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

Class 2

Signal Processing Module (DSP).

- Matlab and Simulink.

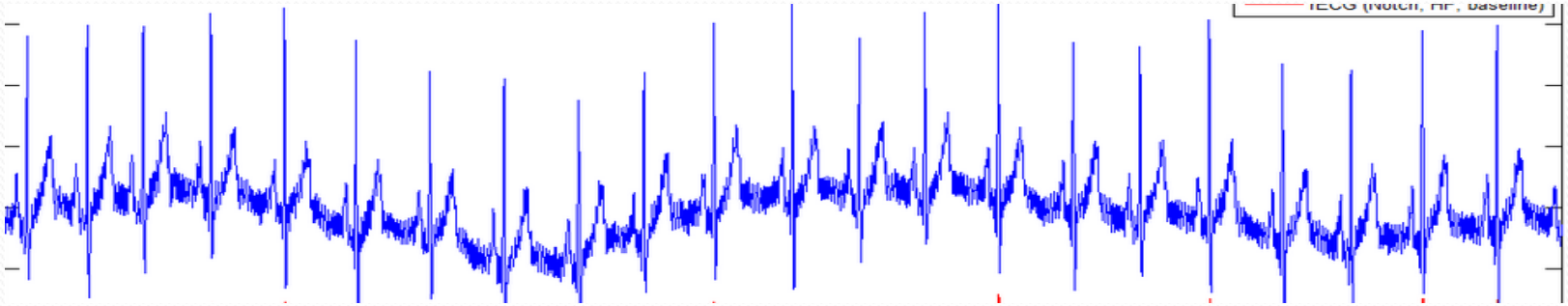
Signal Processing Engineer

- We are detectives of hidden information in signals.
  - Communication signals: obtain the information.
  - Signals from CO2 sensors, heat sensors, etc.
  - Data from Hard Drives.
  - Data from Computers to computers.

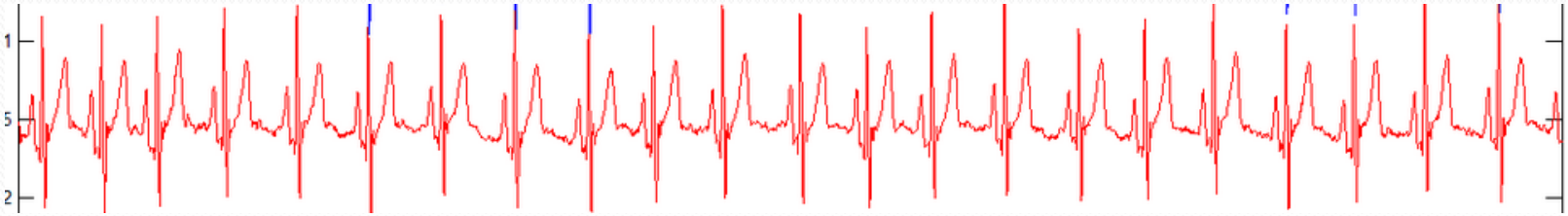
- In order to do that we need to master MATHEMATICS!!

Problem Statement

- Imagine you are an Electrical Engineer Signal Processing proficient. A medical doctor running a Hospital building hires you to solve the following problem:
  - Doc: “Our EKG readings show a garbled signal and I can not diagnose my patients correctly”.

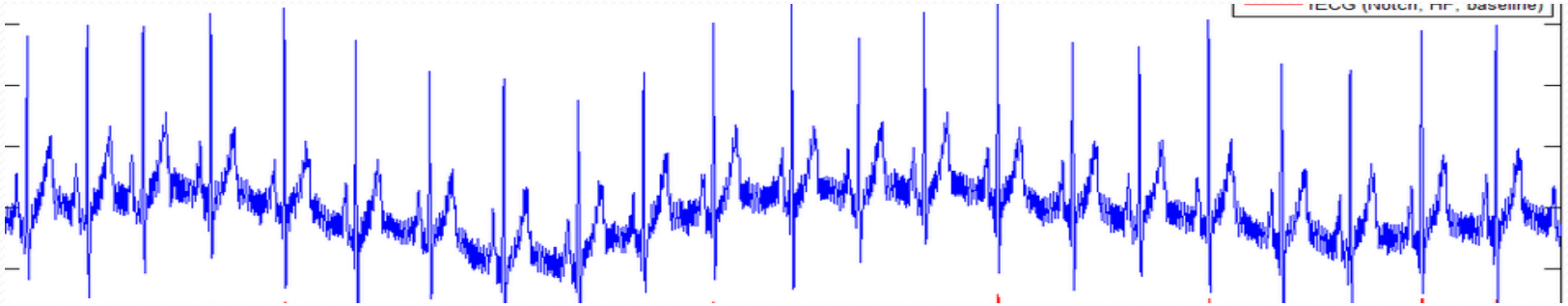


  - He is expecting this kind of signal:

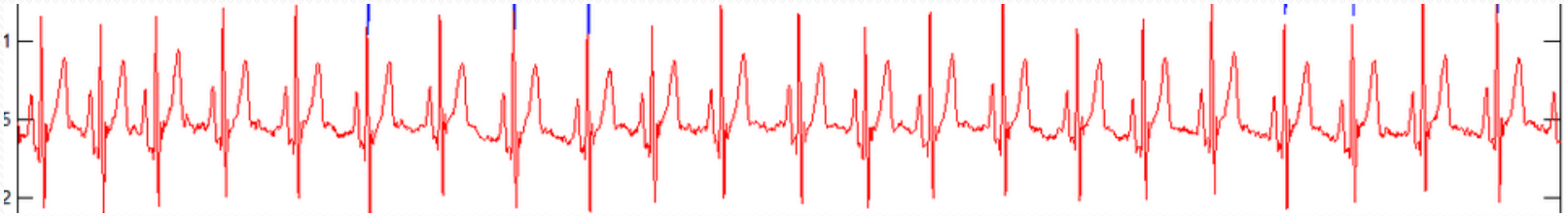


Problem Statement

- What is the difference?.



  - The above signal has and ondulation (low frequency envelope)
  - Besides that, it is noisy (has a lot of ripples or high frequency component)



Matlab

- Matlab is a powerful tool for mathematical/engineering research and development. It is also useful to students to easily computate or solve almost all mathematical and engineering problems.
- This Signal Processing Module will use Matlab as a development and teaching tool.
- Matlab is learned in the course **EE 3220 Software Applications** in Electric Engineering.

Matlab

- We will use the **Simulink** part of Matlab for two main purposes inside this DSP Module:
  - We will emulate/modelate a low frequency signal corrupted with an additive noise (high frequency signal).
  - Design and implement a FILTER that will eliminate a high frequency component (ripple or noise) meanwhile leaving untouched a low frequency sine wave.
- This phenomenon is common in any electrical system, where the 60Hz signal from the power lines corrupt a signal of interest as an Electro EncephaloGram (EEG), Electro CardioGram (EKG), or just a sinewave.

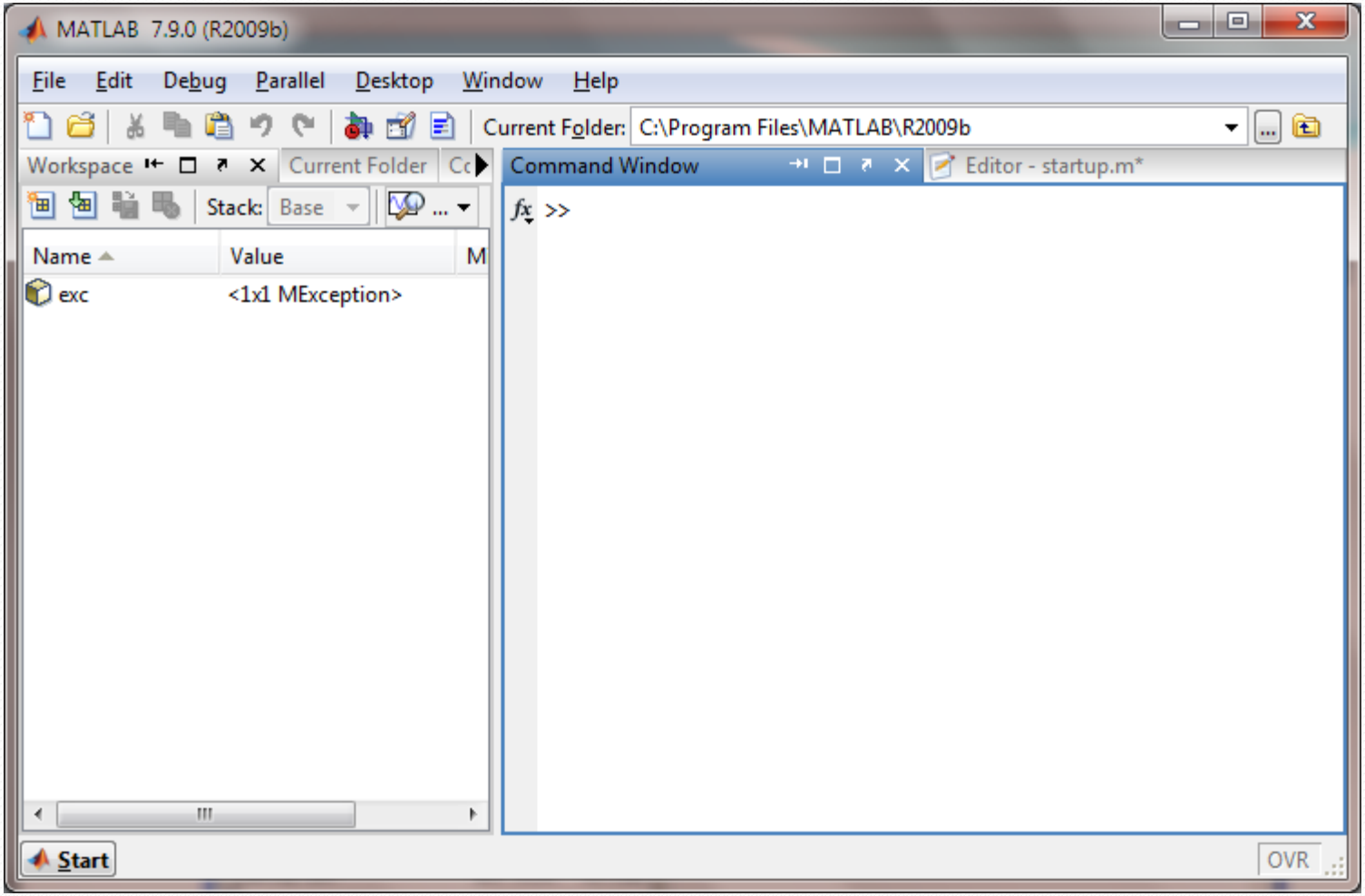
Matlab

- To run Matlab, just double click on the matlab icon.

MATLAB 6.5.lnk

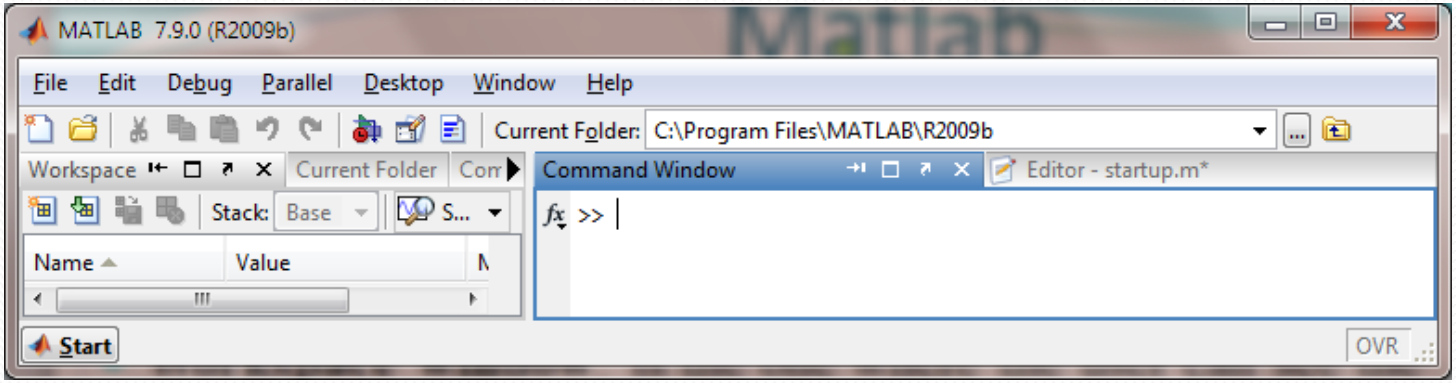
Matlab

- The Matlab window application will appear. We can see different sub-windows:

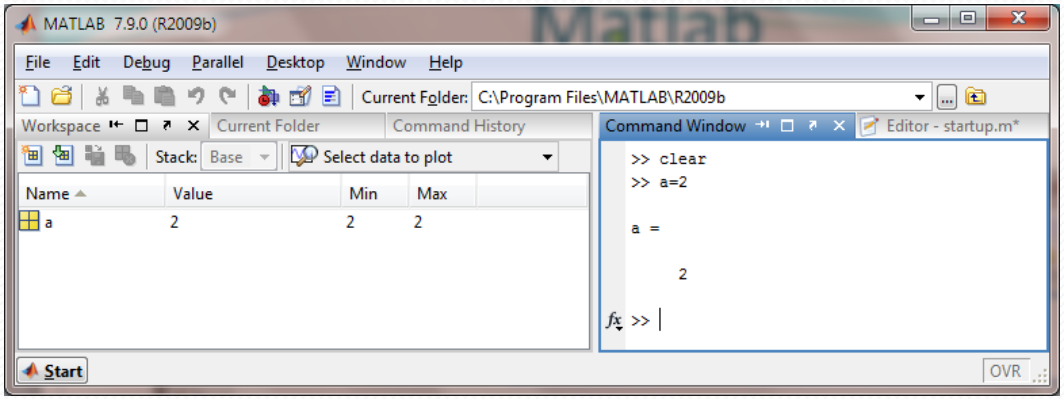


Matlab

- **Command window**: is the one where the user writes the variables and where the results are displayed

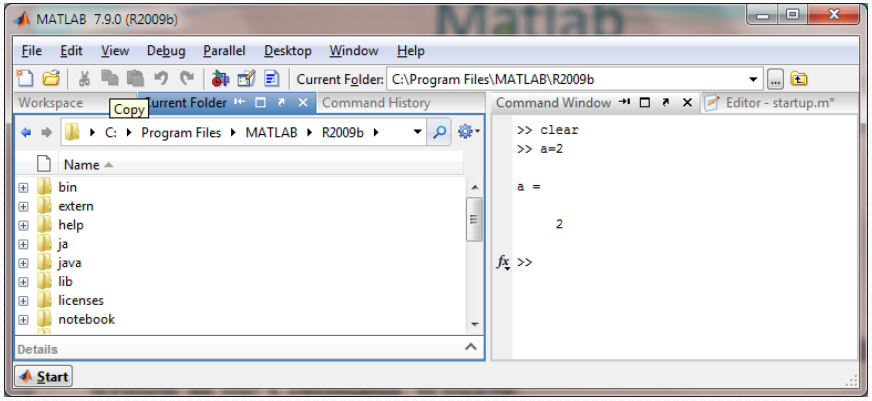


- **Workspace window**: is the one where the user can see the variables.

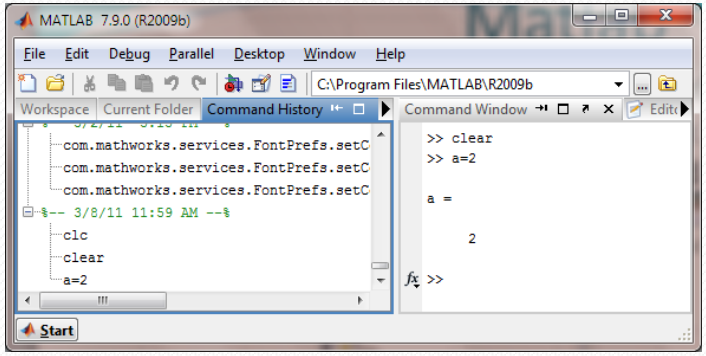


Matlab

- **Current Folder window**: is the one where the user can see the different files stored in the current directory. The current directory is the one that Matlab uses to store and read files.



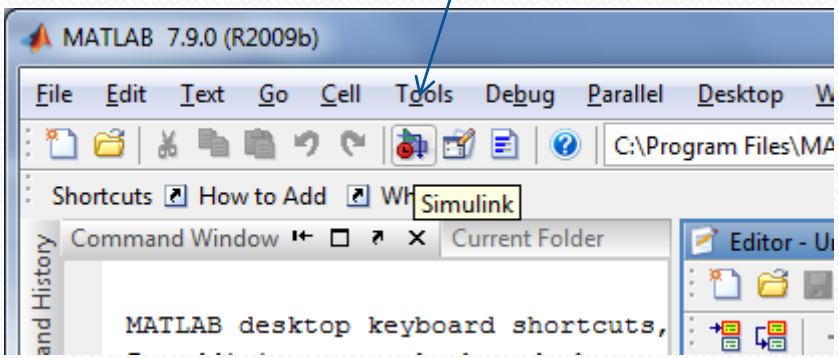
- **Command History window** that shows us all the commands written in the Command Window.

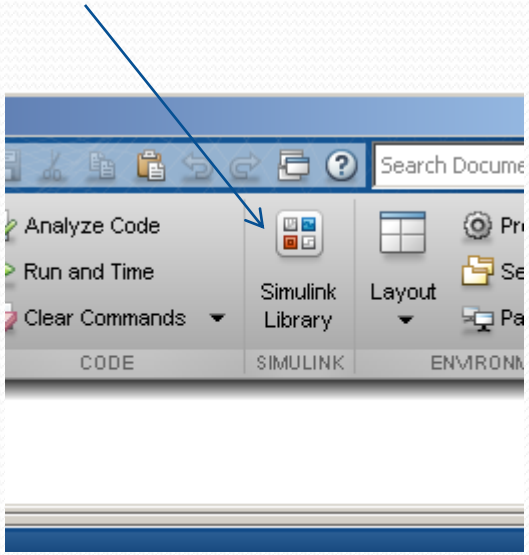


Simulink

- Simulink is part of Matlab. Simulink works by interconecting blocks. Each block is in fact a matlab function with input and output parameters.

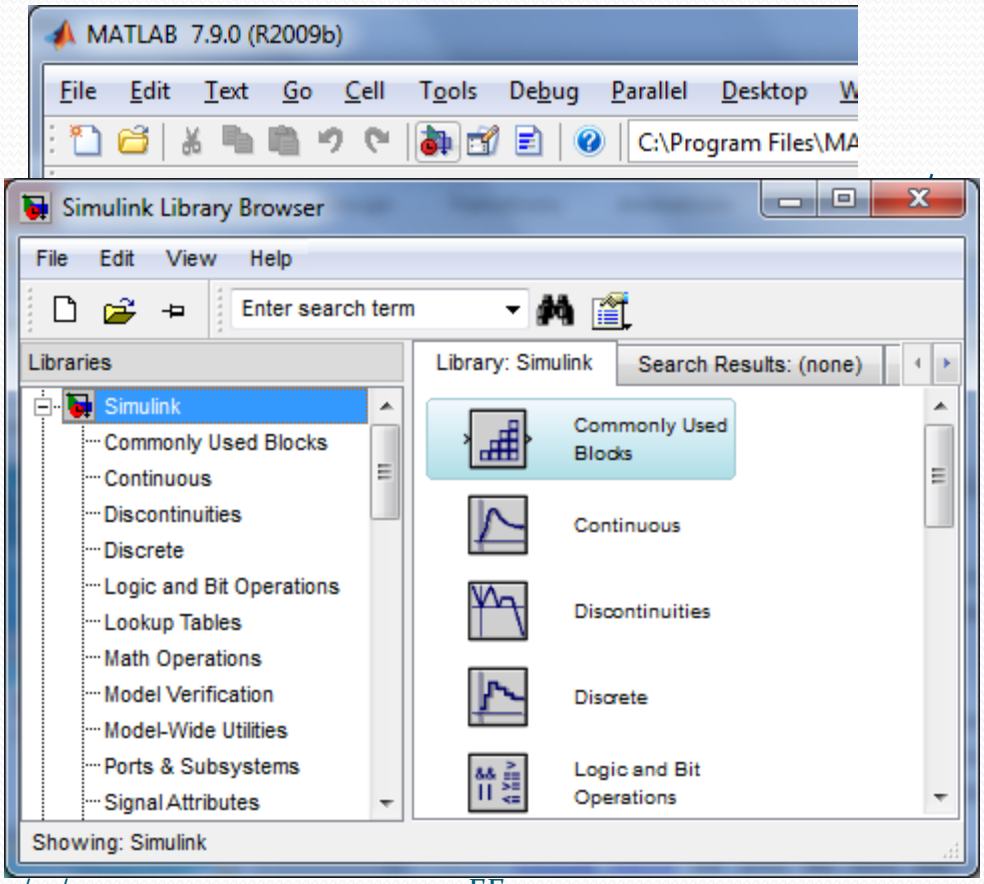
- To open simulink click on the Simulink icon as shown:

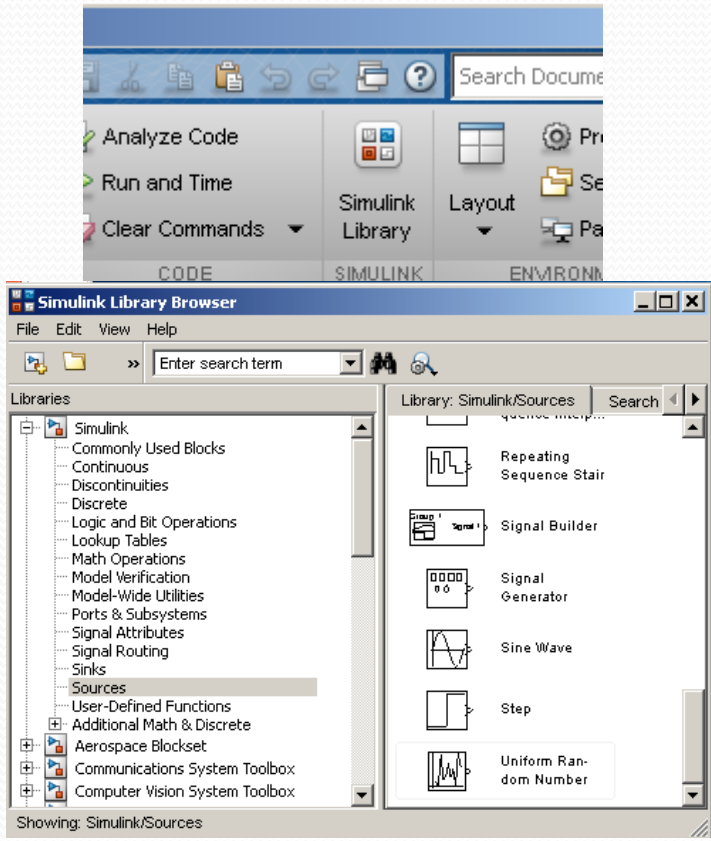




Simulink

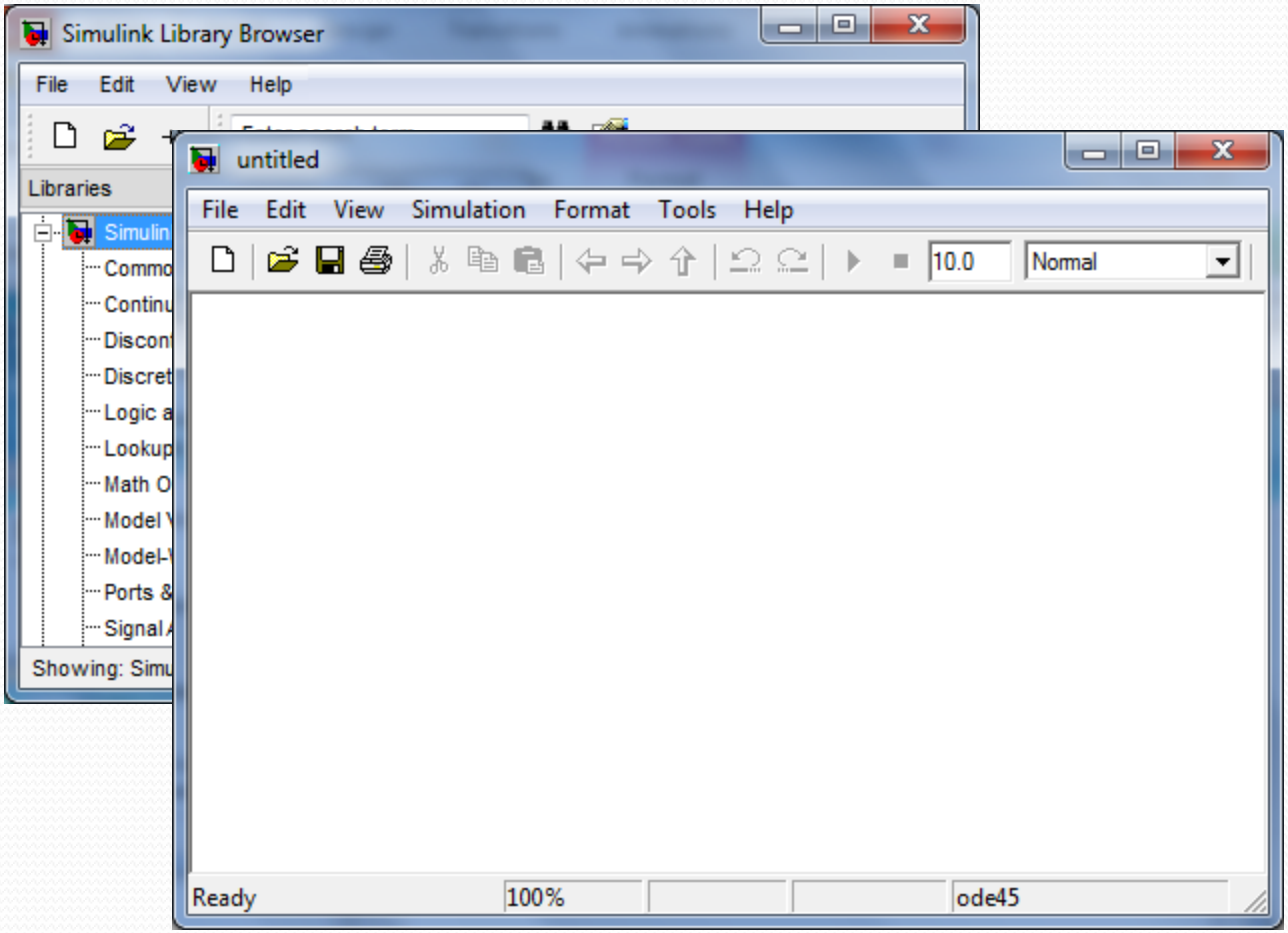
- A new window appears, the library Browser: Notice Continuous, Math Operations, Source, and Sink groups that we will be using





Simulink

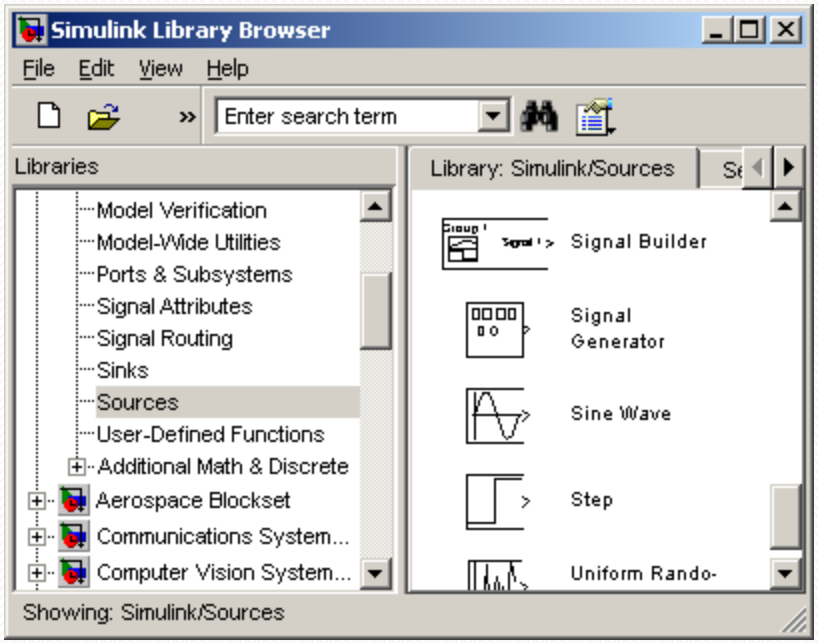
- Click on the new document to open:

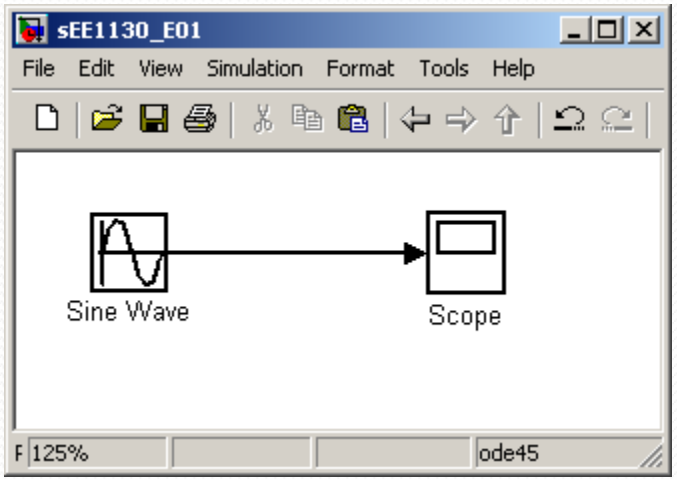


Simulink: Sinewaves.

- Lets build an easy example of a sinewave and scope it:

$$x(t) = \sin(2\pi ft) \quad \text{with} \quad f = 1$$



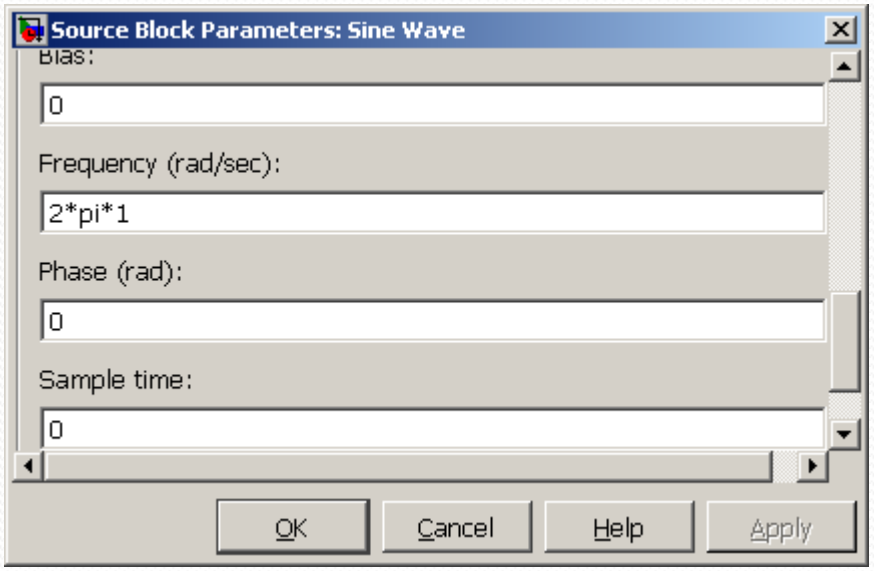


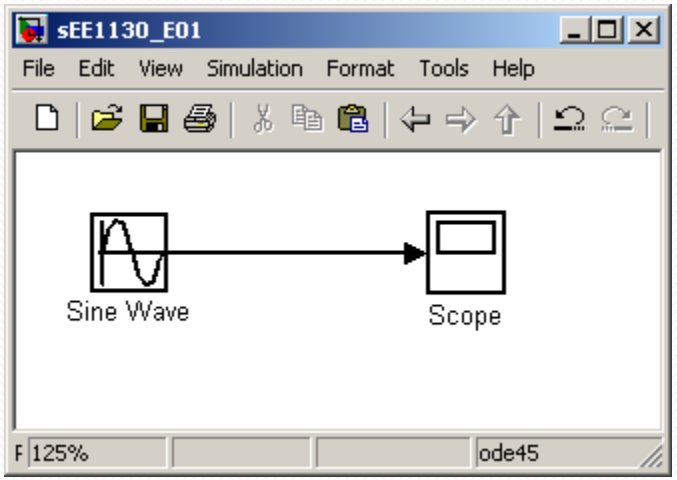
- We inserted a sinewave from the sources, and a scope from the sink library groups respectively.

Simulink: Sinewaves.

- By double clicking on the Sine Wave box we open its properties box and set the angular frequecy to 2*pi*1:

$$x(t) = \sin(2\pi f t) \quad \text{with} \quad f = 1$$

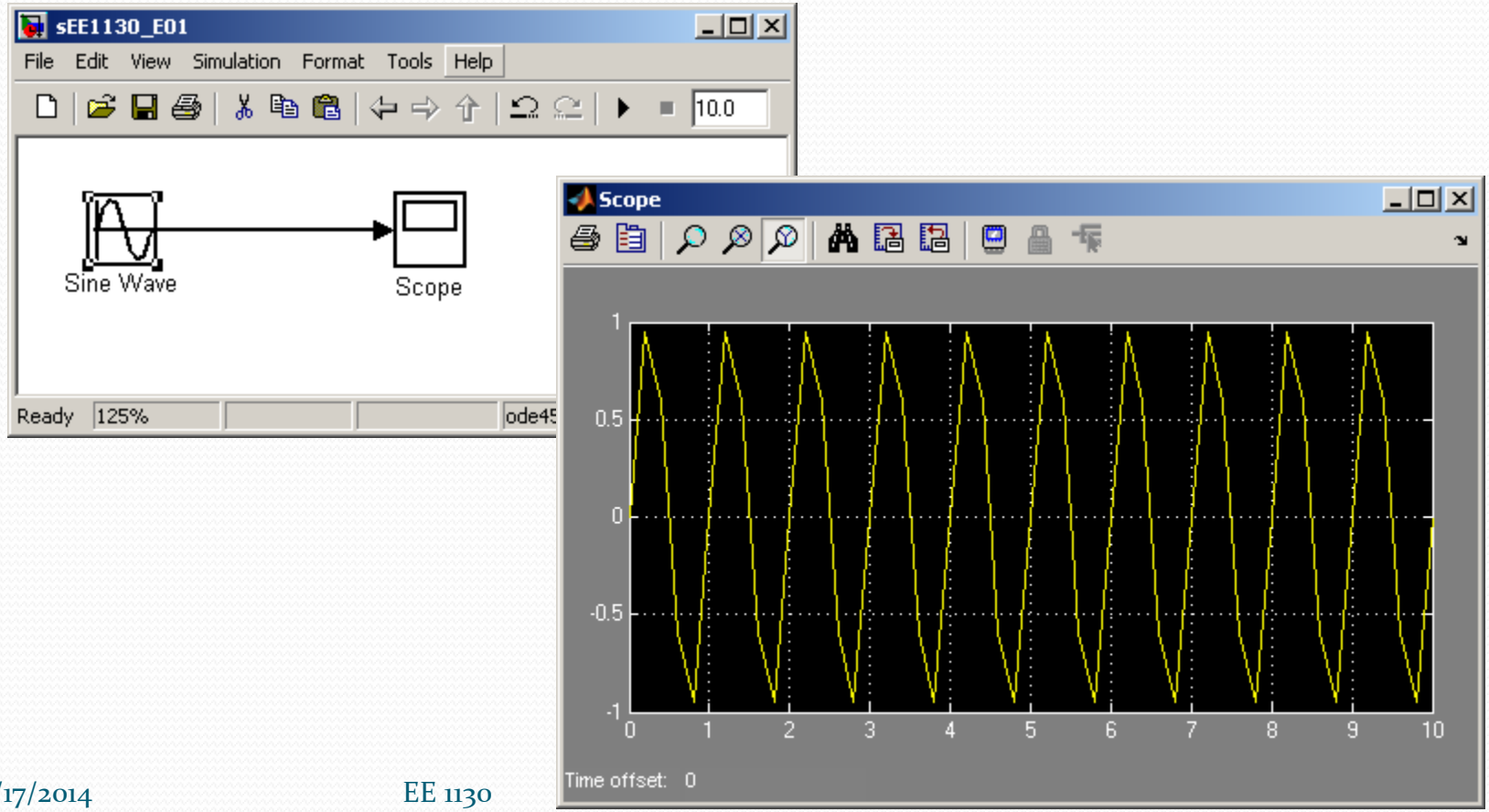




Simulink: Sinewaves.

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

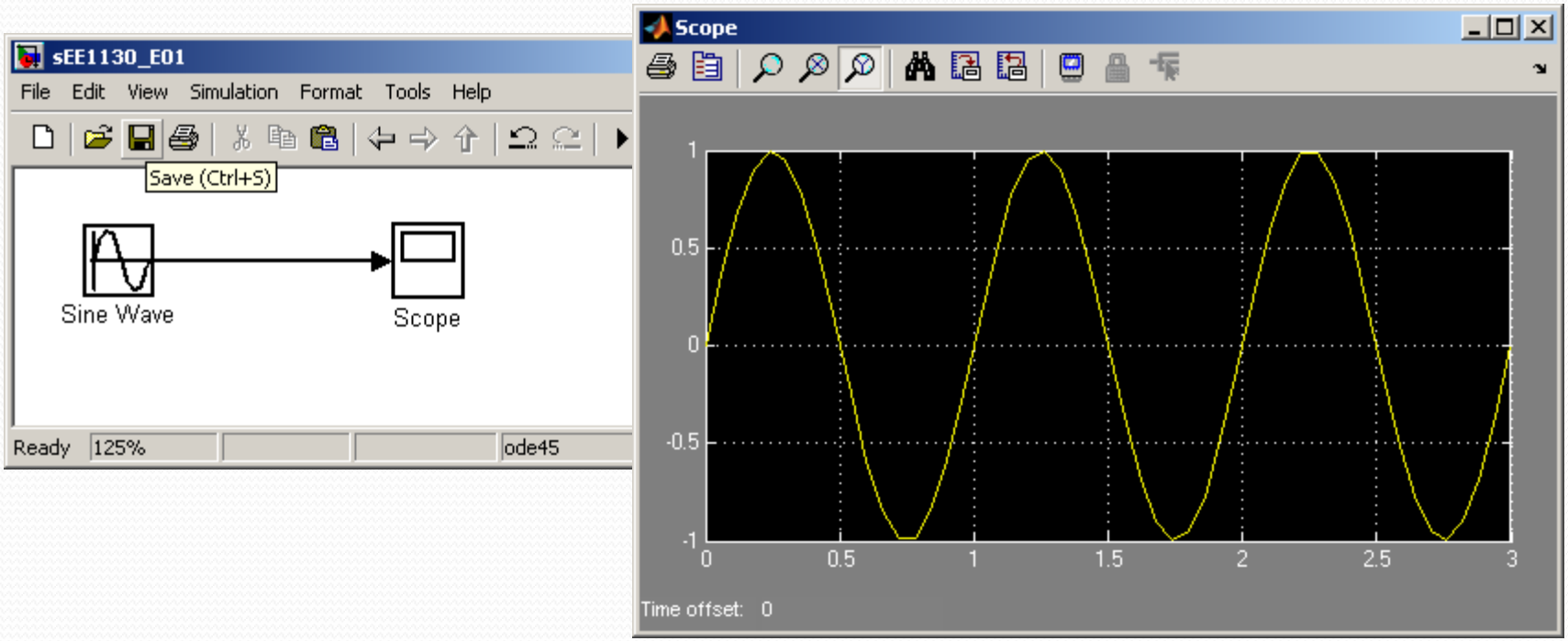
$$x(t) = \sin(2\pi f t) \quad \text{with} \quad f = 1$$



Simulink: Sinewaves.

- To obtain a better view of the sinewave, we reduce the running time to 3 seconds as shown in next figure:

$$x(t) = \sin(2\pi ft) \quad \text{with} \quad f = 1$$

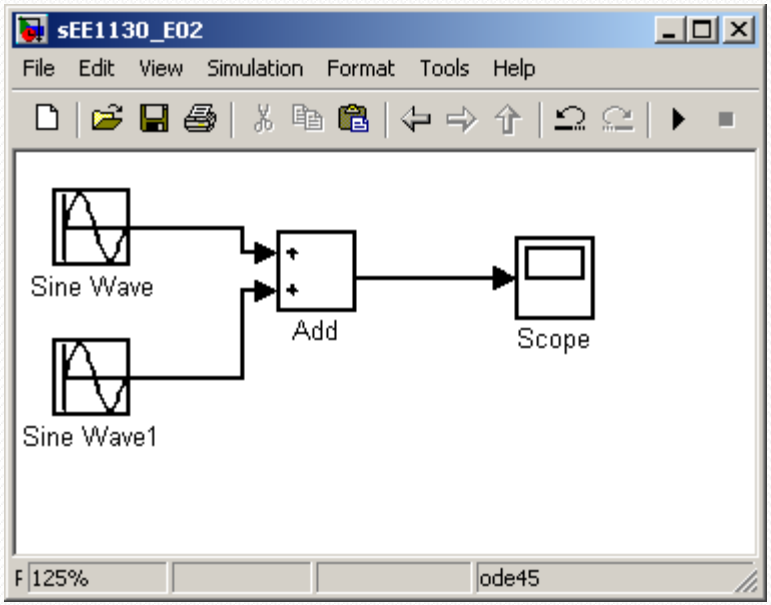


- Click on the binoculars to zoom the signal so it fills the Scope.

Simulink: Sum of Sinewaves.

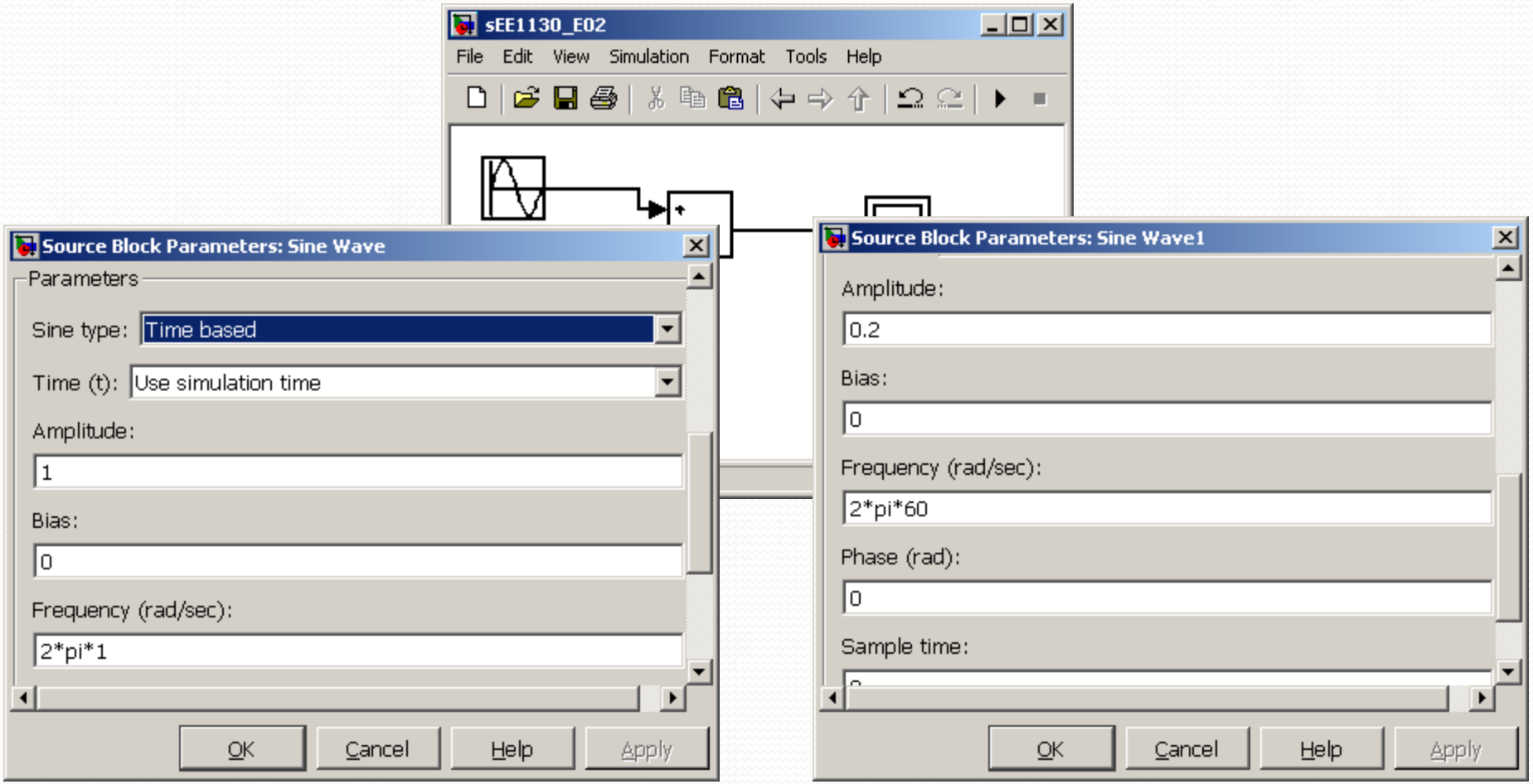
- Lets perform a summation of two sinewaves. One of 1Hz of frequency and 1 volt of amplitude and another of 60Hz frequency and 0.2 volts of amplitude:

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



Simulink: Sum of Sinewaves.

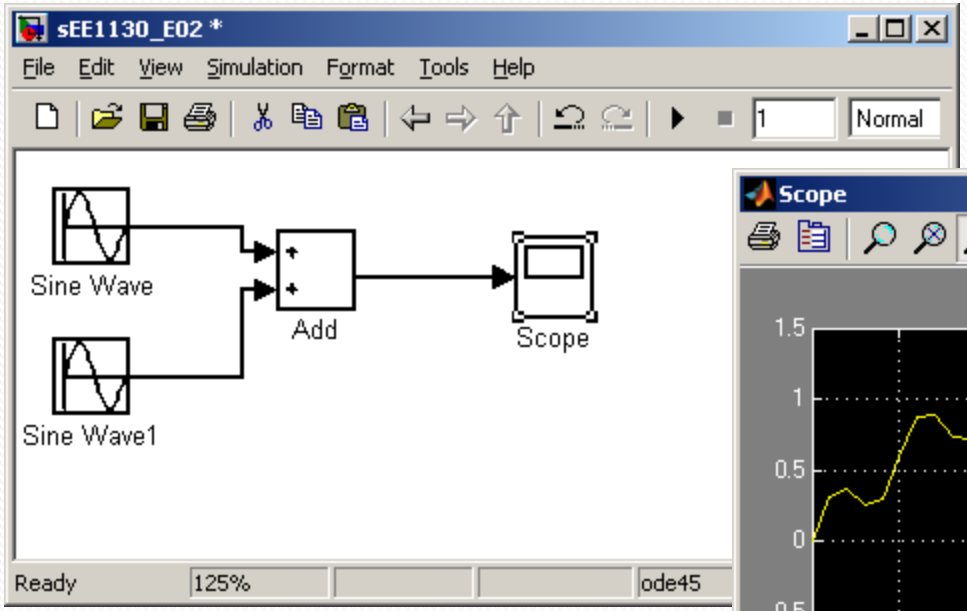
- We need to click on each sinewave box to set up the amplitude and frequencies: $x(t) = \sin(2\pi 1t) + 0.2\sin(2\pi 60t)$

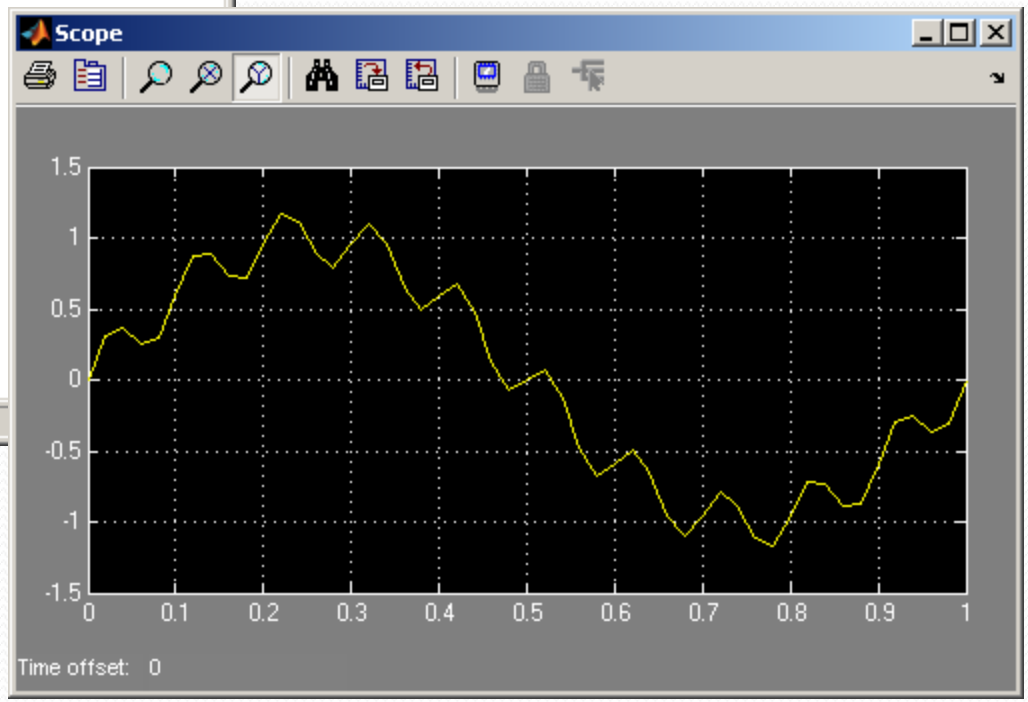


Simulink: Sum of Sinewaves.

- Set the running time to 1 second, double click on the Scope box to open it up and hit play:

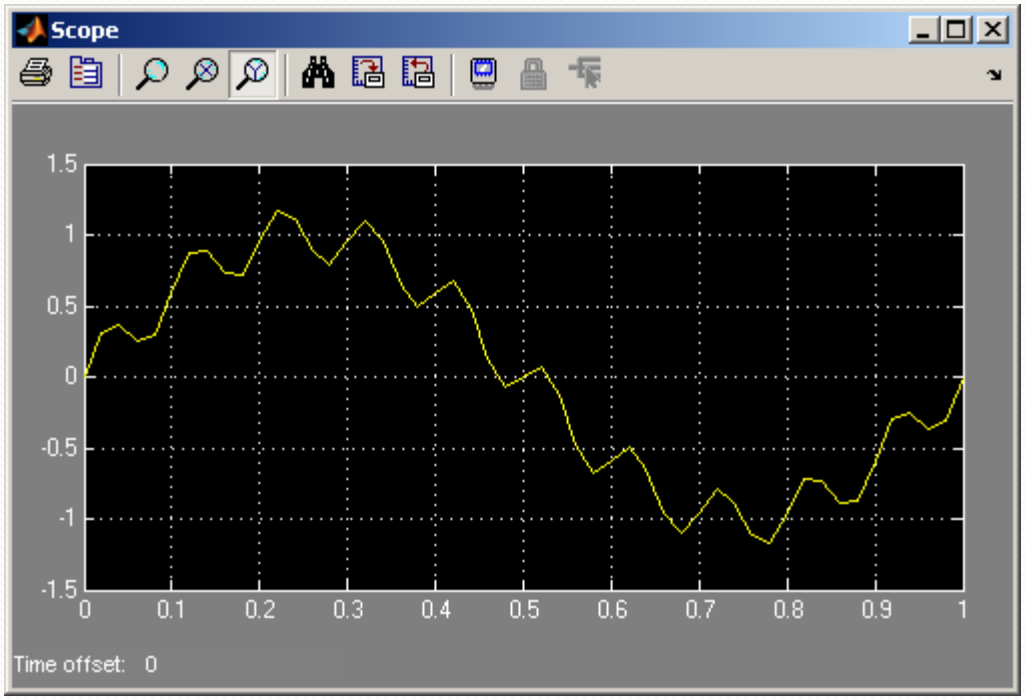
$$x(t) = \sin(2\pi 1t) + 0.2\sin(2\pi 60t)$$





Simulink: Sum of Sinewaves.

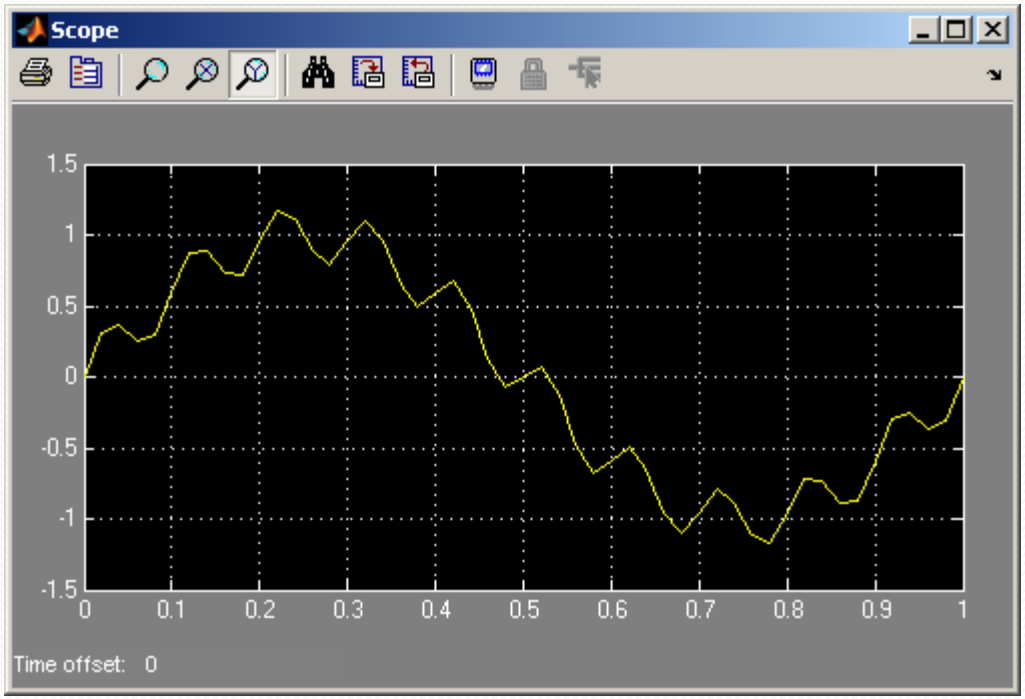
- We see the 60Hz wave riding on the 1Hz wave. This is called 60Hz noise or ripple.



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

Simulink: Sum of Sinewaves.

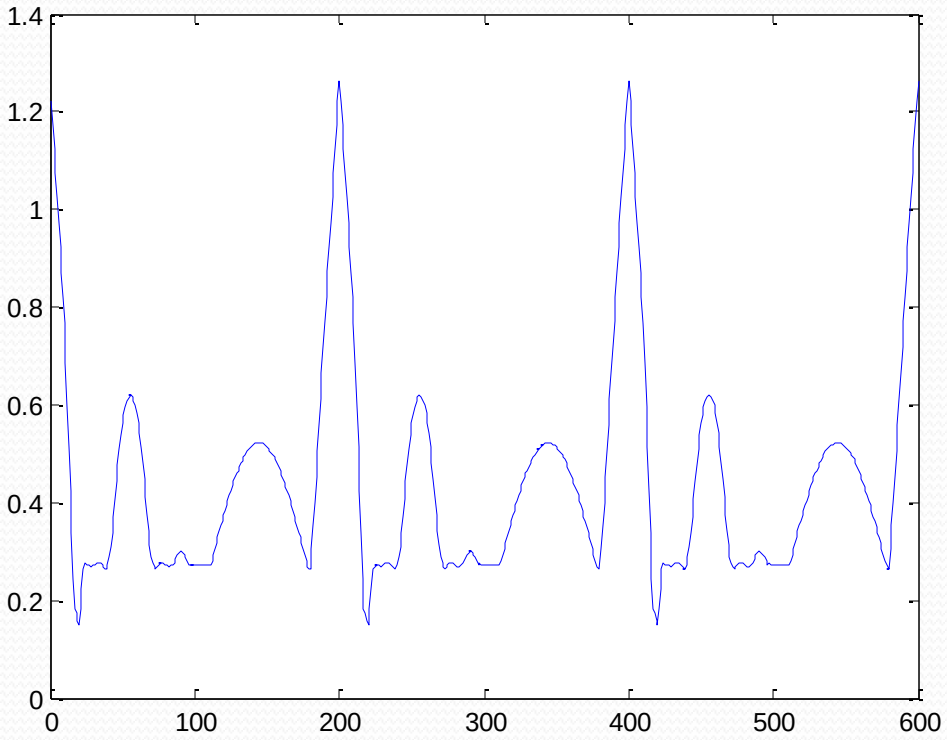
- The objective of next class is to get rid of the ripple and keep the 1Hz sinewave clean of noise!!! This is Signal Processing!!



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

Emulation of an EKG signal

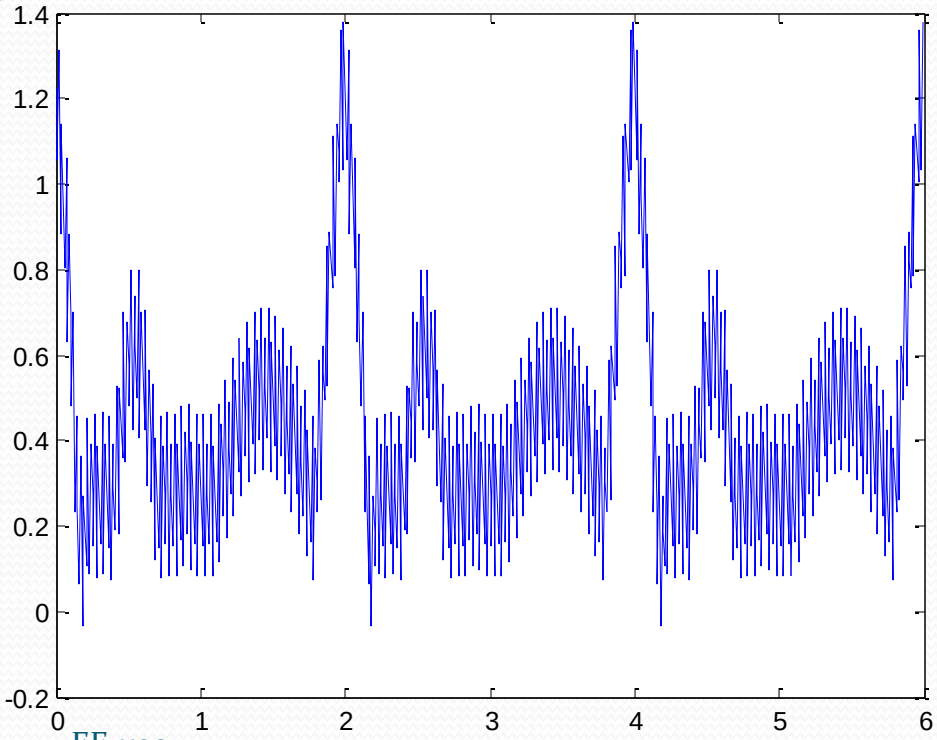
- There is a code that I copied emulaing a EKG signal!!



Emulation of an EKG signal

- The location is!!

- I modified the code to create a noisy signal. This is the power of Matlab!!



End of Class