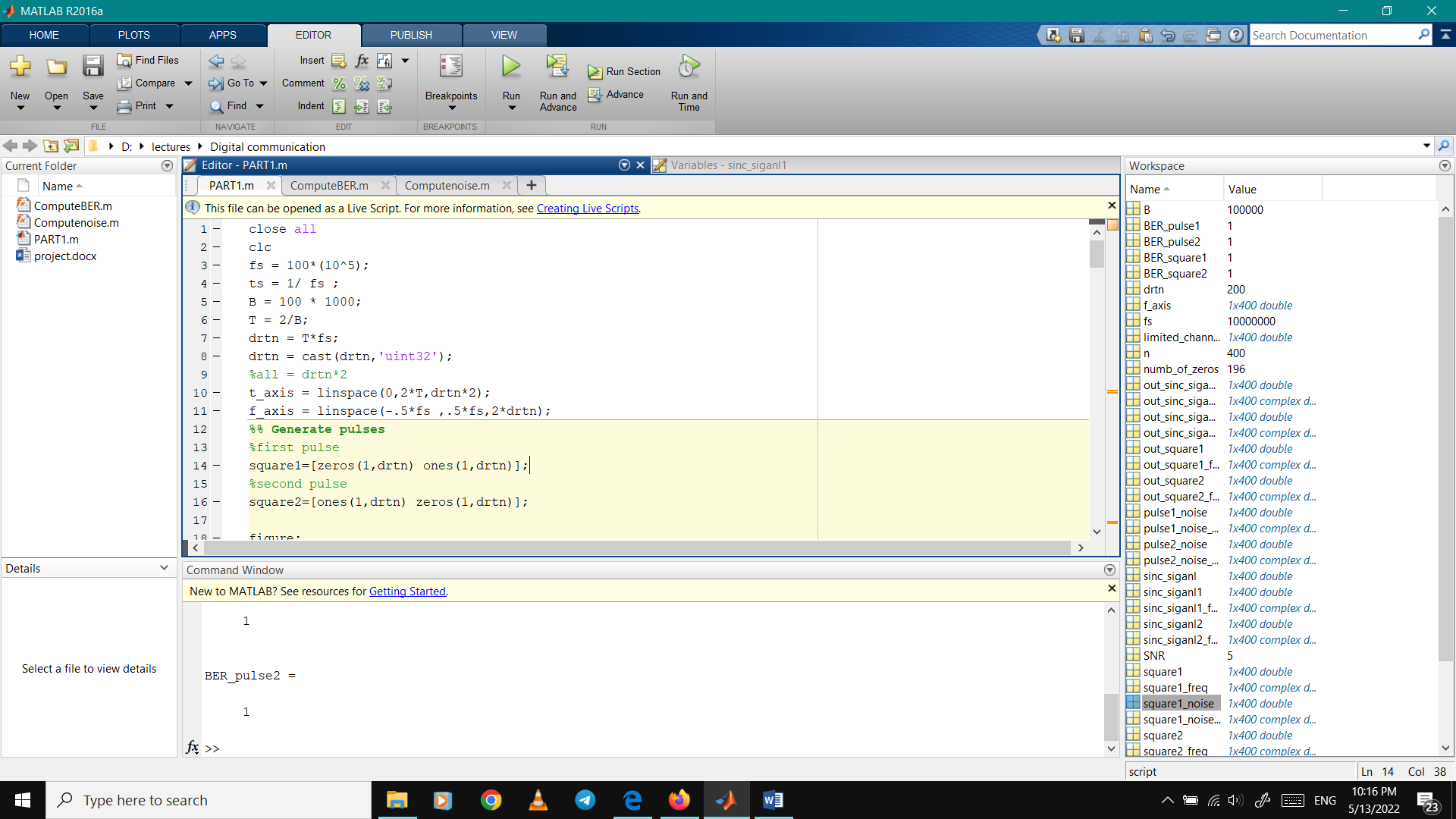
Part 1 : Inter-Symbol Interference due to band-limited channels :

In this part we define sampling rate, the Bandwidth of the channel and the duration of the pulse

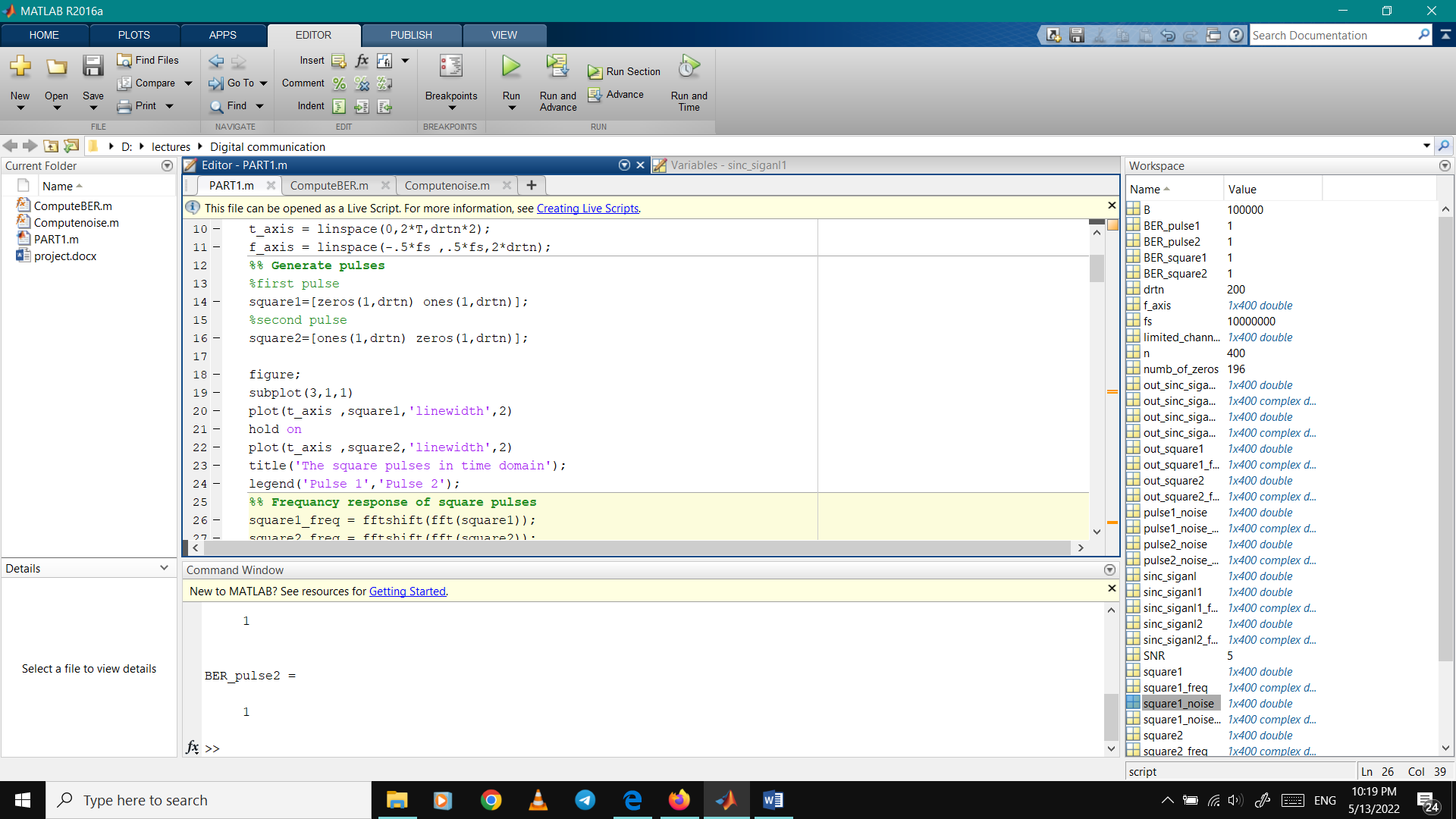


In this part we will generate square pulses and sinc signals “for zero ISI”

We will start with the square pulses :

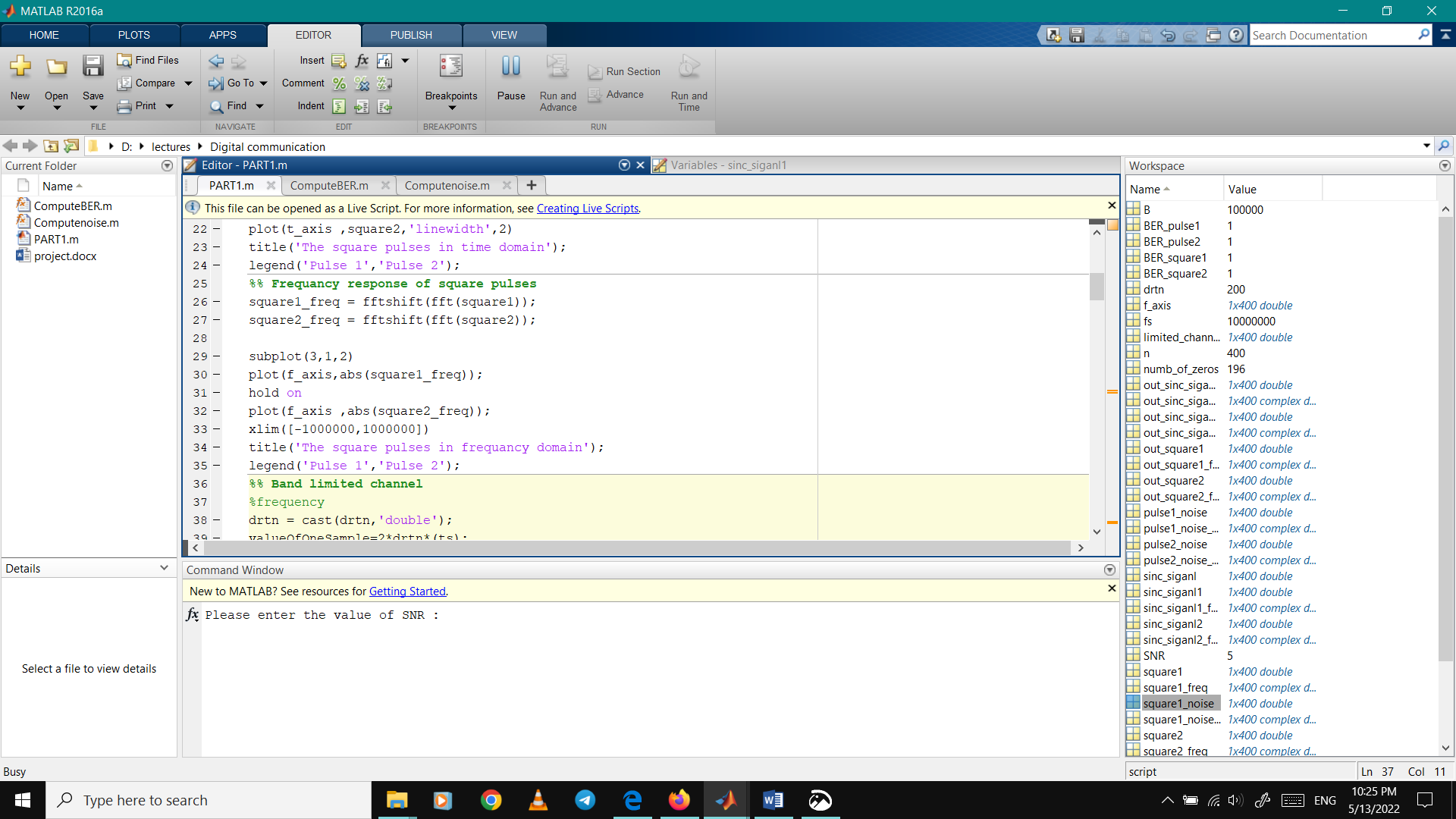
In time domain:

We generated square pulses with zeros and ones



In frequency domain :

By using FFT for the square pulse in time domain it turned to frequency domain

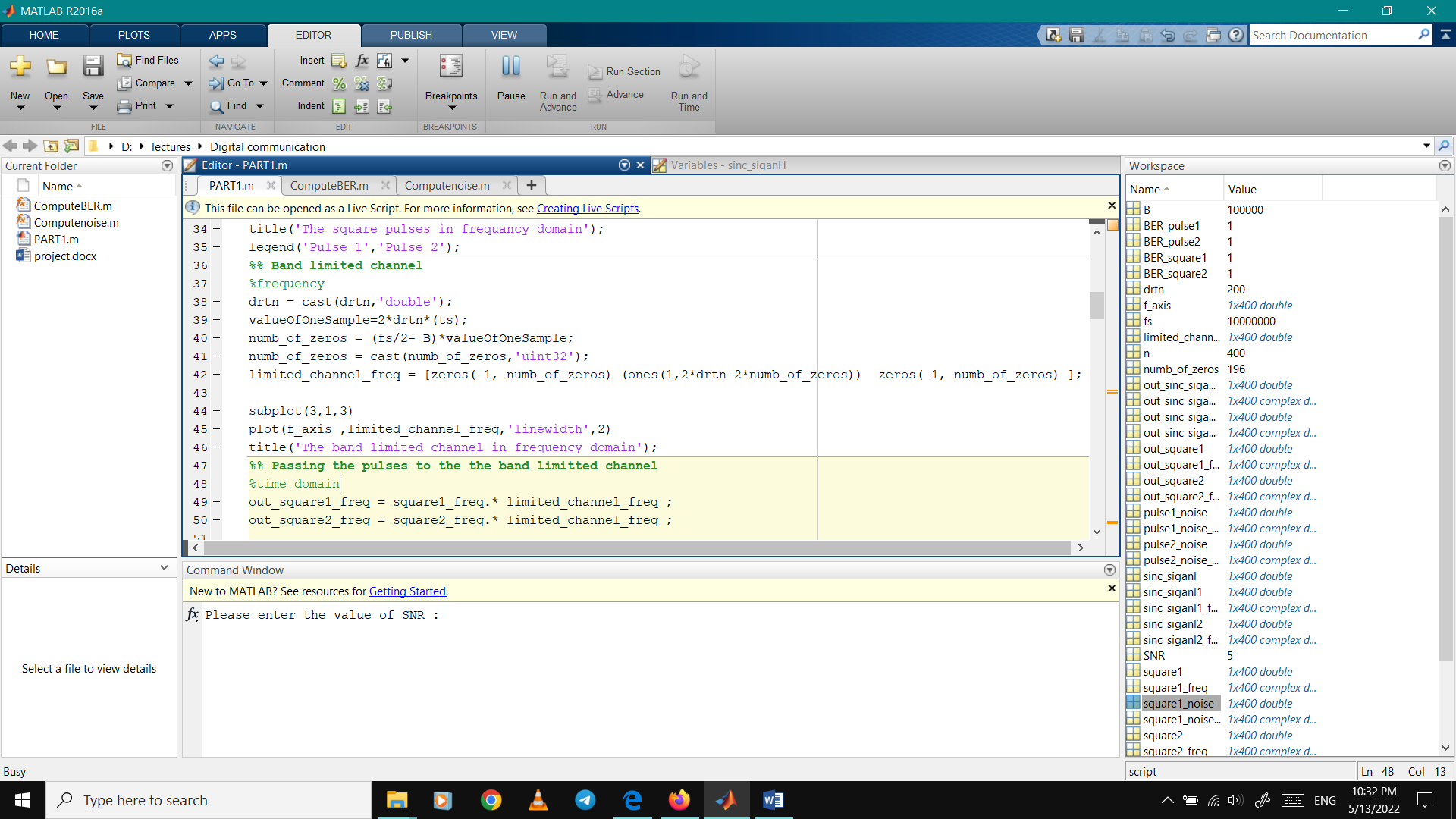


Creating the band-limited channel :

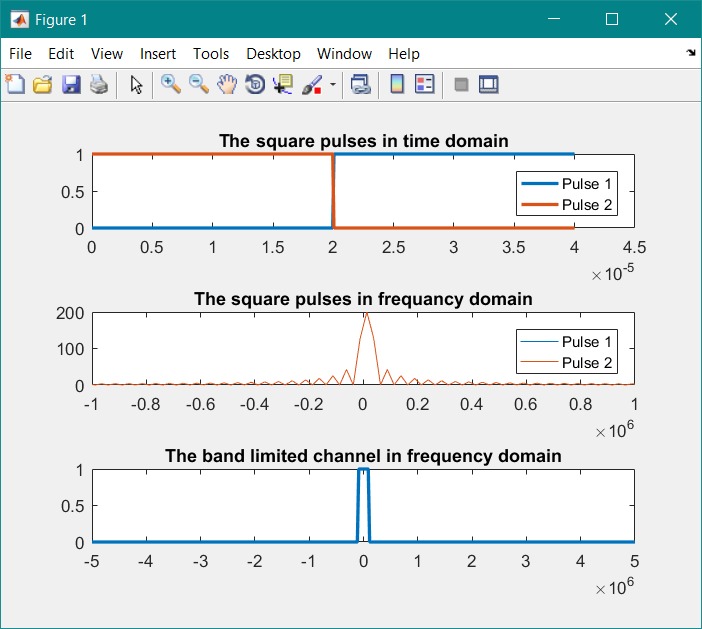
In this part we implement the band-limited channel by introducing some variables

* ValueOfOneSample which carries the time taken for each sample
* no\_of\_zeros which carries the number of zero samples of the filter where it is equal to half of the number of samples minus the band width of the channel filter

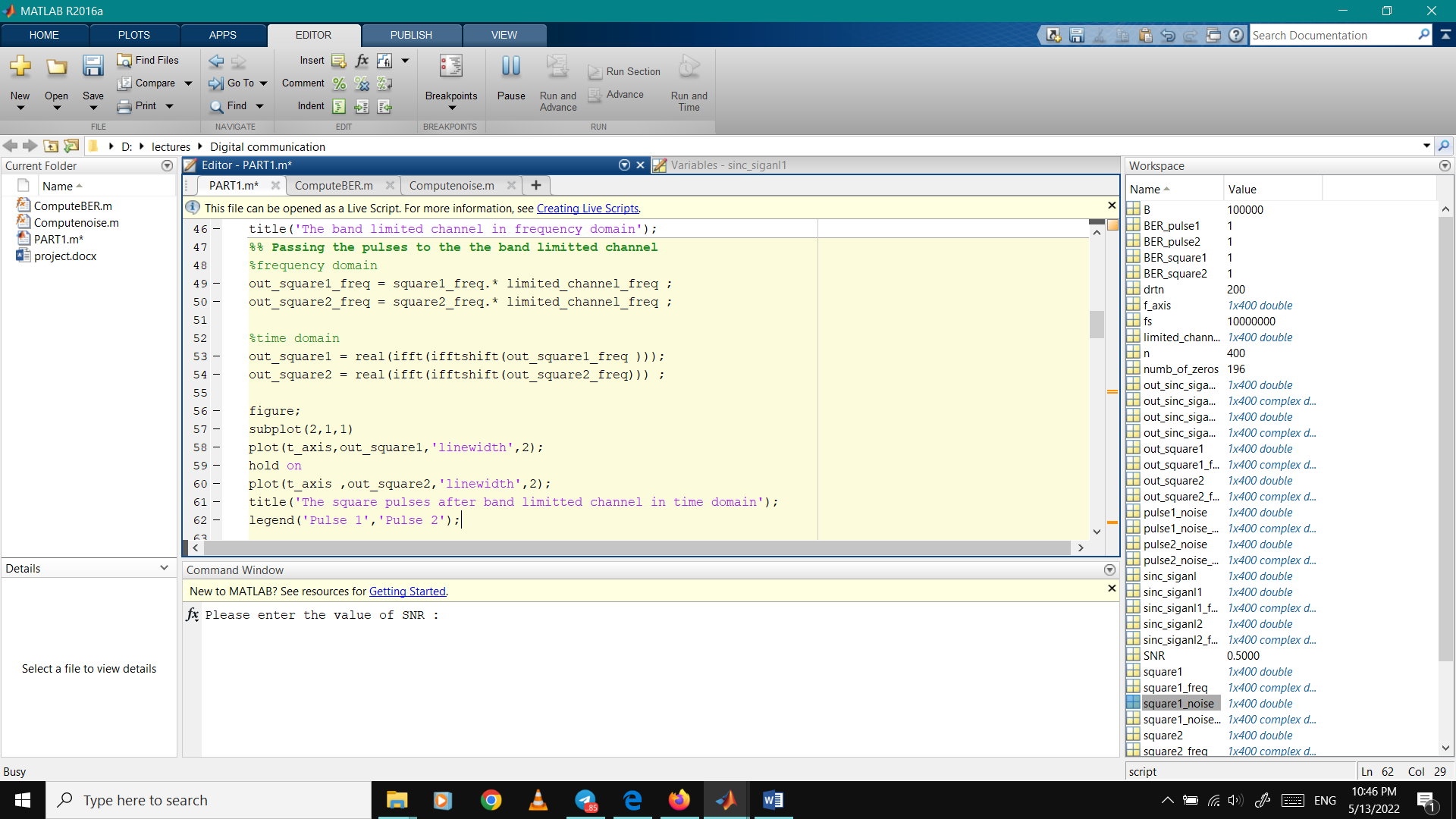
Finally we implement our channel by the ones and zeros functions to get the desired filter

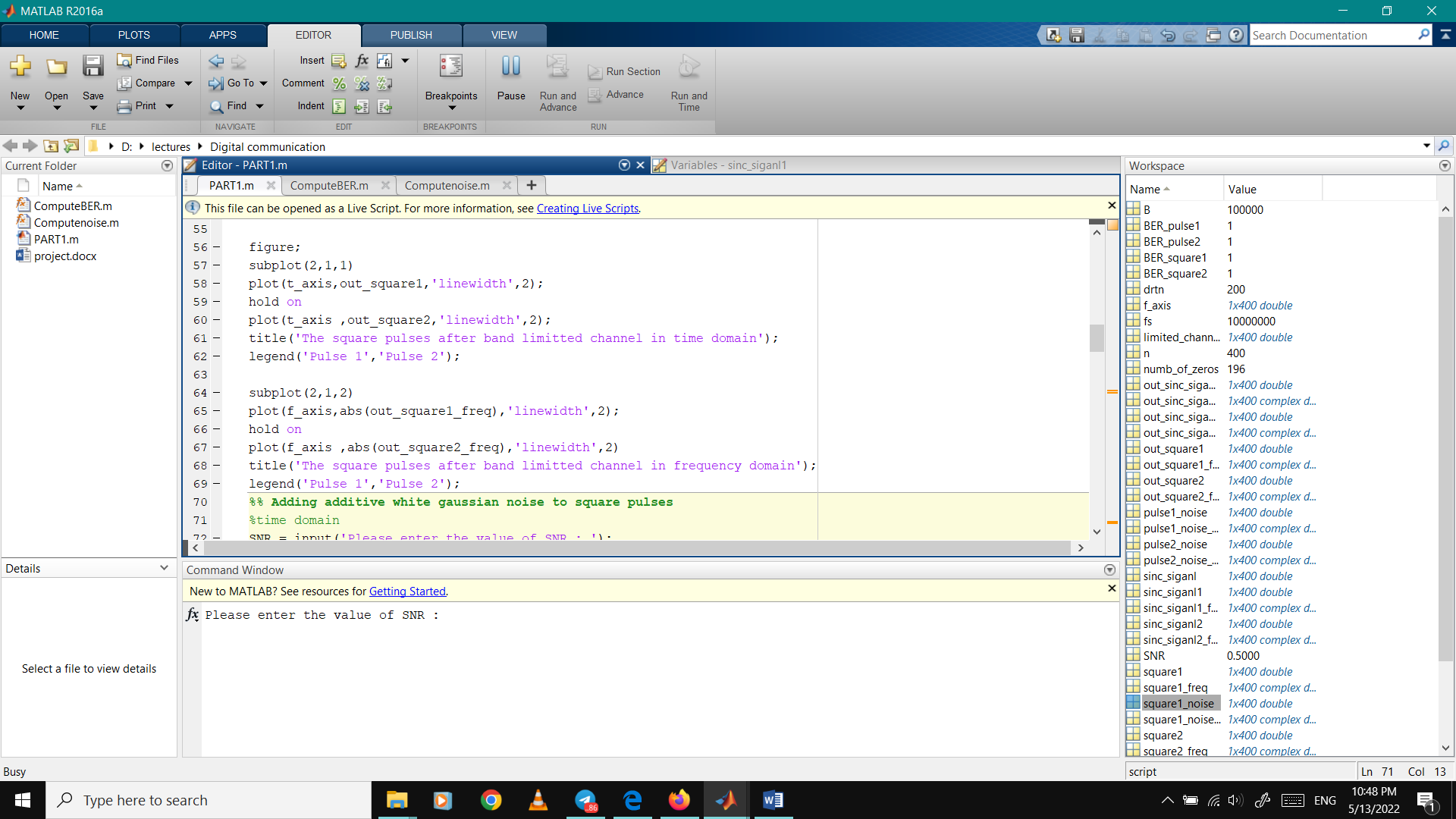


Plots of square pulses in time and frequency domain and the band-limited channel :

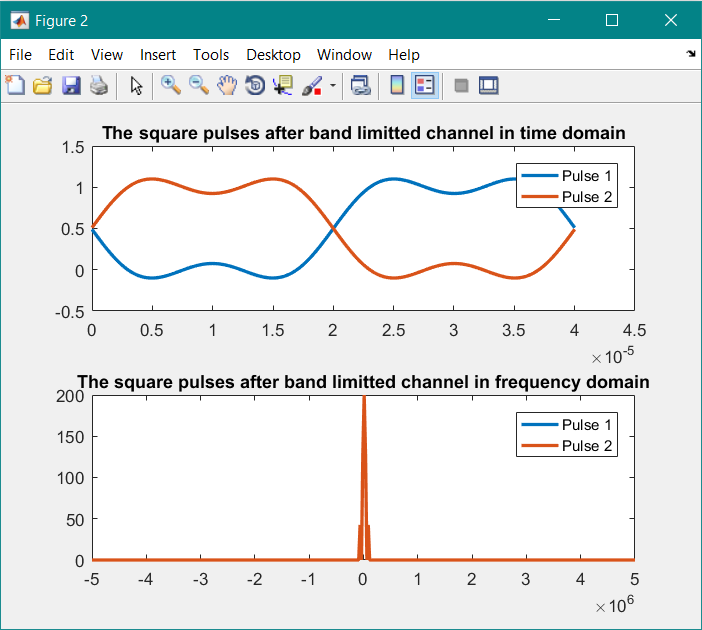


Next step, we pass the square wave in frequency domain into the channel and by using IFFT it converted to time domain :

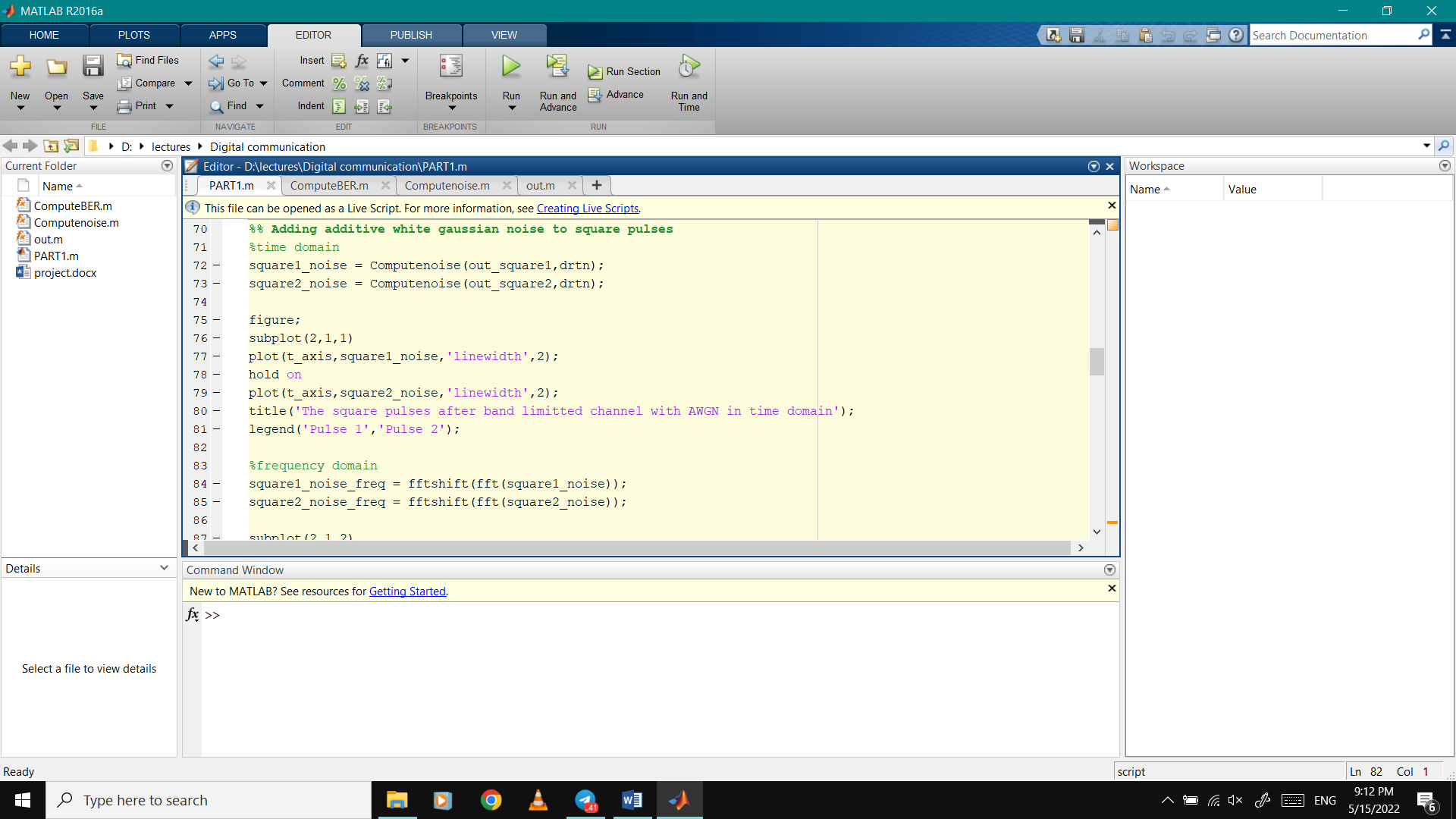


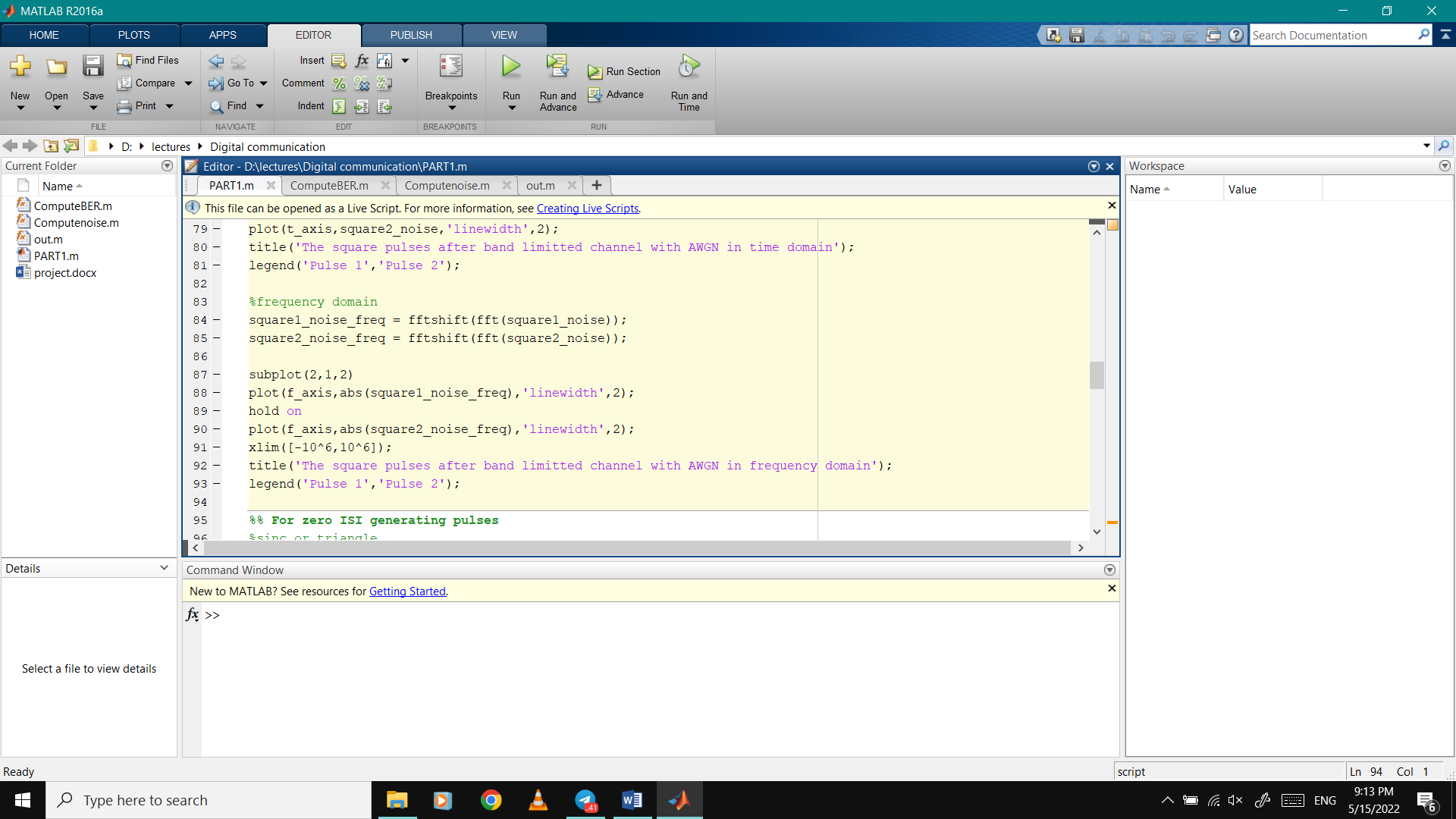


Plots after passing two square pulses through the band limited channel :



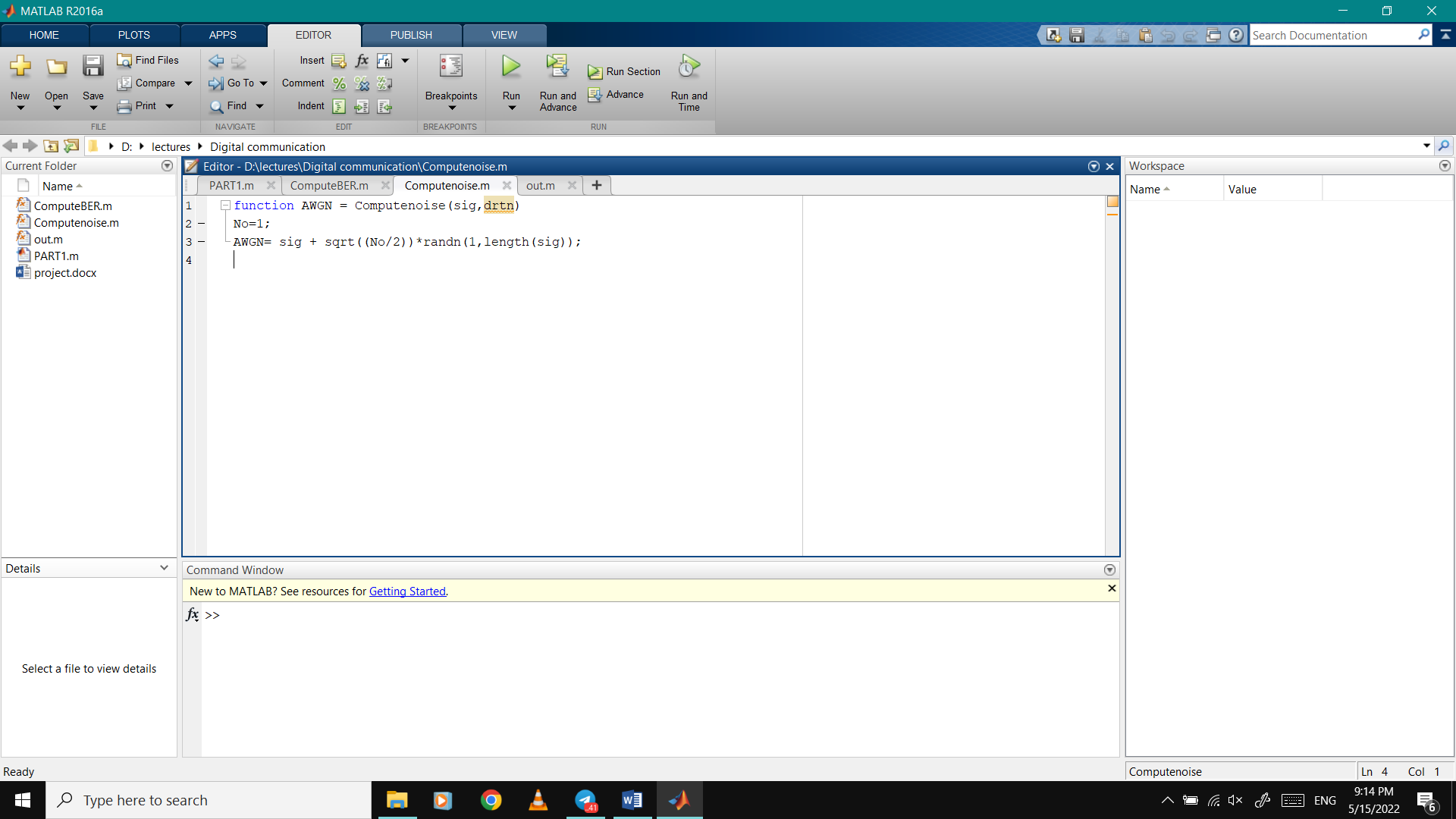
Adding additive white gaussian noise to square pulses:



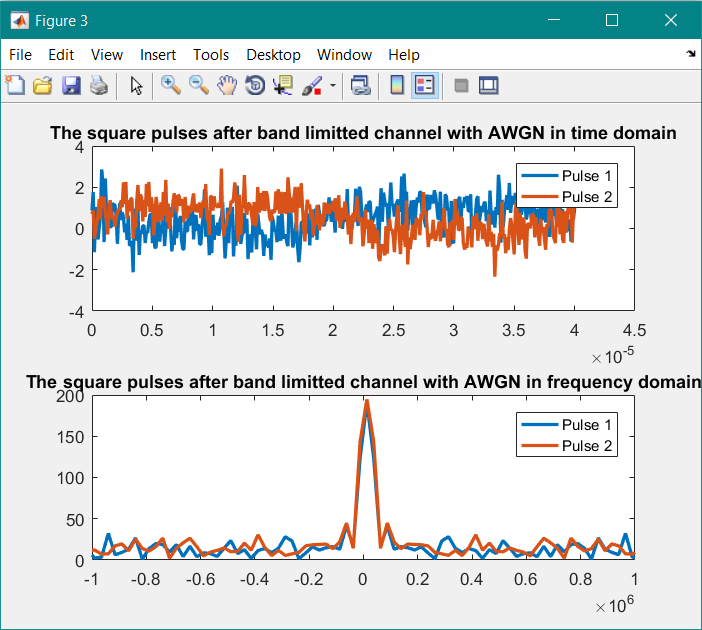


We create a function for the AWGN and called it Computenoise:

To add noise to the pulses



Plots after adding AWGN to the pulses in time and frequency domain:



For no ISI we generated sinc signal :

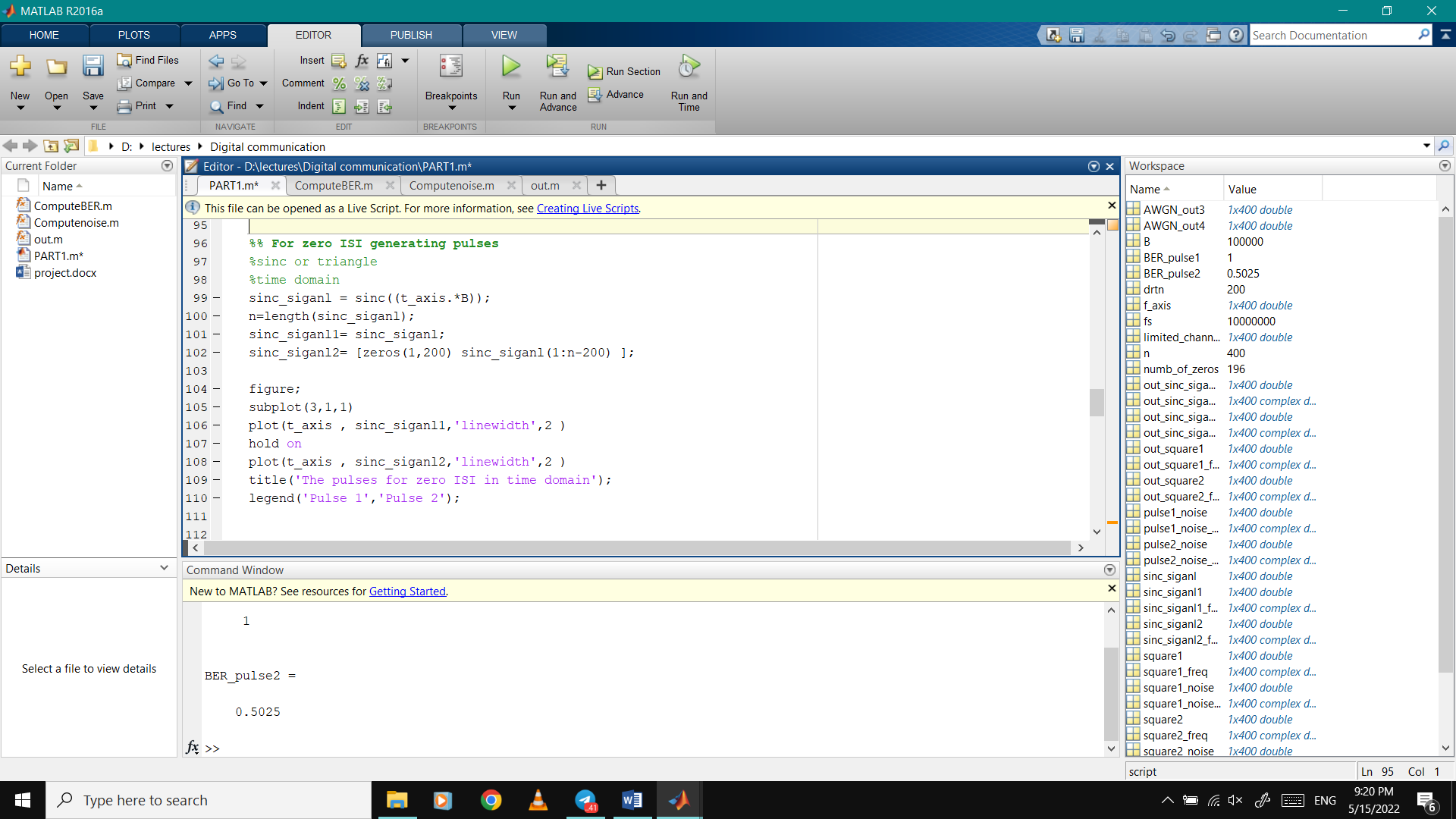
Explain what is the mathematical criterion that ensures no ISI.

For nullifying the ISI terms, with an impulse of unit value applied at t=0 to the combined filters h(t), the samples of the h(t) at the output of the filter combination should be 1 at the sampling instant t=0 and zero at all other sampling instants kT\_b (k !=0).

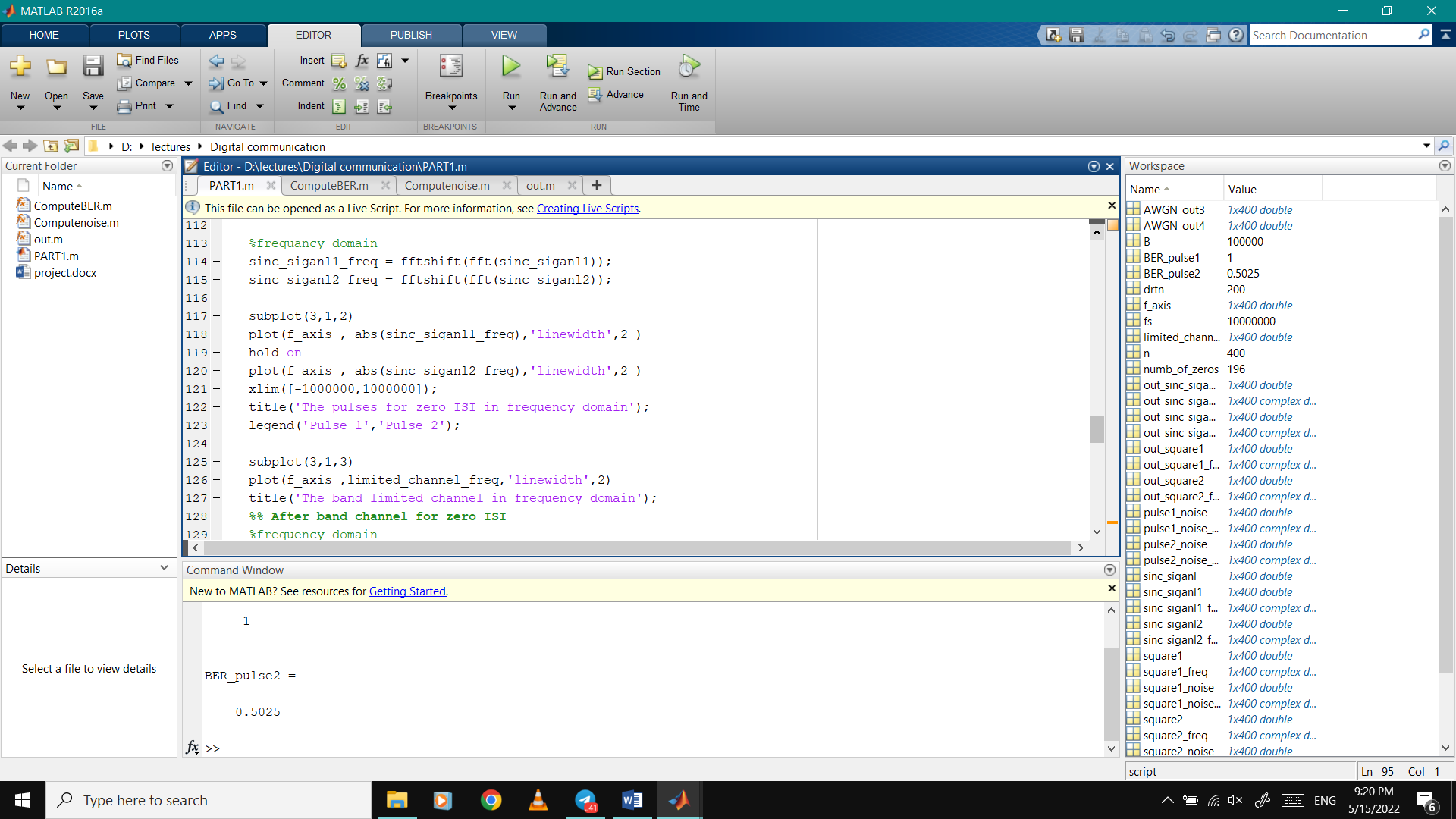
Describe one or more pulse shapes that ensure that such condition is met

* the signal that avoids ISI with the least amount of bandwidth is a sinc pulse of bandwidth F/ 2.
* any triangular waveform („Δ‟ function) with a width that is less than 2Tb will also satisfy the condition.
* raised–cosine pulses that satisfy the Nyquist criterion and require slightly larger bandwidth than what a sinc pulse (which requires the minimum bandwidth ever) requires.

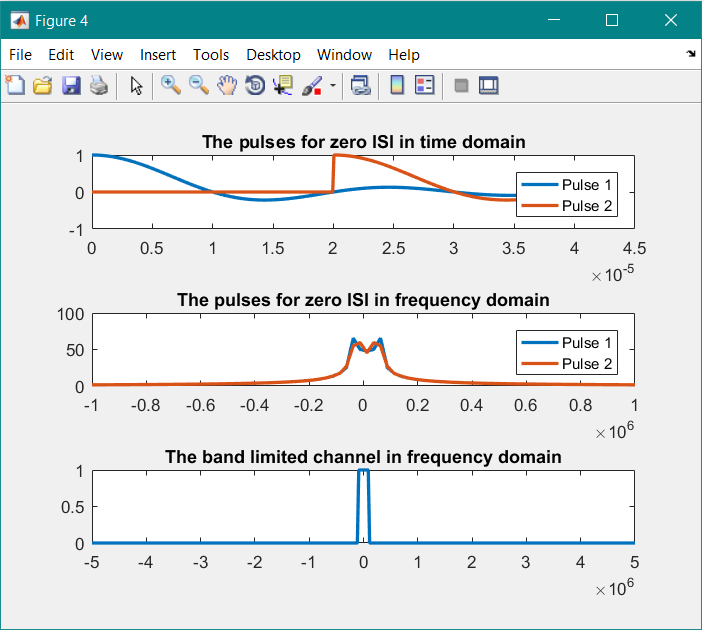
First sinc signal starts from zero and second starts from 201 after a vector of zeros in time domain



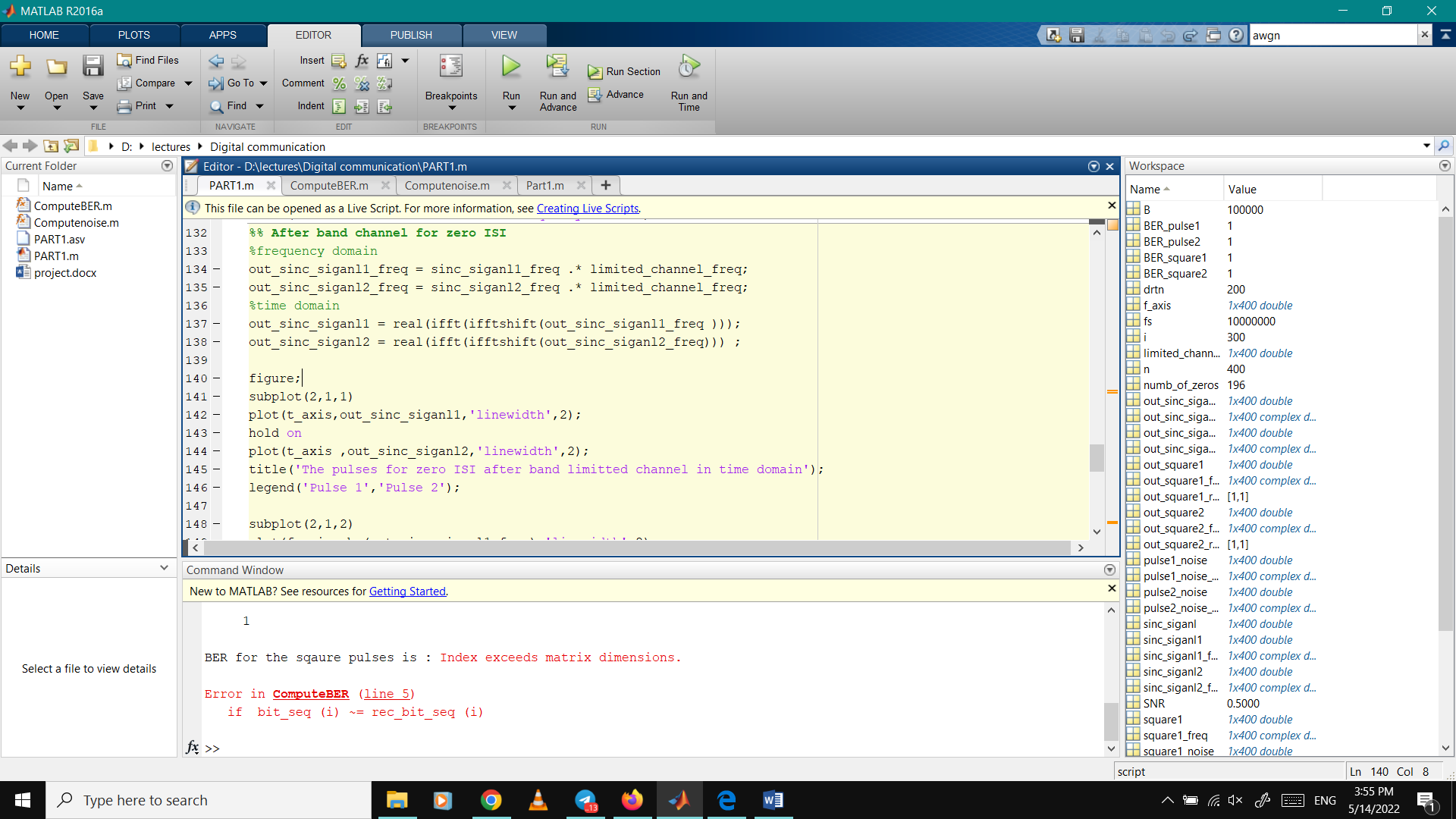
And in frequency domain we converted it by FFT

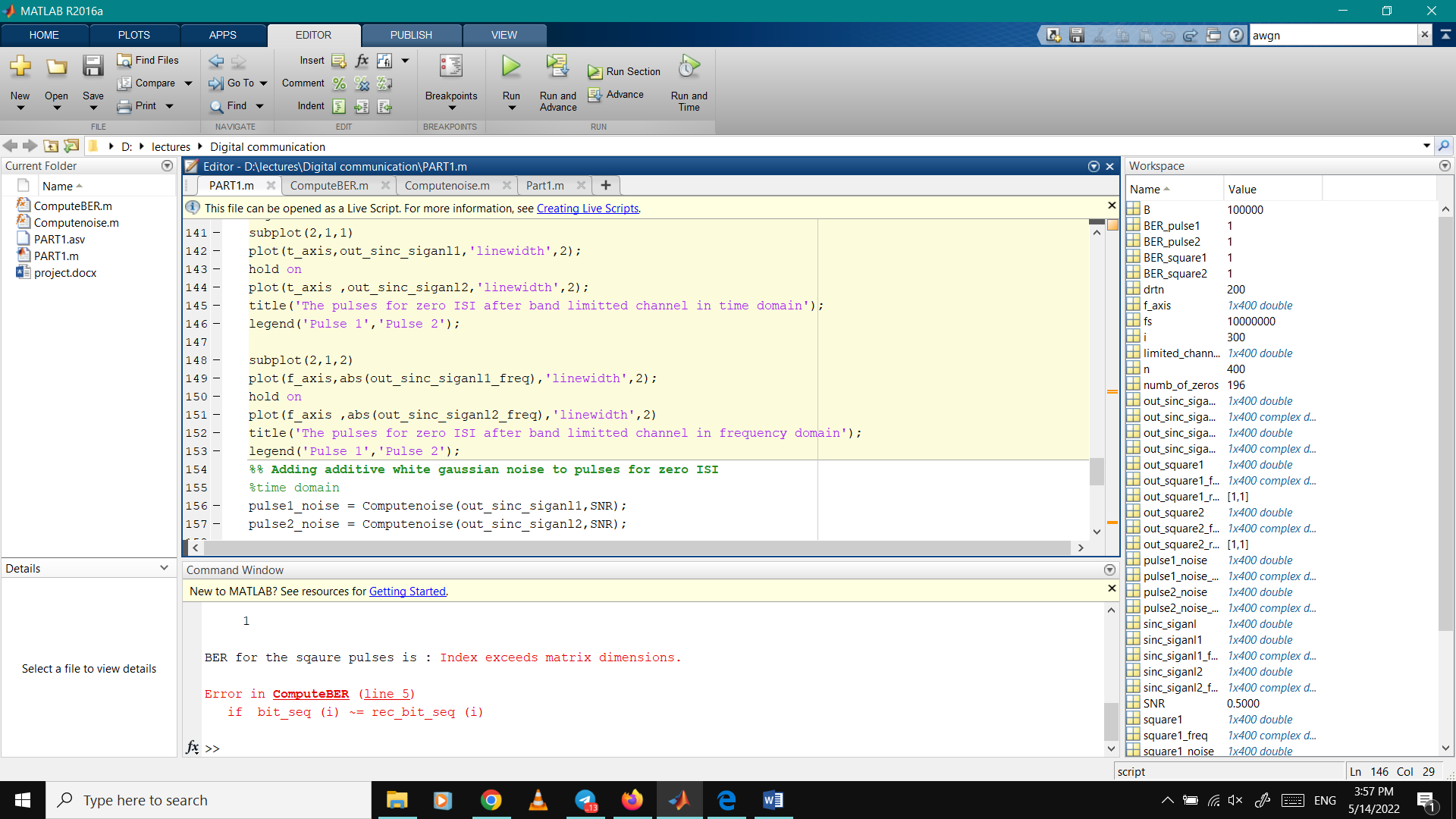


Plots of both sinc signals in time and frequency domain

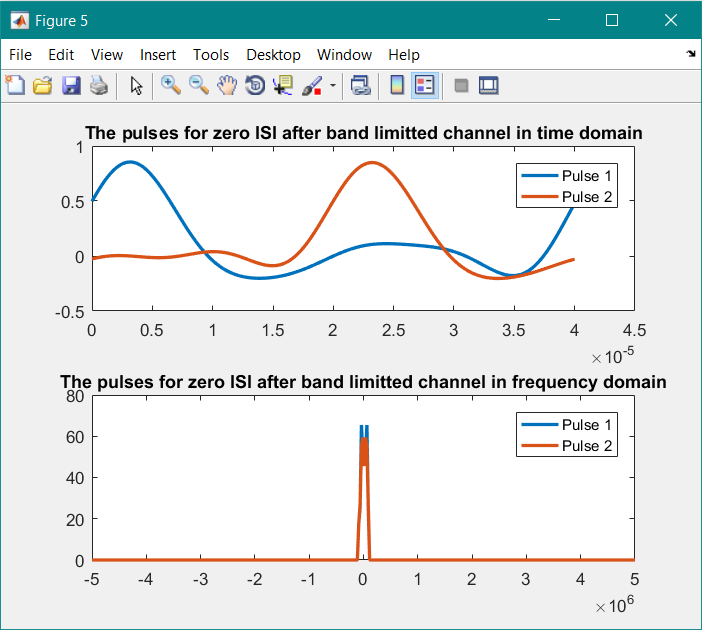


And after passing both of sinc signals through the band limited channel in frequency domain and time domain

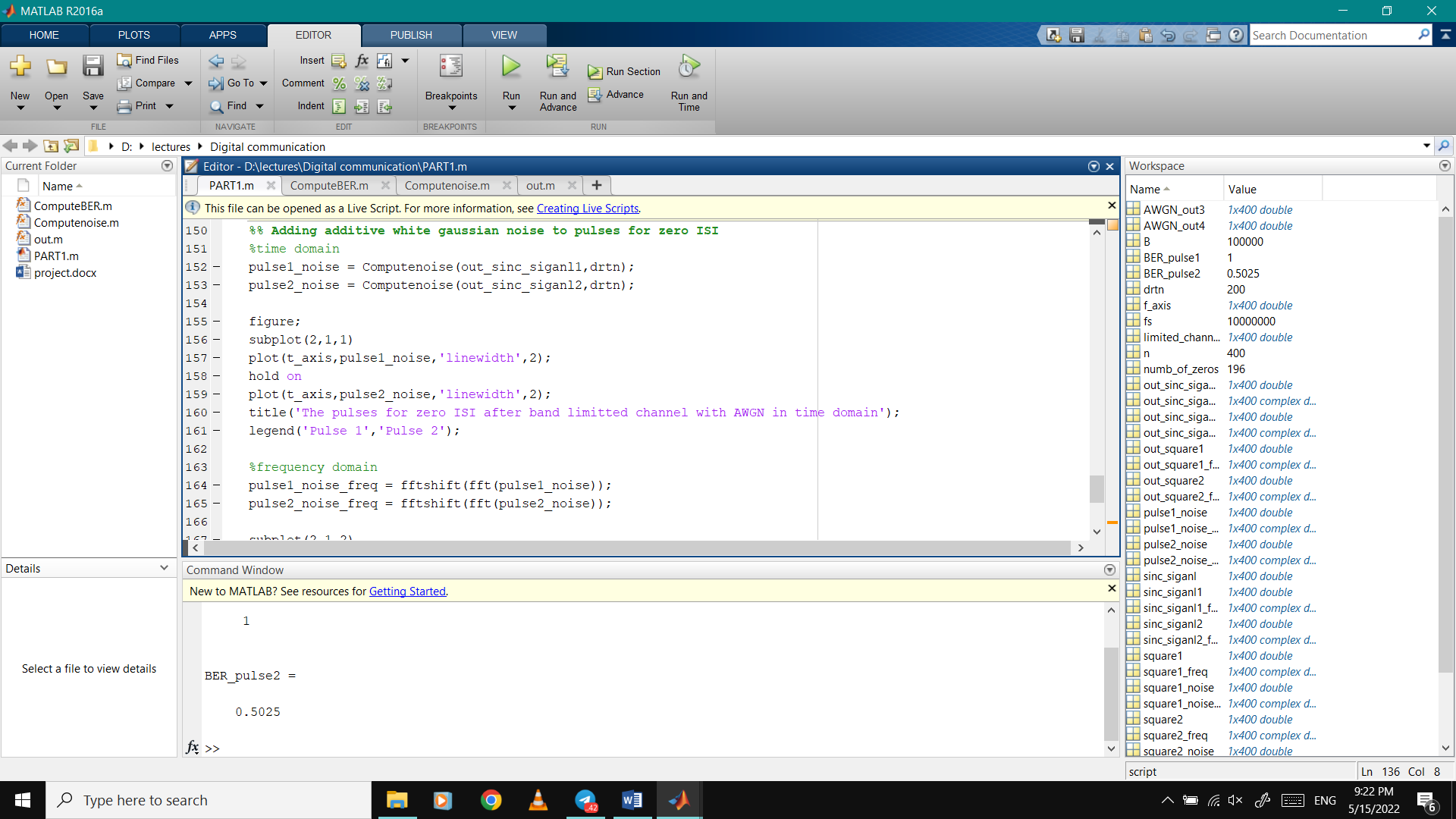


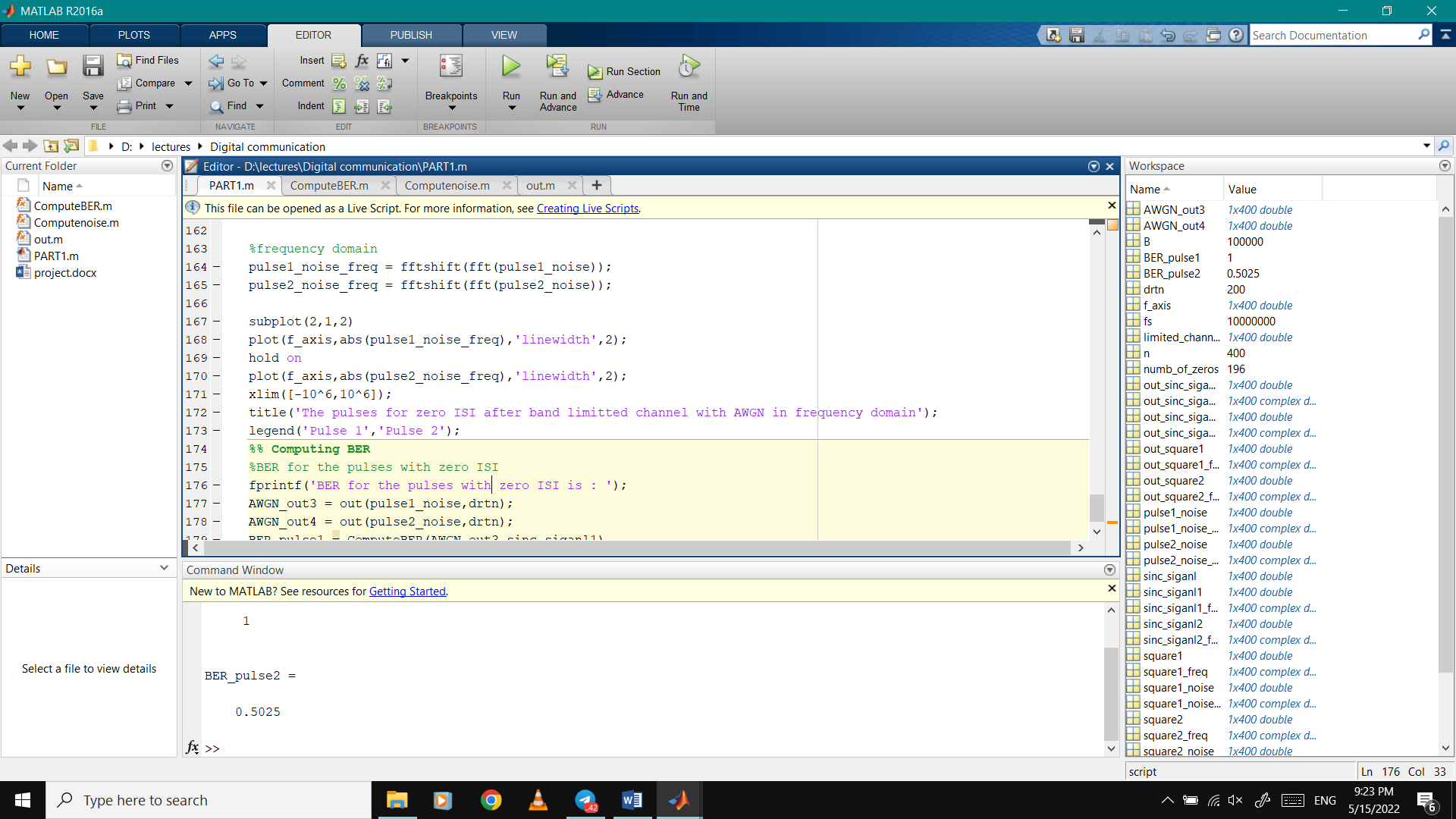


Plots after passing both of sinc signals through the band limited channel in frequency domain and time domain

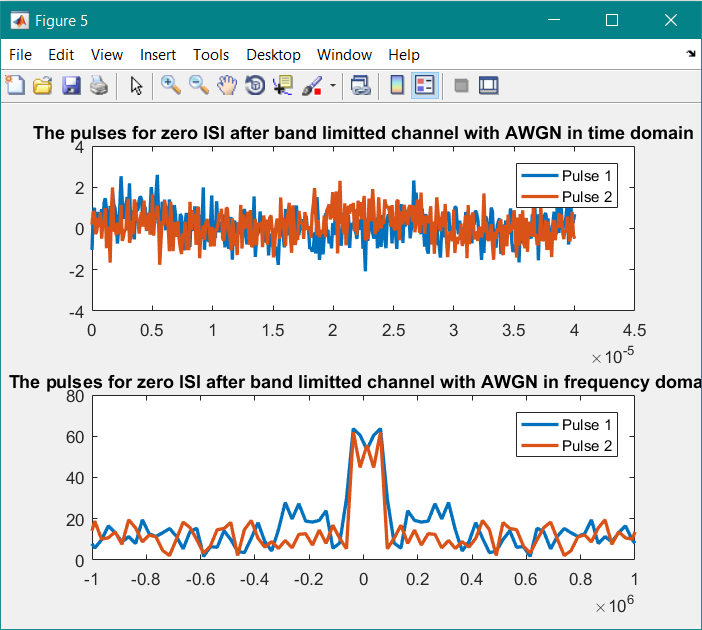


passing both of sinc signals through the AWGN channel in frequency domain and time domain



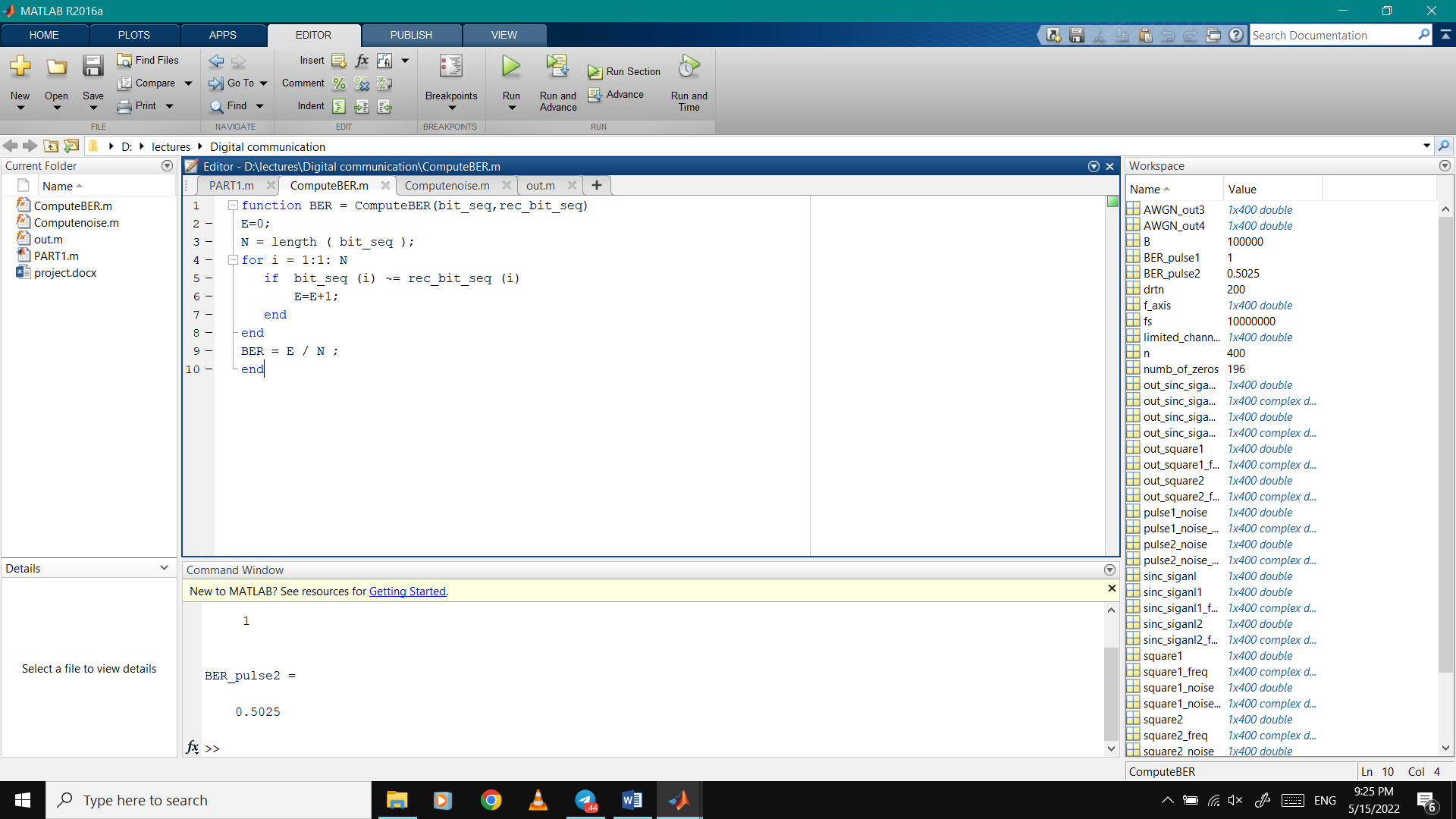


Plots after passing both of sinc signals through the AWGN channel in frequency domain and time domain

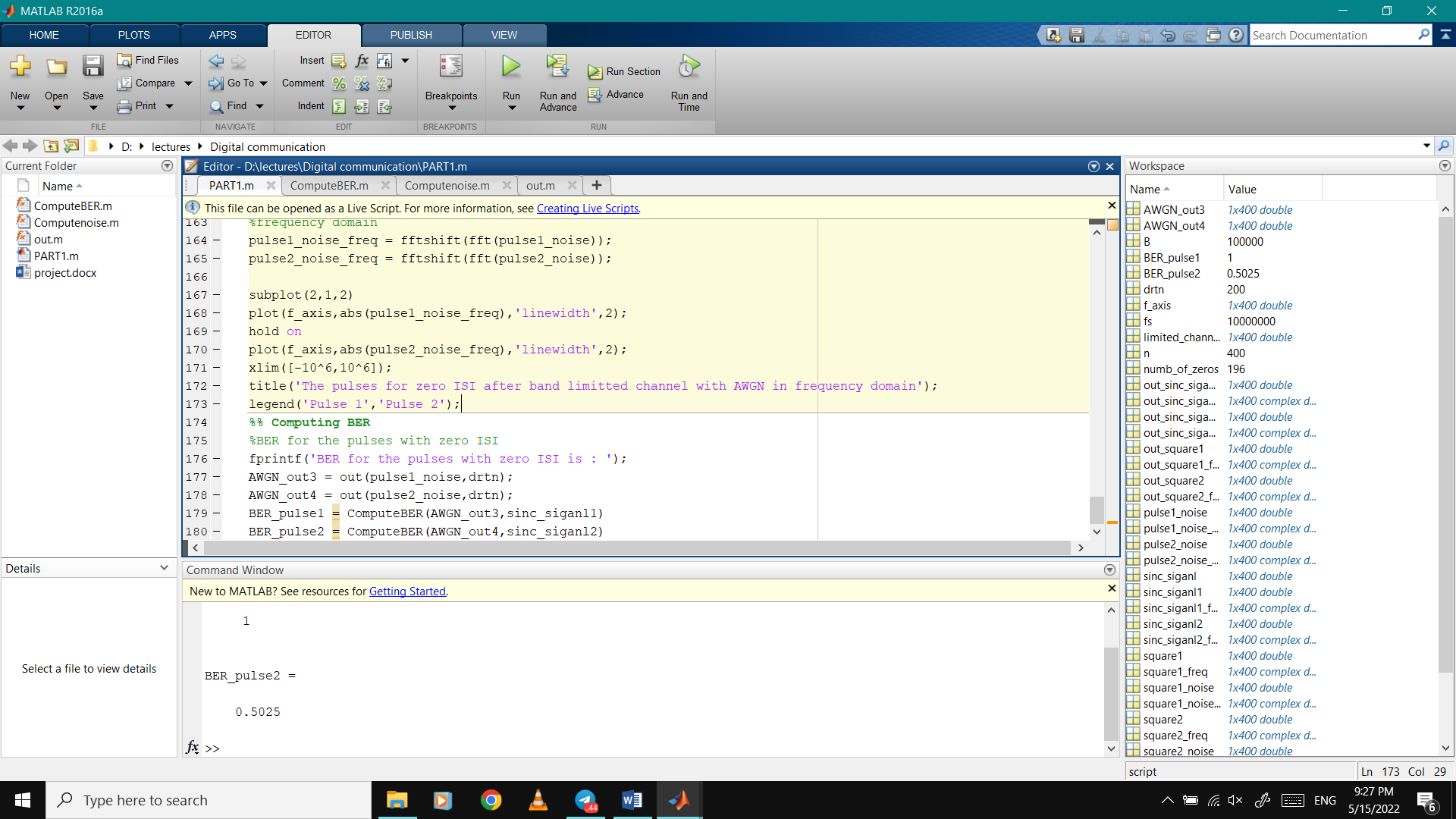


To compute BER we created function and called it ComputeBER :

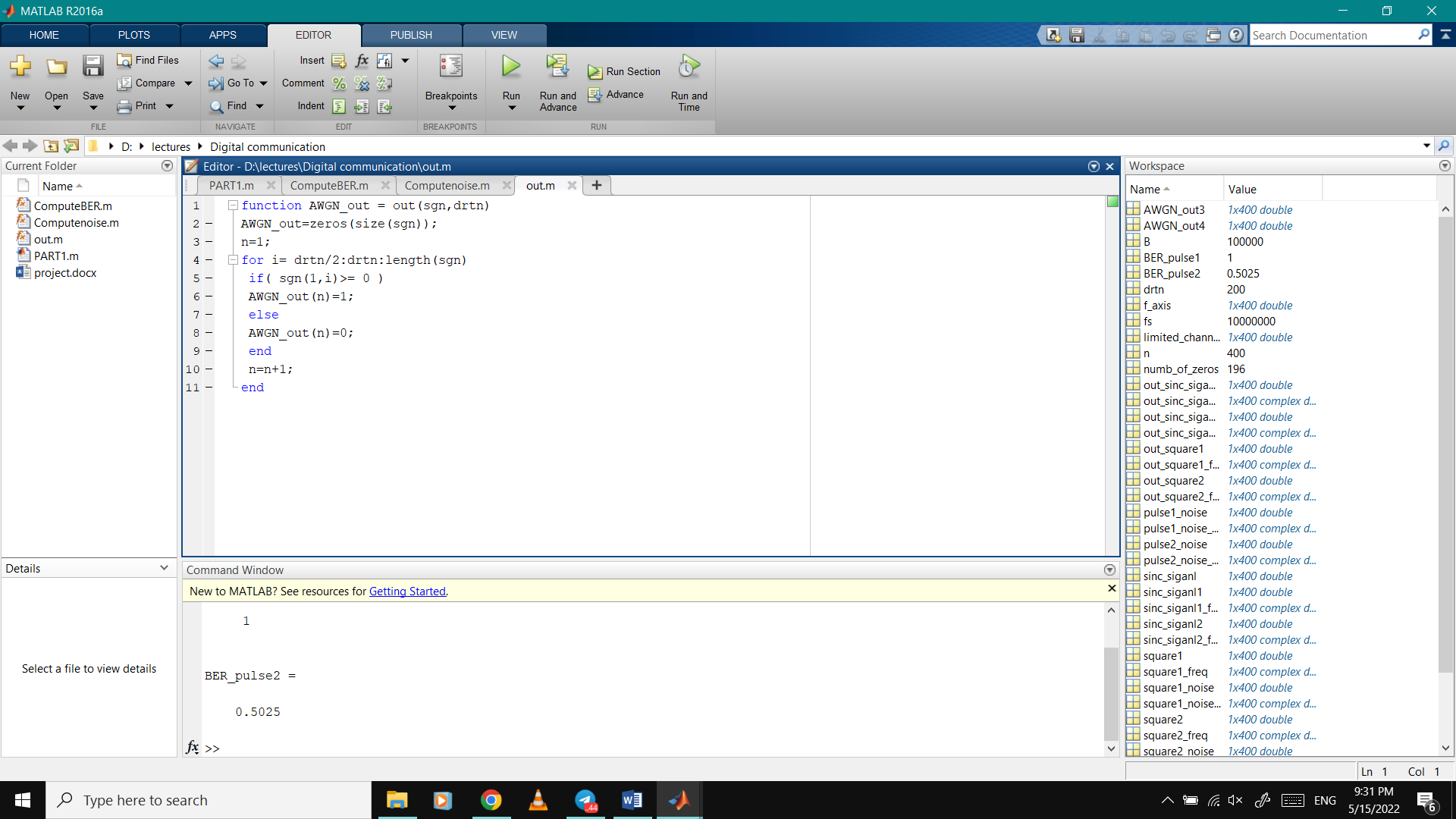
It compares the bit in the generated sinc signal before noise and the same bit after passing it through the noise.



In the main code :



We created a function called out to convert the signal bits into zeros and ones “ any value greater than or equal zero will be one , and other values will be zero “



And BER will be :

