

Matlab laboratory on Signal Theory

Run Matlab.

For those who are not familiar with Matlab:

- enter “intro” in the command window and press Enter to have a short introduction to Matlab potentialities;
- explore the “help” menu, especially the “documentation” and the “examples” submenu;
- explore the “help” command (and more specifically the “help elmat” command to see which elementary functions are available).

1) “rand” and “plot” commands and “for” cycles

Read the “rand” help (command “help rand”).

Generate a vector of 100 or more random numbers.

Read the “plot” help.

Generate a graph of the random function saved in the vector.

Read the “for” help to learn the syntax of “for” cycles.

Use the “for” command to generate a non-random function (e.g., a cosine function using the “cos” command).

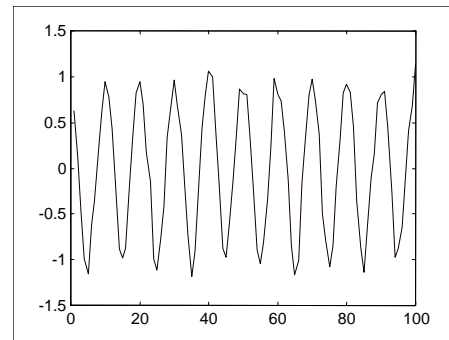
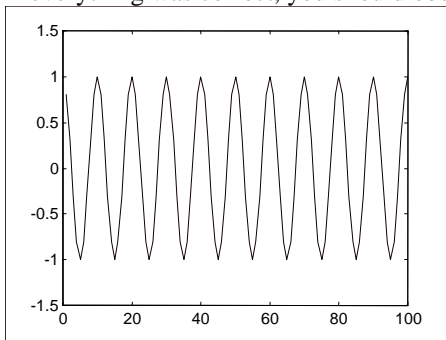
Plot the result.

Add a random noise (“rand” command) ranging from -0.2 to 0.2 saving the new function in a new vector.

Explore the “figure” help to learn how to manage multiple figures.

Compare the graphs of the two functions (with and without noise)

If everything was correct, you should observe a result similar to:



Explore the variables saved in your workspace with the command “who” or “whos”.

Clear the workspace with the “clear” command.

2) Creating a periodic signal and observing the spectrum

Try to solve this exercise without using cycles.

Generate a periodic function with 200 samples by adding three or more sine waves with different frequency and different phase (assume that the time interval between samples is $T_s=5\text{ms}$ which corresponds to a sampling frequency of 200Hz; thus the signal described by the 200 samples is 1 second long). So, the suggestion is to start by generating a vector of 200 elements containing the time axis sampled every 5ms. Then use this vector to generate the periodic function.

Plot the periodic function.

Explore the help of “fft” and “abs” commands.

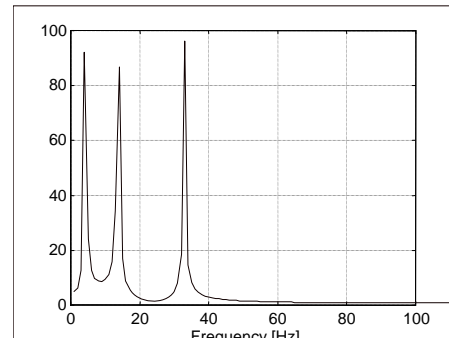
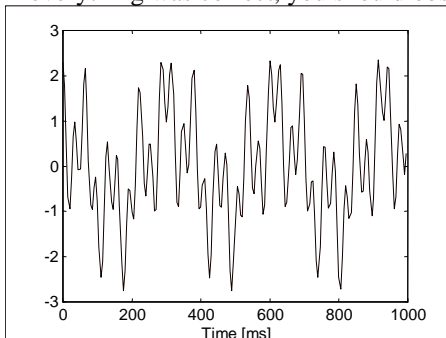
Calculate and plot the amplitude spectrum of the periodic function.

Read the “axis” help.

Use the “axis” command to plot only the frequencies from 0 to Nyquist (i.e., up to 100Hz).

Apply “grid” for better observation of the amplitude spectrum.

If everything was correct, you should observe a result similar to:



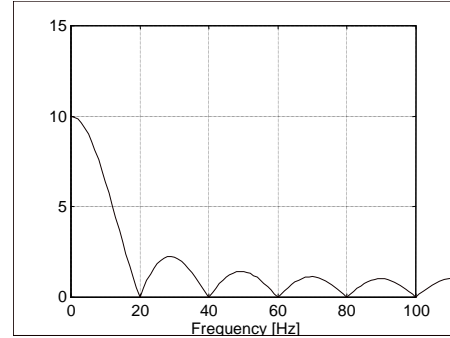
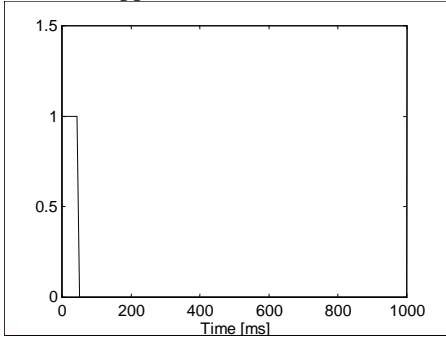
3) Spectrum of a rectangular function and a delta function

Explore the help of “zeros” and “ones” commands.

Using “zeros” and “ones” commands, generate a rectangular function 50ms long with amplitude 1 (use a vector of 200 samples assuming $T_s=5\text{ms}$).

Plot the rectangular function and the amplitude spectrum from 0 to Nyquist frequency.

You are supposed to see a result as follows:



Observe how the amplitude spectrum changes while progressively reducing the duration of the rectangular function.

Reduce the duration till the rectangular function becomes a delta function. Plot both the amplitude and the phase spectra (“angle” command). If everything was correct, a constant amplitude spectrum and a null phase spectrum are expected.

Clear the workspace.

4) Delta function and white noise

Generate the spectrum of a delta function (it might be worth using “ones”).

Return to the time domain by applying the Inverse Fourier Transform (“ifft”) and check that the result is really a delta function (also check that the imaginary part of the resulting function is null).

Generate the spectrum of a white noise, i.e., a constant amplitude spectrum and a random phase spectrum (it might be worth using “exp” or “^” and the imaginary unit “j”). Take care to generate the amplitude and phase spectra with the expected symmetrical properties. Be careful in assigning amplitude and phase values at zero frequency and Nyquist Frequency.

Inverse transform the white noise spectrum and plot the result. If everything was correct, a random sequence is expected. Check that the imaginary part of the resulting function is null. If this is not the case, check the symmetric properties of the phase spectrum.

Clear the workspace.

5) Convolution in time domain

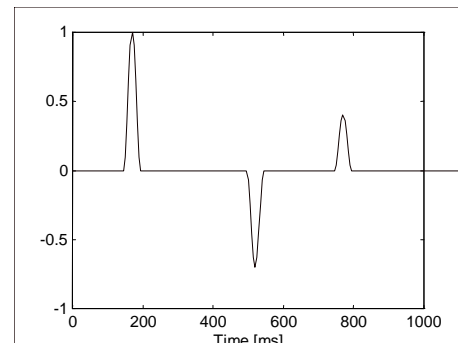
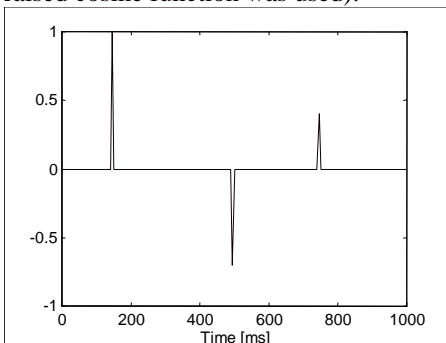
Generate a pulse (e.g., a triangle function or a raised cosine function) 50ms long followed by null samples (use a vector of 200 samples assuming $T_s=5\text{ms}$). The commands “triang” or “hann” might help.

Generate a reflectivity function 1s long which contains 3 or 4 delta functions with amplitudes between -1 and +1.

Explore the “conv” help.

Convolve the pulse with the reflectivity function and plot the result.

If everything was correct, the reflectivity function and the convolution output might be similar to the following results (where a raised cosine function was used):



6) Convolution in frequency domain

Calculate the spectra of the pulse and the reflectivity function generated in the previous step.

Perform the convolution of the pulse with the reflectivity function in the frequency domain by spectra multiplication.

Apply the inverse transform and compare the convolution result with the result obtained in the time domain.

Clear the workspace.

7) Filter

Generate a periodic signal (with 200 samples) combining sine components with frequencies in the 4-20Hz range. Assume a sampling interval $T_s=5\text{ms}$.

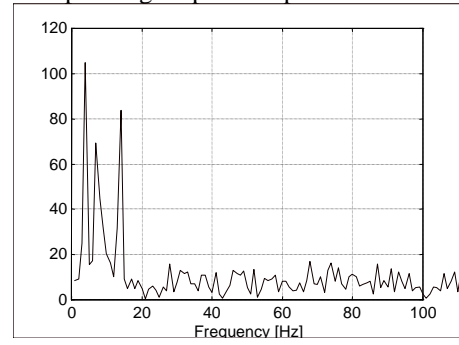
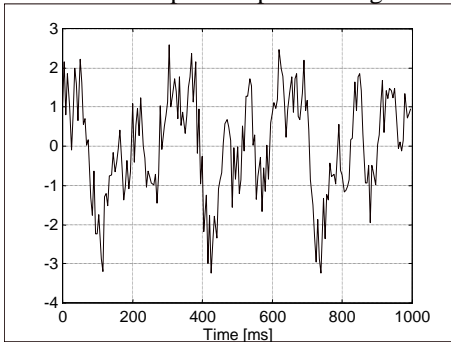
By using the “rand” command, generate a 200 sample noise sequence with zero mean value and with max amplitude approximately equal to the 30% of the periodic signal max amplitude.

Plot the signal and the noise.

Sum noise and signal and plot the result.

Calculate and plot the amplitude spectrum of the sum result.

Below an example of a periodic signal disturbed by noise and of the corresponding amplitude spectrum.



Generate a 200 sample vector representing the transfer function of a zero-phase low-pass filter designed to filter the high frequency components of the noise. (Take care to generate a symmetrical transfer function corresponding to a real filter). Apply the filter to the noise disturbed signal (by multiplying the spectrum and the transfer function) and compare the result (after inverse transform) with the unfiltered signal. Clear the workspace.

8) Autocorrelation in time domain

By using “rand”, generate a noise sequence (200 samples at 5ms intervals) with amplitudes in the -0.5 to +0.5 range. Explore the “xcorr” help.

Calculate the autocorrelation of the noise signal and plot the result.

9) Autocorrelation in frequency domain

Calculate the Fourier Transform of the noise signal and plot both amplitude and phase spectra.

Generate the spectrum of the noise autocorrelation by a multiplication in the frequency domain (the “conj” command might help).

Plot both amplitude and phase spectra of the autocorrelation and compare with noise spectra.

Apply the inverse transform to the autocorrelation spectrum and compare the result with the autocorrelation obtained in time domain. Discuss the differences.

Clear the workspace.

10) Test for periodicity

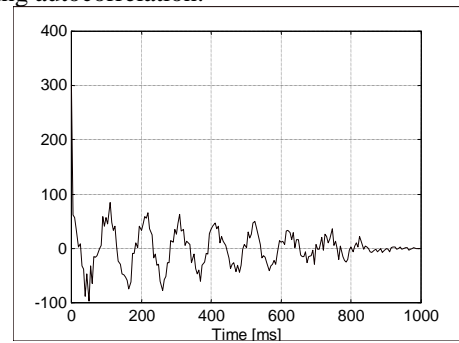
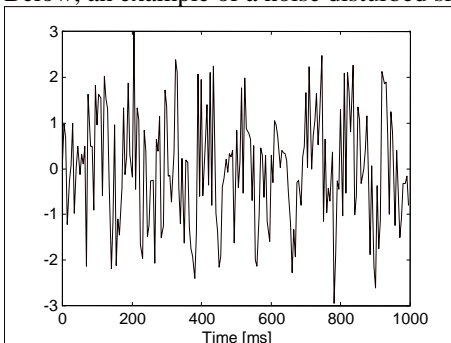
Generate a sinusoidal signal (200 samples at 5ms intervals) with amplitude equal to 1.

By using “rand”, generate a noise sequence (200 samples at 5ms intervals) with amplitudes in the -2 to +2 range.

Sum the signal with the noise and plot the result.

Apply the autocorrelation to detect the periodic component of the sum signal.

Below, an example of a noise disturbed sinusoid and the corresponding autocorrelation:



Clear the workspace.

11) Autocorrelation of a sweep signal

Generate a linear sweep between 10 and 60 Hz sampled at 5ms and 1s long.

Plot the sweep.

Apply the Fourier Transform and plot the amplitude spectrum.

Calculate the autocorrelation of the sweep signal and plot the result.

12) Matched filter for a sweep source

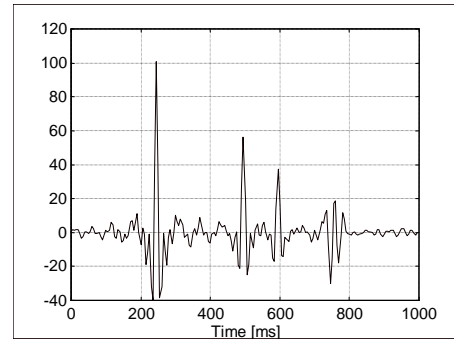
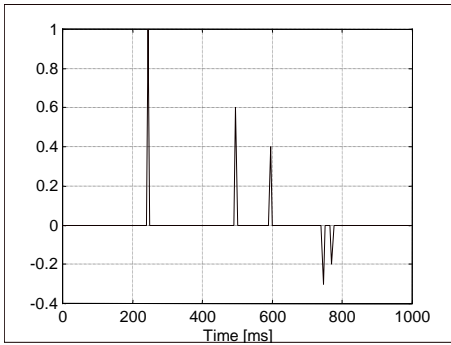
Generate a reflectivity function 1s long containing some pulses of different amplitudes (within -1 and +1). Make sure that two pulses are separated by 100ms and that others have a larger separation while others a smaller separation.

Convolve the sweep signal with the reflectivity function and plot the result.

By using the “xcorr” command, apply the matched filter of the sweep to the result of the convolution.

Plot the result and evaluate the resolution that is offered by a 10-60Hz sweep.

If everything was correct, the reflectivity function and the filtered signal might be similar to:



13) Matched filter for a random source

By using “rand”, generate a noise sequence (200 samples at 5ms intervals) with amplitudes in the -0.5 to +0.5 range. Convolve the noise signal with the reflectivity function of the previous step and plot the result.

By using the “xcorr” command, apply the matched filter of the noise sequence to the result of the convolution. Plot the result.

Clear the workspace.

14) Transfer function of a geophone pattern

Generate a 200 sample vector representing the impulse response in horizontal direction for a pattern of 13 geophones regularly distributed with a 6m interval (assume a velocity of 300m/s for the surface wave and use a sampling interval of 5ms).

Transform the impulse response to obtain the transfer function of the geophone pattern and plot the amplitude to evaluate the filtered band. If everything was correct, the filtered band should be about 4-46 Hz.

Clear the workspace.

15) Aliasing

Generate a 1s long sinusoidal signal at 180Hz sampled at 1ms interval.

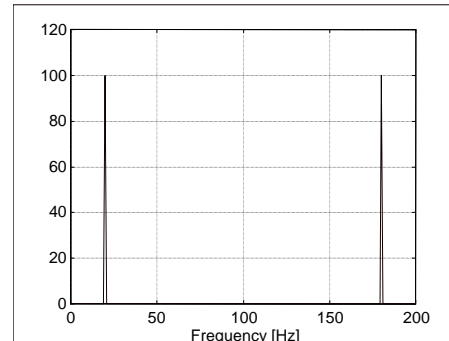
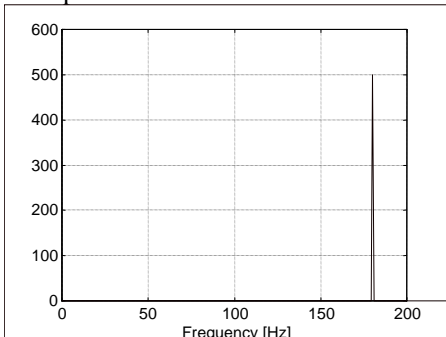
Plot the signal and its amplitude spectrum.

Decrease the sampling rate by a factor of 5, i.e., keep every 5th sample of the original sequence (the “downsample” command might help).

Compare the original signal and the downsampled version.

Compare also the respective amplitude spectra in the range 0-200Hz.

The spectra should be as follows:



Clear the workspace.

16) 2D Fourier Transform

Generate a 2D ideal pulse (e.g., a 32x32 matrix containing a unit sample).

Explore the “surf” help. Use “surf” to observe the 2D function.

Calculate the 2D Fourier Transform by using the “fft2” command. Use “surf” to observe the amplitude spectrum.

Generate other simple 2D functions and observe the corresponding amplitude spectra.

Clear the workspace

17) Envelope

Generate a 1s long reflectivity function sampled at 5ms interval containing 5 ideal pulses with amplitudes in the -1 to +1 range.

Generate the transfer function of a zero-phase band-pass filter by using a Hann window (“hann” command) of 31 samples centered at 25Hz. Plot the transfer function of the filter.

Apply the filter to the reflectivity function and plot the result.

Extract the envelope of the filtered signal by calculating the amplitude of the analytic signal obtained in the frequency domain (suggestion: the spectrum of the analytic signal is null at negative frequencies and is twice the original spectrum at positive frequencies).

Overlap the envelope and the filtered signal on the same graph.

Repeat the envelope extraction by using the “hilbert” command and check graphically that the result is the same.