Lab 1 - The Sampling Theorem and Fourier Analysis

Name 1: Ainhoa Arnaiz **Student ID 1:** H00301693
Name 2: Marta Gorecka **Student ID 2:** H00302212

Instructions: Provide your answers in the spaces below. Supplement with figures/codes where appropriate.

Part B

Question 1

One can construct the sequence shifted dirac delta sequence by using "*x1a = (n == m)*" where m is the shift amount. Explain why the above method for constructing the unit sample sequence works. (5%)

The above method for constructing the unit sample sequence works, because the delta dirac function operates only on zeros and a one. The "x1a = (n==m)" does the same as it is a logical array.

The base point for delta dirac function (with no shift, meaning m=0) is when 1 exists only when n=0. As the function is shifted, the "1" is being shifted as well. Therefore, the logical statement (n == m) is TRUE (gives 1) only when the value of m matches the number of a sample (value of n).

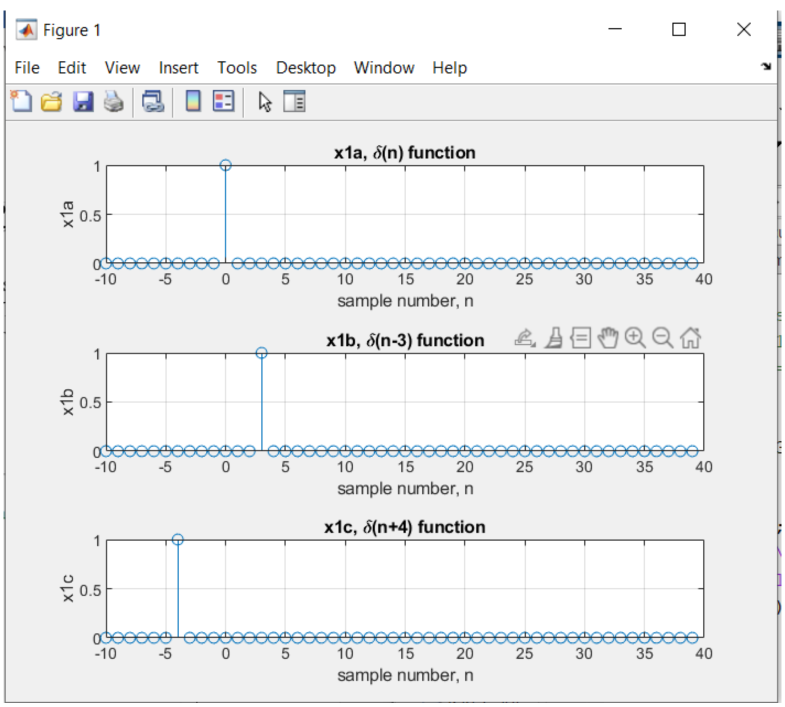

Figure 1 Exercise 1, graphs

```
n = [-10:39]; %number of samples
%% Part 1
impulse = @(n)  0*((n<0)|(n>0)) + 1*(n==0); %unit impulse funct

x1a = impulse(n);
x1b = impulse(n-3);
x1c = impulse(n+4);

%alternative methods for x1a
%
% x1a = zeros(1,50);
% x1a(11) = 1;
% x1a = (n == 0);

figure(1)
tiledlayout(3,1)

nexttile
stem(n, x1a);
title('x1a, \delta(n) function');
xlabel('sample number, n');
ylabel('x1a');
grid;

nexttile
stem(n, x1b);
title('x1b, \delta(n-3) function');
xlabel('sample number, n');
ylabel('x1b');
grid;

nexttile
stem(n,x1c);
title('x1c, \delta(n+4) function');
xlabel('sample number, n');
ylabel('x1c');
grid;
```

Figure 2 Exercise 1, code

Question 2

Explain why x2a = sign(1 + sign(n)) is equivalent to the unit step sequence. (5%)

Sign(n) gives (-1) as an output for negative values, in this case that would be for all n from -10 up to -1. The output is 0 when n=0 and, lastly, it gives 1 for all positive values of n (in this case: 1=< n =< 39). When adding 1 to these results, the received values will change to 0, 1, 2, respectively for these intervals. Therefore, the function sign(1+sign(n)) would give zeros as its output for all the values of n < 0 and ones for all the other, non-negative values (n>=0), as their sign is positive, ending up being a precise definition of the unit step sequence.

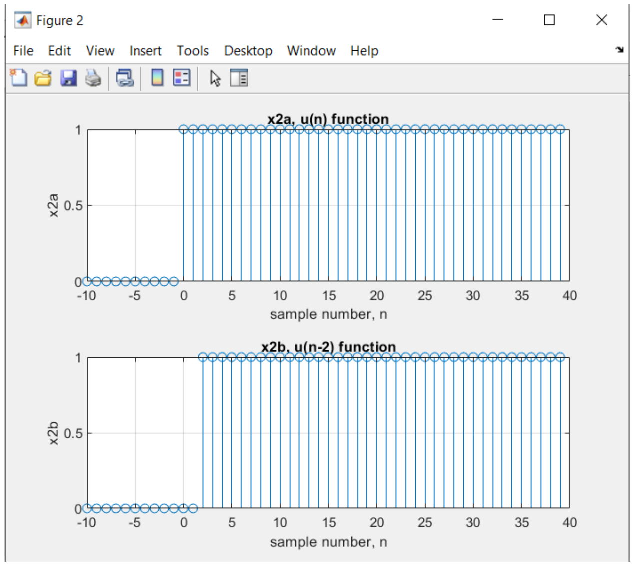

Figure 3 Exercise 2, graphs

```
%% Part 2
step = @(n) 0*(n<0)+1*(n>=0); %creating a unit step function

x2a = step(n);
x2b = step(n-2);
%OR
%x2a = (n>=0);
%x2b = (n>=2);
%OR
% x2a = sign(1 + sign(n));
% x2b = sign(1 + sign(n-2));

figure(2)
tiledlayout(2,1)

nexttile
stem(n, x2a);
title('x2a, u(n) function');
xlabel('sample number, n');
ylabel('x2a');
grid;

nexttile
stem(n, x2b);
title('x2b, u(n-2) function');
xlabel('sample number, n');
ylabel('x2b');
grid;
```

Figure 4 Exercise 2, code

Question 3

What is the appropriate sequence required for *p* in Part B Question 5) part (b)? (5%)

The appropriate sequence required for p is a pulse function, existing between 0 and 10, in order to ‘cut off’ the parts of the sawtooth wave for n<0 and n >10.

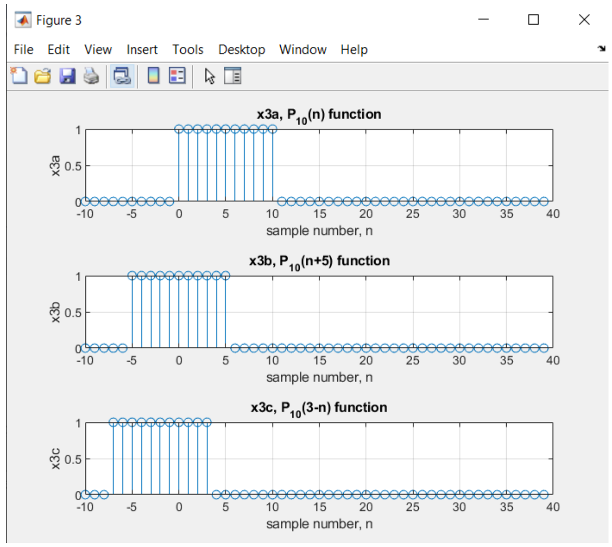

Figure 5 Exercise 3, graphs

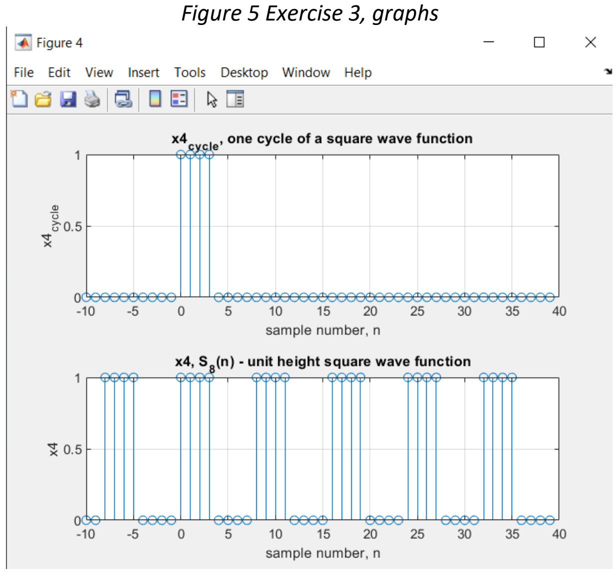

Figure 6 Exercise 4, graphs

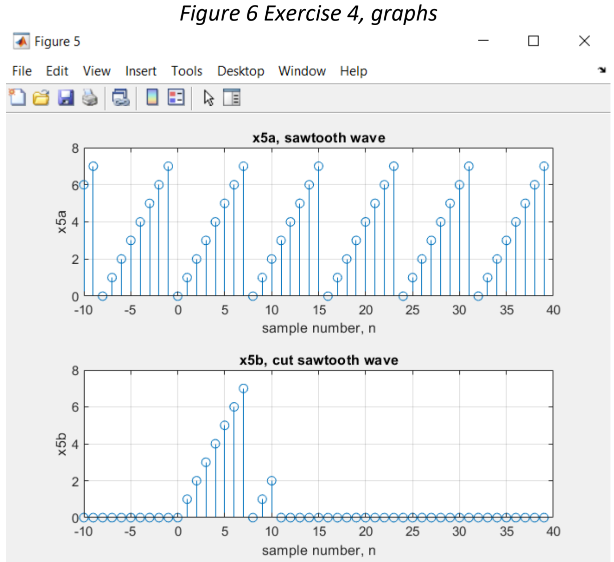

Figure 7 Exercise 5 graphs

```
%% Part 5
sawtooth_period = @(n) n.*((n>=0)&(n<8)); %one cycle

%repeating for obtaining a full sequence
x5a = 0;
for i = -5:5
    x5a = x5a + sawtooth_period(n + 8*i); %creating a sequence
end

%setting to 0 all the values except for the ones between 0 and
x5b = x5a.*pulse(n, 10);

figure(5);
tiledlayout(2,1);

nexttile
stem(n, x5a);
title ('x5a, sawtooth wave');
xlabel('sample number, n');
ylabel('x5a');
grid;

nexttile
stem(n, x5b);
title('x5b, cut sawtooth wave');
xlabel('sample number, n');
ylabel('x5b');
grid;
```

Figure 8 Exercise 5, code

Part C

Question 4

Explain your observations in Part C) Question (b) and (c) by considering the problem of sampling a harmonic signal. (15%)

Since x=sin(t*8), the cutoff frequency, fc, equals 8. It is known that for the sinusoidal functions the "Nyquist rate" would be equal twice the value of the frequency. Therefore, in this case the sampling frequency is 2*8=16.

In (b), when the interval, T, is equal to half of the value of the Nyquist rate (0.5*fs = 8), the signal is nearly ideally recreated, however, some imperfections can be observed there – the signal is not fully recovered.

In (c) for T = fs = 16, further imperfections can be observed, compared to (b) and when T = 3/2*fs = 24, significant differences between the original and reconstructed signals can be noticed.
This leads to a conclusion that the greater the interval, the less accurate the reconstructed signal is. This is in line with theory and expected results for this experiment as the greater the interval step becomes, the fewer samples can be fit in the length of the signal.

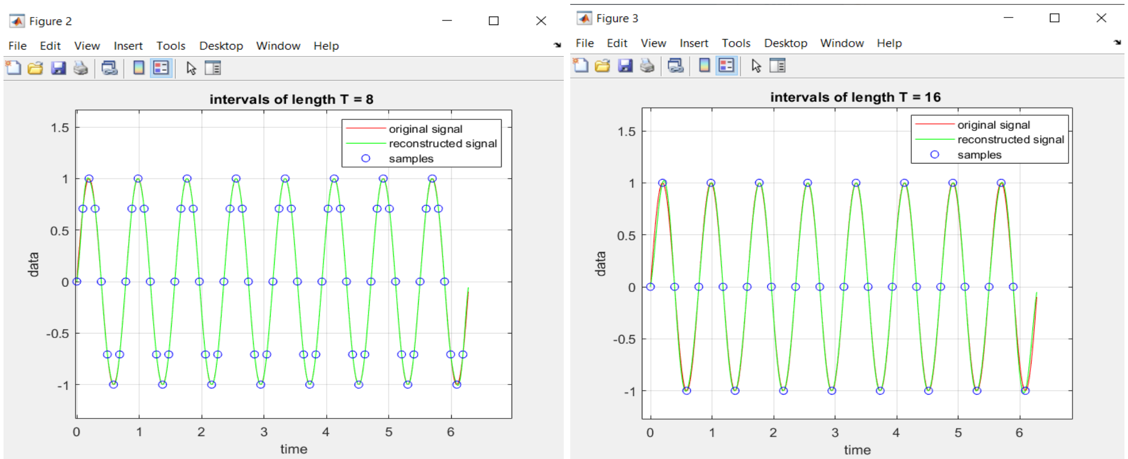

Figure 9 Part D, graph 1 and graph 2

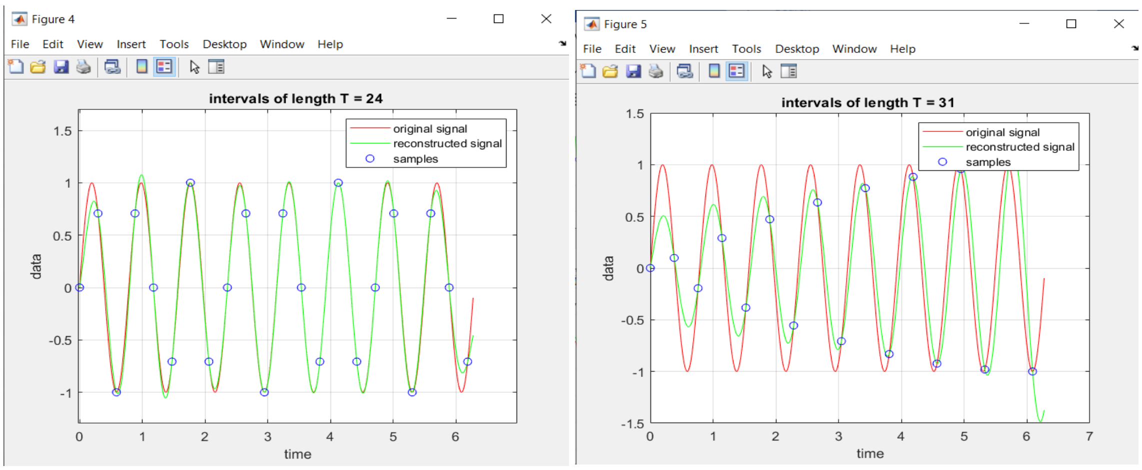

Figure 10 Part D, graph 3 and graph 4

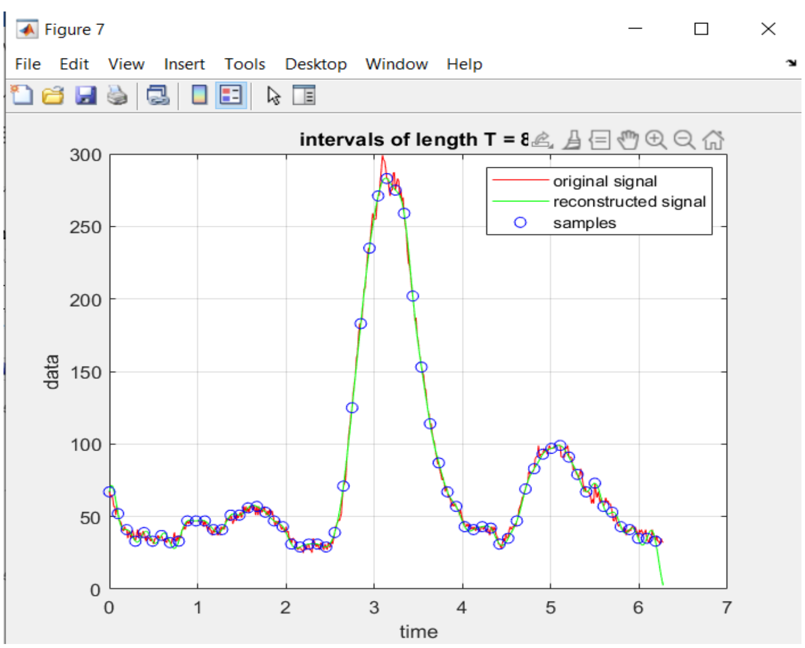

Figure 11 Part D, bv graph

```
%% a
t = [0:511]*2*pi/512;
fc = 8;
x = sin(t.*fc);

figure(1)
plot(t, x);
grid;
%% b
fs = 2*fc; %Nyquist rate
sub_sampling_interval = [16:32]; %sub-sampling interval
T = 0.5*fs;

shannon(x, t, T, 2);
%% c
shannon(x, t, fs, 3);
shannon(x, t, 3/2*fs, 4);
shannon(x, t, 31, 5);

%% d
r = [ones(1,64) zeros(1,64)];
x = [r r r r];

shannon(x, t, T, 8);
%% e
bv
shannon(x, t, T, 6);
%% functions
function y = shannon(x,t,T, i) ...
```

Figure 12 Part D, code

Question 5

Explain why the reconstructions in (d) are expected to be more distorted than those at the same sampling rate in (b), even though the fundamental frequency in (d) is less than that in (b). (15%)

The reconstruction in (d) is expected to be more distorted than those in (b) because of the Gibbs phenomenon occurring at the discontinuities of the square-wave function presented in (d). Since the reconstruction process is based on sinusoidal functions, 'overshoots' occur at discontinuities of the square-wave function. Thanks to the fact the first signal is sinusoidal (continuous), it is much easier to reconstruct it with a method making use of a sinusoidal signal too.

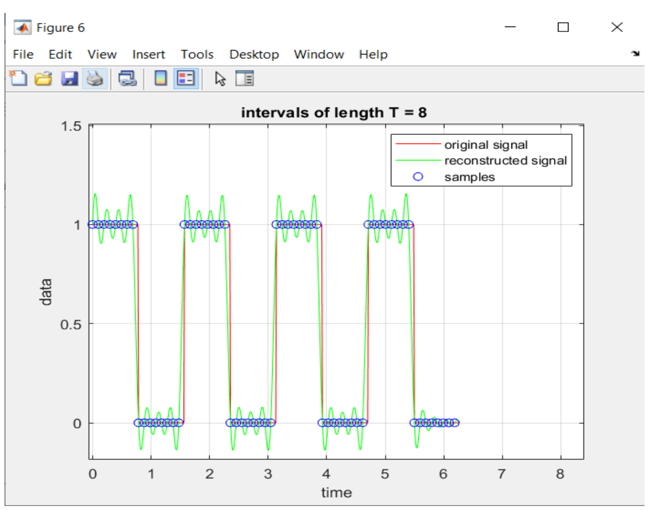

Figure 13 Part D, square function, graph

Part D

Instruction: You need to provide all the m-files used for this Part separately as .m files. Do not attach your source code. A software plagiarism checker such as MOSS may be used to check your programs.

Z = ceil(X+Y/2) > Z = ceil(2+3/2) = 4 > Wav file name: pin_04.wav

PIN decoded: 6 4 5 8
Sampling Frequency: 8000Hz
Pulse width: 100ms
Pulse spacing: 150ms

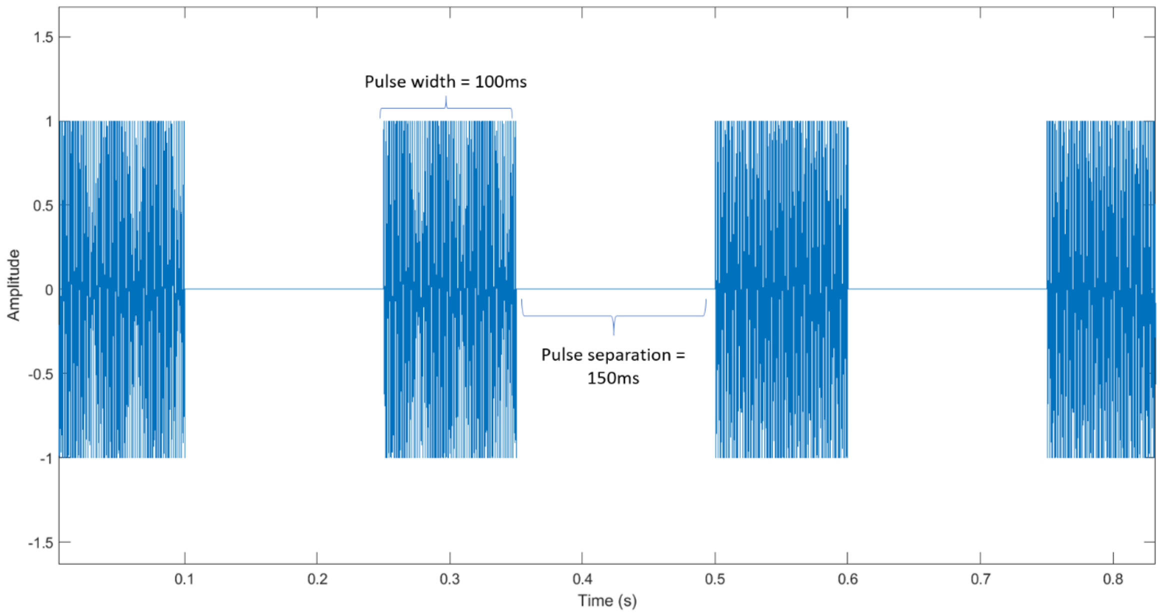

Figure 14 Representation of the pulses in time domain

Magnitude Spectra for each digit in the Pin. Use as much space as you need.

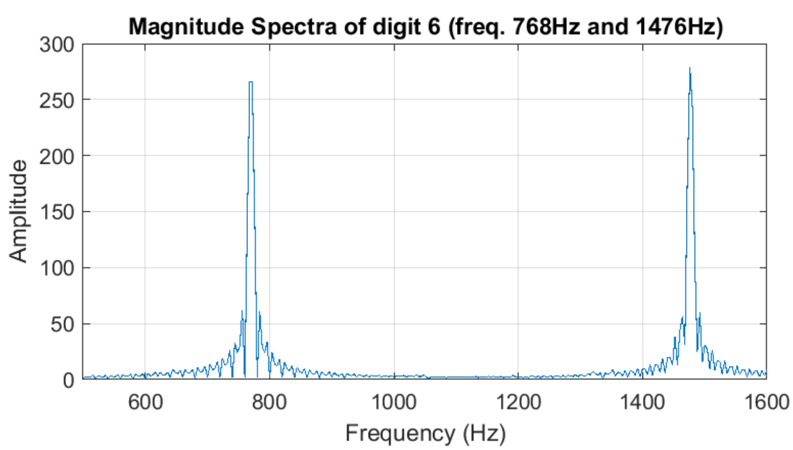

Figure 15 Magnitude spectra, digit 6

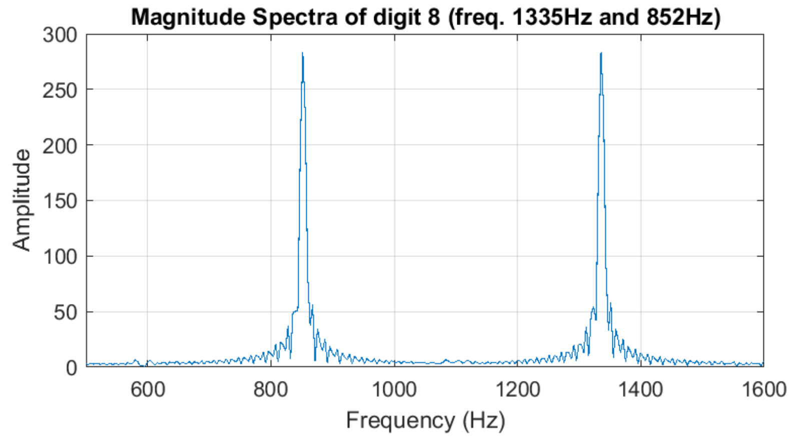

Figure 16 Magnitude spectra, digit 4

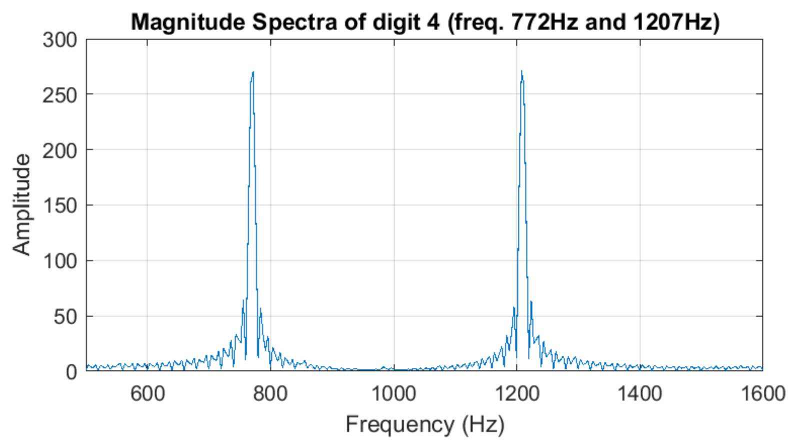

Figure 17 Magnitude spectra, digit 5

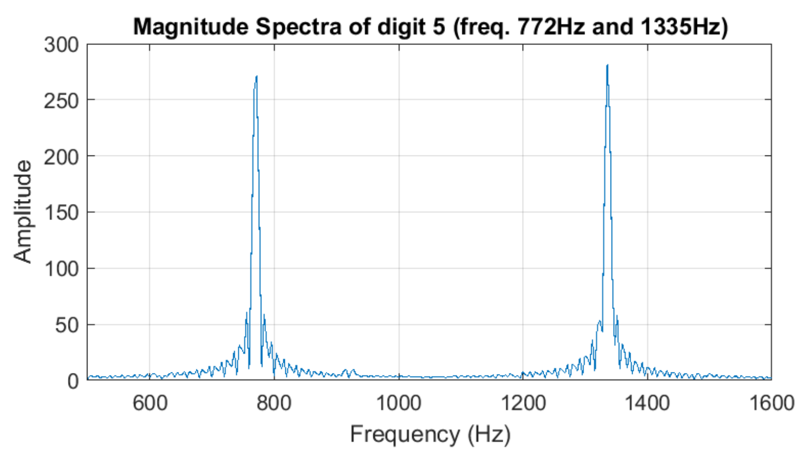

Figure 18 Magnitude spectra, digit 8

Choice of window function and window width and why:

Leakage cannot be avoided, but it can be reduced by applying windowing functions to the data prior to performing an FFT. The three main window functions are the Hanning window, Hamming window and the rectangular window. When choosing one, the two primary factors that describe a window function need to be considered:

1. Width of the main lobe (i.e., at what frequency bin is the power half that of the maximum response)
2. Attenuation of the side lobes (i.e., how far away down are the side lobes from the main lobe). This reveals the spectral leakage in the window.

The rectangular window yields the narrowest main lobe, providing the best frequency resolution for spectral analysis. However, the first side lobe for the rectangular window drops only 13 dB below the peak of the main lobe, which is not good because it increases leakage. On the other hand, the Hanning and Hamming windows reduce the levels of the side lobes, but their main lobes are almost twice as wide as the main lobe of the rectangular window. Wider main lobes degrade frequency resolution, yet their small side lobes greatly reduce leakage. The loss of frequency resolution can be overcome by sampling the signals faster during the sample period.

The Hamming window has the lowest first side lobe level of all three types of windows, but after its first side lobe, its remaining side lobes decay slowly relative to those of the Hanning window. The slow decay means that leakage two or three bins away from a signal's centre frequency is lower for the Hamming window than for the Hanning window. But leakage a half dozen or so bins away from a signal's centre frequency is much lower for the Hanning window than for the Hamming window. Therefore, Hanning window became our choice among all the options, as it results in a signal with smoother transitions overall. However, the Hamming window would also be a suitable option (see Figure 19.).

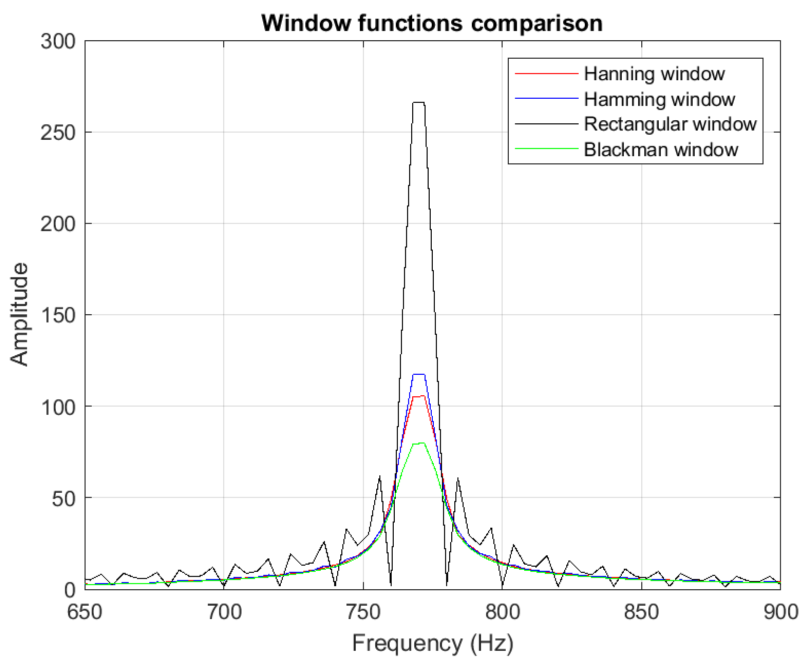

Figure 19 Window functions comparison

Windowing works by forcing your data smoothly to zero at exactly the start and the end of the sequence, but not before. Shortening the window destroys information unnecessarily. Thus, the window length should match the length of the sample sequences. For instance, each digit consists of 2000 samples, so the window length should be 2000. However, as it can be observed from the graphs below, if we decrease the number of samples to 1200, it smooths down the originally spiky representation of the signal, keeping the most important parts in place. Even though this might seem to be a less accurate solution, compared to having 2000 samples, the width of the main lobe remains the same, as well as the leakage of the side-lobes. The only noticeable difference is that the amplitude of the frequency is kept closer to the original. For that reason, using 1200 samples seems to be a reasonable choice, as the most important information is well preserved.

Of course, as observed from figures 21 and 23, the number of samples must be reasonably chosen and cannot be too small either, as the graph loses its shape and accuracy.

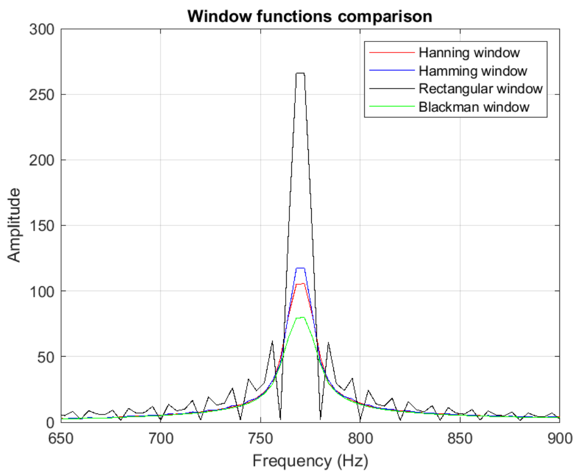

Figure 20 Window functions comparison, 2000 samples

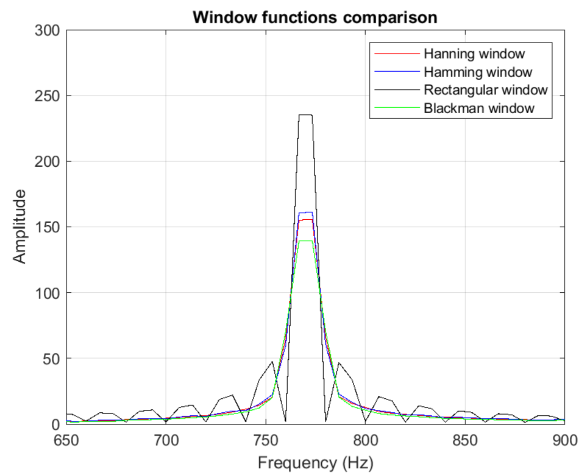

Figure 22 Window functions comparison, 1200 samples

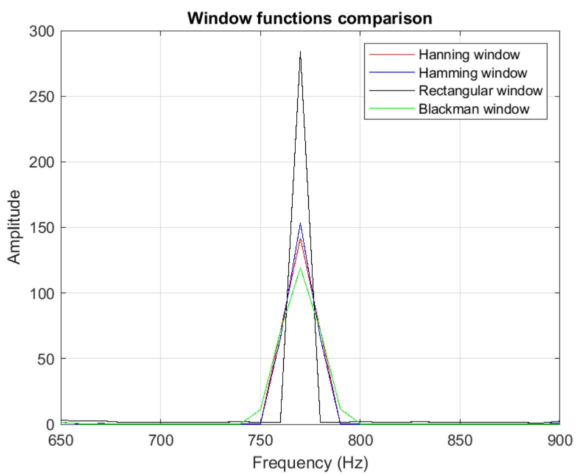

Figure 21 Window functions comparison, 800 samples

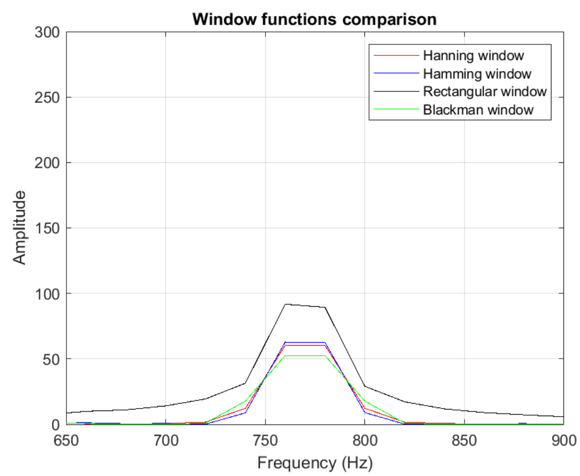

Figure 23 Window functions comparison, 400 samples

Magnitude spectra after Hanning windowing (1200 samples):

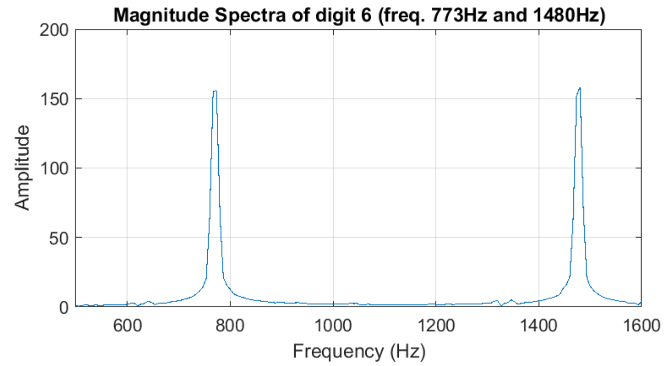

Figure 24 Magnitude spectra after Hanning windowing, digit 6

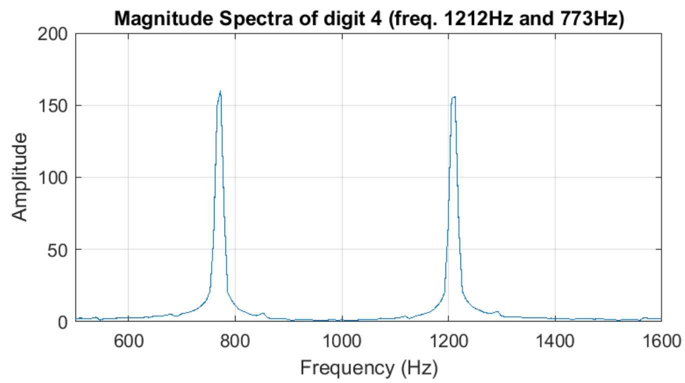

Figure 25 Magnitude spectra after Hanning windowing, digit 4

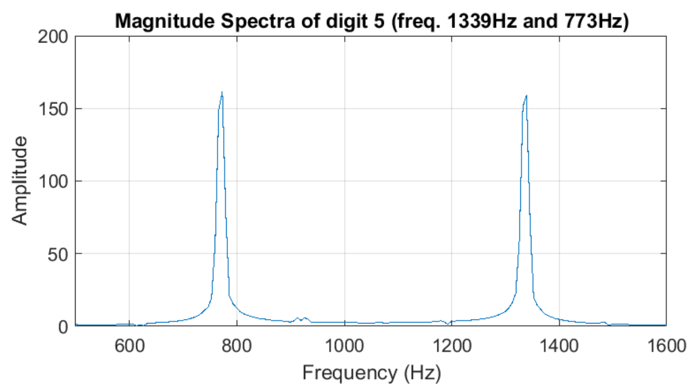

Figure 26 Magnitude spectra after Hanning windowing, digit 5

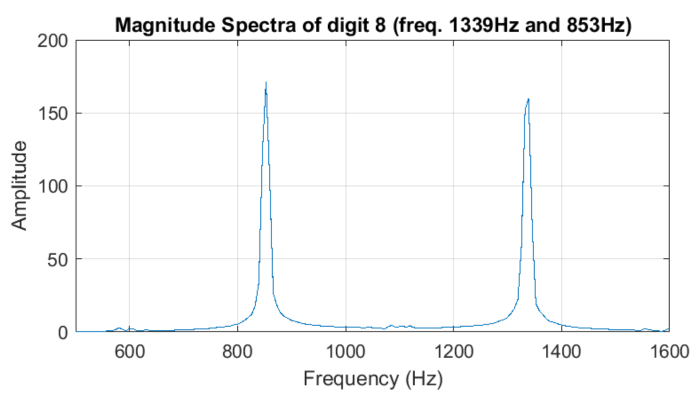

Figure 27 Magnitude spectra after Hanning windowing, digit 8