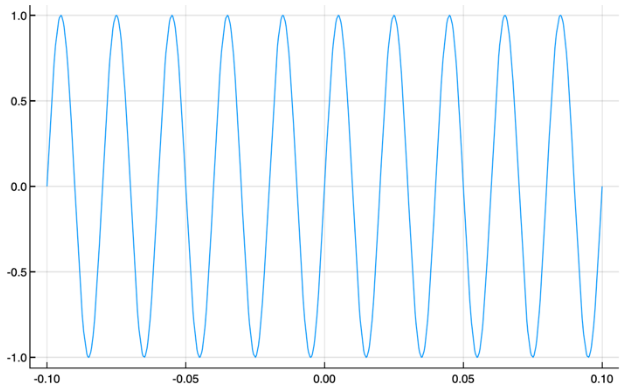
Assignment 2 Output Plots

Sampled Signals, Discrete Fourier Transform and Matched Filter

0.1 Julia Exercise 2.5.1 – Visualising Sampled Sinusoid

0.1.1 Simulate a sinusoidal signal over enough time to see several cycles.

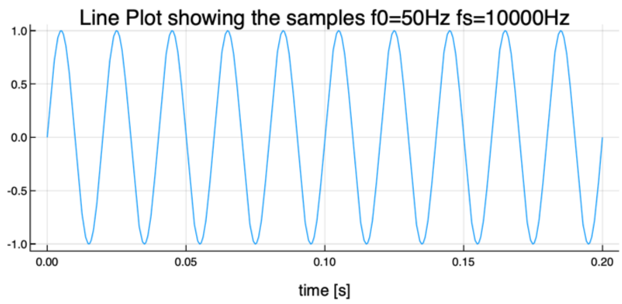
Out[2]:



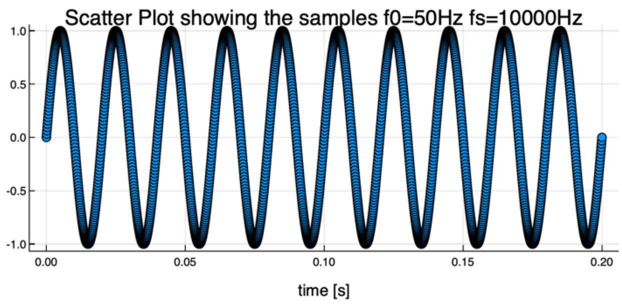
0.1.3 Plot the sampled waveforms in order to show the visual effect of sampling at:

0.1.4 100x the Nyquist rate

Out[5]:

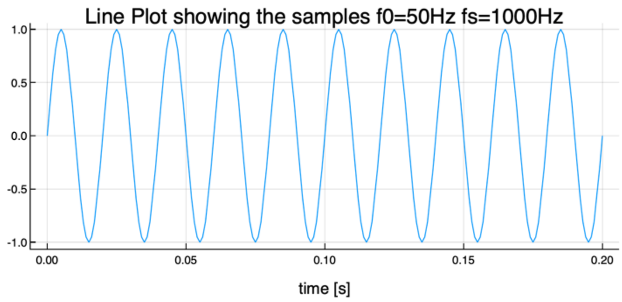


Out[6]:

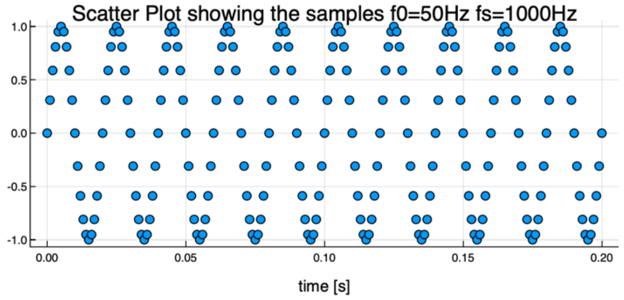


0.1.5 10x the Nyquist rate

Out[7]:

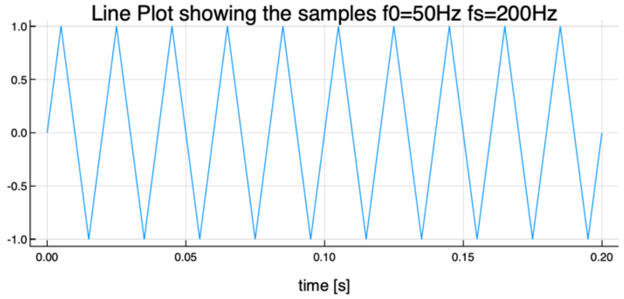


Out[8]:

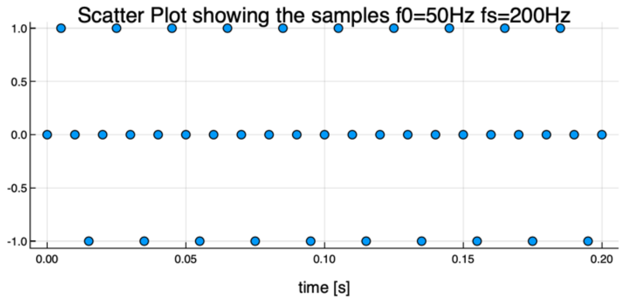


0.1.6 2x the Nyquist rate

Out[9]:

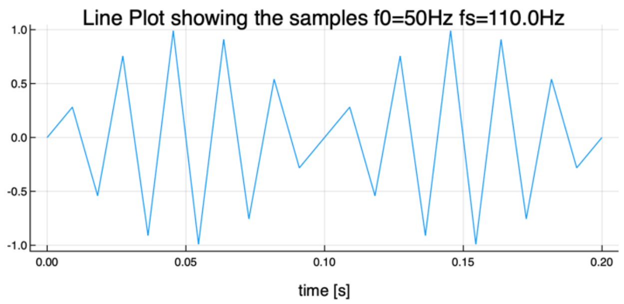


Out[10]:

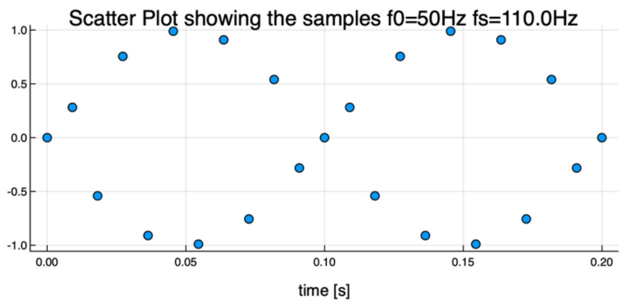


0.1.7 1.1x the Nyquist rate

Out[11]:

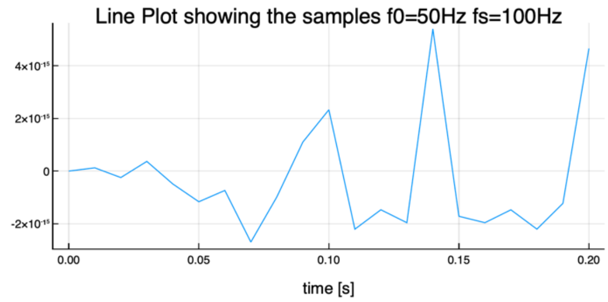


Out[12]:

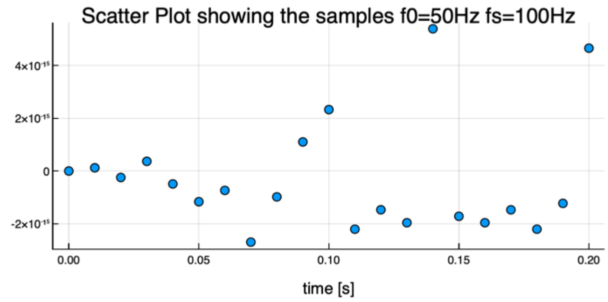


0.1.8 On the Nyquist rate

Out[13]:

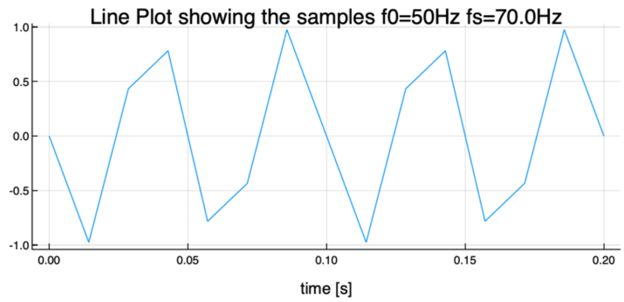


Out[14]:

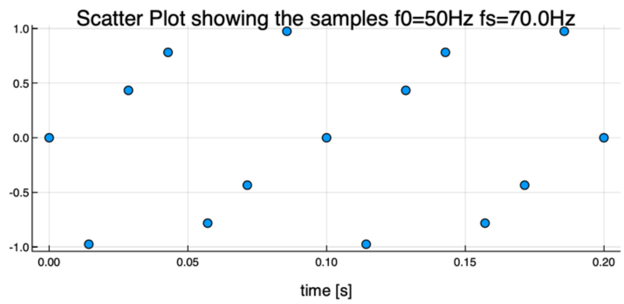


0.1.9 0.7x the Nyquist rate

Out[15]:

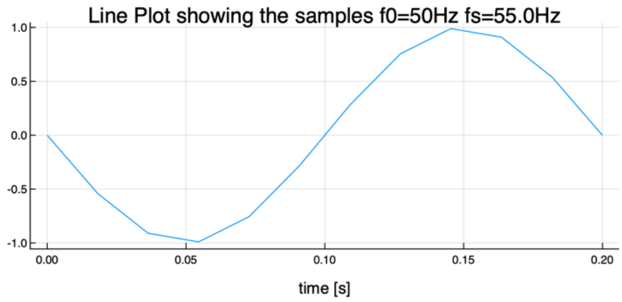


Out[16]:



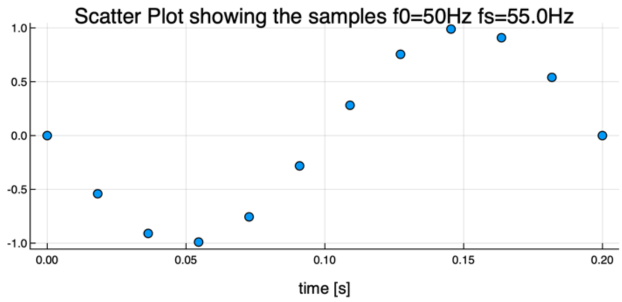
0.1.10 0.55x the Nyquist rate

Out[18]:



Line Plot showing the samples f0=50Hz fs=55.0Hz
1.0
0.5
0.0
-0.5
-1.0
0.00
0.05
0.10
0.15
0.20
time [s]

Out[19]:

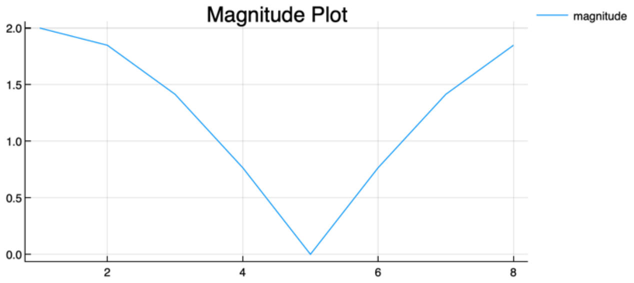


Scatter Plot showing the samples f0=50Hz fs=55.0Hz
1.0
0.5
0.0
-0.5
-1.0
0.00
0.05
0.10
0.15
0.20
time [s]

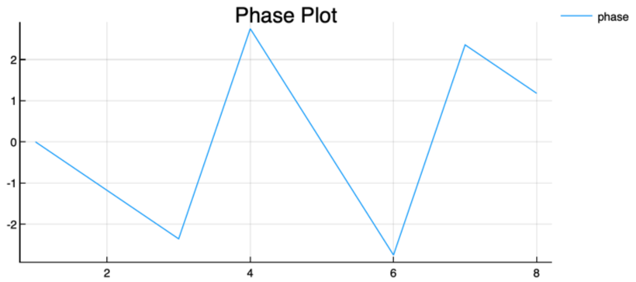
0.2 Julia Exercise 2.5.2 – DFT / FFT Introduction

0.2.3 Plot the magnitude and phase

Out[26]:



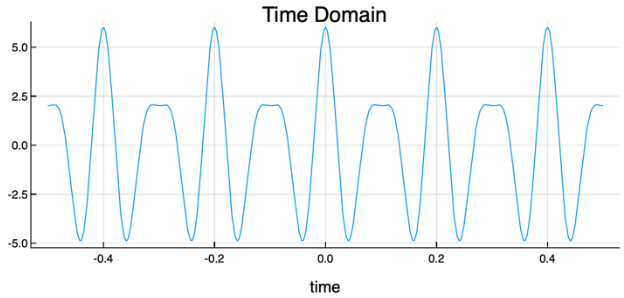
Out[27]:



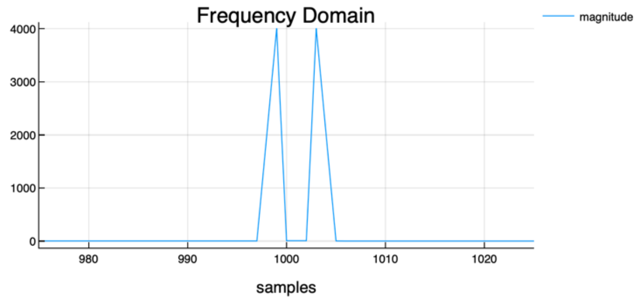
0.3 Julia Exercise 2.5.3 – FFT of a sine wave

0.3.1 Time domain: v(t) = 4 cos (20 t) + 2 cos (30 t)

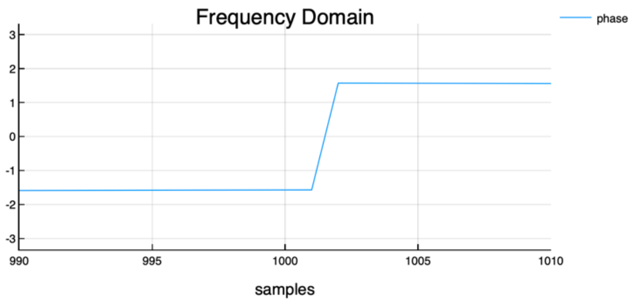
Out[36]:



Out[38]:

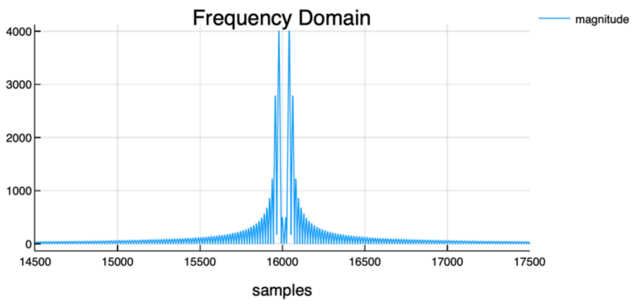


Out[39]:

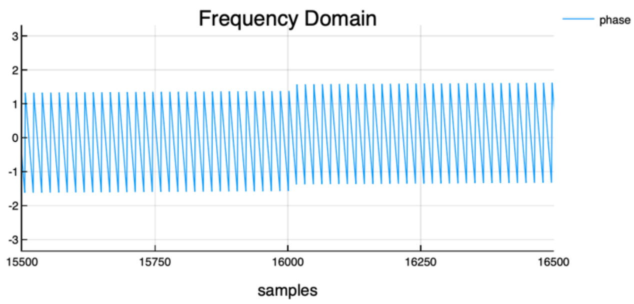


0.3.2 Applying zero padding in the time domain

Out[40]:



Out[41]:

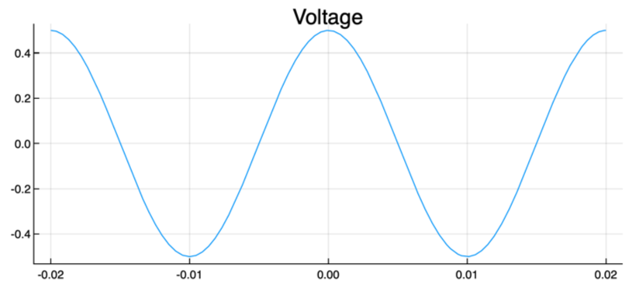


0.4 Julia Exercise 2.5.4 – Effect of ADC quantization

0.4.1 Simulate a sinusoid voltage

v = cos.(2pif0*t) that lies in the range: Amin = -0.5 to Amax = 0.5

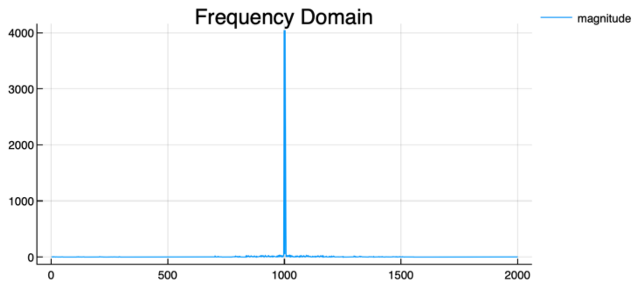
Out[42]:



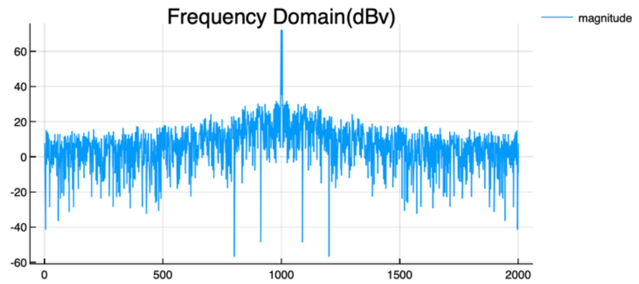
0.4.2 Quantize the signal into a power-of-2 levels

0.4.3 Number of Bits = 2

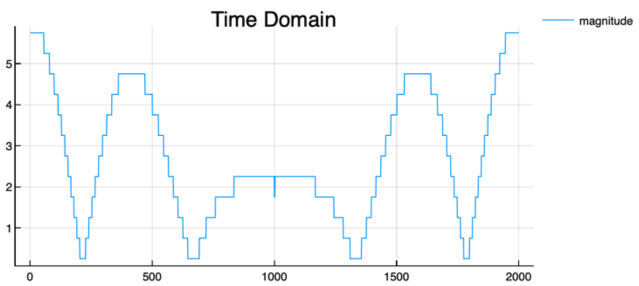
Out[45]:



Out[46]:

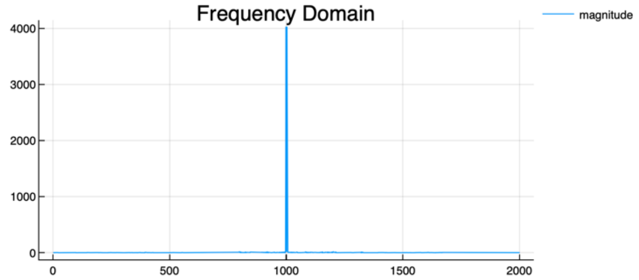


Out[47]:

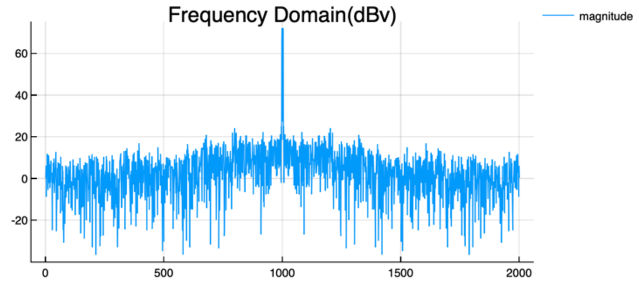


0.4.4 Number of Bits = 3

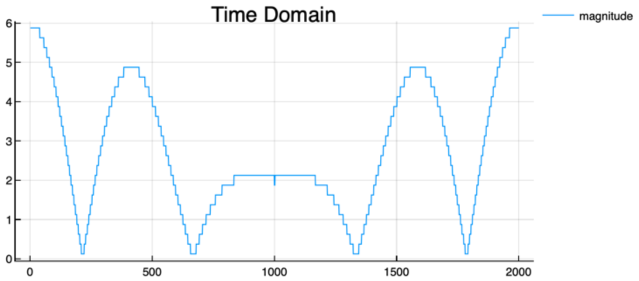
Out[48]:



Out[49]:

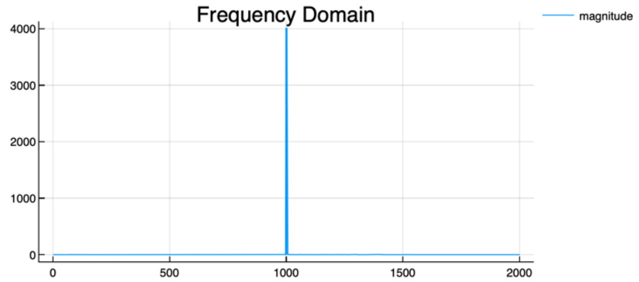


Out[50]:

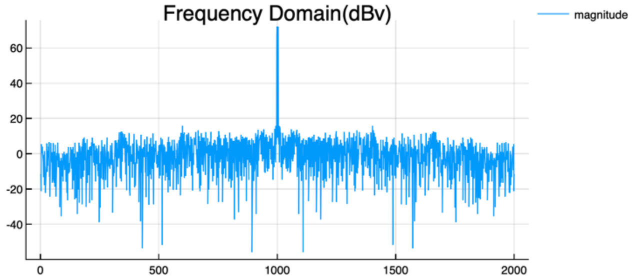


0.4.5 Number of Bits = 4

Out[51]:

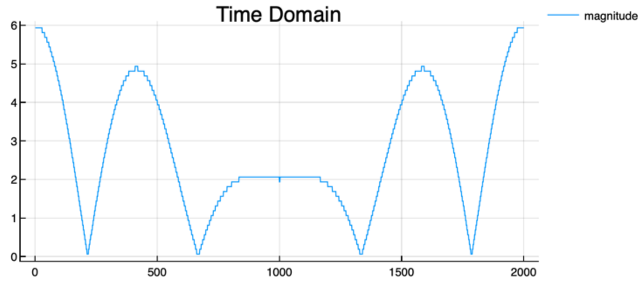


Out[52]:



Frequency Domain(dBv)
magnitude
60
40
20
0
-20
-40
0
500
1000
1500
2000

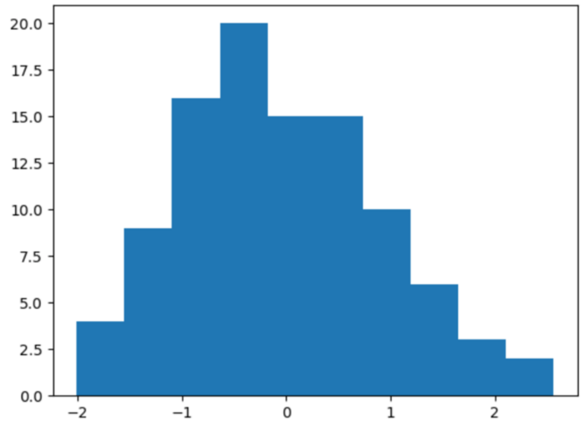
Out[53]:



Time Domain
magnitude
6
5
4
3
2
1
0
0
500
1000
1500
2000

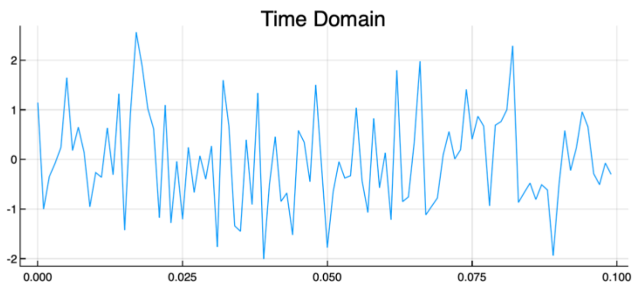
0.5 Julia Exercise 2.5.5 – Simulating bandlimited noise

0.5.1 Noise Simulation

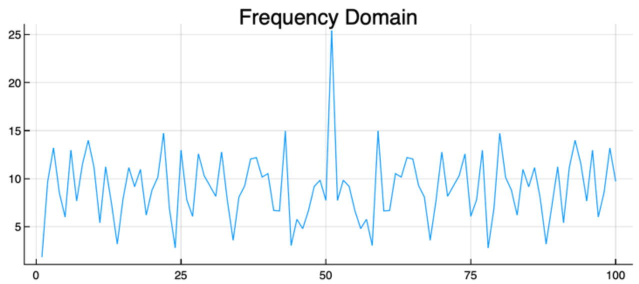


0.5.2 Time and Frequency Domains

Out[56]:

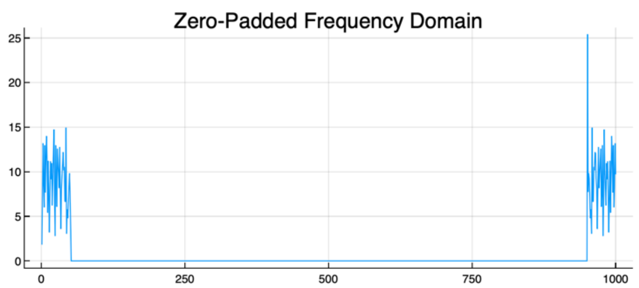


Out[57]:

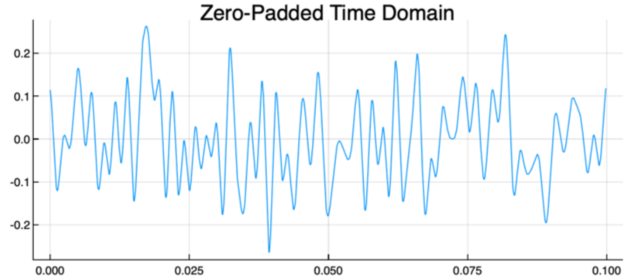


0.5.3 Applying frequency-domain zero padding

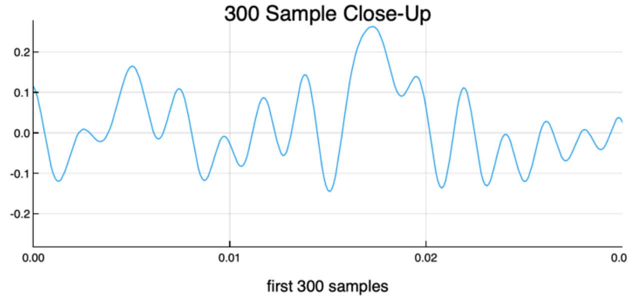
Out[58]:



Out[59]:



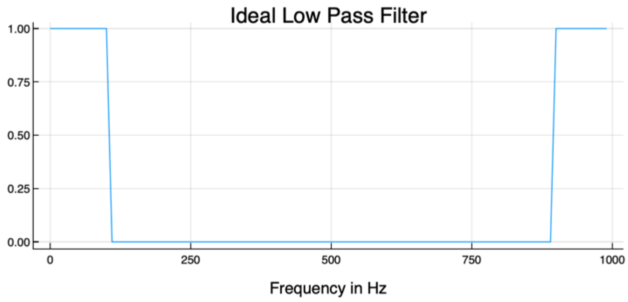
Out[60]:



0.5.4 Bandlimiting Noise using an Ideal LPF

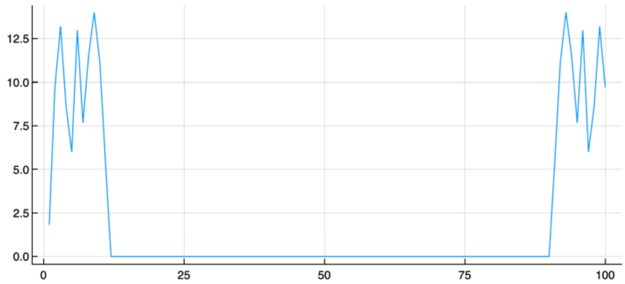
0.5.5 Create and display an ideal LPF

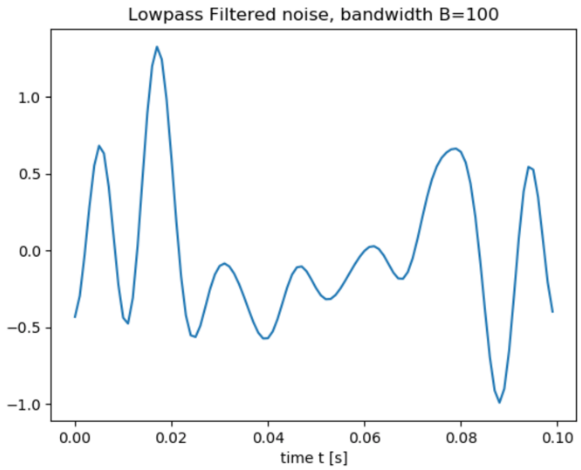
Out[61]:



0.5.6 Apply Filter to Noise

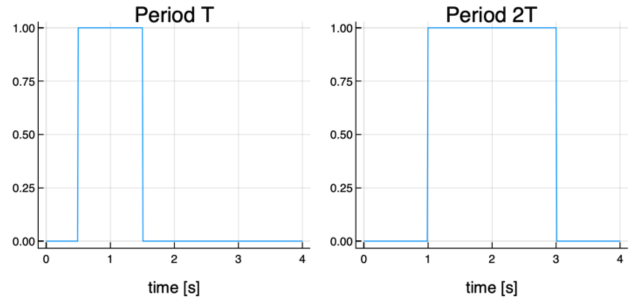
Out[62]:



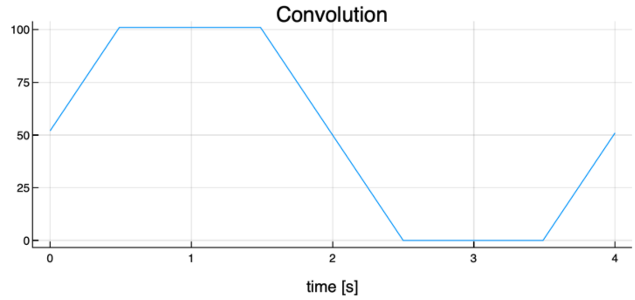


0.6 Julia Exercise 2.5.6 – Discrete fast convolution

Out[65]:



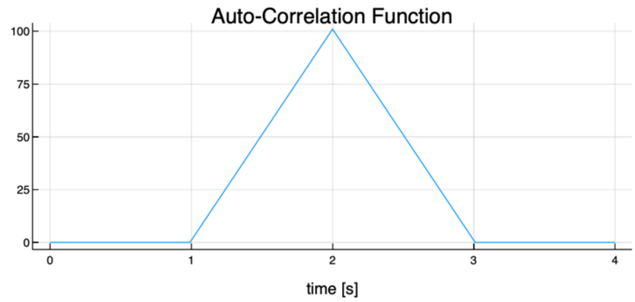
Out[66]:



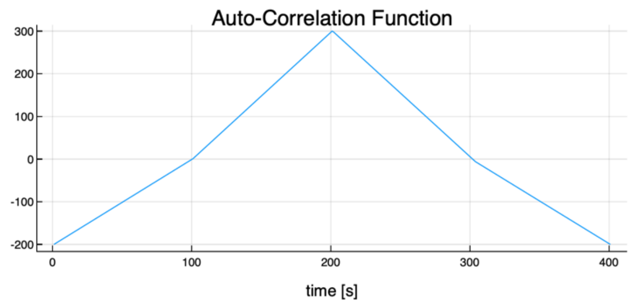
0.7 Julia Exercise 3.6.1 a) – Discrete fast correlation

0.7.1 Produce an auto-correlation function: Rxx(t') = conj(X( ))*X( )

Out[67]:



Out[68]:

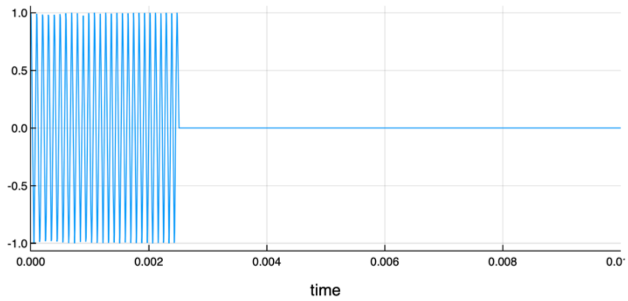


0.8 Julia Exercise 3.7.1 – Matched Filter

0.8.1 Define a chirp function.

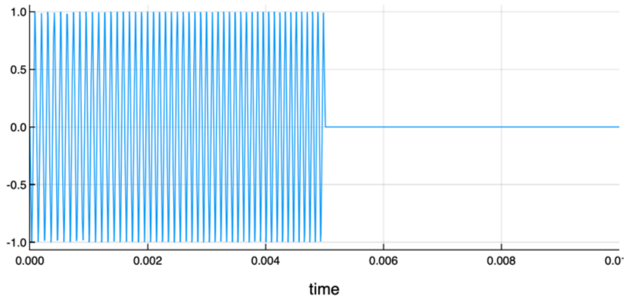
0.8.2 Setting: centre frequency of 10 kHz; bandwidth of 2 kHz; pulse length of 5 ms

Out[69]:



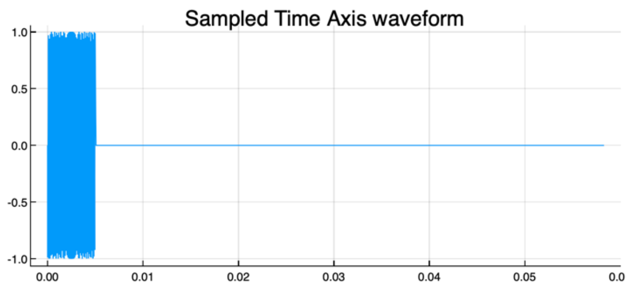
0.8.3 Define a physically realizable - delayed - chirp function.

Out[70]:



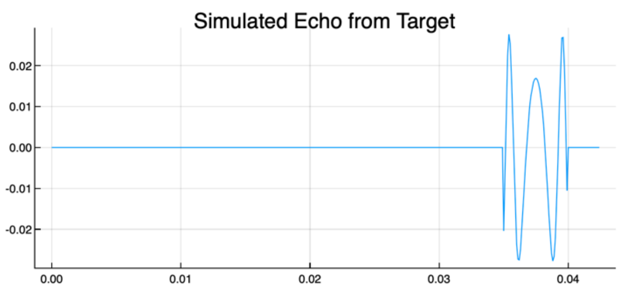
0.8.4 Sampled Time Axis

Out[72]:



0.8.5 Simulated Echo from Target

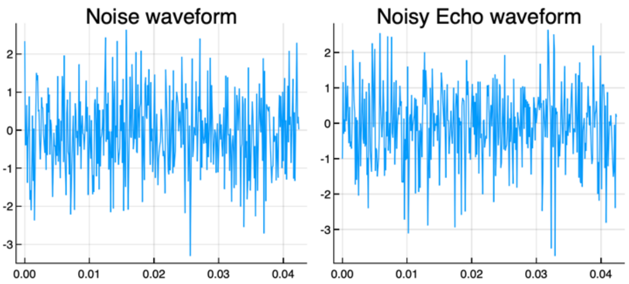
Out[73]:



0.8.6 Plot time domain waveforms, showing what goes into and out of the matched filter

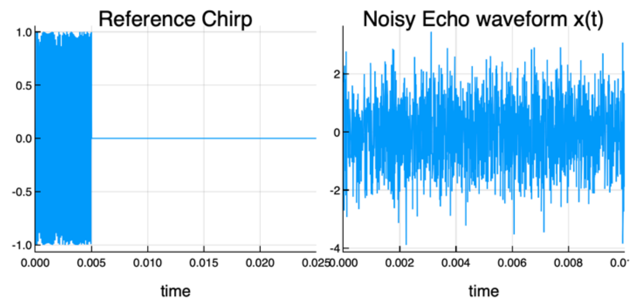
0.8.7 Echo from Target with Additive Noise

Out[74]:

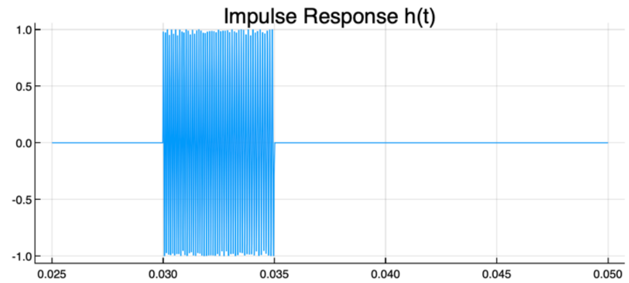


0.8.8 Apply a matched filter which is created from the reference chirp

Out[75]:

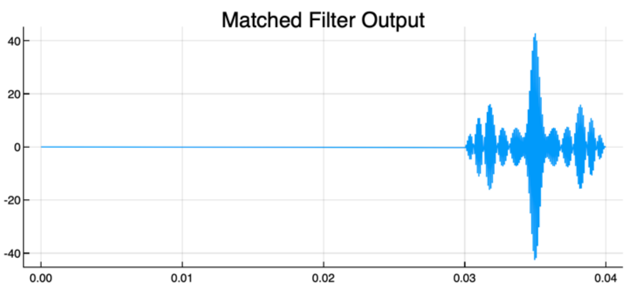


Out[76]:



0.8.10 Plot of output as a function of time

Out[79]:



0.8.11 Plot the magnitude of the FFT of the pulse, and also of the matched filter.

Out[80]: