EE 1130 Freshman Eng. Design for Electrical and Computer Eng.

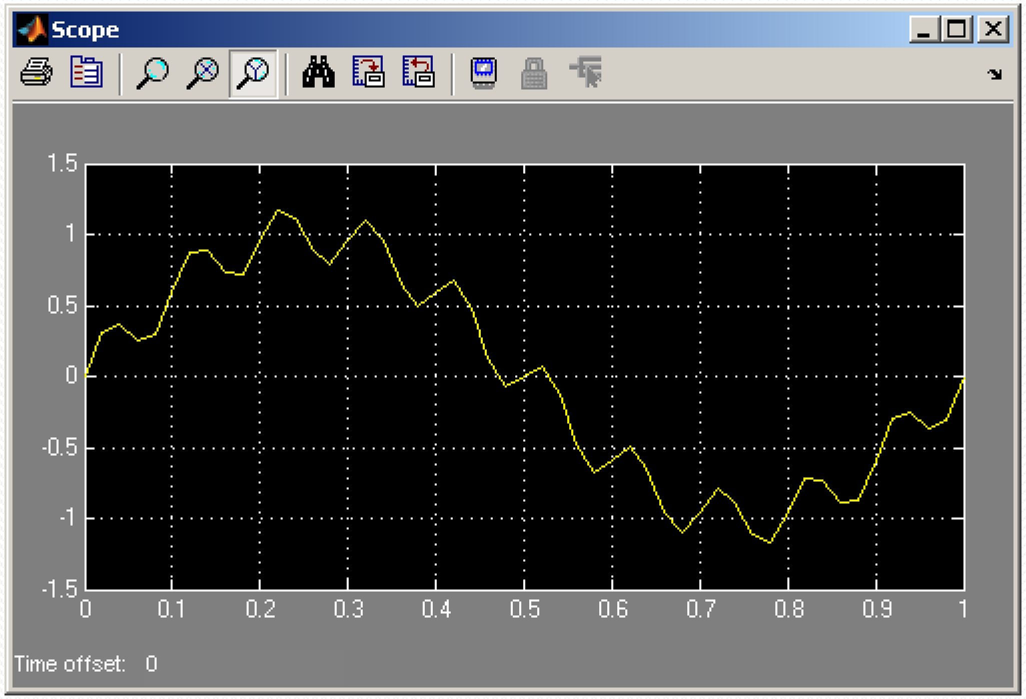
Class 4

Signal Processing Module (DSP).

- Matlab and Simulink.

Simulink: Signal Processing.

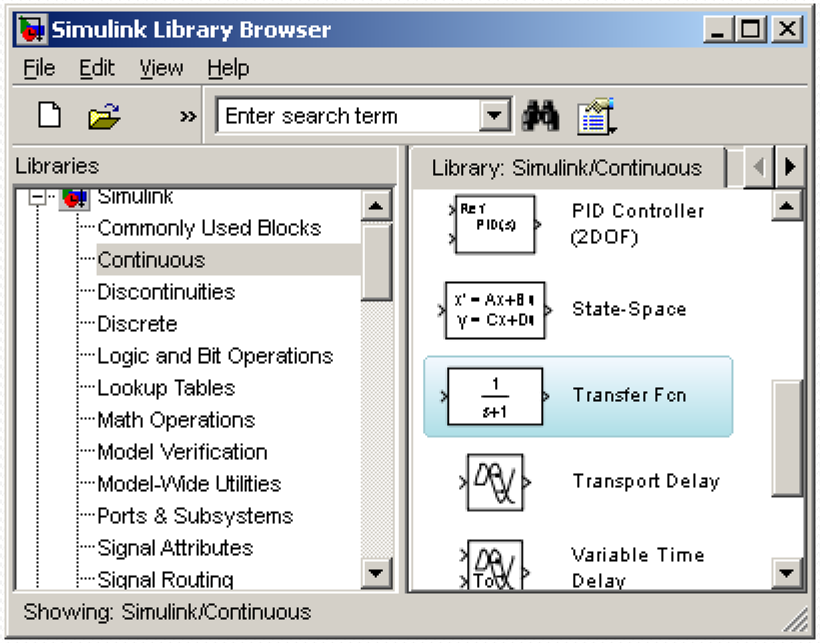
- Last lecture we ended up with a noisy signal as next figure shows:

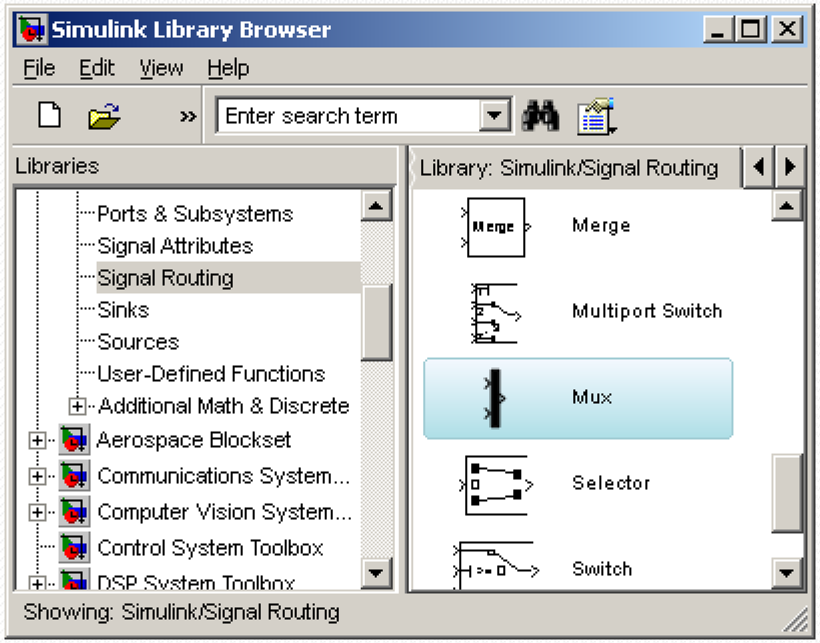


$$x(t) = \sin(2\pi 1t) + 0.2\sin(2\pi 60t)$$

Simulink: Signal Processing.

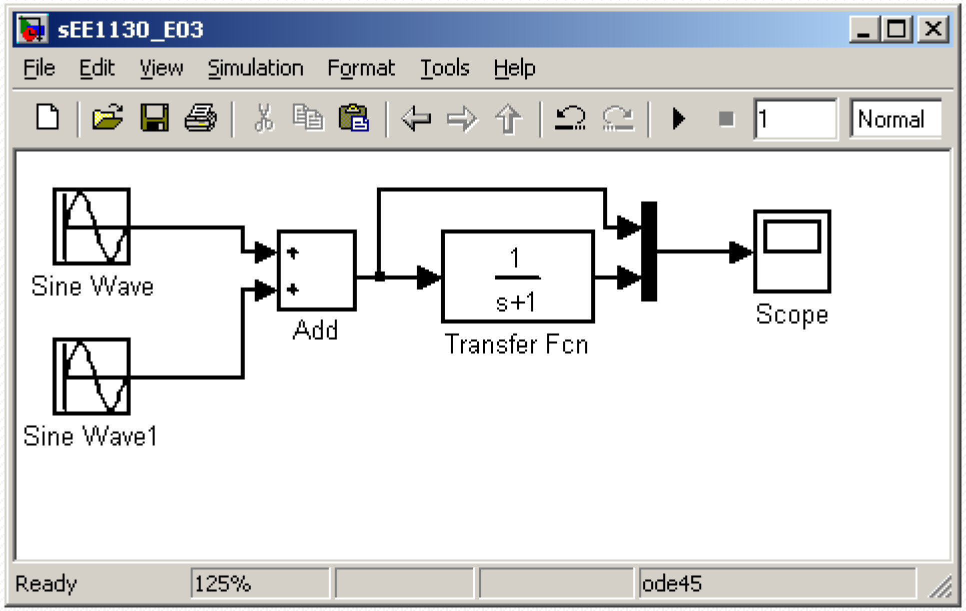
- We will insert a system that will filter out the ripple.
- First option is to insert from the continuous library group a Transfer Function block.
- We also add a Mux from Signal Routing library group.





Simulink: Signal Processing.

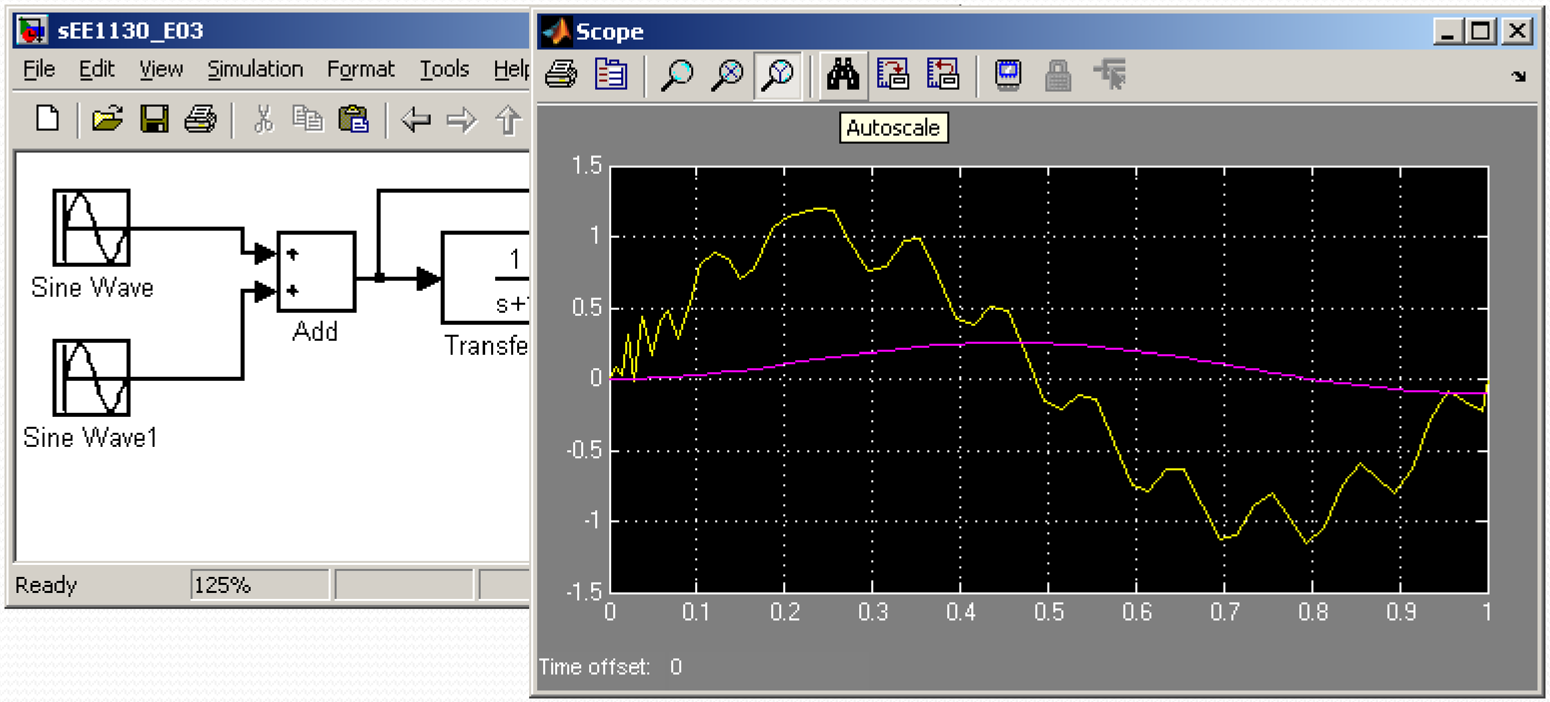
- We insert the Transfer Function after the summator and before the Mux.
- The Mux will allow the Scope to show two traces:



- Now, hit play and see:

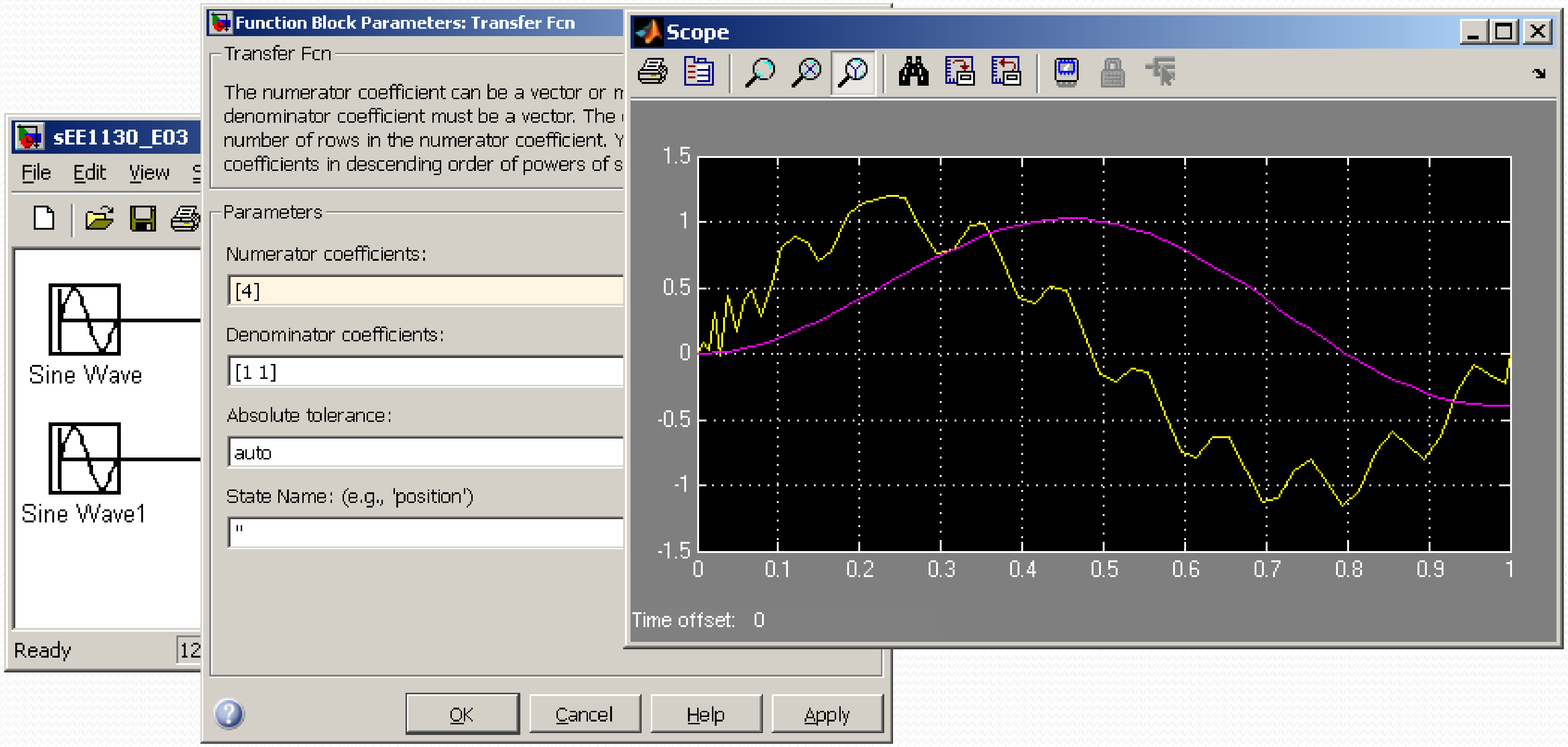
Simulink: Signal Processing.

- Now, hit play and see:
- Somehow we cleaned the signal, but we need to amplify its gain by a factor of 4. We open the transfer function and set 4 the numerator to do this.



Simulink: Signal Processing.

- Somehow we cleaned the signal, but we need to amplify its gain by a factor of 4. We open the transfer function and set 4 the numerator to do this.



The numerator coefficient can be a vector or m
denominator coefficient must be a vector. The
number of rows in the numerator coefficient. Y
coefficients in descending order of powers of s

Parameters

Numerator coefficients:
[4]

Denominator coefficients:
[1 1]

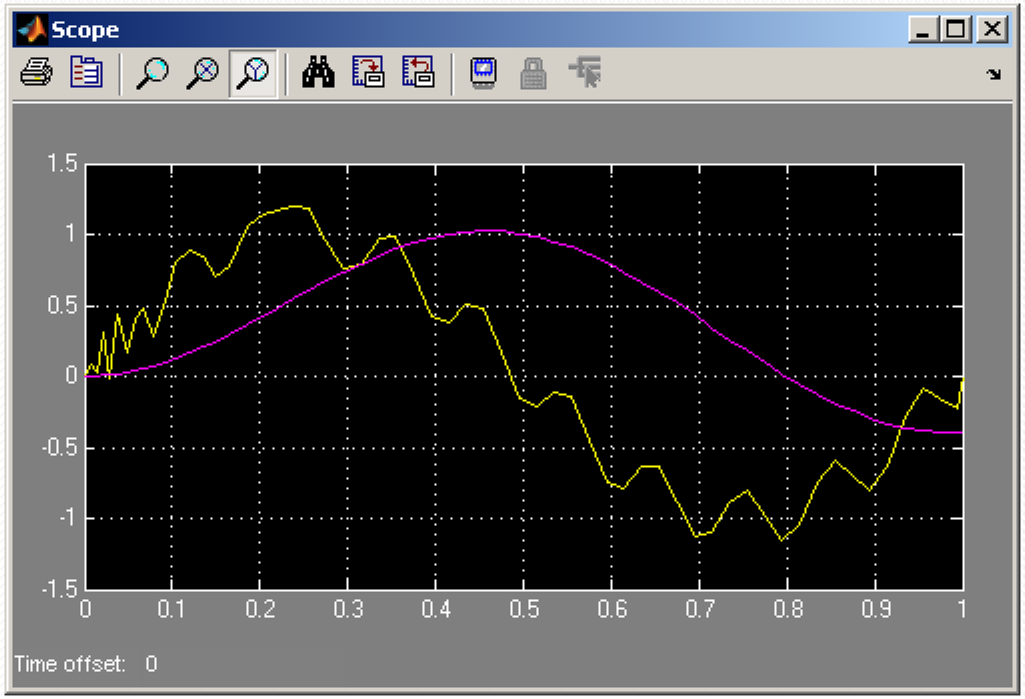
Absolute tolerance:
auto

State Name: (e.g., 'position')
''

Time offset: 0

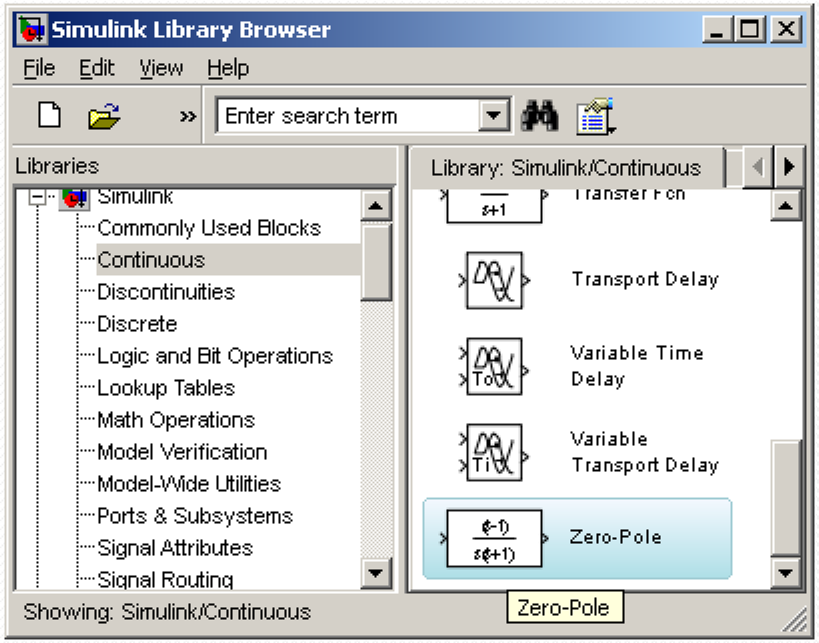
Simulink: Signal Processing.

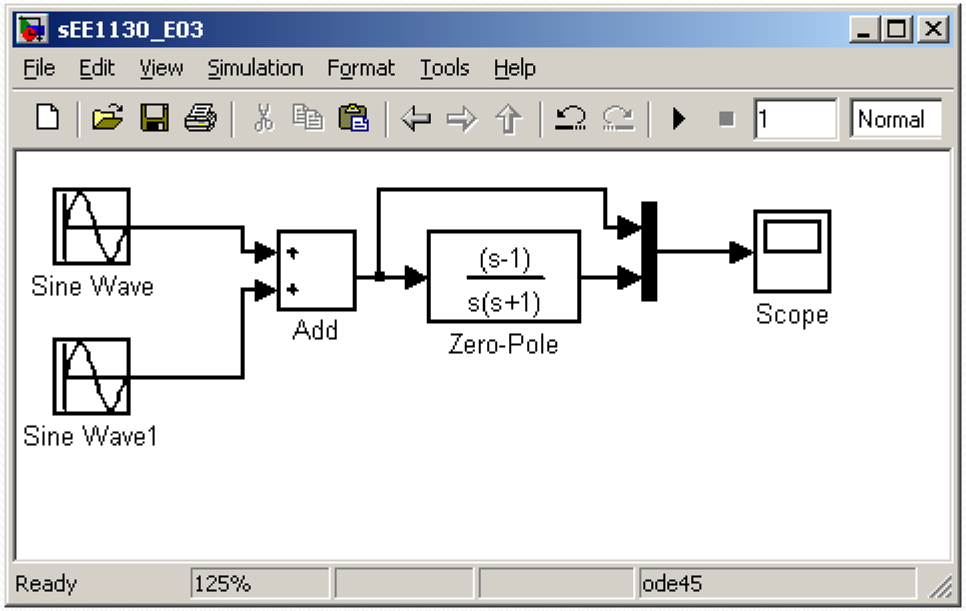
- We have cleaned the signal, but introduced a time delay (time shift). This is common in any kind of filters.



Simulink: Signal Processing.

- Now, lets design a filter that particularly eliminates the signal of 60Hz. We do that using the Zero-Pole Transfer function





Simulink: Signal Processing.

- When studing Filter Theory you will learn that the roots of the numerator must be (s-$2\pi 60j$) and (s+$2\pi 60j$). The use of complex conjugated roots is to have real coefficients because:

$$(s-2\pi 60\,j)(s+2\pi 60\,j)=s^2+4\pi^2 60^2$$

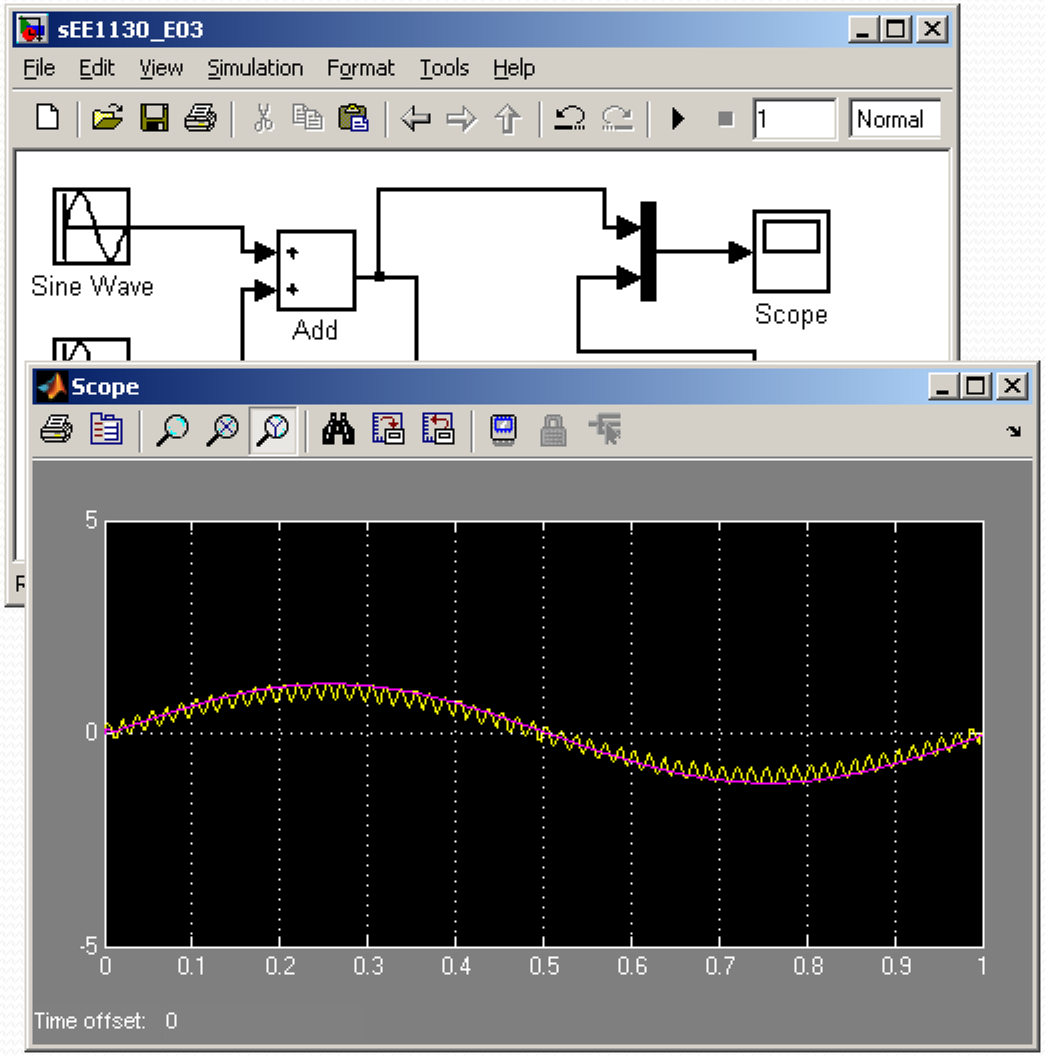
- At the denominator we just set roots to.

$$(s+350)(s+350)$$

- If you set smaller roots, the output becomes too large. Please try other values to check out by yourself

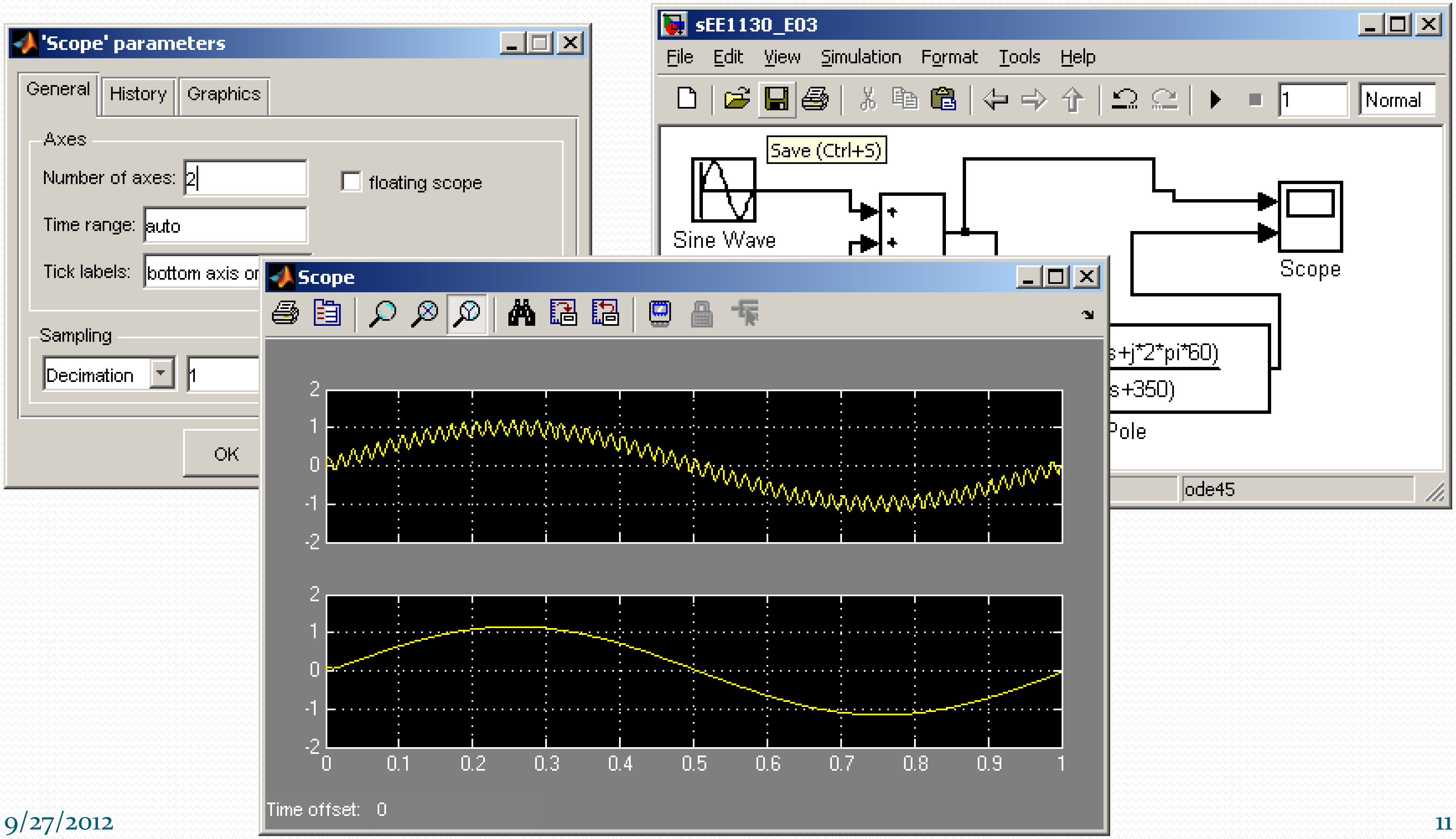
Simulink: Signal Processing.

- Now we hit play and compare input and output in the Scope



Simulink: Signal Processing.

- We notice the dark trace is completely clean of noise. We could add another trace to the scope and see both signals separated:


'Scope' parameters

General | History | Graphics

Axes

Number of axes: 2

floating scope

Time range: auto

Tick labels: bottom axis or

Sampling

Decimation 1

Simulink: Diff Equ.

- As a last example lets implement the output of a First Order Differential Equation:

- The objective is to analyze the output of the following Differential Equation

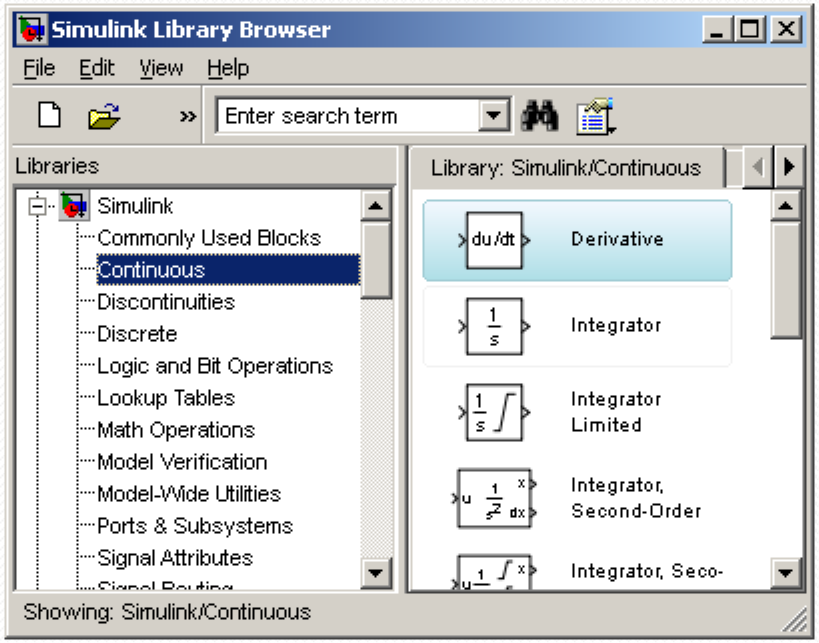
$$\frac{dy}{dt} = -2y(t) - 1$$

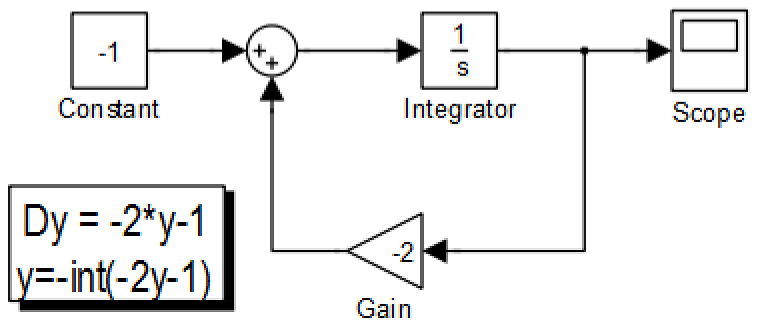
- This equation is equivalent to the following integral equation:

$$y = \int \left(-2y(t) - 1\right) dt$$

Simulink: Diff Equ.

- Next figure shows the equivalent block diagram:
- Here we inserted a constant box (from Sources Library), an integrator (from Continuous), a scope (from Sinks), and a gain and summator (from Math).

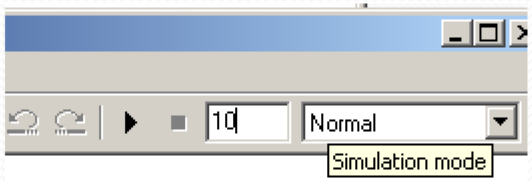


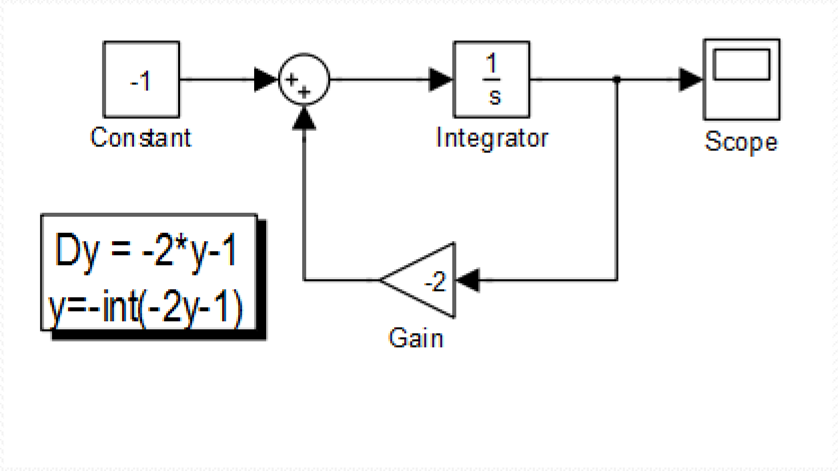


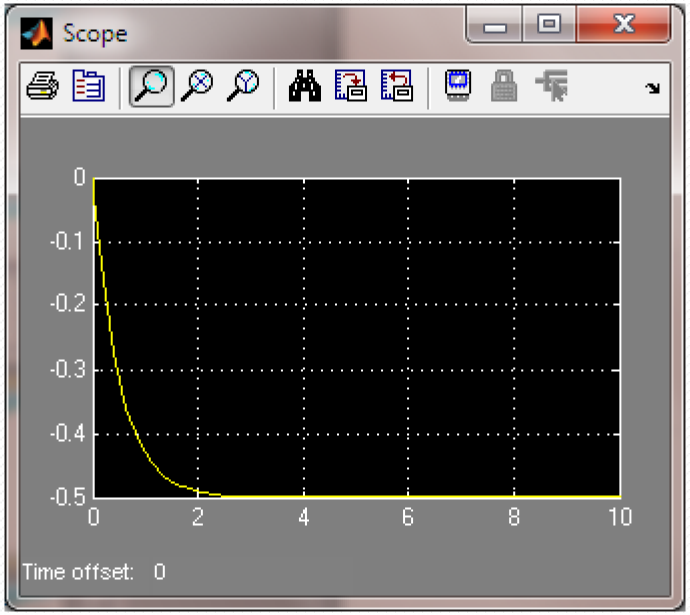
- We also created an information box by double clicking anywere in the blank space and start typing.

Simulink: Diff Equ.

- To open the scope we double click on the Scope box.
- We hit the play icon to run the simulation.

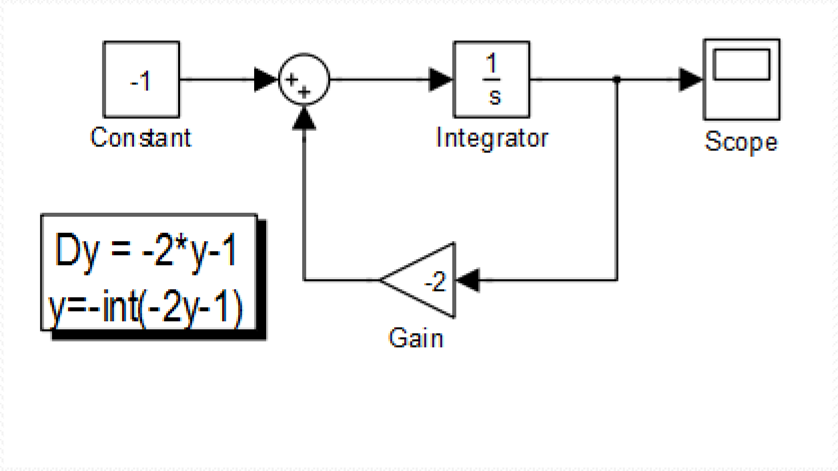


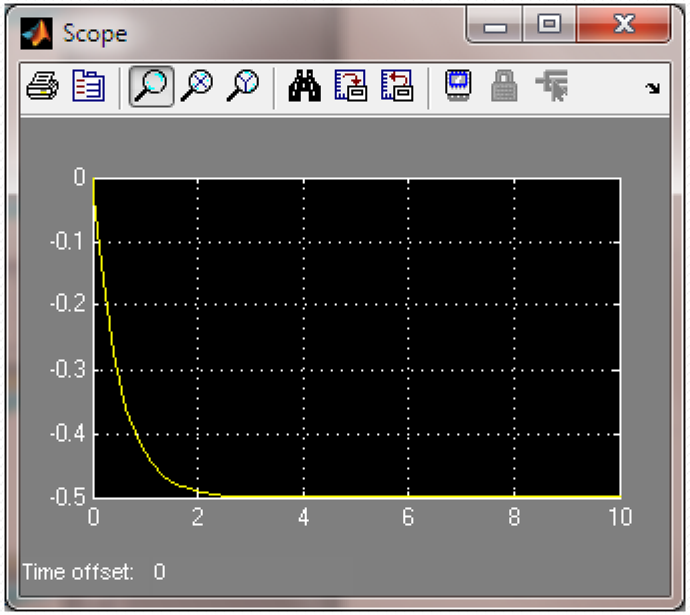




Simulink: Diff Equ.

- The solution of this Differential Equation is a decaying exponential function as shown in the Scope figure.
- Simulink is good to evaluate Differential Equations and much more!!!





End of Class