Лектор: к.т.н., доцент ОЭЭ ИШЭ ТПУ Воронина Н.А.

Дисциплина:
Компьютерное моделирование электротехнических устройств, комплексов и систем

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

- Simulink-функции (S-функции, S-functions) являются описанием блока на одном из языков программирования: MATLAB, C, C++, Ada, или Fortran.
- С помощью языков программирования пользователь может создать описание сколь угодно сложного блока и подключить его к Simulink-модели, при этом, с точки зрения взаимодействия пользователя с моделью, блок на основе S-функции ничем не отличается от стандартного библиотечного блока Simulink.
- Создаваемые блоки могут быть непрерывными, дискретными или гибридными.
- S-функции, созданные на C, C++, Ada или Fortran, компилируются в исполняемые (*.dll) файлы, за счет чего обеспечивается повышенная скорость выполнения таких блоков.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

- S-функции обладают дополнительными возможностями, которые включают работу с разными типами данных (целыми, действительными и комплексными числами различной степени точности), использование матриц в качестве входных и выходных переменных (MATLAB S-функции могут оперировать только векторами в качестве входных и выходных переменных), а также больший набор внутренних функций (callback-методов).
- S-функции используются при создании новых библиотечных блоков, обеспечивающих взаимодействие Simulink с аппаратными средствами компьютера, при создании блоков на основе математических уравнений, блоков реализующих анимационные возможности MATLAB, а также при подключении к модели Simulink существующего программного кода языков высокого уровня.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

S-функция подключается к модели Simulink с помощью библиотечного блока S-Function (библиотека Functions & Tables)

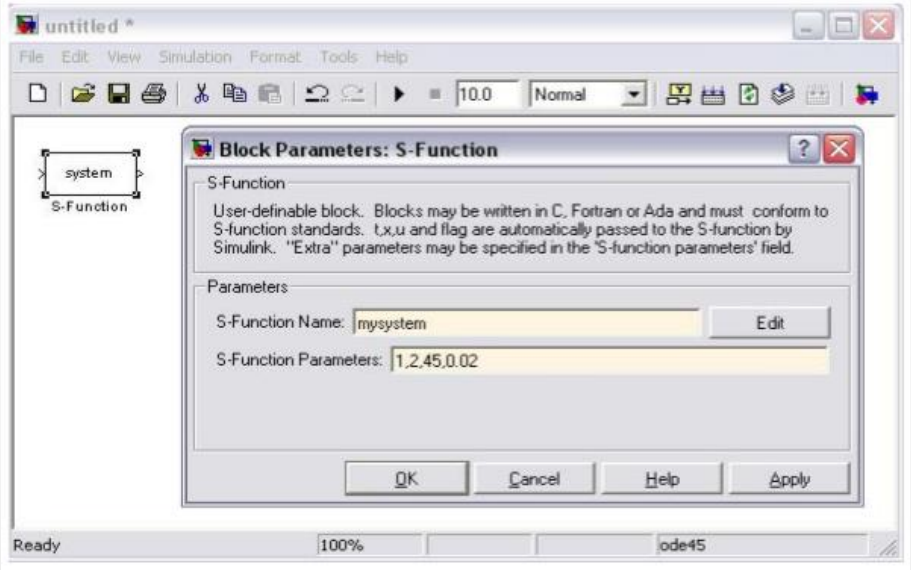
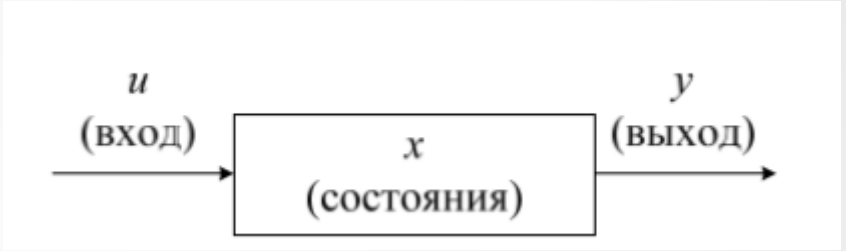


Рис.1 Окно модели с блоком S-function и его окно диалога

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Математическое описание S-функции:

Simulink-блок описывается наборами входных переменных *u*, переменных состояния *x* и выходных переменных *y*.



- Рис.2 Общий вид Simulink-блока

В математической форме блок можно описать в общем виде следующей системой уравнений:

$$y = f_0(t,x,u) \text{ (выходы)}$$

$$\dot{x}_c = f_d(t,x,u) \text{ (производные непрерывных переменных состояния)}$$

$$x_{d_{k+1}} = f_u(t,x,u) \text{ (дискретные переменные состояния), где } x = x_c + x_d.$$

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Этапы моделирования :

На первом этапе:

- выполняется инициализация модели:
  - ✓ подключение библиотечных блоков к модели,
  - ✓ определение размерностей сигналов, типов данных, величин шагов модельного времени,
  - ✓ оценка параметров блоков,
  - ✓ определяется порядок выполнения блоков,
  - ✓ выполняется выделение памяти для проведения расчета.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Этапы моделирования :

На втором этапе:

- выполняется цикл моделирования

✓ на каждом цикле моделирования (временном шаге) происходит расчет блоков в порядке, определенном на этапе инициализации;

✓ для каждого блока Simulink вызывает функции, которые вычисляют переменные состояния блока *x*, производные переменных состояния и выходы *y* в течение текущего шага модельного времени

✓ процесс продолжается пока моделирование не будет завершено.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

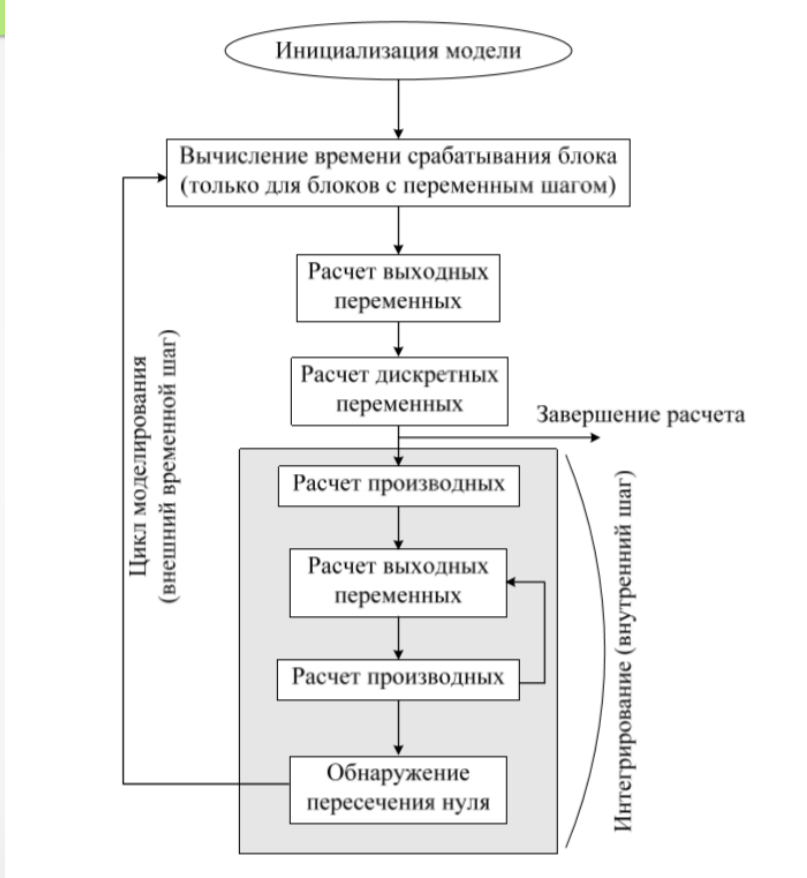


Рис. 3. Диаграмма, иллюстрирующая процесс моделирования

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Callback-методы S-функции:

Каждая задача при вызове S-функции в процессе моделирования решается с помощью специальной внутренней функции (callback-метода).

используются следующие методы:

1. mdlInitializesizes – инициализация.

В течение этого этапа Simulink:

– инициализирует структуру с именем SimStruct, содержащую информацию S-функции;

– устанавливает количество и размерность входных и выходных портов;

– устанавливает шаг модельного времени для блока;

– выделяет память для хранения переменных и устанавливает размерность массивов.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Callback-методы S-функции:

2. mdlGetTimeOfNextVarHit – вычисление времени следующего срабатывания блока (для блоков с дискретным переменным шагом расчета)
3. mdlOutputs – вычисление значений выходных сигналов на внешнем шаге моделирования (рассчитанные выходные сигналы блока передаются на его выходные порты).

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Callback-методы S-функции:

4. mdlUpdate – расчет дискретных переменных состояния на внешнем шаге моделирования. Дискретные переменные состояния сохраняют свое значение до следующего цикла моделирования.
5. mdlDerivatives – расчет производных переменных состояния.
6. mdlTerminate – завершение работы S-функции.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Основные понятия S-функции :

Для того чтобы создать S-функцию правильно, необходимо определить основные понятия, используемые в технологии создания S-функций.

К ним относятся:

- Direct feedthrough – прямой вход. Проход входных сигналов на выход. Установка правильного значения параметра Direct feedthrough очень важна, поскольку именно с помощь него Simulink определяет наличие в модели замкнутых алгебраических контуров.
- Dynamically sized inputs – динамическая размерность входов. Чтобы задать динамическую размерность какой-либо переменной, нужно задать значение размерности для этой переменной равное -1 в соответствующем поле структуры sizes.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

- Setting sample times and offsets - установка шагов модельного времени и смещений. Simulink обеспечивает следующие варианты задания шага модельного времени:
- Continuous sample time – непрерывное модельное время.
- Continuous but fixed in minor time step sample time – непрерывное модельное время с фиксированным шагом во внутреннем цикле.
- Discrete sample time – дискретное модельное время. Пользователь должен задать шаг модельного времени sample time и смещение (задержку) offset, чтобы определить моменты времени, в которые Simulink должен вызвать на выполнение данный блок. Время срабатывания блока определяется выражением: TimeHit = (n * sample_time) + offset, где n – целое число шагов расчета. Если задано дискретное модельное время, то Simulink обращается к методам mdlUpdate и mdlOutputs на каждом внешнем шаге моделирования.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

- Variable sample time – дискретный переменный шаг расчета. В начале каждого шага моделирования S-функция должна определить значение времени следующего срабатывания. Для этого используется mdlGetTimeOfNextVarHit метод.
- Inherited sample time – наследуемый шаг расчета. Например, для блока Gain не имеет значения, какой шаг модельного времени реализован – блок выполняет усиление входного сигнала для любого варианта sample time. В подобных случаях параметр sample time может быть унаследован от предыдущего или последующего блока, либо от блока, имеющего наименьший шаг расчета.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Создание S-функций на языке MATLAB

Наиболее простой и быстрый путь создать S-функцию – это написать ее на языке MATLAB с использованием файла-шаблона.

Создание S-функций на языке MATLAB имеет некоторые ограничения (например, MATLAB S-функция может иметь только по одному входному и выходному порту, а также передаваемые и принимаемые данные через эти порты могут быть только скалярами и векторами типа double), этот способ является наилучшим с точки зрения изучения механизма работы S-функции. Далее приводится шаблон S-функции. Оригинальный файл шаблона sfuntmpl.m находится в папке ...\toolbox\simulink\blocks.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Шаблон MATLAB S-функции:

SFUNTMPL – Базовый шаблон для создания MATLAB S-функции.

С помощью MATLAB S-функции пользователь может задать систему обыкновенных дифференциальных уравнений (ODE), уравнения дискретной системы, и (или) любой алгоритм, описывающий работу Simulink-блока. Базовая форма синтаксиса S-функции выглядит следующим образом:

$[sys, x0, str, ts] = sfunc(t, x, u, flag, p_1, ..., p_n)$

Параметры *S*-функции:

t – текущее время;

x – вектор переменных состояния системы;

u – вектор входных сигналов;

flag – флаг – целое число, определяющее, какая функция внутри *S*-функции выполняется при вызове.

$p_1, ..., p_n$ – параметры *S*-функции, задаваемые в окне диалога блока “*S-function*”.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Шаблон MATLAB S-функции:

Результат, возвращаемый (вычисляемый) S-функцией в момент времени *t*, зависит от значения переменной *flag*, значения вектора состояния системы *x* и текущего значения вектора входного сигнала *u*.

Результаты работы S-функции в зависимости от значения переменной *flag* приведены в таблице 1.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

flag	РЕЗУЛЬТАТ	ВЫПОЛНЯЕМАЯ ФУНКЦИЯ	ОПИСАНИЕ
0	[*sizes*, *x*0, *str*,*ts*]	*mdlInitializesizes*	Инициализация: расчет начальных условий, значений вектора модельного времени времени, размерности матриц
1	*dx*	*mdlDerivatives*	Расчет значений производных вектора *x* состояния системы
2	*ds*	*mdlUpdate*	Расчет значений вектора состояний *x* дискретной системы: *sys* = *x*(*n*+*1*)
3	*y*	*mdlOutputs*	Расчет значений выходного вектора *sys*
4	*Tnext*	*mdlGetTimeOfNextVarHit*	Расчет значения времени для следующей расчетной точки дискретной части системы. Используется при расчете дискретной или гибридной системы с переменным шагом.
5			Зарезервировано для будущего использования.
9	[]	*mdlTerminate*	Завершение расчета

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Параметры блока "S-function" p_1,..., p_n передаются в S-функцию при любом значении переменной flag. При вызове S-функции со значением переменной *flag* = 0 ею рассчитываются следующие величины:

sys(1) – число непрерывных переменных состояния;

sys(2) – число дискретных переменных состояния;

sys(3) – число выходных переменных;

sys(4) – число входных переменных (размерность вектора *u*).

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Любой из первых четырех элементов в sys может быть задан равным -1 (минус один), что означает динамически задаваемую размерность соответствующих переменных. Фактическая размерность при вызове S-функции с другими значениями переменной flag будет равна размерности входного вектора *u*. Например, при sys(3) = -1, размерность выходного вектора будет задана равной размерности входного вектора.

sys(5) – значение зарезервировано «для будущего использования».

sys(6) – значение, равное 1, соответствует прохождению входного сигнала на выход. Значение, равное 0, определяет, что входной сигнал не используется для вычисления выходного сигнала в функции mdlOutputs, вызываемой при значении переменной flag = 3.

sys(7) – размерность вектора модельного времени (количество строк в матрице ts).

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

- *x*0 – задание вектора начальных значений переменных состояния. Если переменных состояния нет – значение параметра задается равным [].
- *str* – зарезервированный параметр. Значение параметра задается равным [].
- *ts* – матрица, размерностью (m×2), задающая модельное время и смещение (*period* и *offset*), где *m* – число строк в матрице *ts*. Если в матрице *ts* заданы несколько значений шага модельного времени, то его значения должны располагаться в порядке возрастания. В том случае, если задано более одного значения шага, требуется проверка времени срабатывания блока в явном виде:

$$abs(round((T - OFFSET) / PERIOD) - (T - OFFSET) / PERIOD).$$

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Обычно задаваемая погрешность при проверке равна 1*e*-8. Она зависит от величин шага модельного времени и времени окончания расчета. Можно также задать, чтобы значение временного шага передавалось в блок «S-Function» из предшествующего блока модели. Для функций, изменяющихся внутри основного временного шага, должно быть задано sys(7) = 1 и *ts* = [-1 0]. Для функций, не изменяющихся внутри основного временного шага, должно быть задано sys(7) = 1 и *ts* = [-1 1]. Ниже лежащие строки показывают базовую структуру S-функции:

```
function [sys,x0,str,ts] = sfuntmpl(t,x,u,flag) 1
switch flag,
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

В зависимости от значения переменной flag происходит вызов того или иного метода:

Инициализация.
case 0, [sys,x0,str,ts]=mdlInitializeSizes;

Расчет производных.
case 1, sys=mdlDerivatives(t,x,u);
Расчет значений вектора состояний дискретной части системы.
case 2, sys=mdlUpdate(t,x,u);

Расчет значений вектора выходных сигналов непрерывной части системы.
case 3, sys=mdlOutputs(t,x,u);

Расчет значения времени для следующей расчетной точки дискретной части системы.
case 4, sys=mdlGetTimeOfNextVarHit(t,x,u);

Завершение расчета.
case 9, sys=mdlTerminate(t,x,u);

Неизвестное значение переменной flag.
otherwise error(['Unhandled flag = ',num2str(flag)]); end

Окончание функции sfuntmpl.
mdlInitializeSizes – функция инициализации: расчет начальных условий, значений вектора модельного времени, размерности матриц.
function [sys,x0,str,ts]=mdlInitializeSizes

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Приведенные значения параметров даны в качестве примера и не отражают реально задаваемых значений:

```
sizessizes.NumContStates = 0;

Число непрерывных переменных состояния.
sizes.NumDiscStates = 0;

Число дискретных переменных состояния.
sizes.NumOutputs = 0;

Число выходных переменных (размерность выходного вектора).
sizes.NumInputs = 0;

Число входных переменных (размерность вектора u).
sizes.DirFeedthrough = 1;
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Параметр, задающий входной сигнала на выход. Этот параметр нужно задавать равным 1, в том случае, если входной сигнал прямо или опосредованно (например, через логическое выражение или алгебраическую операцию) проходит на выход системы (иными словами: если входной сигнал *u*, входит в выражения задаваемые в функции mdlOutputs) или используется метод mdlGetTimeOfNextVarHit. При описании системы в уравнениях пространства состояний этот параметр следует задать равным 1, если матрица D (матрица обхода) не пустая и равным нулю, в противном случае

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

sys = simsizes(sizes);
Второй вызов функции simsizes. Данные о размерностях передаются в *Simulink.*

x0 = [];
Задание вектора начальных значений переменных состояния (начальных условий).

str = [];
Задание параметра str, как пустой матрицы. Параметр заразервирован для будущего использования.

ts = [0 0];
Матрица из двух колонок, задающая шаг модельного времени и смещение.
Окончание mdlInitializeSizes.
mdlDerivatives – функция для расчета значений производных вектора состояния непрерывной части системы.

function sys=mdlDerivatives(t,x,u)
sys = [];
Окончание mdlDerivatives.
mdlUpdate – функция для расчета значений вектора выходных сигналов непрерывной части системы.

function sys=mdlUpdate(t,x,u)
sys = [];
Окончание mdlUpdate.
mdlOutputs – функция для расчета значений вектора выходных сигналов непрерывной части системы.

function sys=mdlOutputs(t,x,u)
sys = [];
Окончание mdlOutputs.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

mdlGetTimeOfNextVarHit – расчет значения времени для следующей расчетной точки дискретной части системы. Функция рассчитывает время (абсолютное значение), по достижении которого значения дискретной части системы передаются в Simulink-модель. Функция используется только в случае моделирования дискретной части системы с переменным шагом (variable discrete-time sample time).

В этом случае параметр *ts* функции mdlInitializeSizes должен быть задан как [-2 0].

```
function sys=mdlGetTimeOfNextVarHit(t,x,u)
sampleTime = 1;

Пример: время срабатывания блока увеличивается на 1 секунду.
sys = t + sampleTime;
Окончание mdlGetTimeOfNextVarHit.
mdlTerminate – функция, выполняющая завершение расчета.
function sys=mdlTerminate(t,x,u)
sys = [];
Окончание mdlTerminate.
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Примеры S-функций языке MATLAB:

Простейшая S-функция

Одним из самых простых примеров S-функций, поставляемых с пакетом MATLAB является функция timestwo. Данная S-функция выполняет умножение входного сигнала на коэффициент. Ниже приведен текст этой S-функции.

function [sys,x0,str,ts] = timestwo(t,x,u,flag) [2]

TIMESTWO – Пример *S*-функции.

Выходной сигнал равен входному, умноженному на 2: $y = 2*u$.

switch flag,

В зависимости от значения переменной *flag* происходит вызов того или иного метода:

case 0 [sys,x0,str,ts]=mdlInitializeSizes;

Расчет значений вектора выходных сигналов.

case 3 sys=mdlOutputs(t,x,u);

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Примеры S-функций языке MATLAB:

Простейшая S-функция

В примере не используются методы для завершения работы S-функции, нет непрерывных и дискретных переменных состояния, поэтому значения переменной flag = 1, 2, 4, 9 не используются. Результатом S-функции в этом случае является пустая матрица

```
case { 1, 2, 4, 9 } sys=[];
otherwise error(['Unhandled flag = ',num2str(flag)]); end
```

Окончание функции timestwo.

mdlInitializeSizes – функция инициализации: расчет начальных условий, значений вектора шагов модельного времени, размерности матриц.

```
function [sys,x0,str,ts] = mdlInitializeSizes()
sizes = simsizes;
sizes.NumContStates = 0;
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Примеры S-функций языке MATLAB:

Простейшая S-функция

В примере не используются методы для завершения работы S-функции, нет непрерывных и дискретных переменных состояния, поэтому значения переменной flag = 1, 2, 4, 9 не используются. Результатом S-функции в этом случае является пустая матрица

```
case { 1, 2, 4, 9 } sys=[];
otherwise error(['Unhandled flag = ',num2str(flag)]); end
```

Окончание функции timestwo.

mdlInitializeSizes – функция инициализации: расчет начальных условий, значений вектора шагов модельного времени, размерности матриц.

```
function [sys,x0,str,ts] = mdlInitializeSizes()
sizes = simsizes;
sizes.NumContStates = 0;
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

sizes.NumDiscStates = 0;
Число дискретных переменных состояния.

sizes.NumOutputs = -1;
Число выходных переменных (размерность выходного вектора).
Динамическая размерность выходного вектора.

sizes.NumInputs = -1;
Число входных переменных (размерность входного вектора).
Динамическая размерность входного вектора.

sizes.DirFeedthrough = 1;
Прямой проход. Есть проход входного сигнала на выход.

sizes.NumSampleTimes = 1;
Размерность вектора шагов модельного времени.

sys = simsizes(sizes);
str = [];
Параметр заразервирован для будущего использования.

x0 = [];
Задание вектора начальных значений переменных состояния. Переменных состояния нет, поэтому значение параметра – пустая матрица.

ts = [-1 0];
Матрица из двух колонок, задающая шаг модельного времени и смещение.
Шаг наследуется из предшествующего блока.
Окончание mdlInitializeSizes.
mdlOutputs – функция для расчета значений вектора выходных сигналов.
function sys = mdlOutputs(t,x,u)
sys = u*2;
Выходной сигнал блока есть входной сигнал, умноженный на коэффициент 2.
Окончание mdlOutputs.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Модель непрерывной системы:

Данная S-функция моделирует непрерывную систему с двумя входами, двумя выходами и двумя переменными состояния. Параметры модели (значения матриц A, B, C, D) задаются в теле S-функции и передаются в callback-методы через их заголовки в качестве дополнительных параметров. Ниже приведен текст этой S-функции.

function [sys,x0,str,ts] = csfunc(t,x,u,flag) [3]

С помощью уравнений пространства состояния моделируется непрерывная система:

$x' = Ax + Bu$

$y = Cx + Du$

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Модель непрерывной системы:

Значения матриц передаются в callback-методы через их заголовки в качестве дополнительных параметров. Задание матриц:

Матрица системы.
```
A=[-0.09 -0.01
        1 0];
```

Матрица входа.
```
B=[ 1 -7
        0 -2];
```

Матрица выхода.
```
C=[0 2
        1 -5];
```

Матрица обхода.
```
D=[-3 0
        1 0];
```

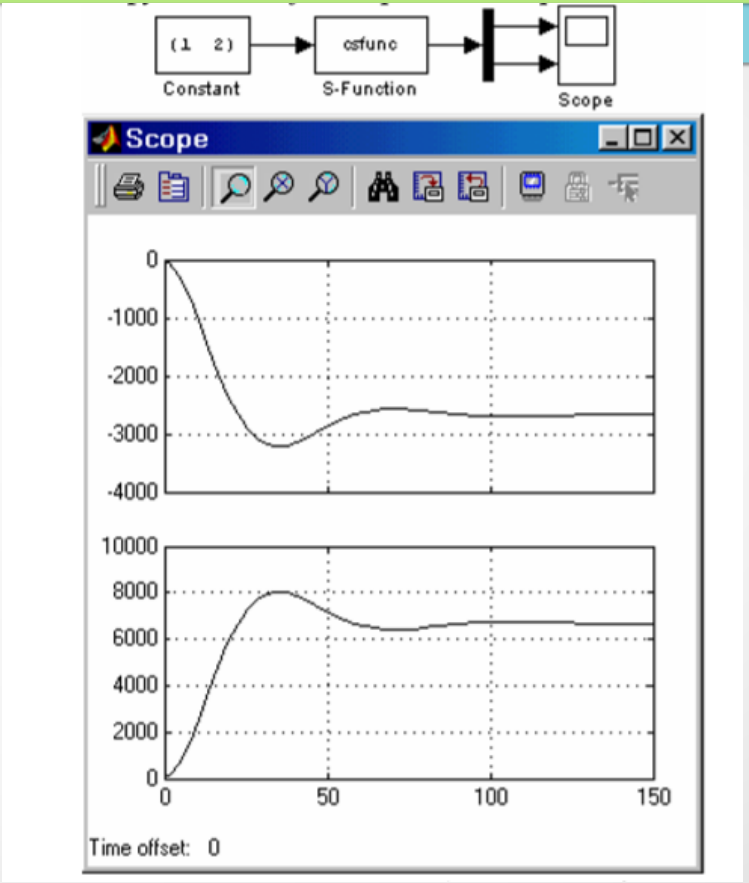
```
switch flag,

case 0, [sys,x0,str,ts]=mdlInitializeSizes(A,B,C,D);

case 1, sys=mdlDerivatives(t,x,u,A,B,C,D);

case 3, sys=mdlOutputs(t,x,u,A,B,C,D);

case { 2, 4, 9 }

sys=[];

otherwise error(['Unhandled flag = ',num2str(flag)]); end

function [sys,x0,str,ts]=mdlInitializeSizes(A,B,C,D)

sizes = simsizes;

sizes.NumContStates = 2;

sizes.NumDiscStates = 0;

sizes.NumOutputs = 2;
```

```
sizes.NumOutputs = 2;

sizes.NumInputs = 2;

sizes.DirFeedthrough = 1;

sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
x0 = zeros(2,1);
```

```
str = [];

ts = [0 0];

function sys=mdlDerivatives(t,x,u,A,B,C,D)

sys = A*x + B*u;

function sys=mdlOutputs(t,x,u,A,B,C,D)

sys = C*x + D*u;
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Пример модели с S-функцией csfunc



Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Модель дискретной системы:

Данная S-функция моделирует дискретную систему с двумя входами, двумя выходами и двумя переменными состояния. Параметры модели (значения матриц A, B, C, D) задаются в теле S-функции и передаются в callback-методы через их заголовки в качестве дополнительных параметров. Ниже приведен текст этой S-функции.

```
function [sys,x0,str,ts] = dsfunc(t,x,u,flag)
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

С помощью уравнений пространства состояния моделируется дискретная система:

$x(n+1) = Ax(n) + Bu(n)$

$y(n) = Cx(n) + Du(n)$

Задание матриц:

```
A = [0.9135 0.1594
     -0.7971 0.5947];

B = [0.05189 0
     0.4782 0];

C = [0 1
     1 0];

D = [0.01 0
     0 -0.02];
```

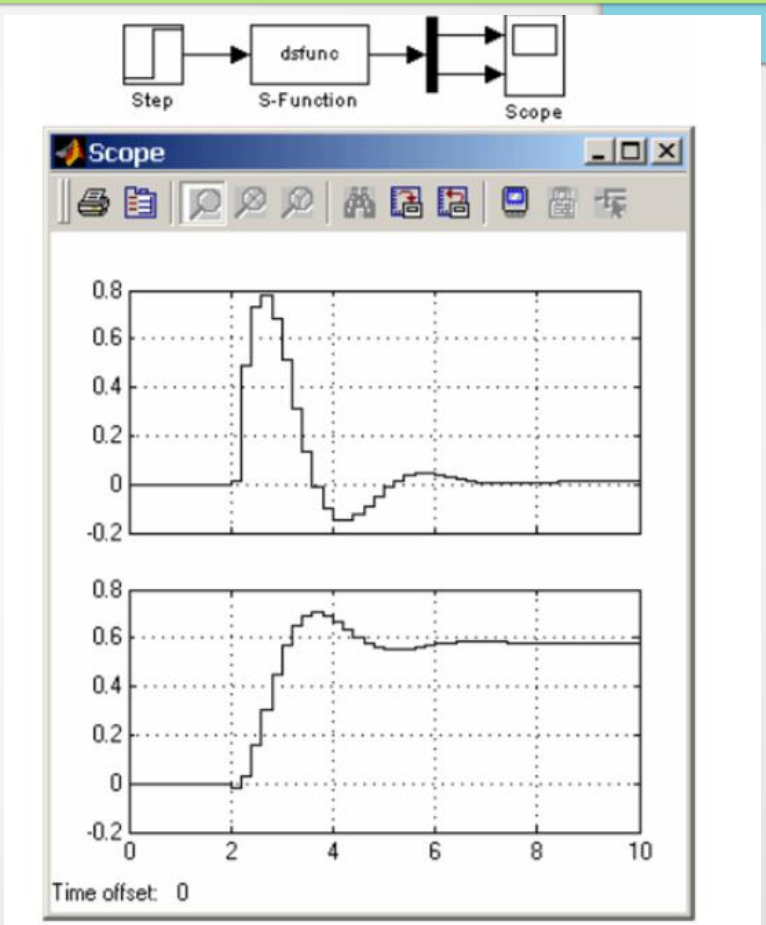
```
switch flag,

case 0, [sys,x0,str,ts] = mdlInitializeSizes(A,B,C,D);

case 2, sys = mdlUpdate(t,x,u,A,B,C,D);

case 3, sys = mdlOutputs(t,x,u,A,C,D);

case { 1, 4, 9 }

sys=[];
otherwise error(['unhandled flag = ',num2str(flag)]); end
```

```
function [sys,x0,str,ts] = mdlInitializeSizes(A,B,C,D)

sizes = simsizes;

sizes.NumContStates = 0;

sizes.NumDiscStates = size(A,1);

sizes.NumOutputs = size(D,1);

sizes.NumInputs = size(D,2);

sizes.DirFeedthrough = 1;

sizes.NumSampleTimes = 1; sys = simsizes(sizes);

x0 = zeros(sizes.NumDiscStates,1);

str = []; ts = [0.2 0];

function sys = mdlUpdate(t,x,u,A,B,C,D)
```

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

```
sys = A*x+B*u;

function sys = mdlOutputs(t,x,u,A,C,D)

sys = C*x+D*u;
```

Пример модели с S-функцией dsfunc:



Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Рекомендации по созданию S-функции ДПТ НВ:

Для создания S-функции для ДПТ НВ за основу берётся пример модели непрерывной системы csfunc. Принципиальные отличия:

1. Для расчета матриц уравнений пространства состояния A, B и C используются параметры передаваемые в S-функцию через окно диалога блока S-function. Эти параметры (L, R, J, Cm, Cw, Fi) записываются в конце списка параметров в заголовке Sфункции: function [sys, x0, str, ts] = dpt_sfunc_1(t, x, u, flag, L, R, J, Cm, Cw, Fi).
2. Для исключения повторяющихся вычислений расчет матриц A, B и C выполняется в методе mdlInitializeSizes. Для этого параметры блока (L, R, J, Cm, Cw, Fi) передаются в метод mdlInitializeSizes через его заголовок: [sys, x0, str, ts] = mdlInitializeSizes(L, R, J, Cm, Cw, Fi). Поскольку инициализация модели происходит лишь один раз, то и расчет матриц A, B и C будет выполнен также один раз, что значительно повысит скорость моделирования.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

3. Передача рассчитанных в mdlInitializeSizes матриц выполняется с помощью глобальных переменных. Для этого объявления вида: global A B C выполнены в теле Sфункции, методе mdlInitializeSizes (где выполняется расчет этих матриц), а также методах mdlDerivatives и mdlOutputs (где эти матрицы используются для расчетов).
4. Поскольку в уравнениях пространства-состояния матрица D отсутствует (входные переменные не участвуют в формировании выходных переменных), то параметр sizesDirFeedthrough в методе mdlInitializeSizes задан равным нулю.

Тема: Инструментарий создания собственных функций в среде Matlab/Simulink

Список рекомендуемой литературы:

1. Дьяконов В. П. MATLAB 6/6.1/6.5 + Simulink 4/5: Основы применения: Полное руководство пользователя / В. П. Дьяконов. — М.: СОЛОН-Пресс, 2002. — 768 с.
2. Дьяконов В. П. MATLAB; Анализ, идентификация и моделирование систем: Специальный справочник / В. П. Дьяконов, В. В. Круглов. — СПб.: Питер, 2002. — 448 с.

Спасибо за внимание!