Computer Aided Design (CAD)

Lecture 10

Introduction to Simulink (3)

Dr.Eng. Basem ElHalawany

Schedule (Updated 28-10)

Topics	Estimated Duration (# Lectures)
Introduction	1
Introduction to Matlab Environment	1
Matlab Programing (m-files) (1)	5
Modeling using Matlab Simulink Tool	4 (3/4)
Midterm	7th Week
Communication Systems Simulation (Applications)	3
Introduction to FPGA + Review on Digital Logic/Circuits	2
VHDL Modeling Language	4
VHDL Application	2
Introduction to OPNET Network Simulator	2
Course Closeout / Feedback/ project (s) Delivery	1

The Lecture is based on :

1. **Modeling of Digital Communication Systems using simulink**
2. **Online Tutorials, You can find complete links on Instructor "External Links" on University website**

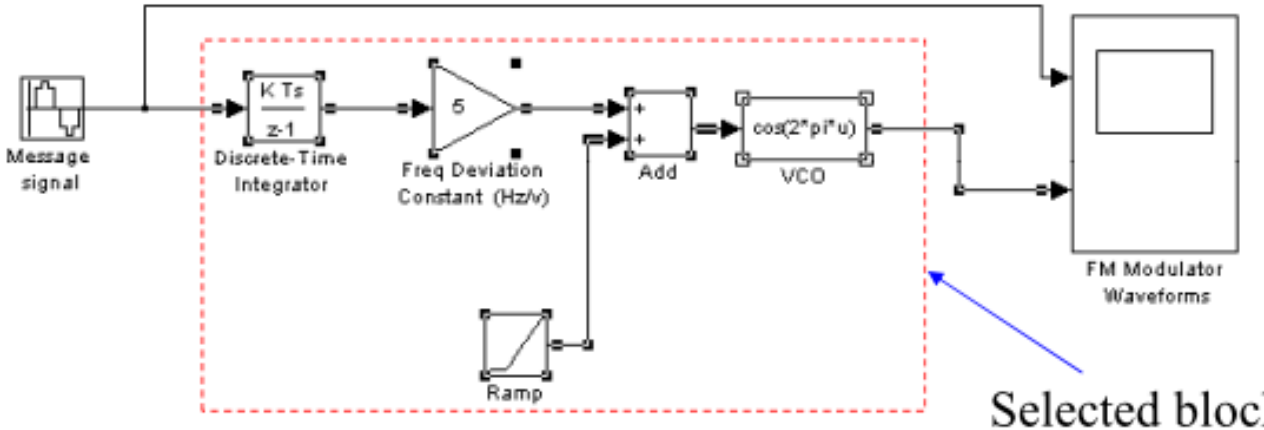
www.bu.edu.eg/staff/basem.mamdoh-external-Links

Subsystems

- A subsystem is a set of blocks that have been replaced by a single block called a **Subsystem** block
- As models increases in size and complexity, they can be simplified by grouping blocks into subsystems
- **Advantages of Subsystems**
  - Reduces the number of blocks displayed in the model window
  - Keeps functionally related blocks together
  - Enables a hierarchical block diagram structure, where a subsystem block is on one layer and the blocks that make up the subsystem are on another

Create a Subsystem

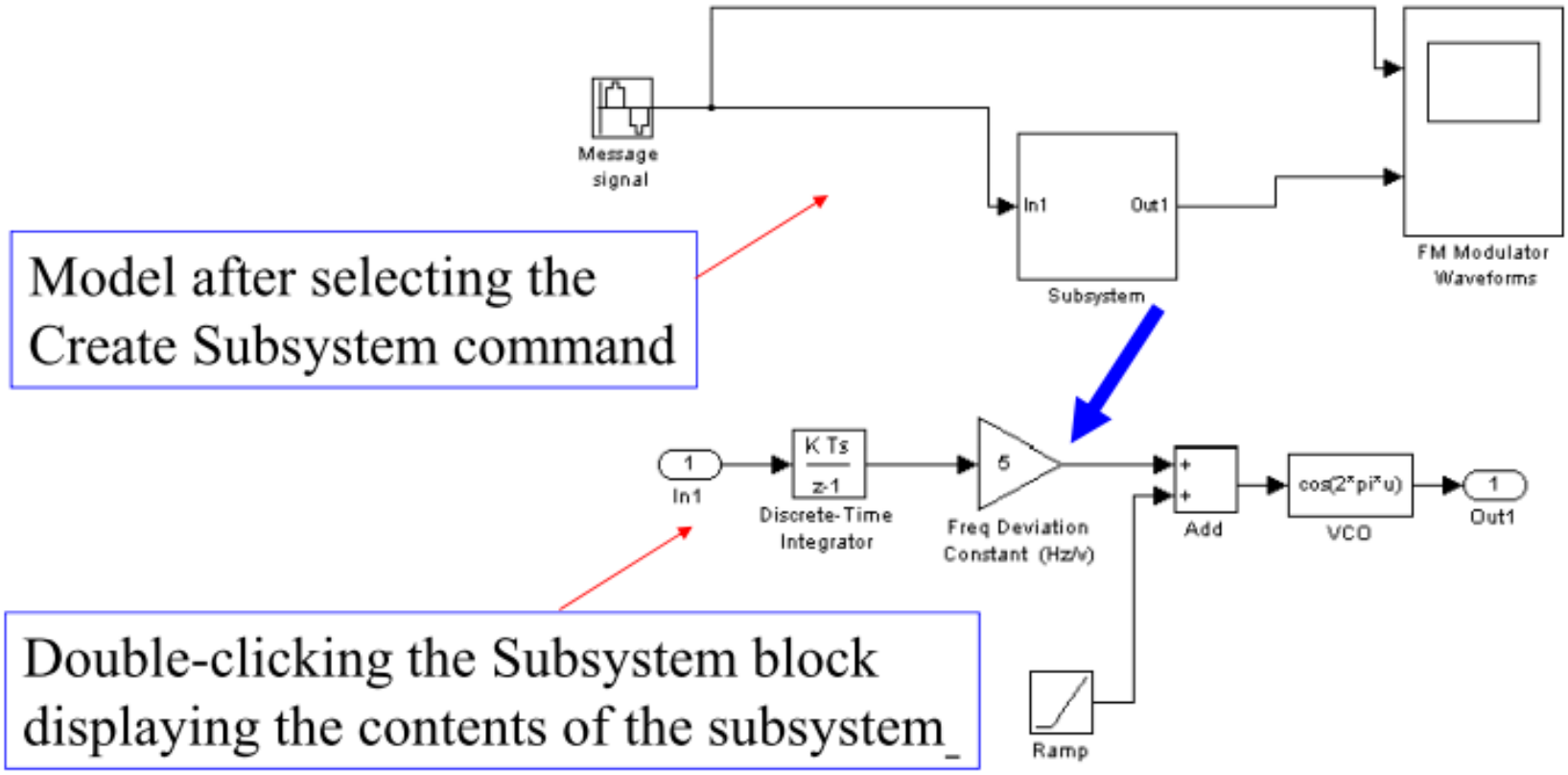
- Enclose the blocks and connecting lines that are included in the subsystem within a bounding box.
  - Define the starting corner of a bounding box by positioning the pointer at one corner of the box, then pressing and holding down the mouse button.
  - Drag the pointer to the opposite corner of the box. A dotted rectangle encloses the selected blocks and lines



Selected blocks and lines constituting the subsystem

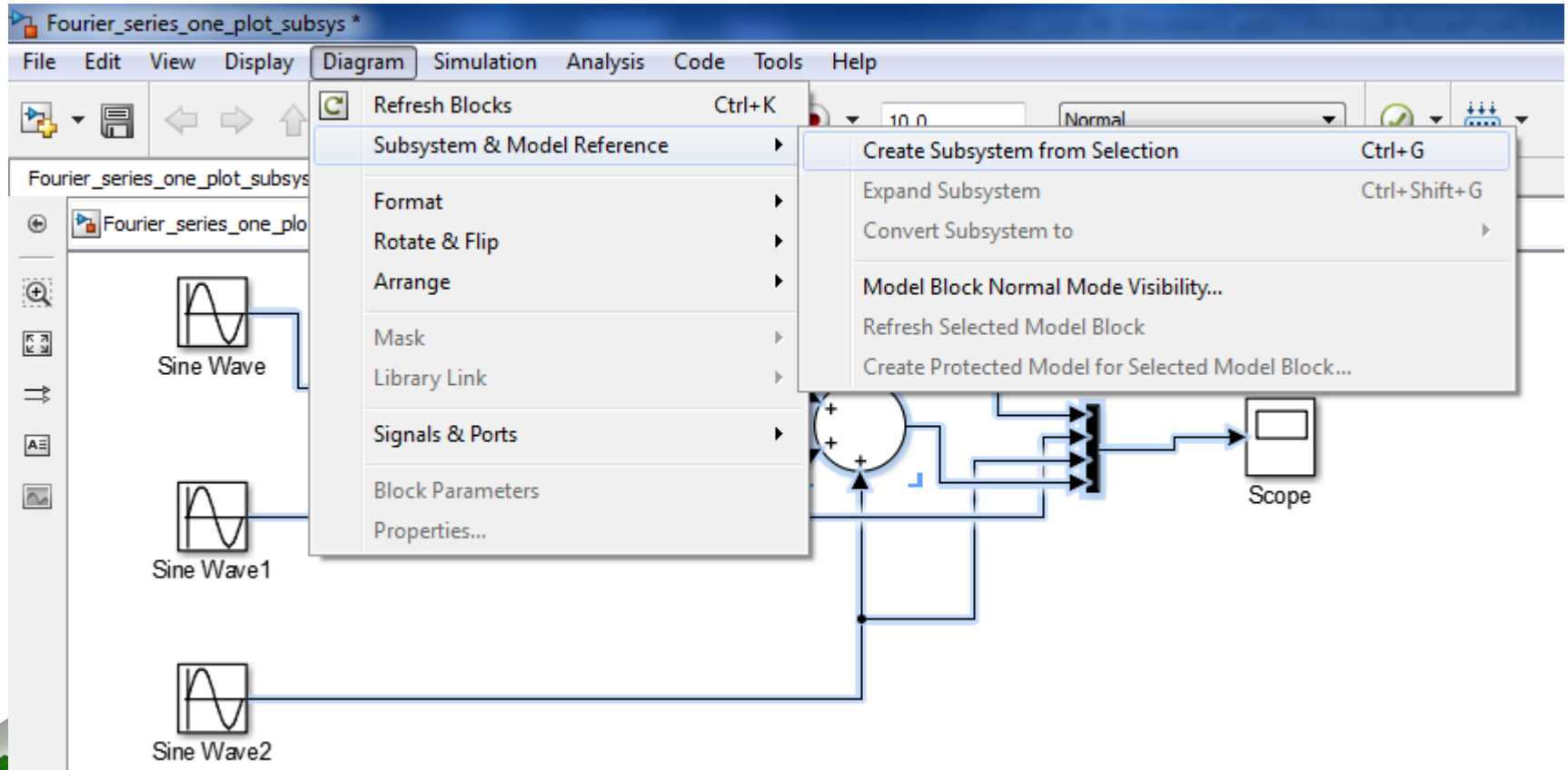
Create a Subsystem (contd)

- Select **Create Subsystem** from the **Edit** menu.
  - A new Subsystem block replaces the selected blocks.



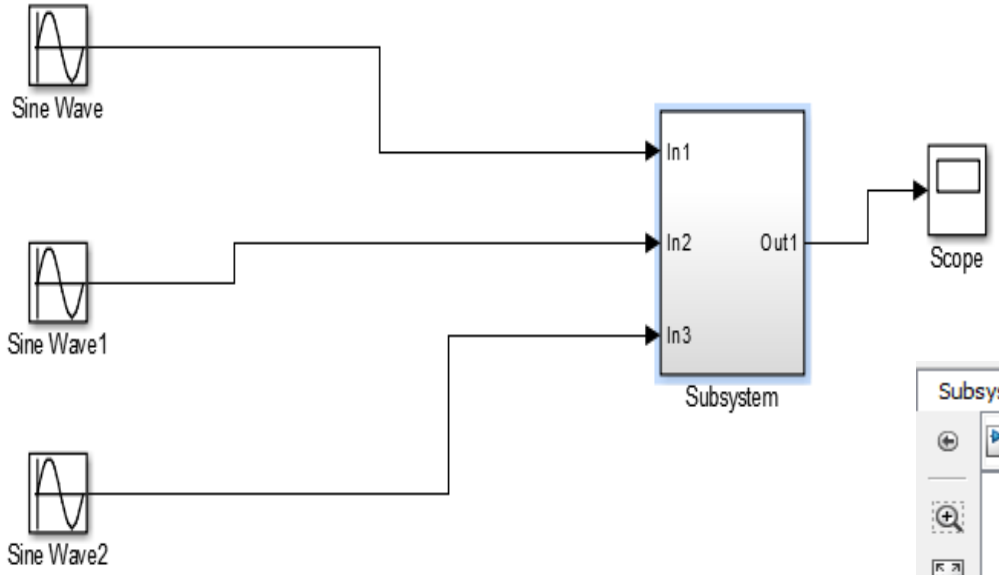
In Matlab 2014

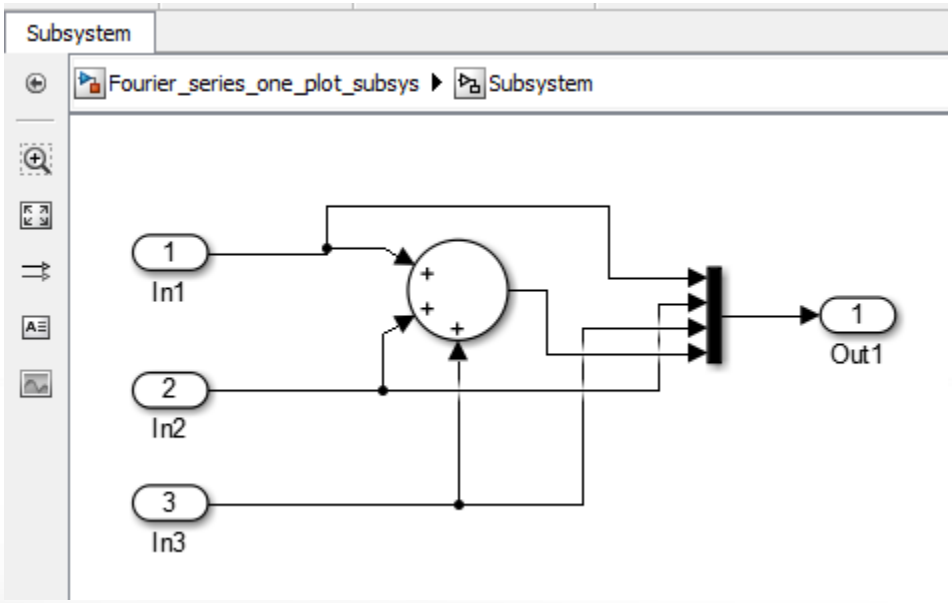
- From Diagram/ Subsystem / Model reference /
- Select Creat Subsystem from selection



In Matlab 2014

- The following figure represents the new subsystem
- The internal construction can be accessed by double-clicking on it



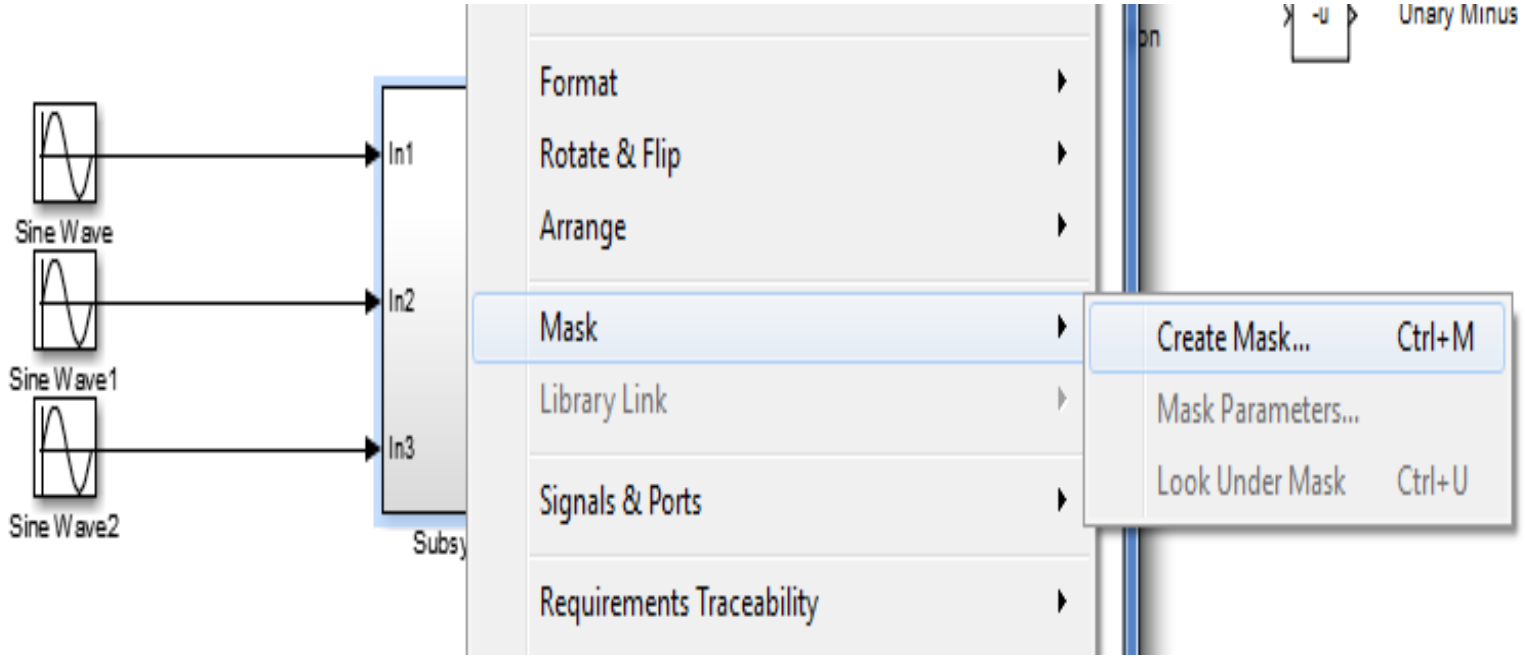


Masking a Subsystem

- A mask is a custom user interface for a subsystem that hides the subsystem's contents, making it appear to the user as a custom block with its own icon and Parameters dialog box
- The Simulink Mask Editor enables us to create a mask for any subsystem. Masking allows us to:
  - Replace the parameter dialogs of a subsystem's contents with a single parameter dialog
  - Replace a subsystem's standard icon with a custom icon that depicts its purpose
  - Prevent unintended modification of subsystems by hiding their contents behind a mask
  - Create a custom block with its own block diagram that defines the block's behavior in a masked subsystem and then placing the masked subsystem in a library.

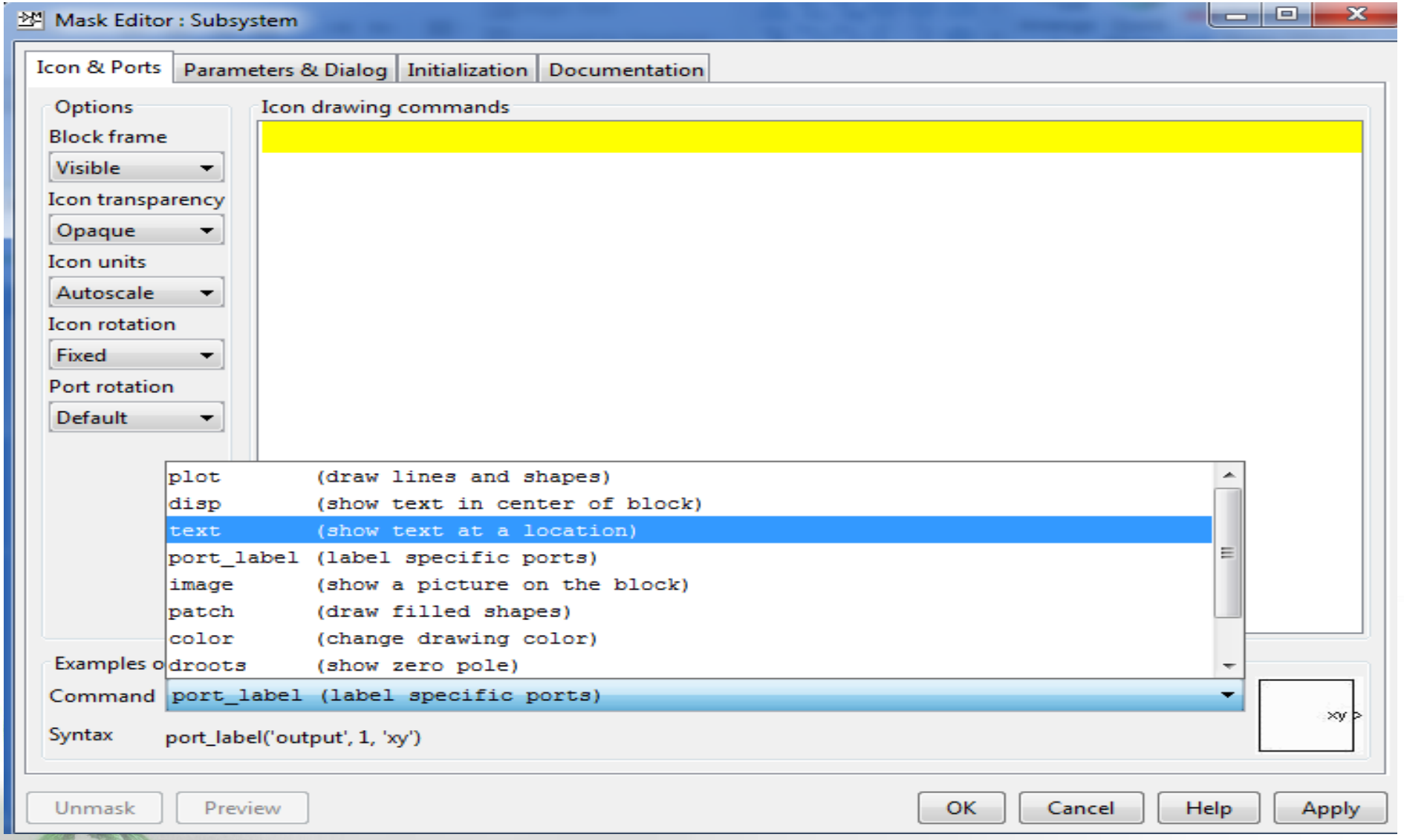
Masking Subsystem

- Right-click on the subsystem, then select “Mask”, then “Create Subsystem”.



Masking Subsystem

- You can create the mask here by using the commands in the lower part according to your design



Masking Subsystem

- Icons and Ports Tab

Options

Block frame: Visible

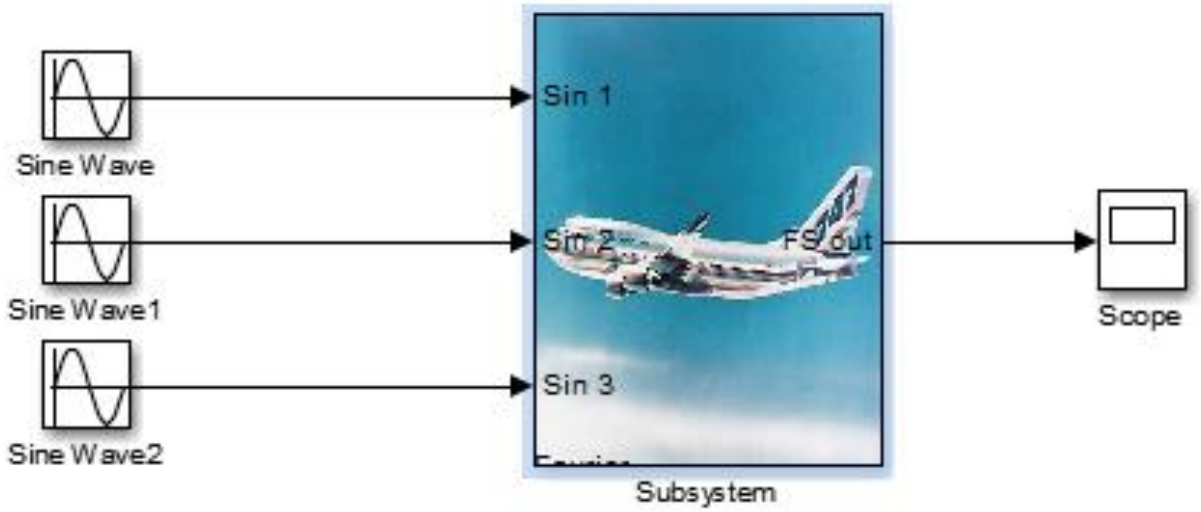
Icon transparency: Opaque

Icon units: Autoscale

Icon rotation: Fixed

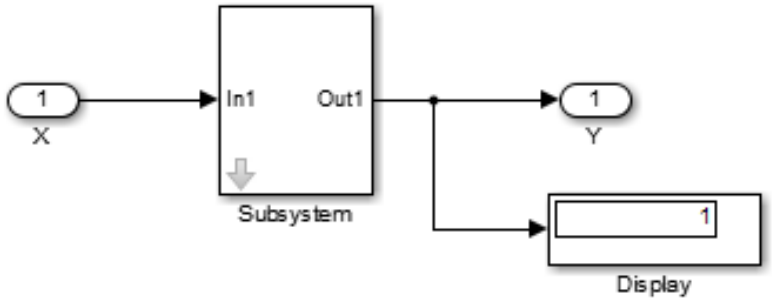
Icon drawing commands

```
image('b747.jpg')
text(5,10, 'Fourier')
port_label('output', 1, 'FS out')
port_label('input', 1, 'Sin 1')
port_label('input', 2, 'Sin 2')
port_label('input', 3, 'Sin 3')
```

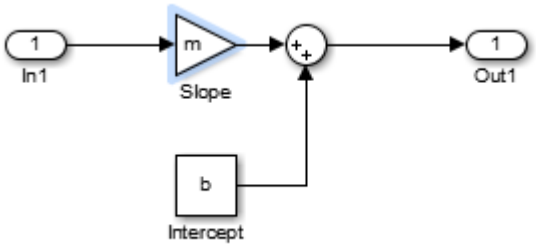


Masking Subsystem

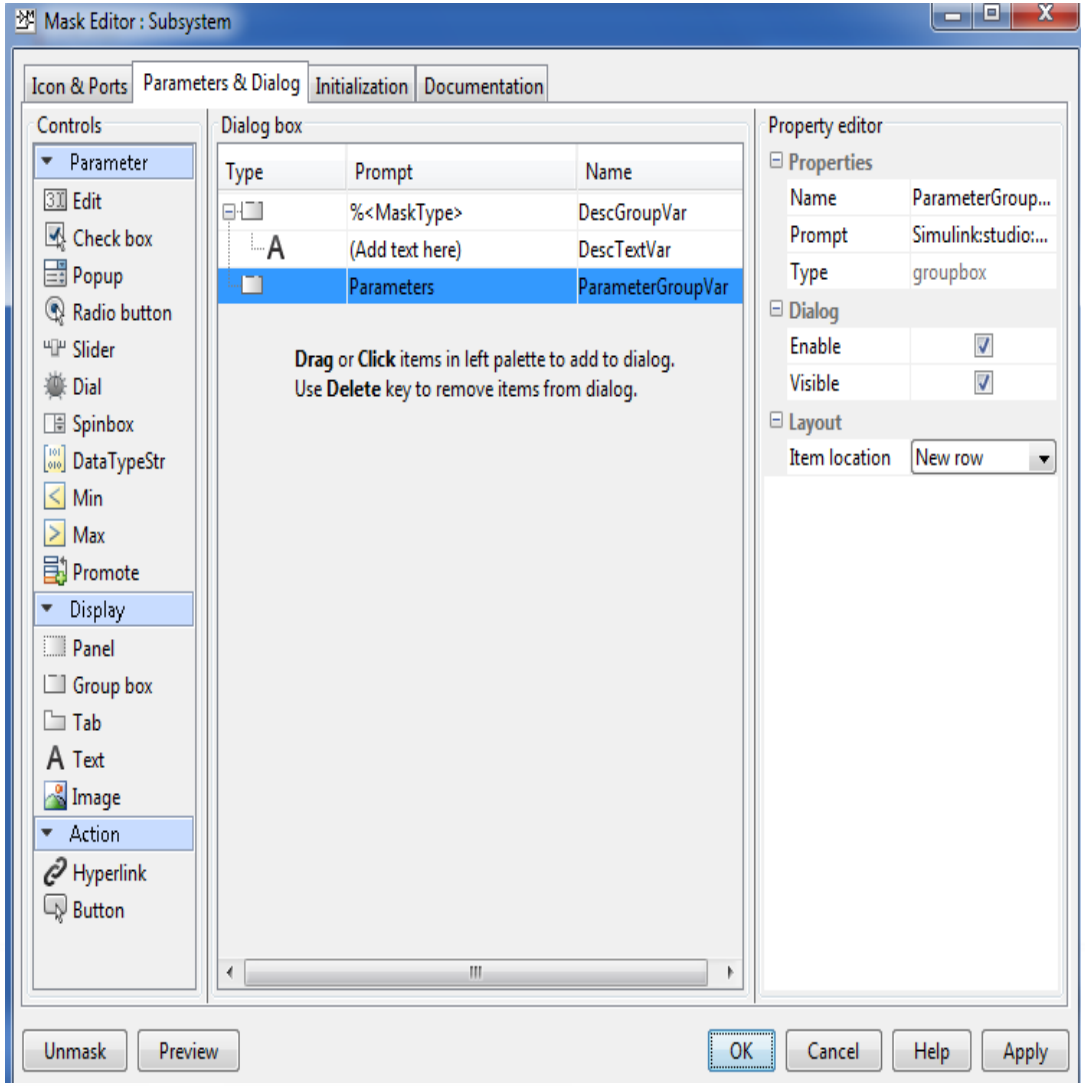
- Parameters and Dialog Tab



System



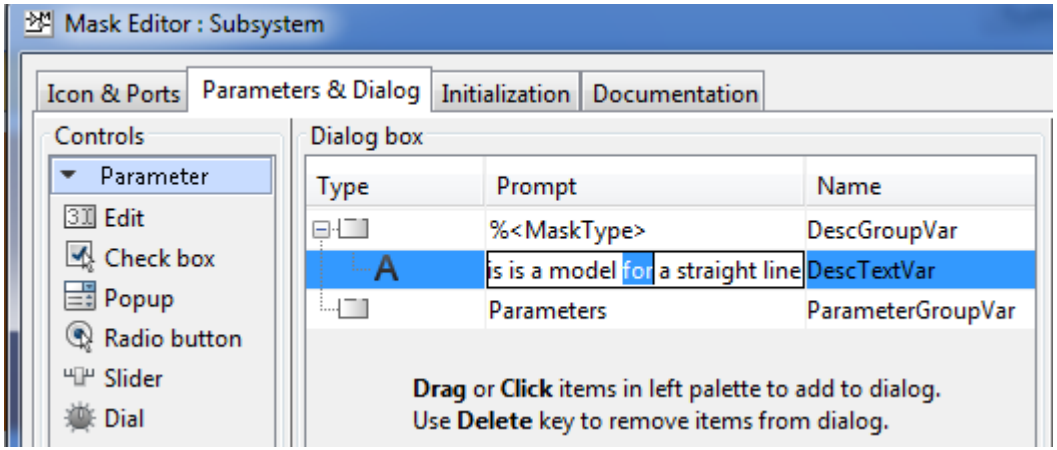
SubSystem

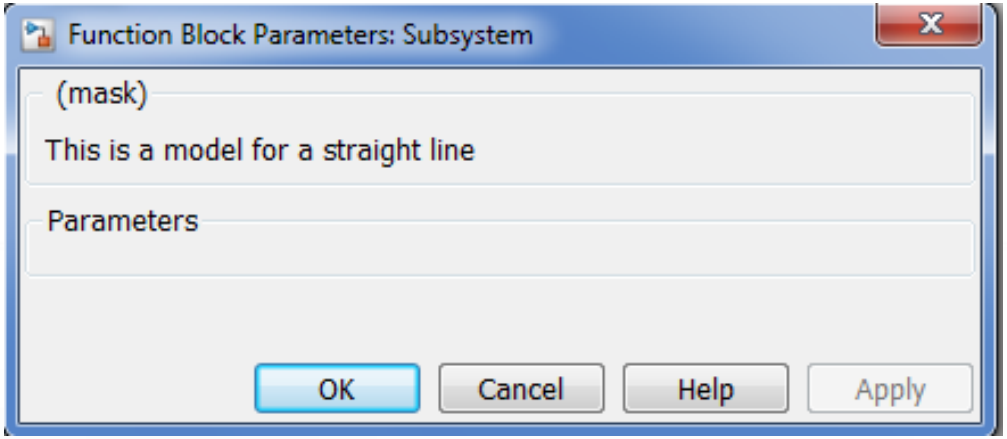


Masking Subsystem

- Parameters and Dialog Tab

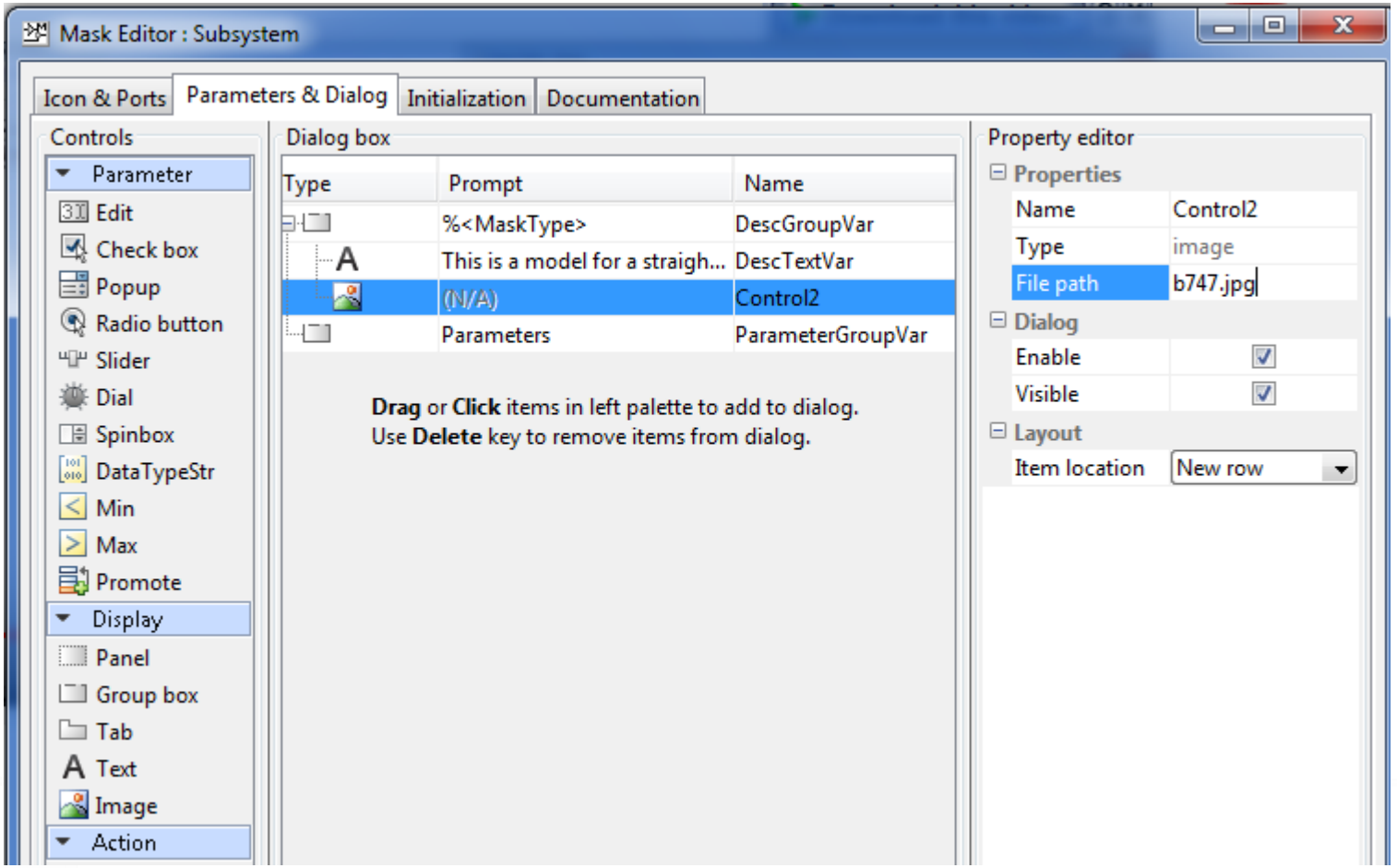
- ✓ Adding introductory text or information in “A”





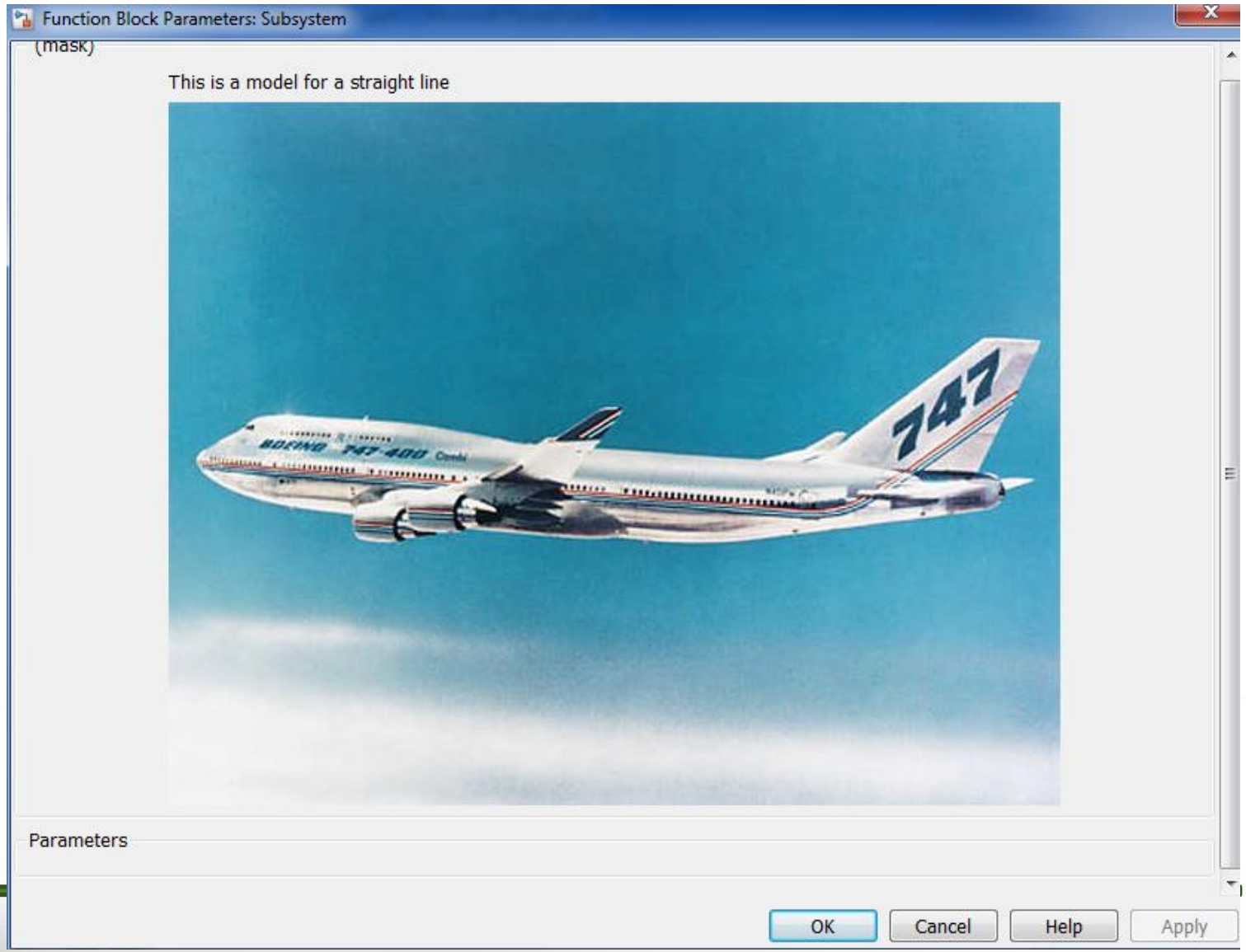
Masking Subsystem

- ➢ Parameters and Dialog Tab
- ✓ Adding Image to the information section



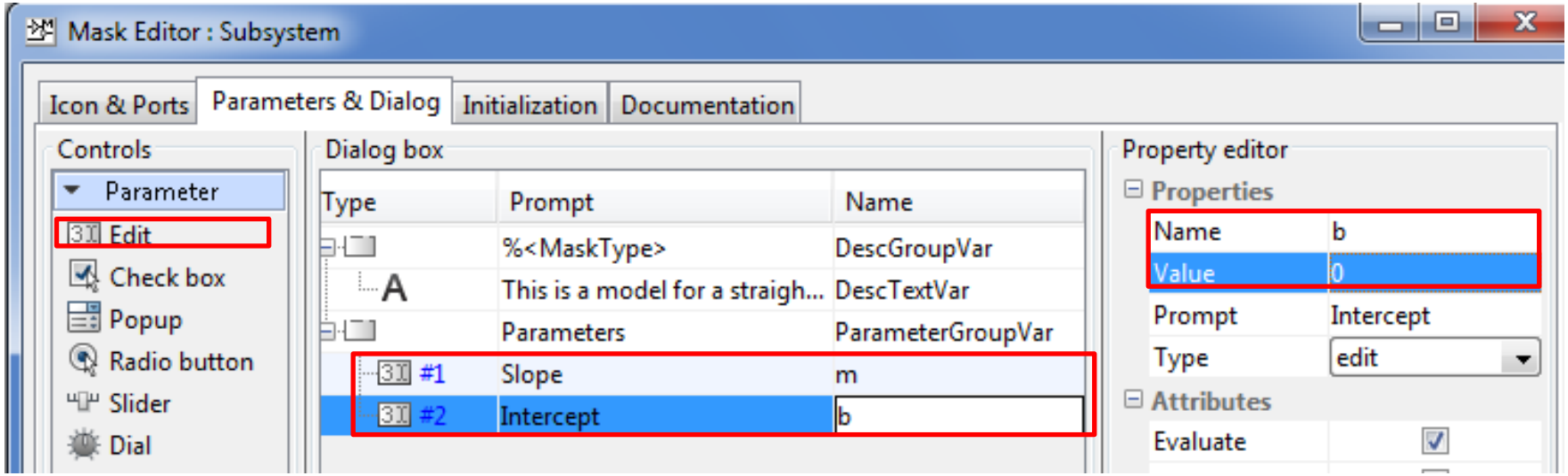
Masking Subsystem

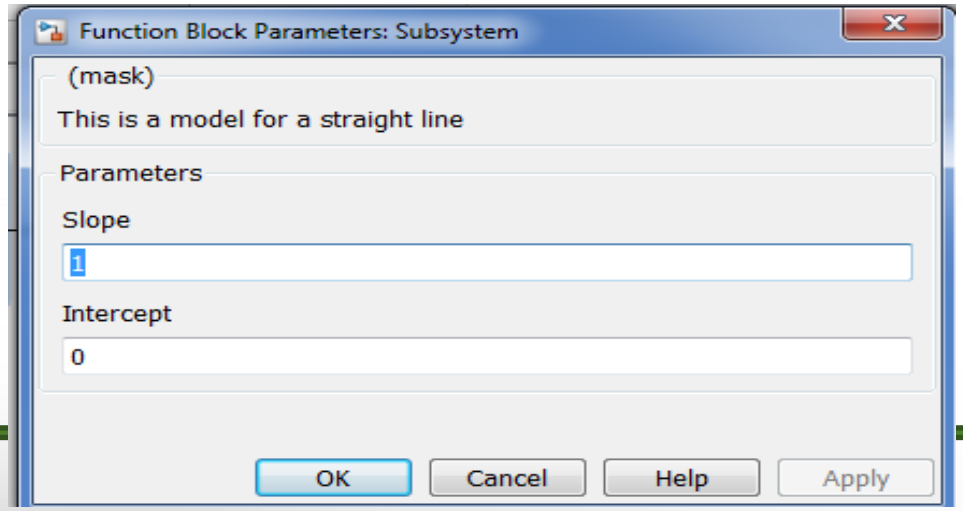
- ➢ Parameters and Dialog Tab
- ✓ Adding Image to the information section



Masking Subsystem

- Parameters and Dialog Tab
- ✓ Adding an input prompt from “Edit”





Masking Subsystem

➢ Initialization

- The Initialization tab allows you to specify initialization commands
- After this, the MATLAB workspace variables are no longer visible

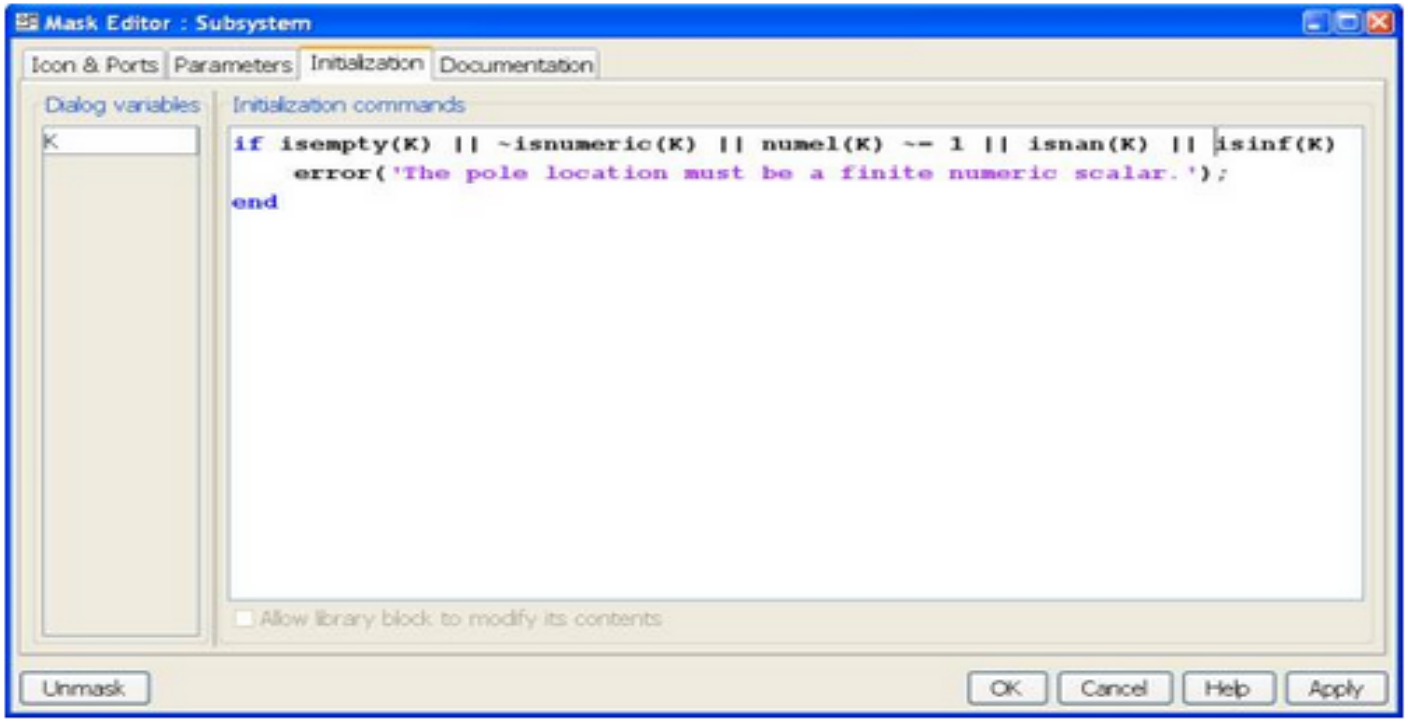
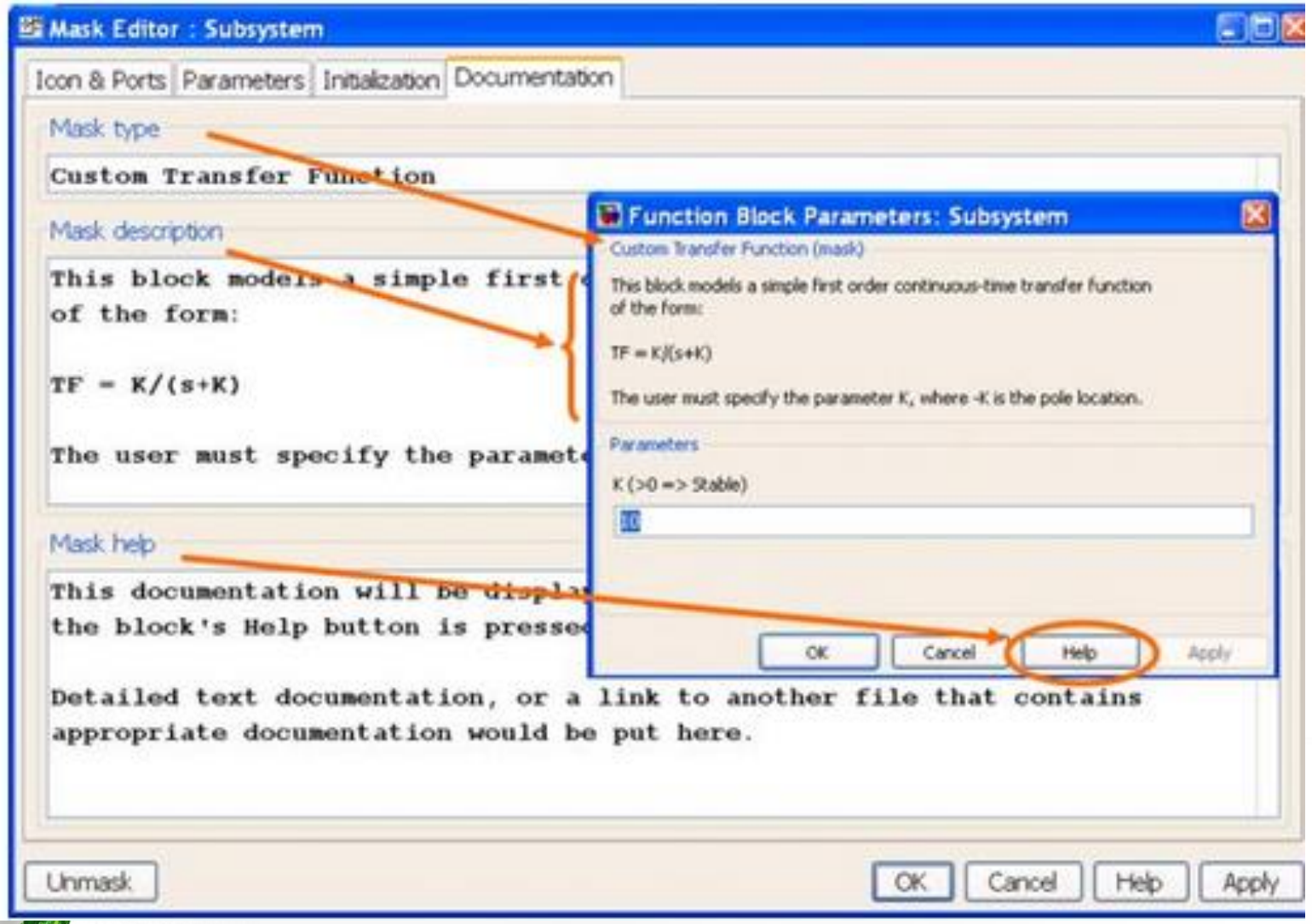


Figure 12: The Initialization Tab.

```
if isempty(K) || ~isnumeric(K) || numel(K) ~= 1 || isnan(K) || isinf(K)
    error('The pole location must be a finite numeric scalar.');
end
```

Masking Subsystem

- Documentation Editor



Running Simulink simulation from a MATLAB m-file

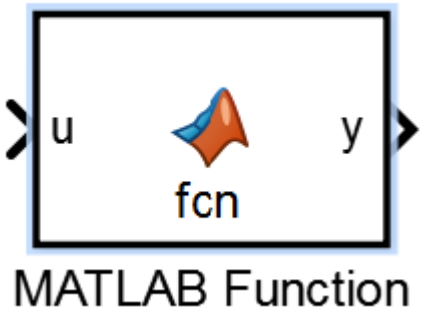
Two commands are used :

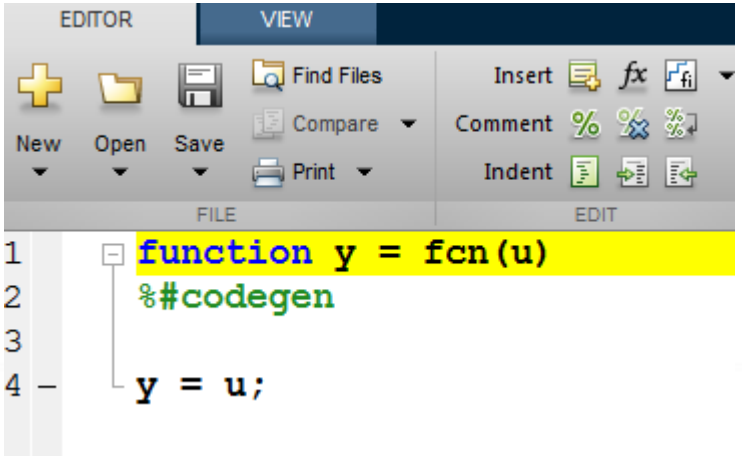
- simset
- sim

```
%% Simulator Settings
   t_stop=100;              % Simulation time
   T_s=t_stop/1000;         % Step Size

%% Simulation options
   options=simset('solver', 'ode5', 'fixedstep', T_s);

%5 Starting simulation
   sim('Simulink_model_name', t_stop, options);
```

Using MATLAB Function Block (old name: Embedded function)

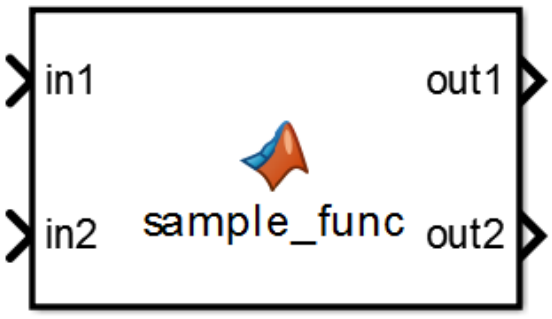
- The MATLAB Function Block is an easy and convenient way to write MATLAB m-code that can be incorporated into a Simulink model.
- The MATLAB Function block is obtained from the *User Defined Functions* Library and is inserted into a model in the same way
- Once in a model the m-code that represents the block's functionality is associated with the block by writing it in the matlab editor and is viewed by double clicking on the block.





Using MATLAB Function Block

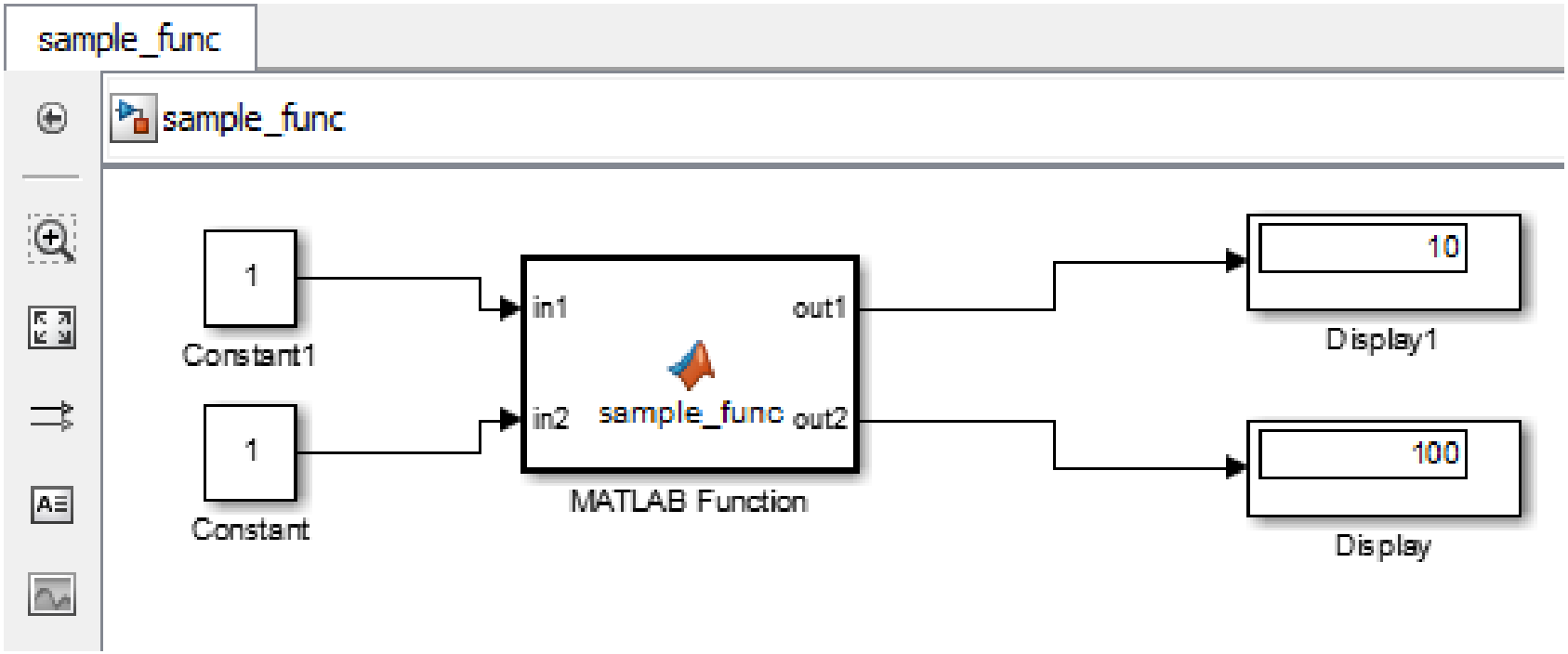
- The function's number of input arguments automatically corresponds to the number of block input ports and
- The function's number of output arguments automatically corresponds to the number of block output ports.
- Similarly the input arguments automatically take on the size and data-type of input signals, and
- The output signals automatically take on the size and data-type of the output variables created in the function.
- This imposes the restriction that the size and data-type of output variables typically needs to be defined at the start of the m-code and not changed during the simulation.




```
1   function [out1, out2] = sample_func(in1,in2)
2   %#codegen
3
4   out1 = in1*10;
5   out2 = in2*100;
6   end
```

Using MATLAB Function Block

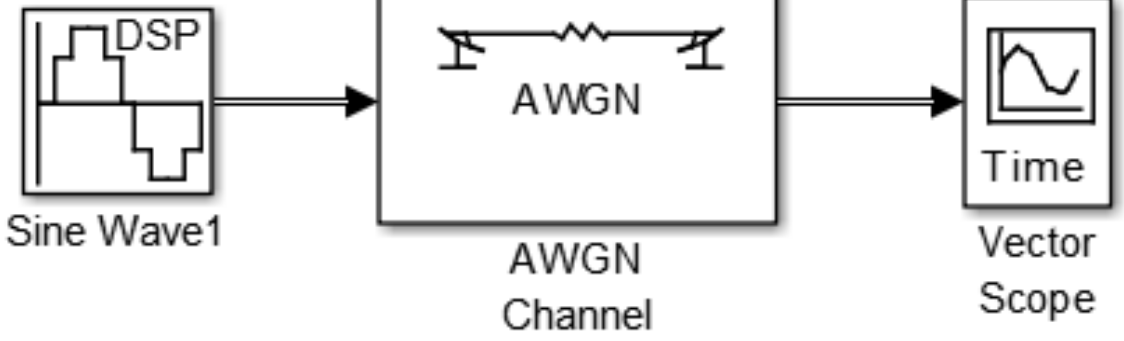
- Example:



Steps for Building a Simple Communication Model

Example of Channel Effect:

- You can add noise to the model using the AWGN Channel block
- The block adds additive white Gaussian noise (AWGN) to the sine wave.

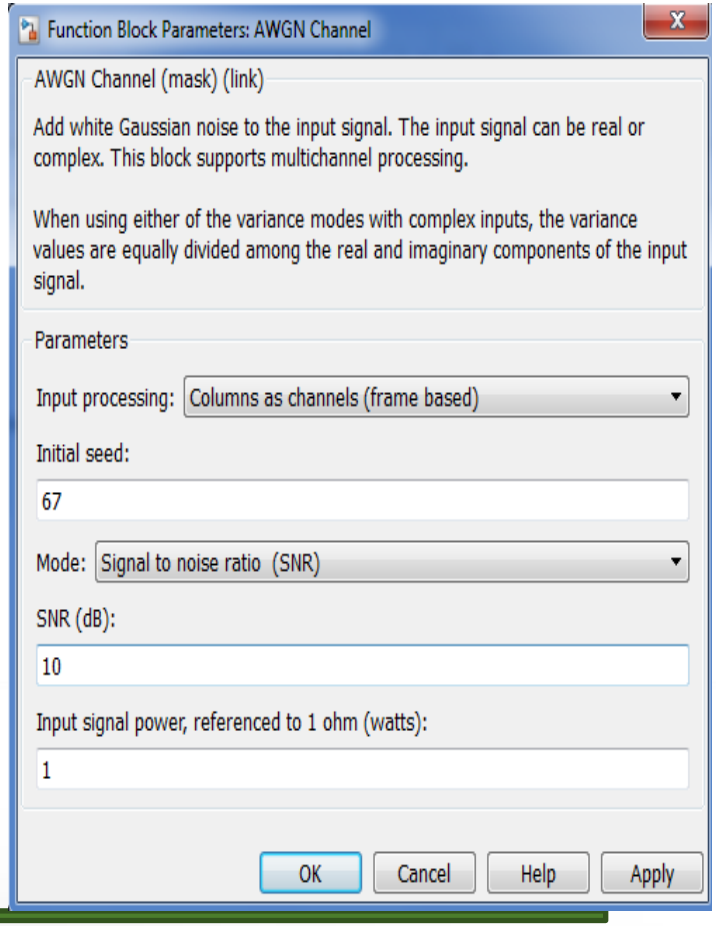


DSP toolbox/ DSP Sources

DSP toolbox/ DSP Sinks

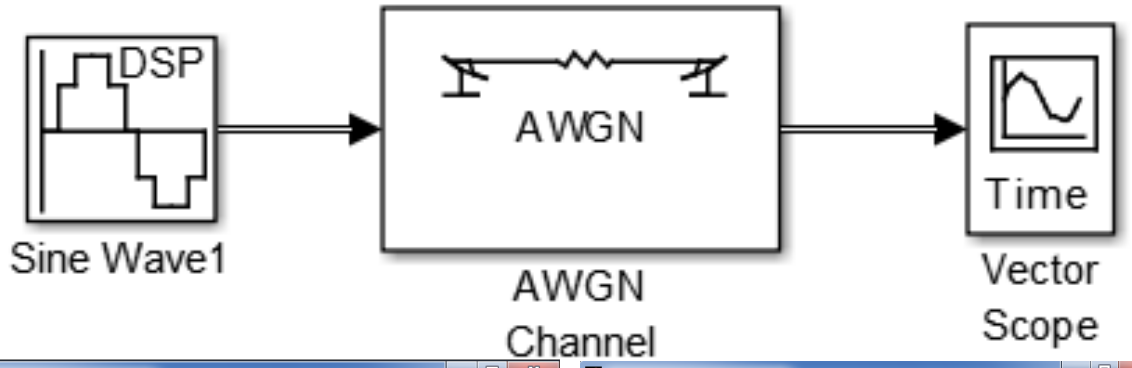
Communications Blockset / Channels library

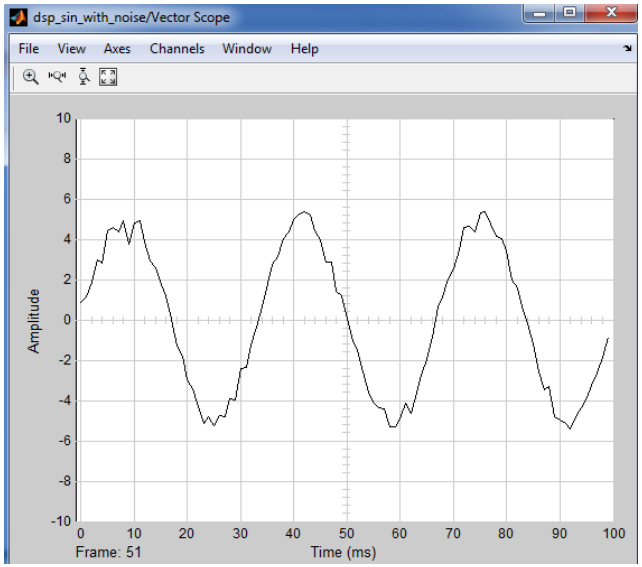
1 Set **Amplitude** to 5.
2 Set **Frequencies** to 30.
3 Set **Samples per frame** to 100.

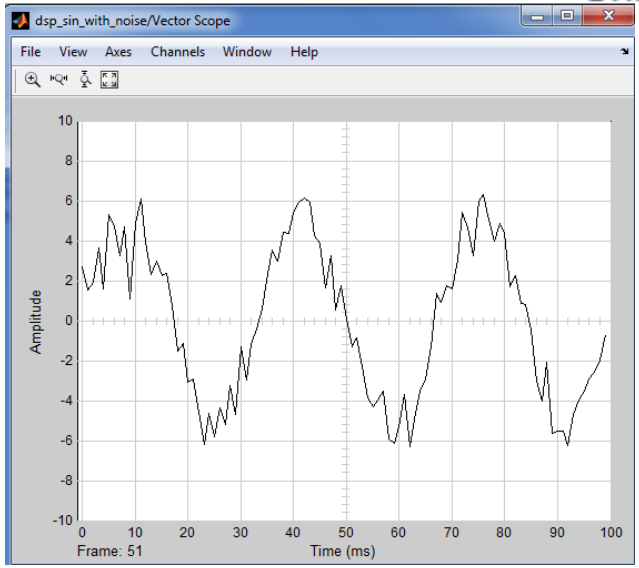


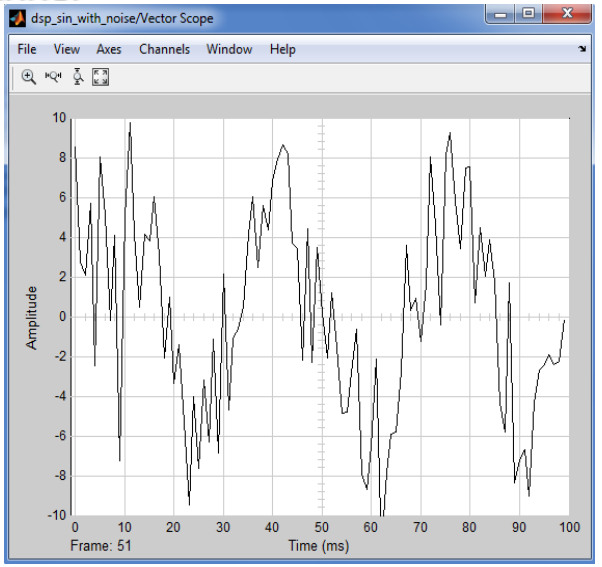
Steps for Building a Simple Communication Model

Example of Channel Effect:









SNR = 10 dB

SNR = 0 dB

SNR = -10 dB