

Subsystem – подсистемы

- Уменьшает количество одновременно отображаемых блоков на экране, что облегчает восприятие модели (в идеале модель полностью должна отображаться на экране монитора).
- Позволяет создавать и отлаживать фрагменты модели по отдельности, что повышает технологичность создания модели.
- Позволяет создавать собственные библиотеки.
- Дает возможность синхронизации параллельно работающих подсистем.
- Позволяет включать в модель собственные справочные средства.
- Дает возможность связывать подсистему с каким-либо **m**-файлом, обеспечивая запуск этого файла при открытии подсистемы (нестандартное открытие подсистемы).

Simulink Library Browser

Файл Правка Вид Помощь

Find

If: IF expression
Run the Action Subsystem connected to 1st of 2 outputs
ELSEIF expression
Run the Action Subsystem connected to 2nd of 2 outputs
ELSE
Run the Action Subsystem connected to last of 2 outputs

- Simulink
 - Continuous
 - Discontinuities
 - Discrete
 - Look-Up Tables
 - Math Operations
 - Model Verification
 - Model-Wide Utilities
 - Ports & Subsystems
 - Signal Attributes
 - Signal Routing
 - Sinks
 - Sources
 - User-Defined Functions
- Aerospace Blockset
- CDMA Reference Blockset
- Communications Blockset
- Control System Toolbox
- DSP Blockset
- Dials & Gauges Blockset
- Embedded Target for Motorola MPC5
- Embedded Target for TI C6000 DSP
- Fixed-Point Blockset
- Fuzzy Logic Toolbox
- MPC Blocks
- NCD Blockset
- Neural Network Blockset
- Real-Time Windows Target
- Real-Time Workshop
- Report Generator
- S-function demos

Готово

Library: simulink/Ports & Subsystems

File Edit View Format Help

Ports & Subsystems

1 In1 1 Out1 Trigger Enable Function-Call Generator

In1 Out1 In1 Out1 In1 Out1

Atomic Subsystem Subsystem Configurable Subsystem

In1 Out1 In1 Out1 In1 Out1

Triggered Subsystem Enabled Subsystem Enabled and Triggered Subsystem

In1 Out1 In1 for{...} Out1 In1 while{...} Out1

Function-Call Subsystem For Iterator Subsystem While Iterator Subsystem

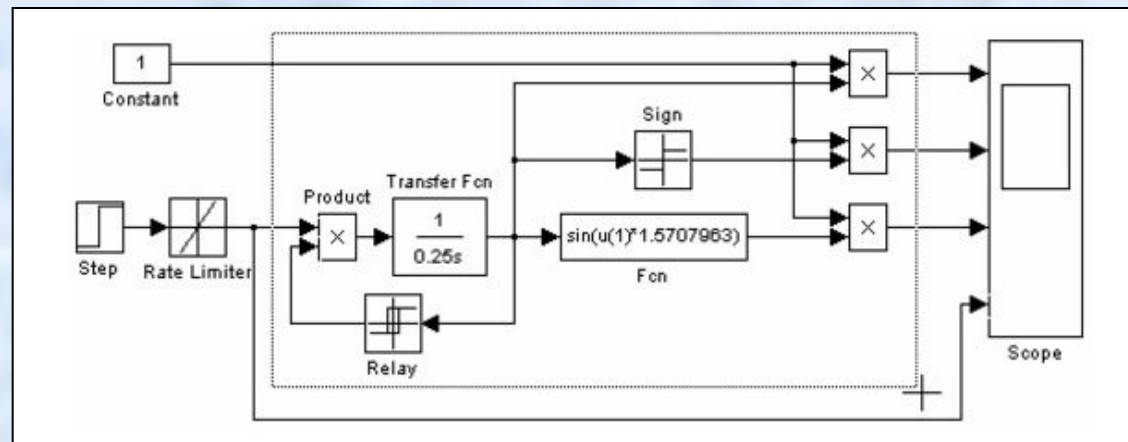
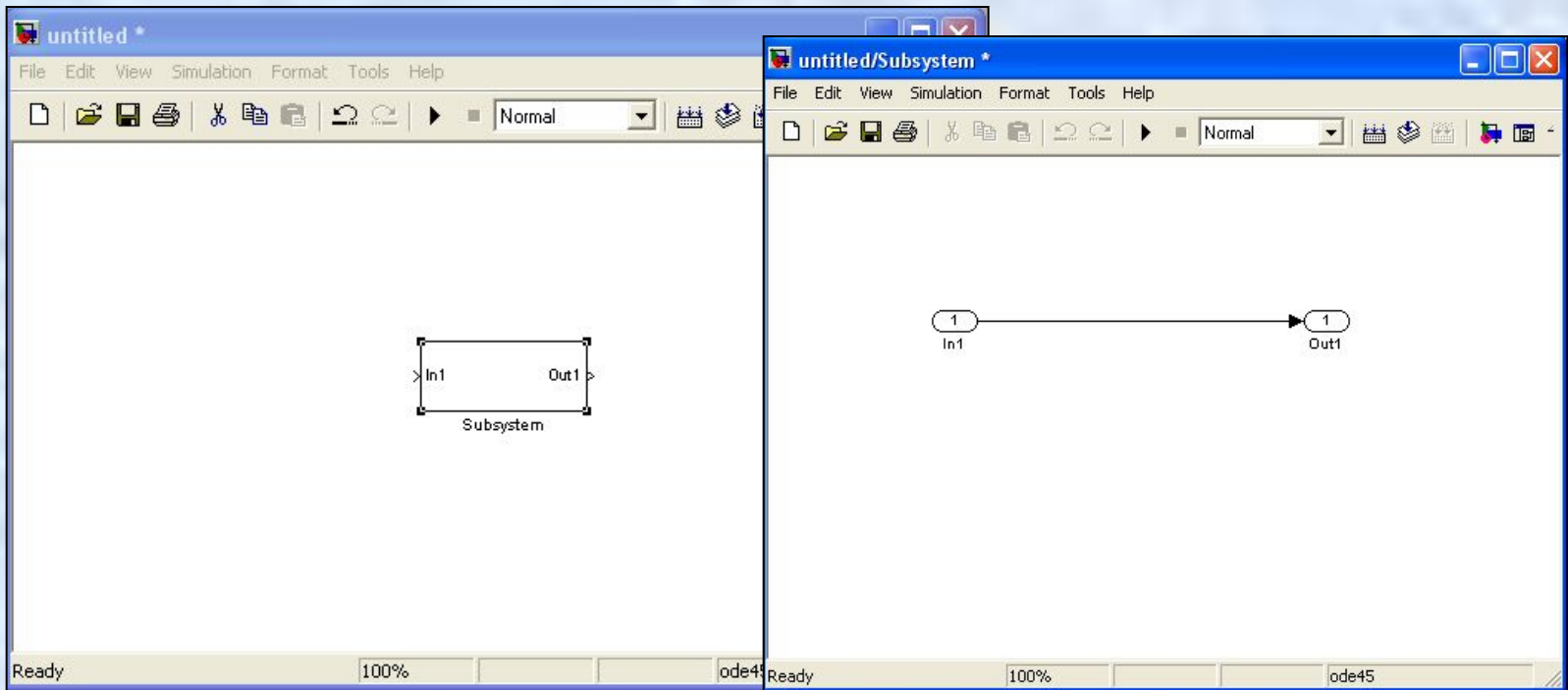
u1 if(u1 > 0) else u1 case [1]: default:

If Switch Case

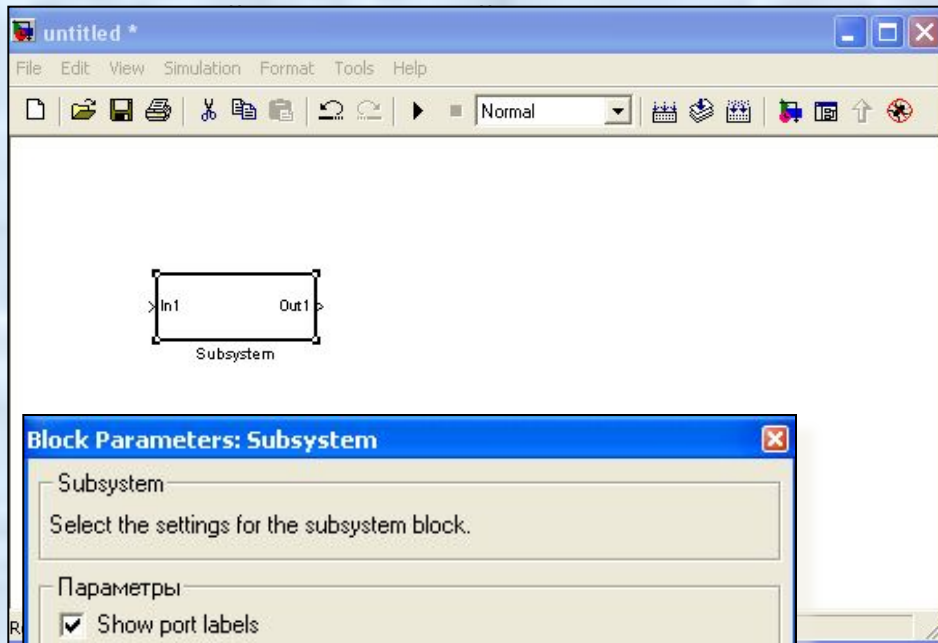
In1 if{ } Out1 In1 case: { } Out1

If Action Subsystem Switch Case Action Subsystem

Subsystem Examples



Treat as atomic unit (флажок) – Считать подсистему монолитной. Таким образом, блоки



т же блок, отличающийся значением

Show port labels – Показать метки портов только для просмотра. Открывать и изменять подсистему.

Treat as atomic unit (флажок) – Считать подсистему монолитной. Таким образом, блоки виртуальной и монолитной подсистем – это один и тот же блок, отличающийся значением данного параметра.

ReadWrite – Пользователь может открывать и изменять подсистему.

ReadOnly – Пользователь может открывать подсистему только для просмотра.

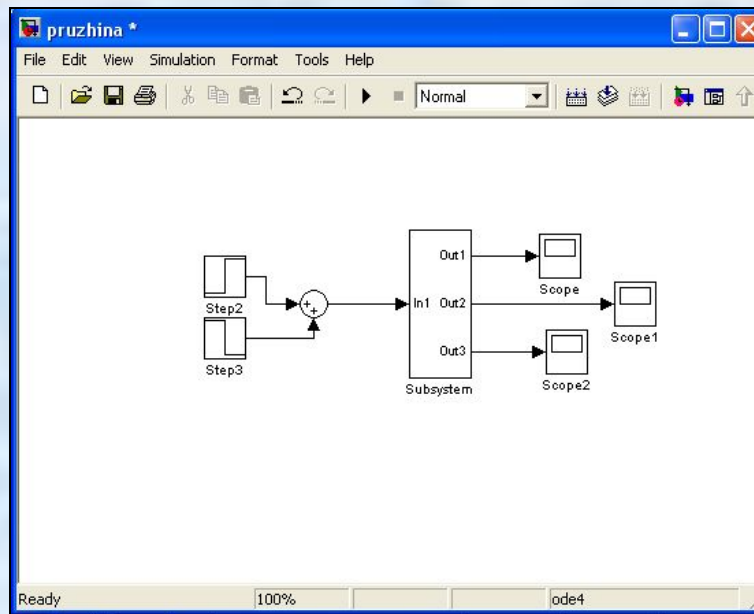
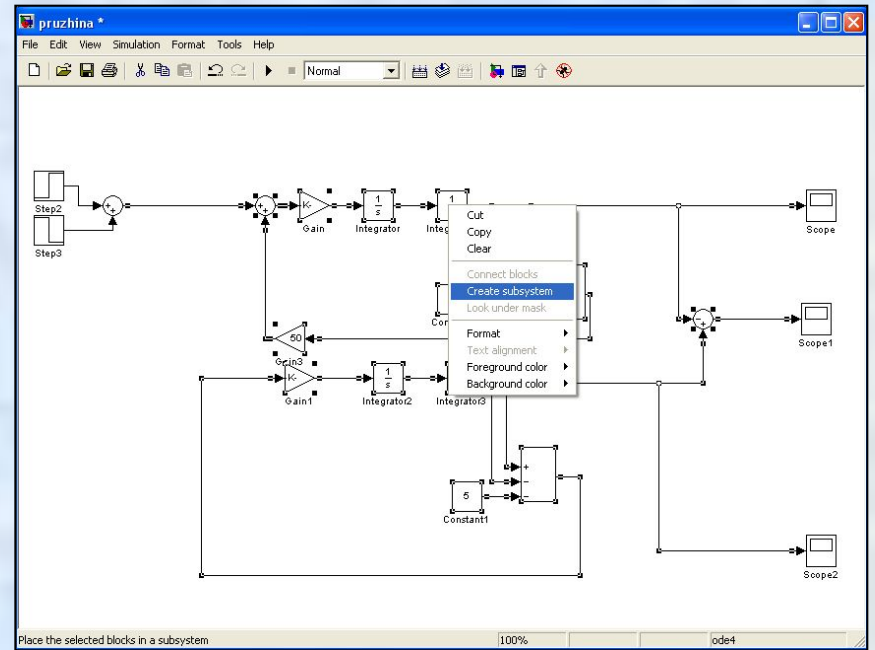
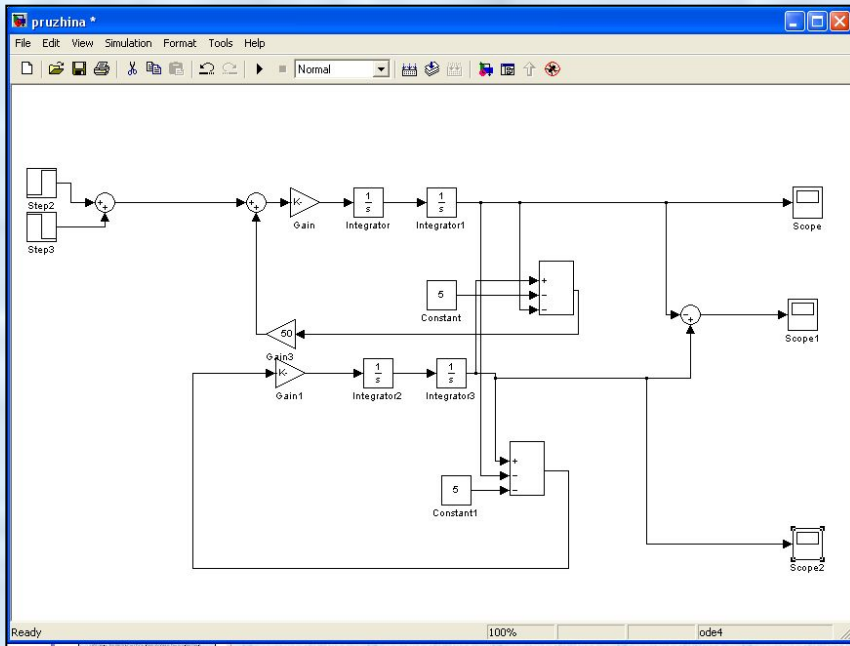
NoReadOrWrite – Пользователь не может открывать и изменять подсистему.

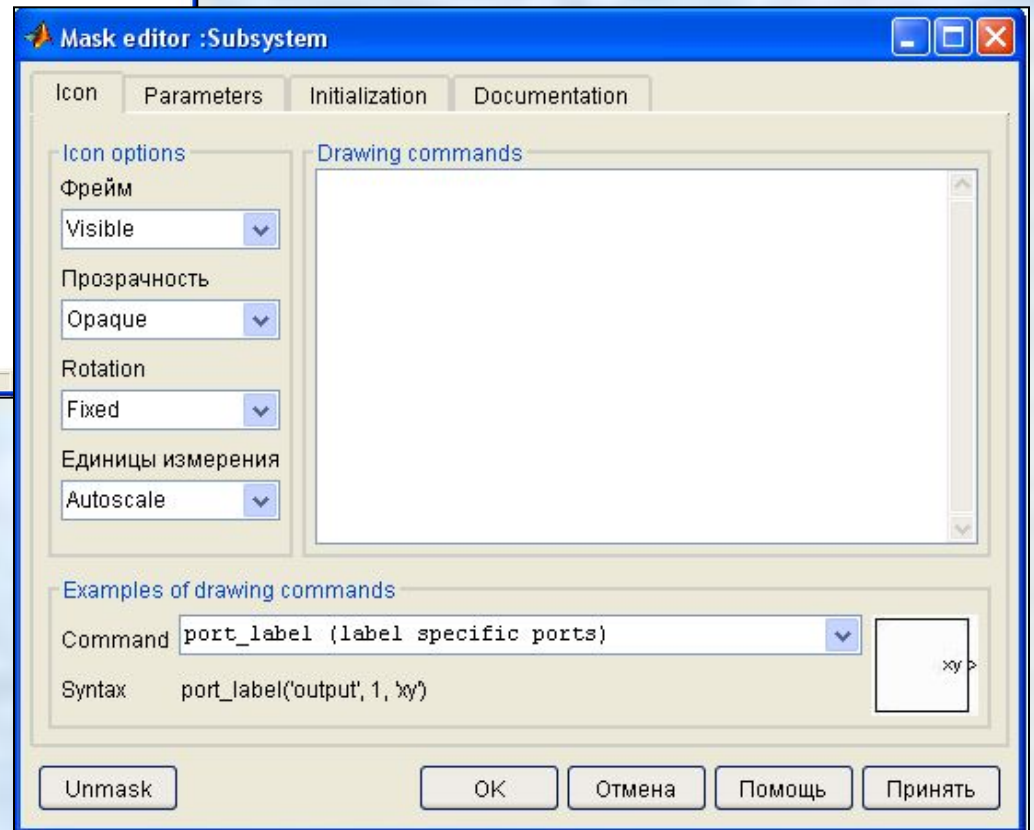
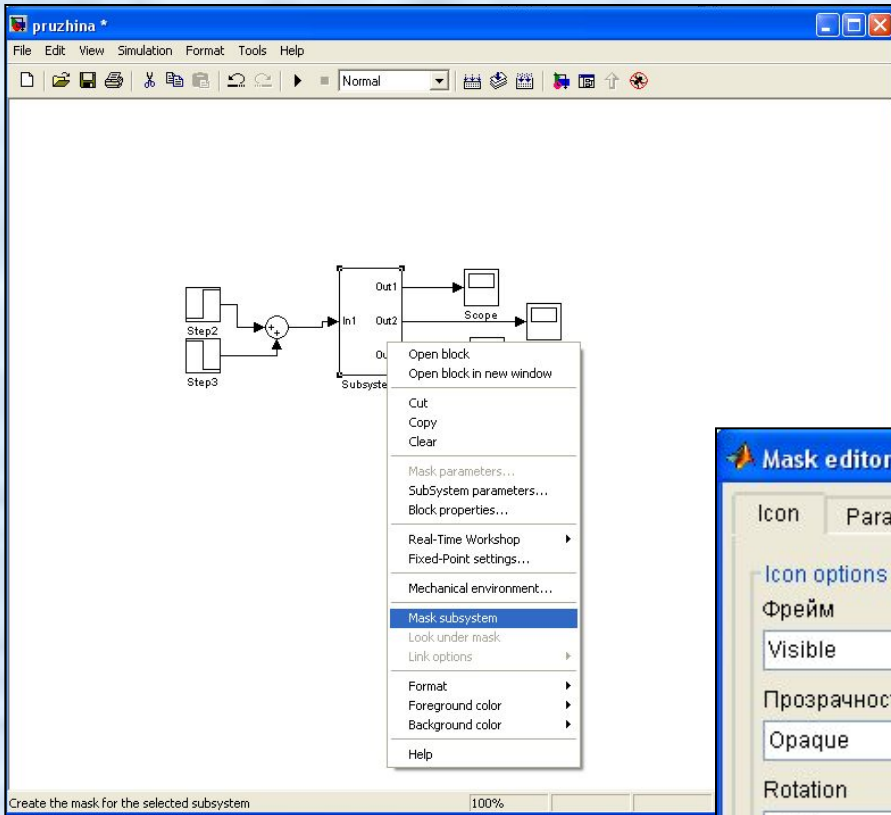
Name of error callback function – Имя функции используемой для обработки ошибок возникающих в данной подсистеме.

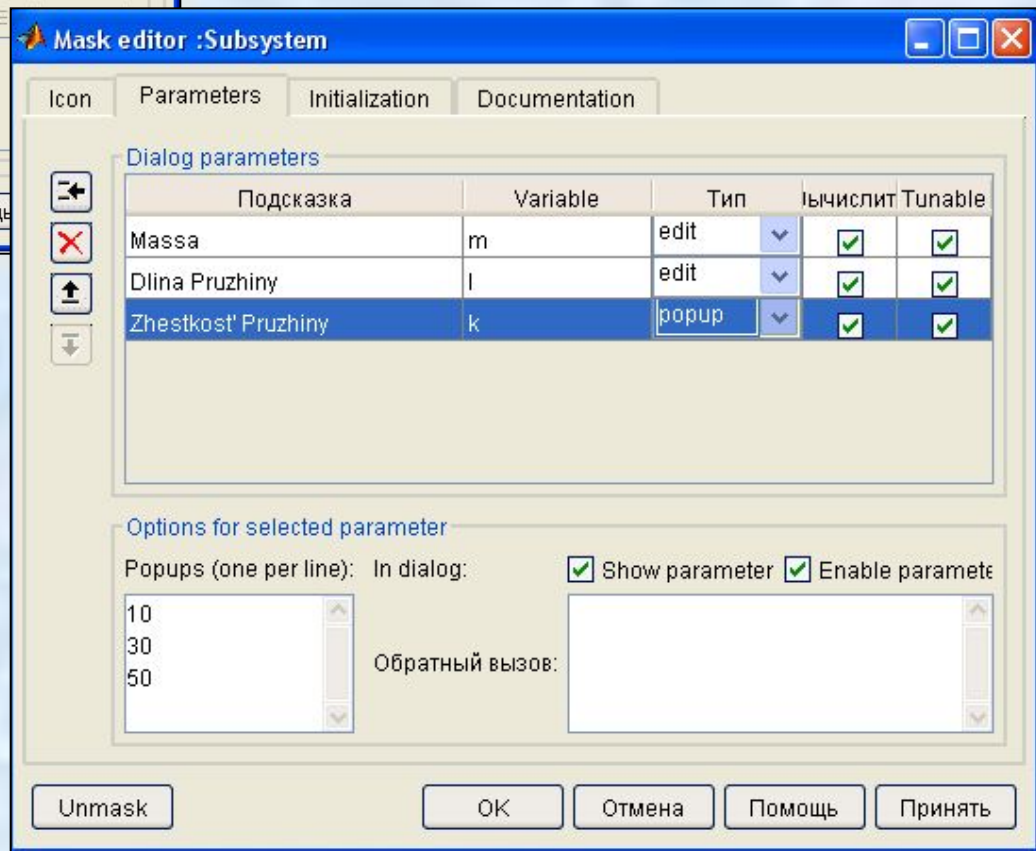
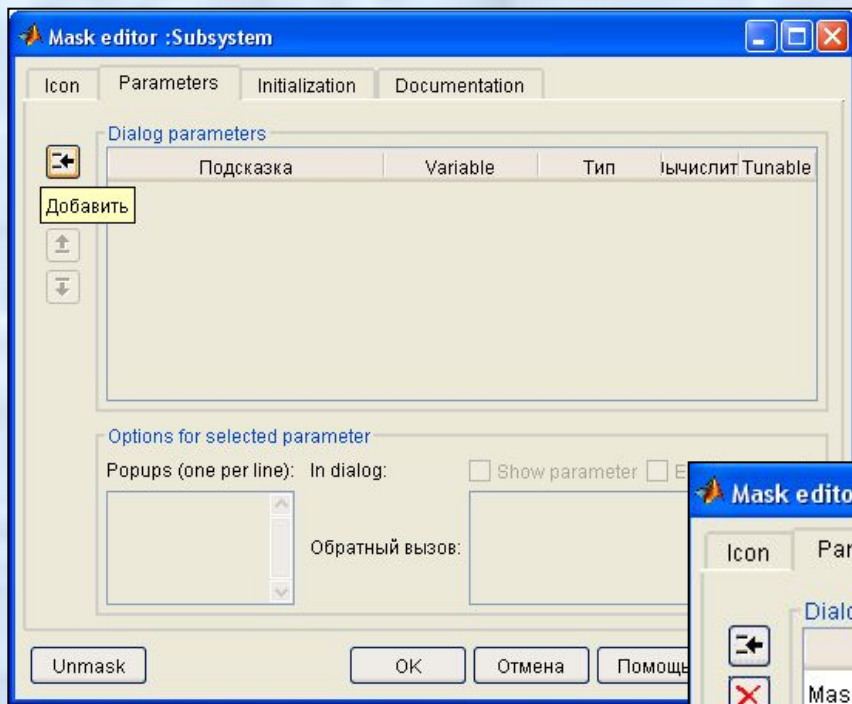
Маскирование подсистем

1. Расширяет возможности пользователя по управлению параметрами модели.
2. Позволяет создавать более понятный интерфейс подсистемы.
3. Повышает наглядность блок-диаграммы.
4. Расширяет возможности построения сложных моделей.
5. Повышает защищенность модели от несанкционированной модификации.

1. Определить, какие параметры подсистемы должны задаваться пользователем в будущем окне параметров. Задать эти параметры в подсистеме с помощью идентификаторов (имен).
2. Определить, каким образом параметр должен задаваться в окне диалога (с помощью строки ввода, выбором из раскрывающегося списка или установкой флажка).
3. Разработать эскиз пиктограммы блока.
4. Создать комментарии (справку) по использованию подсистемы.







pruzhina *

File Edit View Simulation Format Tools Help

Normal

Step2
Step3
+

In1
Subsystem
Out1
Out2
Out3
Scope
Scope1

Block Parameters: Subsystem

Subsystem (mask)

Параметры

Massa
30

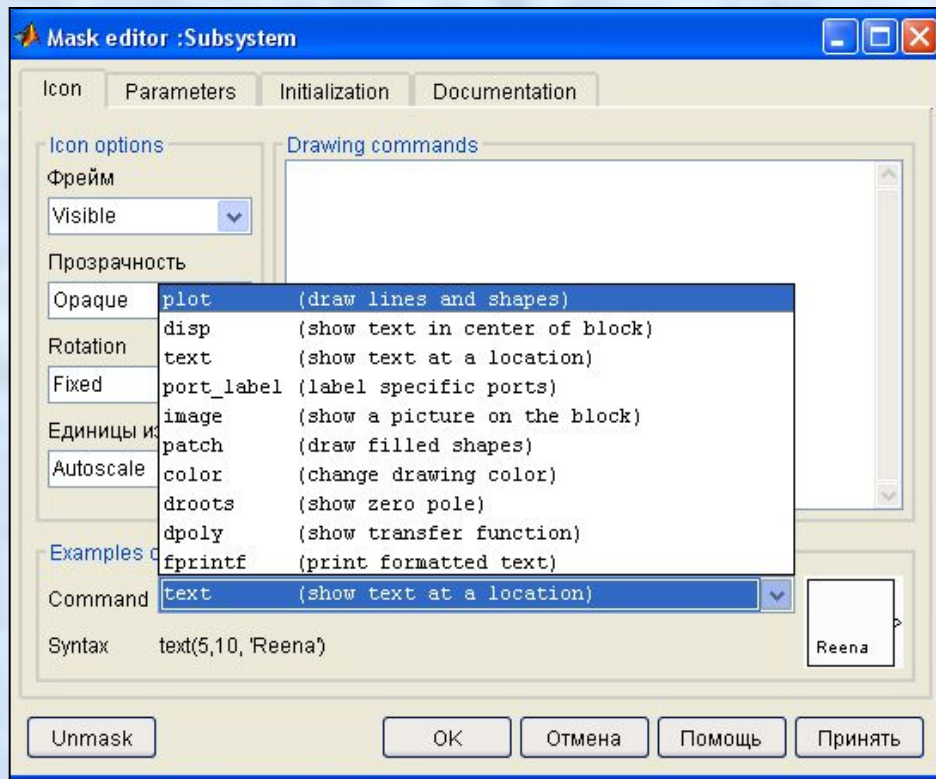
Dlina Pruzhini
5

Zhestkost' Pruzhini 30

10
30
50

OK

Ready 100% ode4



Drawing commands – Область ввода команд рисования. Команды рисования являются выражениями допустимыми в языке **MATLAB**.

Icon frame – Список позволяющий выбрать способ отображения рамки пиктограммы:

Visible – Рамка видна.

Invisible – Рамка не видна.

Icon transparency - Список позволяющий установить прозрачность пиктограммы:

Opaque – Пиктограмма не прозрачна.

Transparent– Пиктограмма прозрачна.

Icon rotation - Список позволяющий задать возможность вращения пиктограммы:

Fixed – Положение пиктограммы фиксировано.

Rotates – Пиктограмма может вращаться вместе с блоком.

Drawing coordinates – Список, задающий условия масштабирования пиктограммы.

Autoscale – Автоматическое масштабирование. Рисунок занимает максимально возможную площадь внутри пиктограммы.

Normalized – Нормализованное масштабирование. Координаты левого нижнего угла пиктограммы **(0,0)**, координаты правого верхнего угла **(1,1)**.

Pixel – Координаты рисунка задаются в пикселах.

Для вывода текста могут использоваться следующие команды:

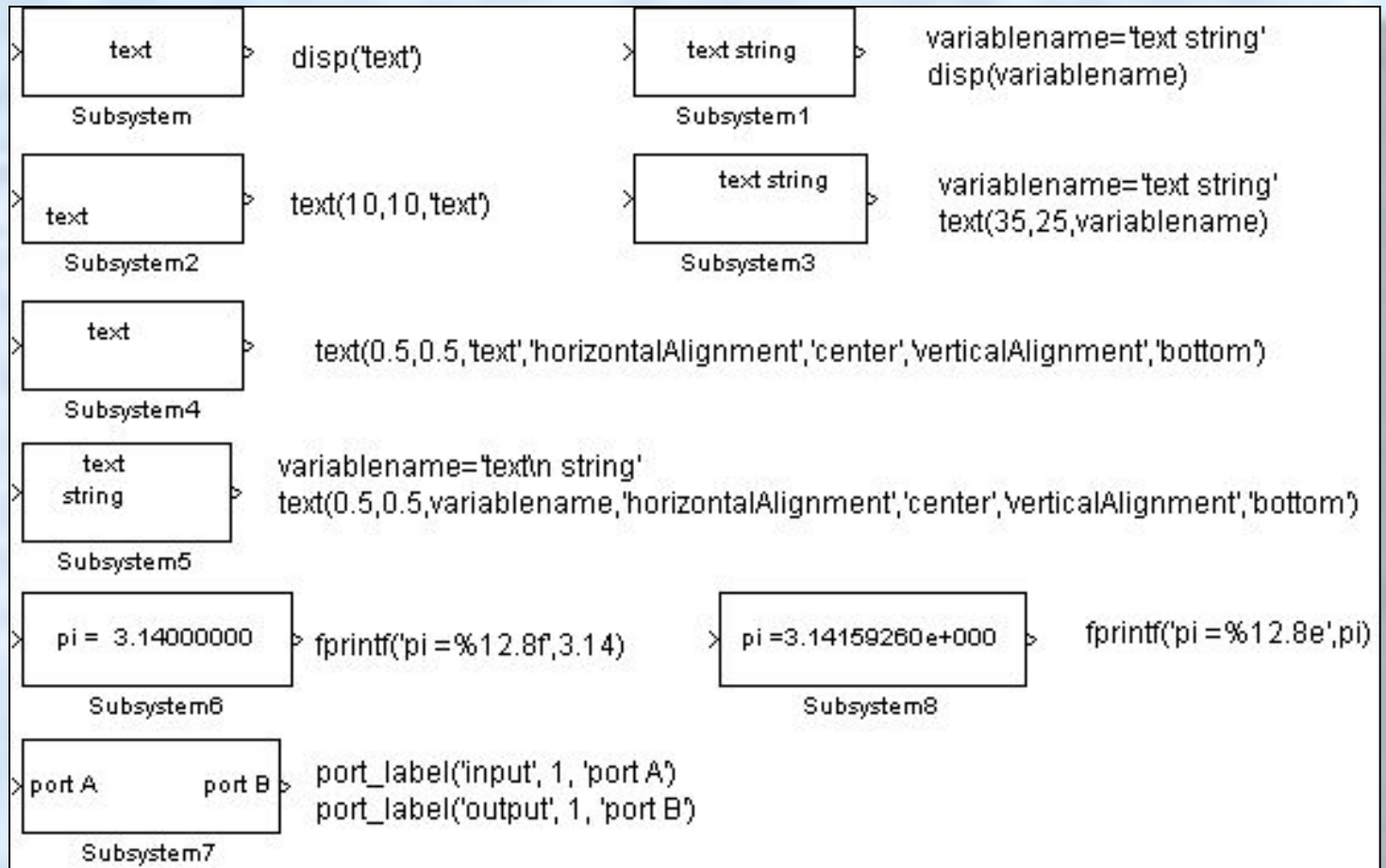
disp('text') или **disp(variablename)** – Вывод текста **'text'** или значения символьной переменной **variablename** в центре пиктограммы.

text(x, y, 'text') или **text(x, y, variablename)** – Вывод текста **'text'** или значения символьной переменной **variablename** начиная с позиции, заданной координатами **x** и **y**.

text(x, y, 'text', 'horizontalAlignment', halign, 'verticalAlignment', valign)-Вывод текста **'text'** в позиции заданной координатами **x** и **y** и с указанием способов выравнивания относительно этой позиции по вертикали или горизонтали. Параметр **halign** может принимать значения: **'left', 'right'** или **'center'**. Параметр **valign** может принимать значения: **'base', 'bottom'** или **'middle'**.

fprintf('text') или **fprintf('format', variablename)** – Форматированный вывод (по правилам языка C) текста **'text'** или значения символьной переменной **variablename** в центре пиктограммы.

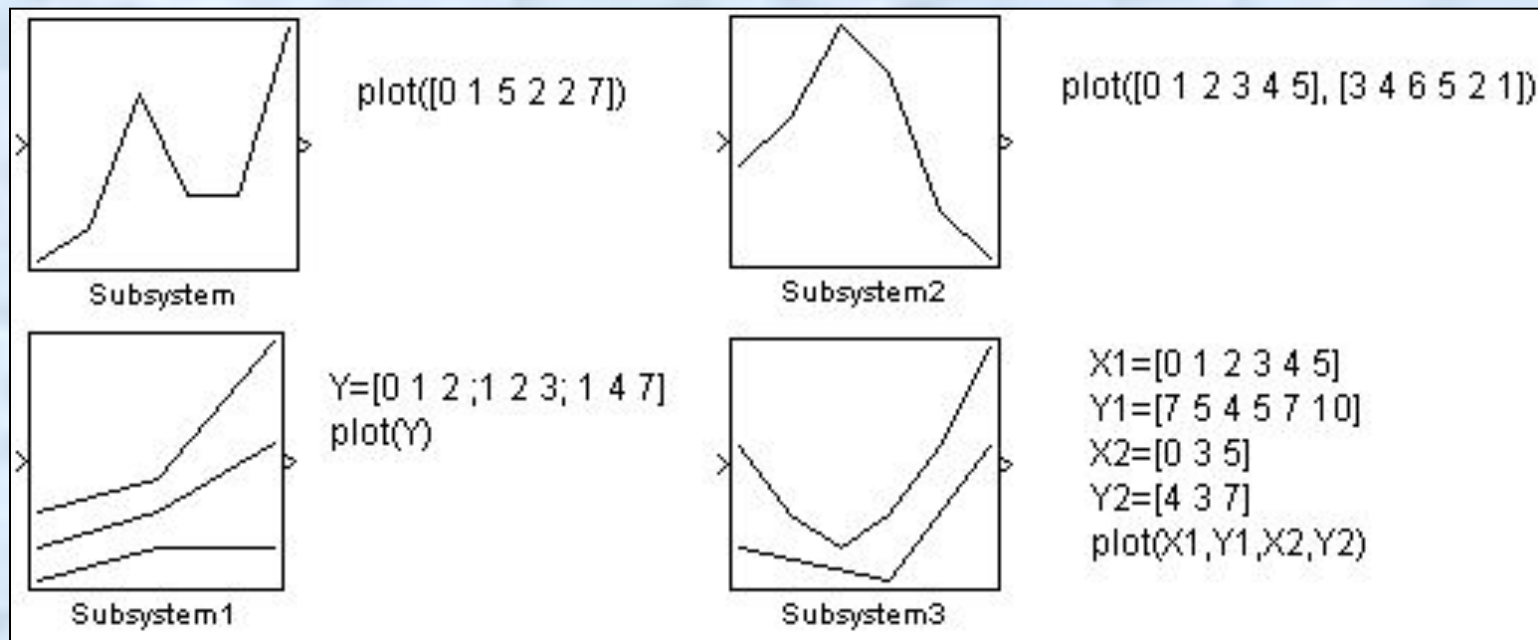
port_label(port_type, port_number, label) – Вывод на пиктограмме метки порта. Например, выражение **port_label('input', 1, 'a')** выводит на пиктограмме метку **a** первого входного порта.



Для построения графиков на пиктограмме могут использоваться следующие команды:

plot(Y) – В том случае, если **Y** является вектором, то строится график по оси абсцисс которого откладывается значение индекса элемента, а по оси ординат значение самого элемента. В том случае если **Y** является матрицей – строятся линии для каждого столбца. По оси абсцисс в этом случае также откладывается значение индекса элемента.

plot(X1,Y1,X2,Y2,...) – Строится графики вида **Y1(X1)**, **Y2(X2)** и т.д.



Для отображения на пиктограмме рисунка из графического файла используются следующие команды:

image(imread('filename')) – Отображение рисунка из файла с полным именем **filename**. Для правильной работы этой команды необходимо поместить рисунок в ту же папку, где находится файл модели, и сделать эту папку рабочей. Допускается также совместно с именем файла указывать его полный путь.

image(a, [x, y, w, h]) – Отображение рисунка содержащегося в переменной **a**. Ширина и высота рисунка задаются параметрами **w** и **h**, соответственно. Левый нижний угол рисунка расположен в точке с координатами **x,y**. Считывание рисунка из файла может быть выполнено командой **a = imread('filename')**.

image(a, [x, y, w, h], rotation) – Команда аналогичная предыдущей, но позволяющая задавать поведение рисунка при вращении пиктограммы. Значение параметра **rotation** равно **'on'** позволяет поворачивать рисунок вместе с пиктограммой подсистемы.

patch(x, y) – Отображение закрашенного многоугольника, координаты которого заданы векторами **x** и **y**. Цвет рисунка – черный.

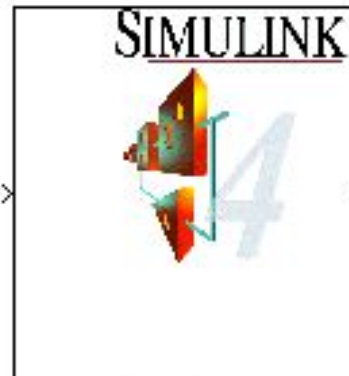
patch(x, y, [r g b]) - Команда аналогичная предыдущей, но позволяющая задавать цвет рисунка. Параметры **r,g** и **b** задают соотношение красного, зеленого и синего цветов в рисунке. Значение параметров должно находиться в пределах от **0** до **1**.

```
image(imread('image_1.bmp'))
```



Subsystem

```
a=imread('image_1.bmp');  
image(a,[0.3 0.3 1 1])
```



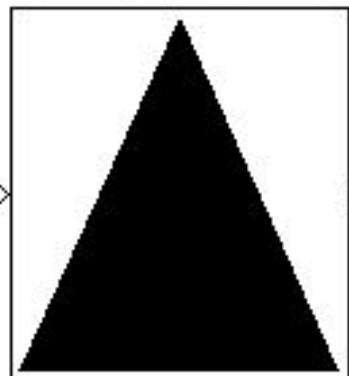
Subsystem4

```
a=imread('image_1.bmp');  
image(a,[0 0 1 1],'on')
```



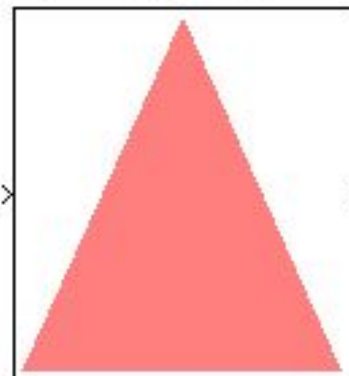
Subsystem1

```
patch([0 .5 1], [0 1 0])
```



Subsystem2

```
patch([0 .5 1], [0 1 0], [1 0.5 0.5])
```



Subsystem3

Использование редактора пиктограмм iconedit

Использование редактора пиктограмм iconedit

