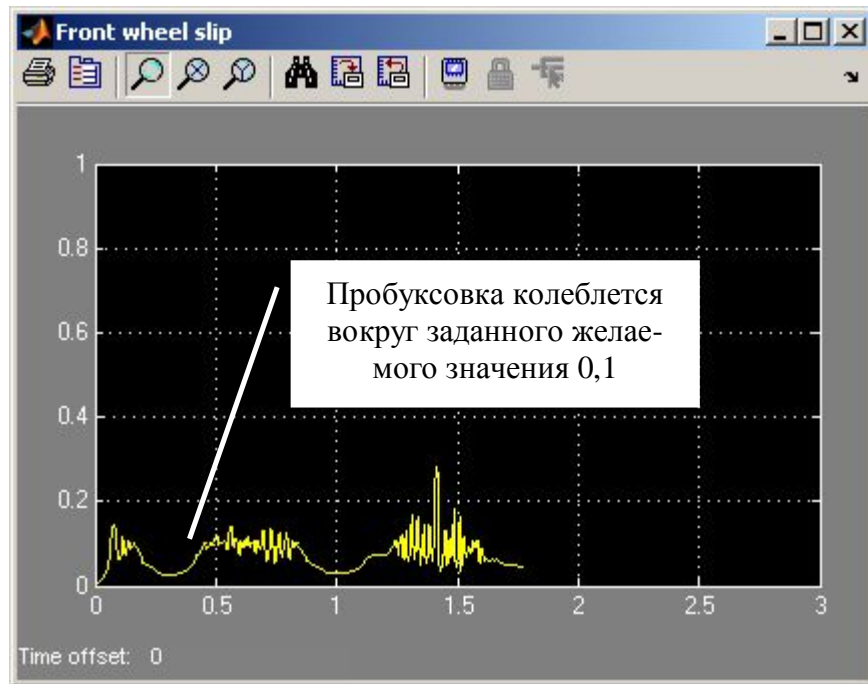


Рис.4.13. Пробуксовка передних колес

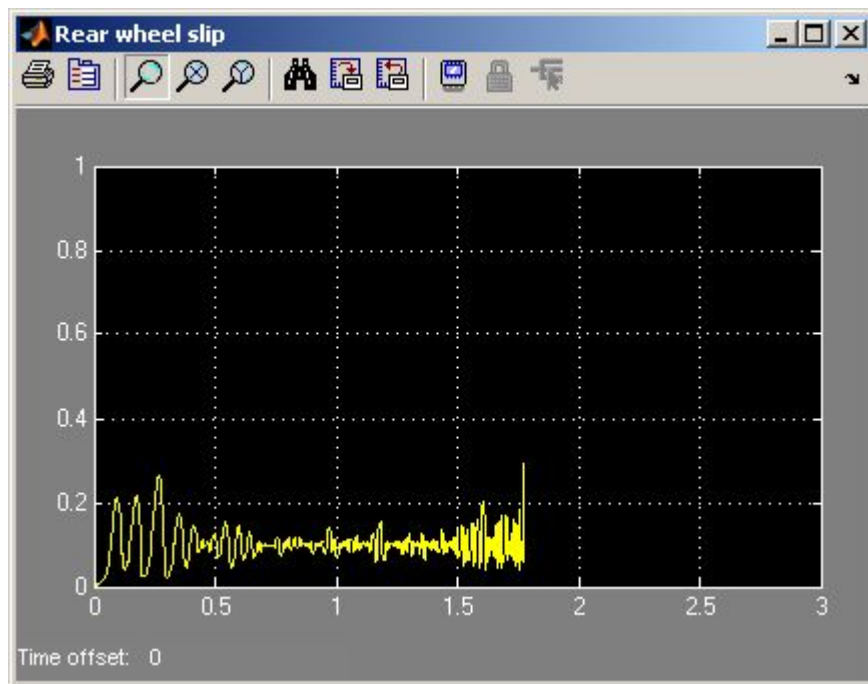


Рис.4.14. Пробуксовка задних колес

4.4.2.1. Результаты моделирования движения без ABS

При отключенной системе ABS графики зависимости представленных выше величин от времени имеют следующий вид.

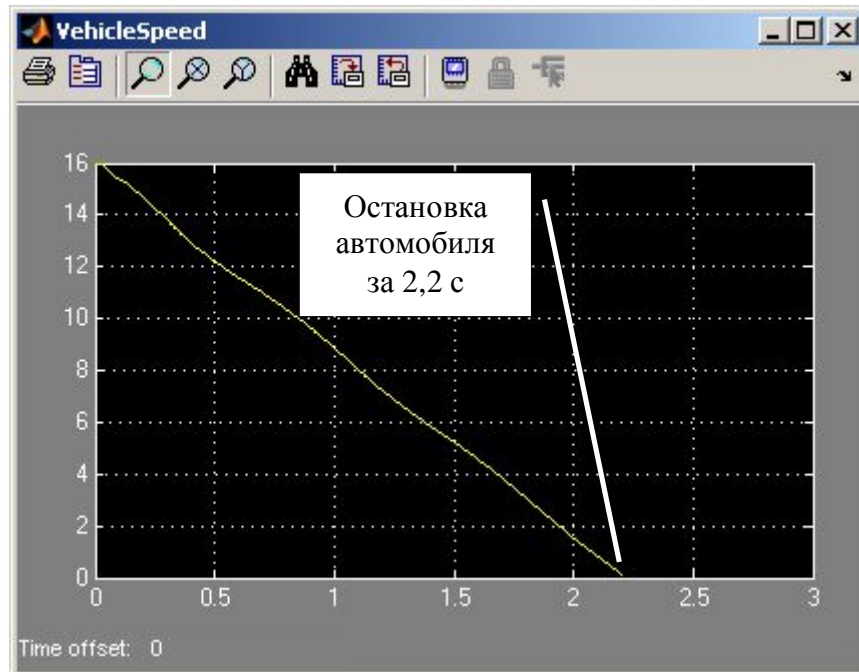


Рис.4.15. Скорость автомобиля с выключенной ABS

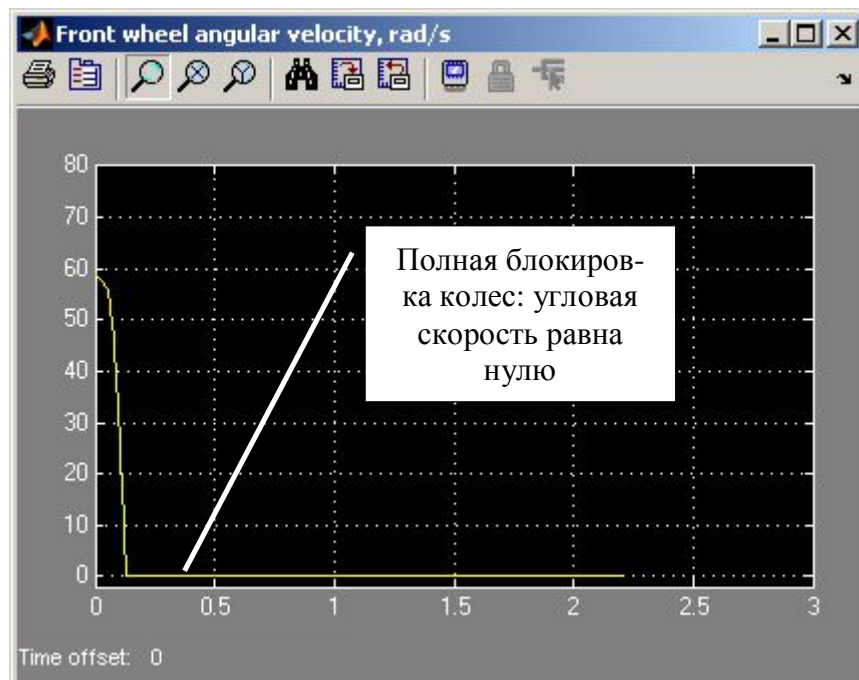


Рис.4.16. Угловая скорость переднего колеса

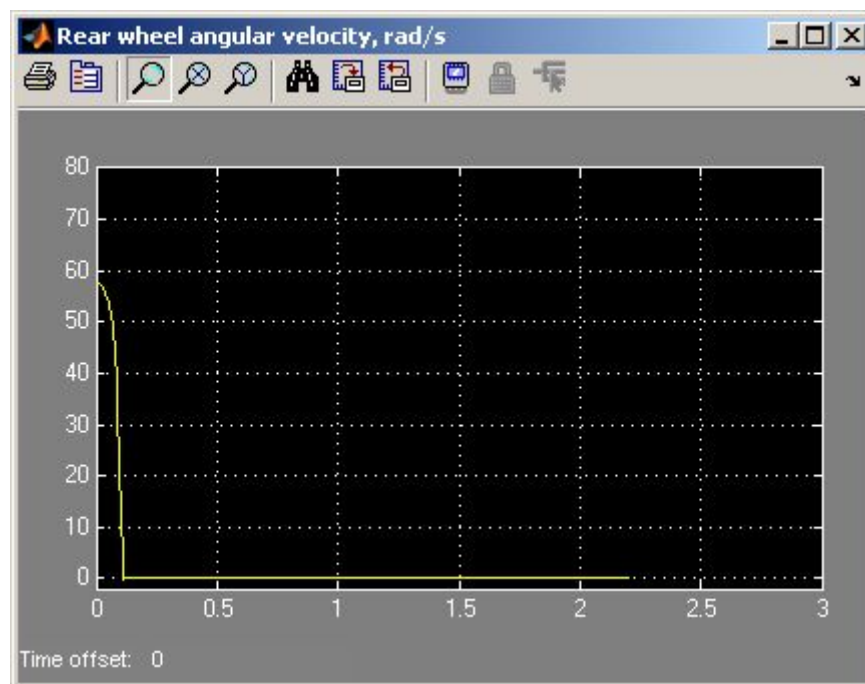


Рис.4.17. Угловая скорость заднего колеса

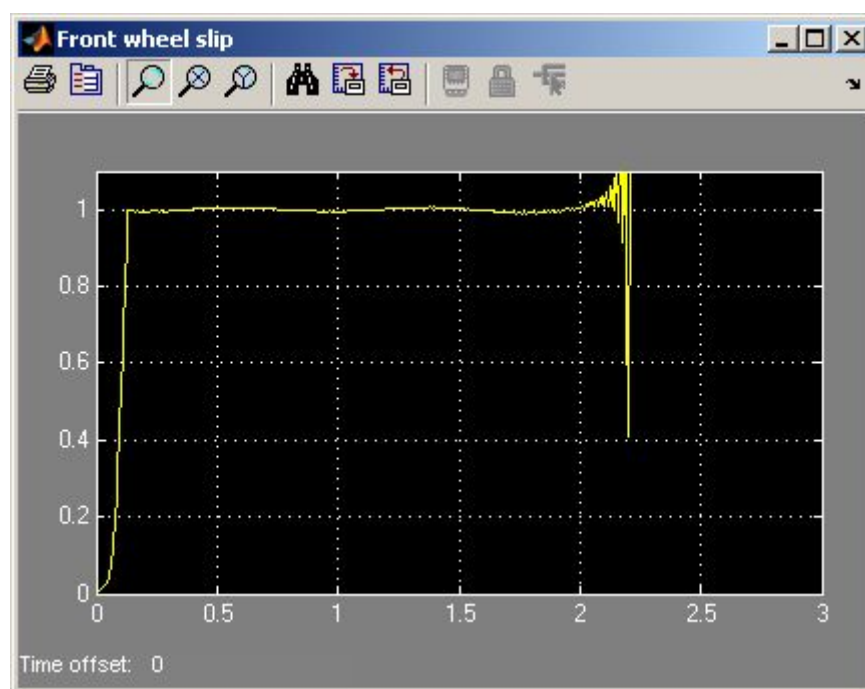


Рис.4.18. Пробуксовка передних колес

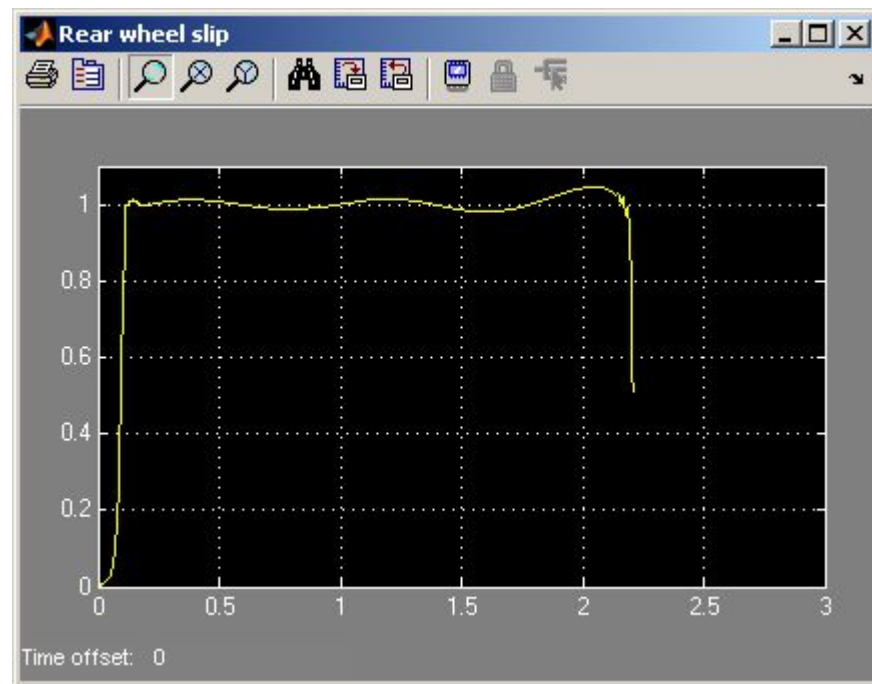


Рис.4.19. Пробуксовка задних колес

При отключенной системе ABS остановка автомобиля происходит за больший промежуток времени: 2,2 с против 1,75 с. Из графиков зависимости угловых скоростей колес автомобиля от времени видно, что при отключенной ABS угловая скорость колес падает, они блокируются, и автомобиль начинает скользить по дороге с полностью заблокированными колесами, что увеличивает тормозной путь.