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 orde to do that we study MATHEMATICS!!

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
 - We will emulate/modelate a low frequency signal corrupted with an additive noise (high frequency signal).
 - Implement a FILTER that will eliminate a high frequency component (ripple or noise) meanwhile leaving untouched a low frecuency sinewave.
- 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.



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



Matlab

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

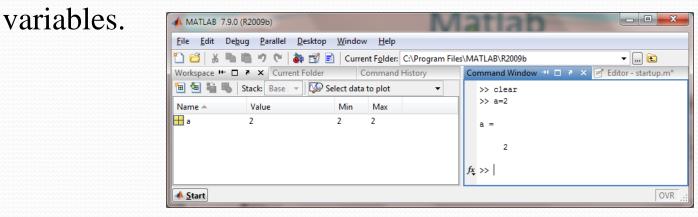
A MATLAB 7.9.0 (R2009b)		
<u>File E</u> dit De <u>b</u> ug <u>P</u> arallel <u>D</u> esktop <u>W</u> ir	ndow <u>H</u> elp	
🗋 🖆 👗 🖺 🛍 🤊 💌 🎒 🖹 🖸	Current Folder: C:\Program Files\MATLAB\R2009b	 €
Workspace	Command Window 💛 🗆 🤻 🔀 Editor - startup.m*	
🛅 🛅 骗 🧠 Stack: Base 👻 🕼 🕶	$f_{x} >>$	
Name 🔺 Value M		
exc <1x1 MException>		
		OVE
A Start		OVR .::



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

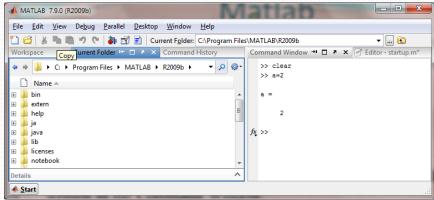
MATLAB 7.9.0 (R2009b)	
<u>F</u> ile <u>E</u> dit De <u>b</u> ug <u>P</u> arallel <u>D</u> esktop <u>W</u> indow <u>H</u> elp	
🞦 🗃 😹 ங 🛍 🤊 🝽 巐 🗊 🖹 Current F <u>o</u> lder: C:\Program Files\MATLAB\R2009b	€
Workspace I → □ ₹ × Current Folder Corr Command Window → □ ₹ × 🖻 Editor - startup.m*	
$\boxed{10}$ $\boxed{10}$ $\boxed{10}$ $\boxed{10}$ Stack: Base \neg $\boxed{10}$ S \neg f_x >>	
Name 🔺 Value N	
<u>▲ Start</u>	OVR .::

• Workspace window: is the one where the user can see the

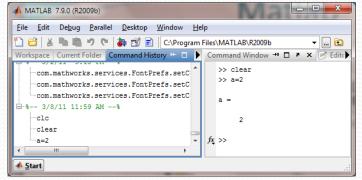


Matlab

• **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.

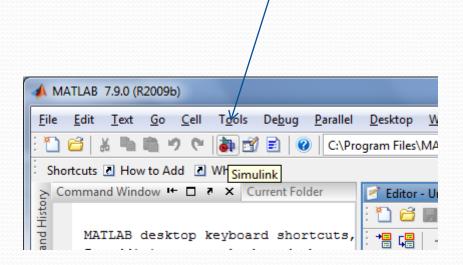


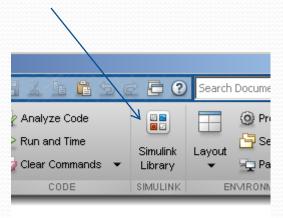
• **Command History window** that shows us all the commands written in the Command Window.



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:

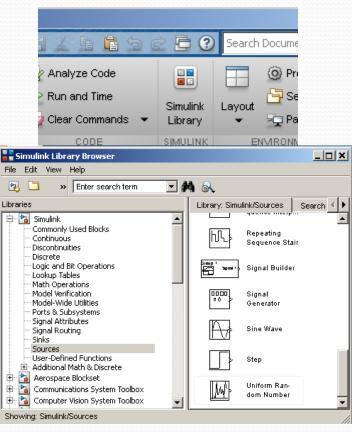




Simulink

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

MATLAB 7.9.0 (R2009b)				
<u>File Edit T</u> ext <u>Go</u> <u>C</u> ell	T <u>o</u> ols De <u>b</u> ug	<u>P</u> arallel <u>[</u>	<u>)</u> esktop <u>W</u>	
- 🔊 🕈 🛍 🖷 👗 🔁 🛄 🤊	🍋 🗹 🖻 🛛	C:\Progr	am Files\MA	
Simulink Library Browser	-	-		x
File Edit View Help				
🗋 🚔 🛥 🛛 Enter search term	- #4	<u> </u>		
Libraries	Library: Simulink	Search R	esults: (none)	4
Simulink Commonly Used Blocks Continuous		ommonly Used ooks		•
··· Discontinuities	Cc	ontinuous		
····Logic and Bit Operations ····Lookup Tables	Pi Di	scontinuities		
Math Operations Model Verification Model-Wide Utilities	Di	screte		
···Ports & Subsystems ···Signal Attributes		ogic and Bit perations		-
Showing: Simulink				d

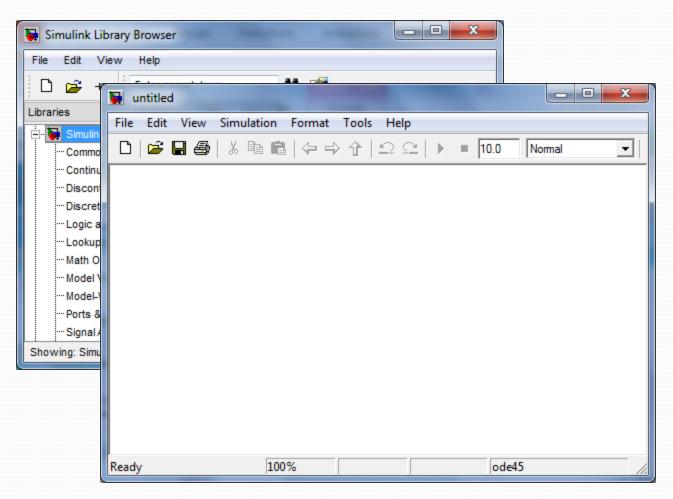


12/16/2013

EE 1130

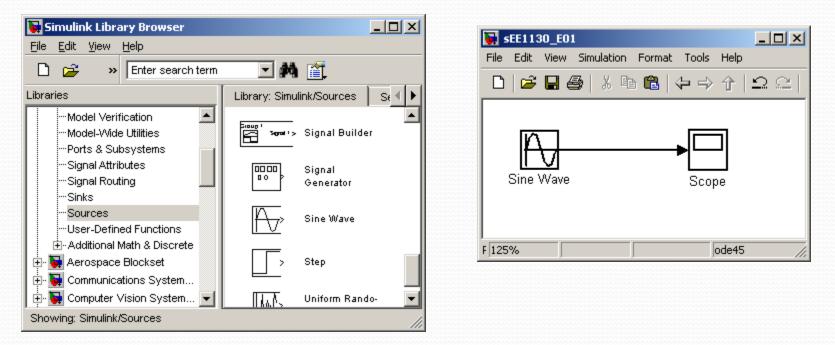
Simulink

• Click on the new document to open:



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

 $x(t) = \sin(2\pi f t)$ with f = 1



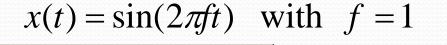
• We inserted a sinewave from the sources, and a scope from the sink library groups respectively.

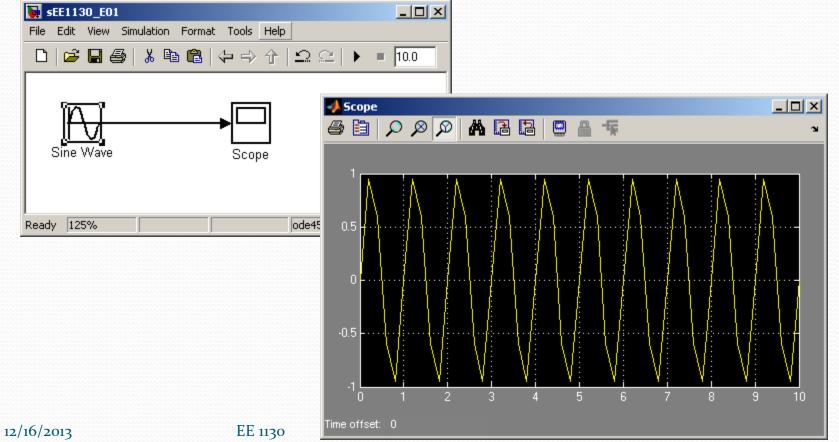
• By double clicking on the Sine Wave box we open its properties box and set the angular frequecy to 2*pi*1:

 $x(t) = \sin(2\pi f t)$ with f = 1

🙀 Source Block Parameters: Sine Wave	×	😽 sEE1130_E01 📃 🗖	×
Blas:		File Edit View Simulation Format Tools Help	
0		□ ☞ 🖩 🚳 % 🖻 🛍 (수 → 수 으 으	:
Frequency (rad/sec):			
2*pi*1			
Phase (rad):			
0		Sine Wave Scope	
Sample time:			
0	_	F 125% ode45	_
•			_//_
<u>O</u> K <u>C</u> ancel <u>H</u> elp <u>Apply</u>	/		

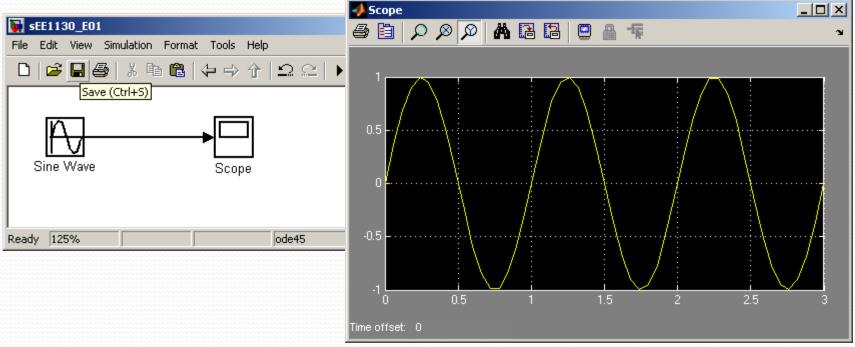
- To open the scope we double click on the Scope box.
- We hit the play icon to run the simulation.





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

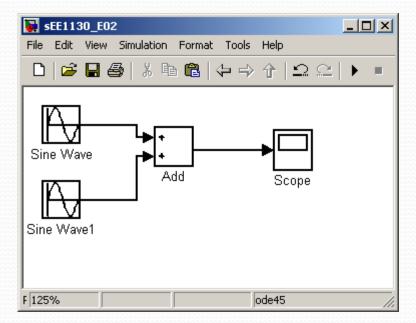
 $x(t) = \sin(2\pi f t)$ with f = 1



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

• 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:

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



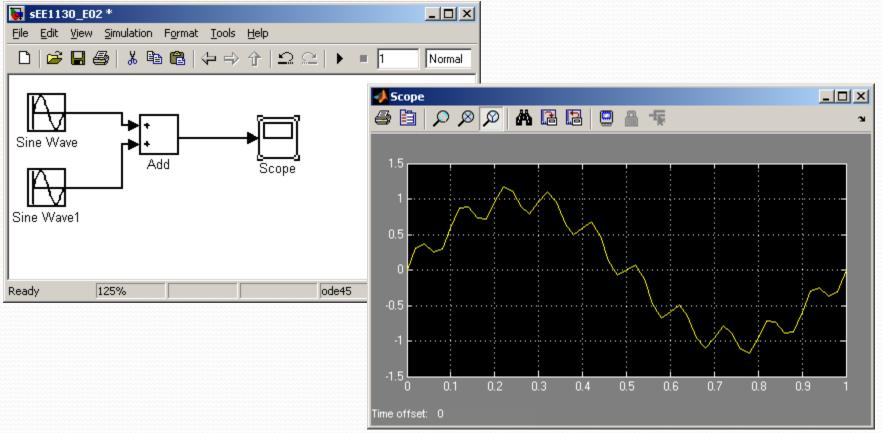
• We need to click on each sinewave box to set up the amplitude and frequencies: $x(t) = \sin(2\pi 1t) + 0.2\sin(2\pi 60t)$

SEE1130_E02	
File Edit View Simulation	Format Tools Help
D 🛩 🖬 🎒 X 🖻	$\blacksquare \Leftrightarrow \Rightarrow \uparrow \mathfrak{Q} \subseteq \bullet \bullet $
Source Block Parameters: Sine Wave	Source Block Parameters: Sine Wave1
Parameters	Amplitude:
Sine type: Time based	0.2
Time (t): Use simulation time	Bias:
Amplitude:	0
1	Frequency (rad/sec):
Bias:	2*pi*60
0	Phase (rad):
Frequency (rad/sec):	0
2*pi*1	Sample time:
QK <u>C</u> ancel <u>H</u> elp <u>Apply</u>	<u>OK</u> <u>Cancel</u> <u>H</u> elp <u>A</u> pply

12/16/2013

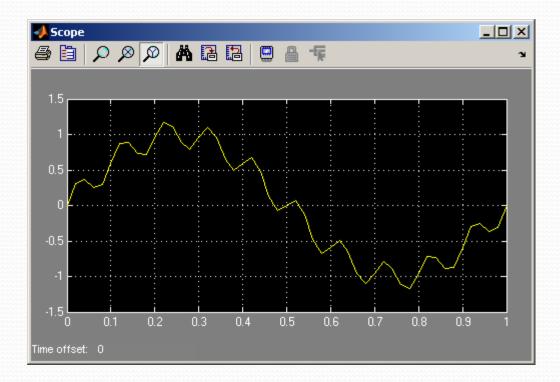
• Set the running time to 1 second, double click on the Scope box to open it up and hit play:

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



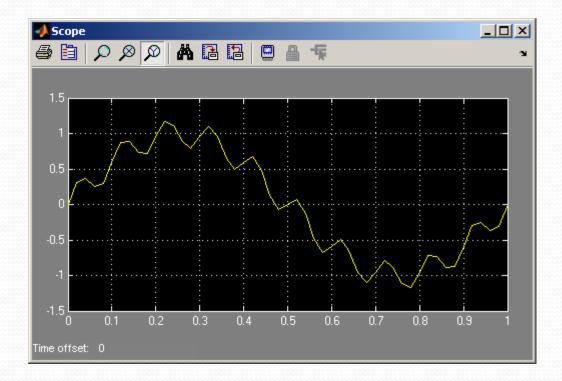
EE 1130

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



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

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



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

End of Class