Laboratory 3A:

**Fourier Transformation and FFT**

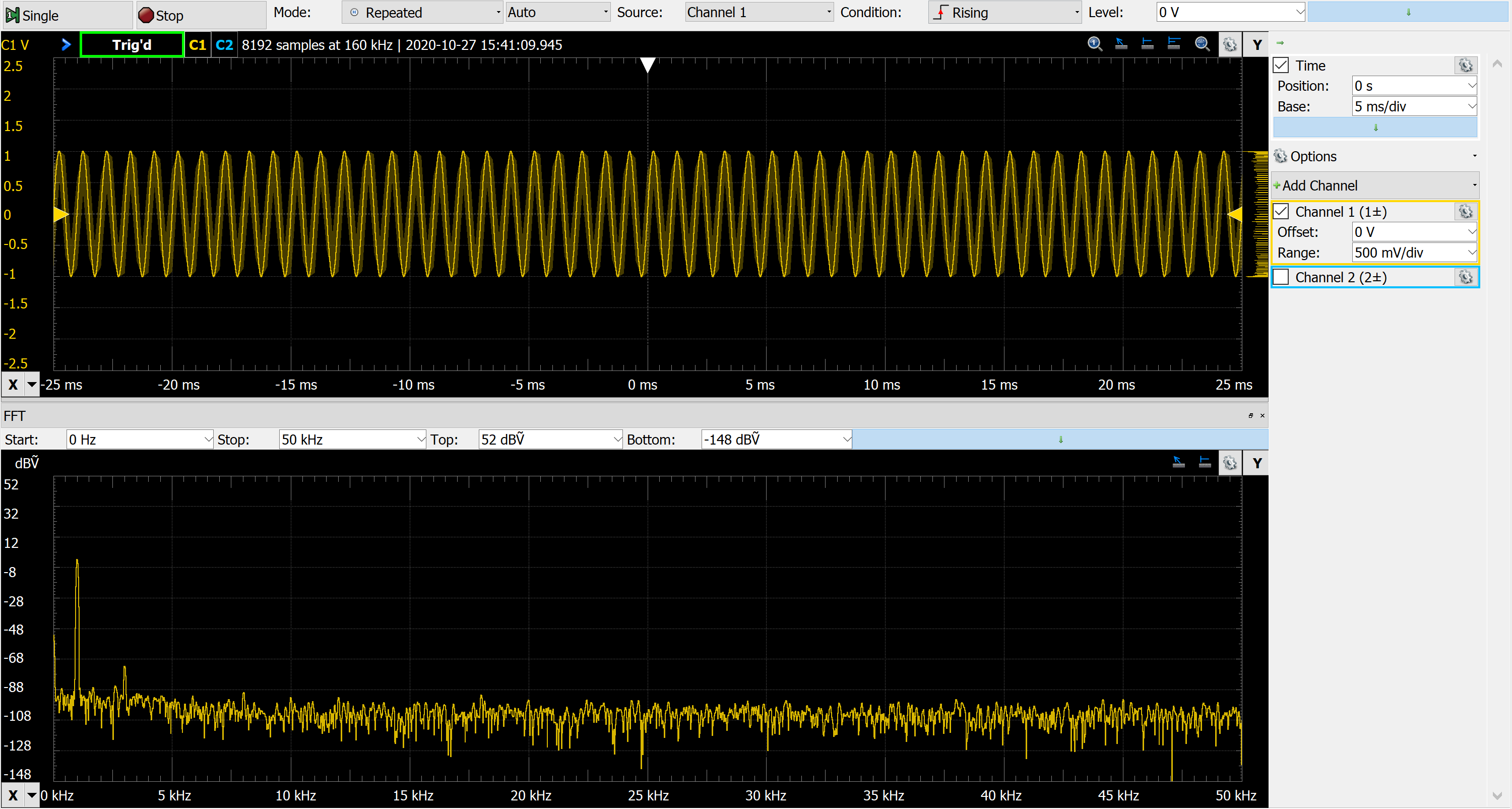
Sample Solution

**Exercise 1: Working with the USB-Scope**

1. Resolution in frequency domain: fstep = 800k / 8192 ~ 100Hz (precisely 97.65… Hz)
2. Calculated resolution above. Can change time-base to check. Please do so.
3. Expect a single harmonic at 1KHz (or as close as resolution allows it).
4. With large frequency range (0-400KHz) seeing 2 peaks around 250kHz. Believe these are image spectra, around a sampling frequency of 250kHz. We will speak of image spectra in chapter 4.

If you change time-base for a larger observation window and smaller freq. range, you address the following question.

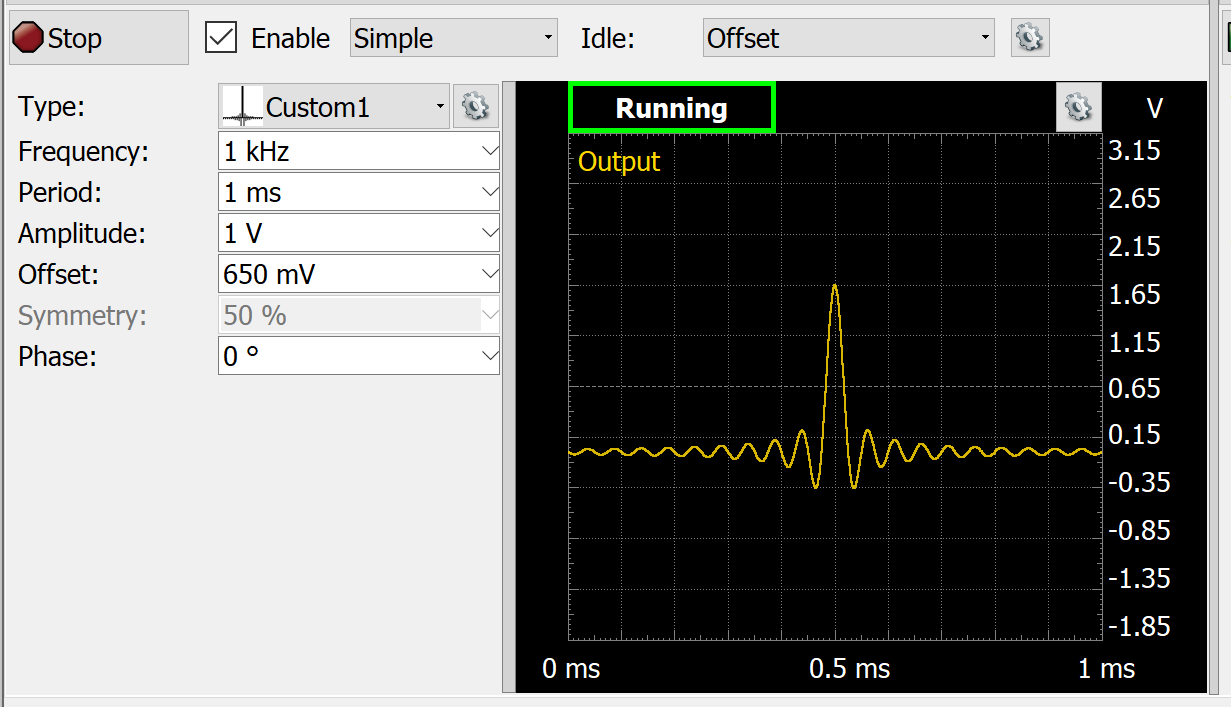




Here we observe some power at 3kHz, which corresponds to the 3rd harmonic, about 60dB below the 1st harmonic level. This happens often when for example, an amplifier presents some non-linearity (e.g. smaller gain closer to rail values) and generates some extra harmonics.

**Exercise 2: Duality Property of the Fourier Transformation**

1. OK.
2. Pay attention that the x range (from/to) will determine the slice of the time function, which will be periodically repeated by the generator.
3. Thinking about the duality property, expect a somehow square shape in the frequency domain (amplitude spectrum).
4. Change the offset in the waveform generator (see snapshot below), to reduce DC-value in the spectrum, and better visualize the square shaped spectrum.



Time Domain Frequency Domain

==============================================================

Snapshot-E : periodic => discrete

T0 = 1ms => f0 = 1kHz

==============================================================

Snapshot-F : periodic “sinc-shape” => discrete “square-shape”

T0 = 1ms f0 = 1kHz

The “” means that the shapes are not perfect.

For example, the sinc is limited (clipped), and

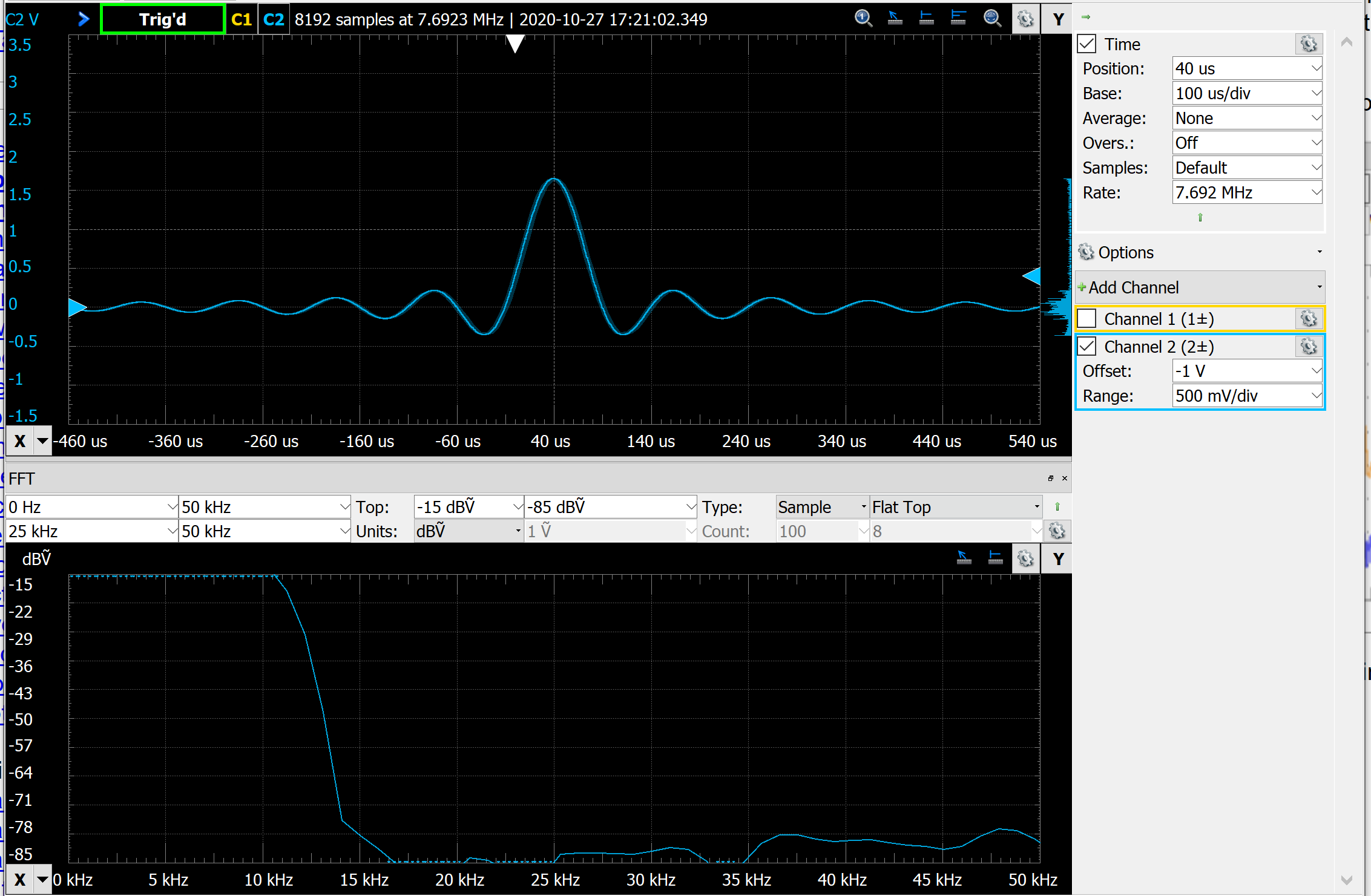
the square gets correspondingly less sharp edges.

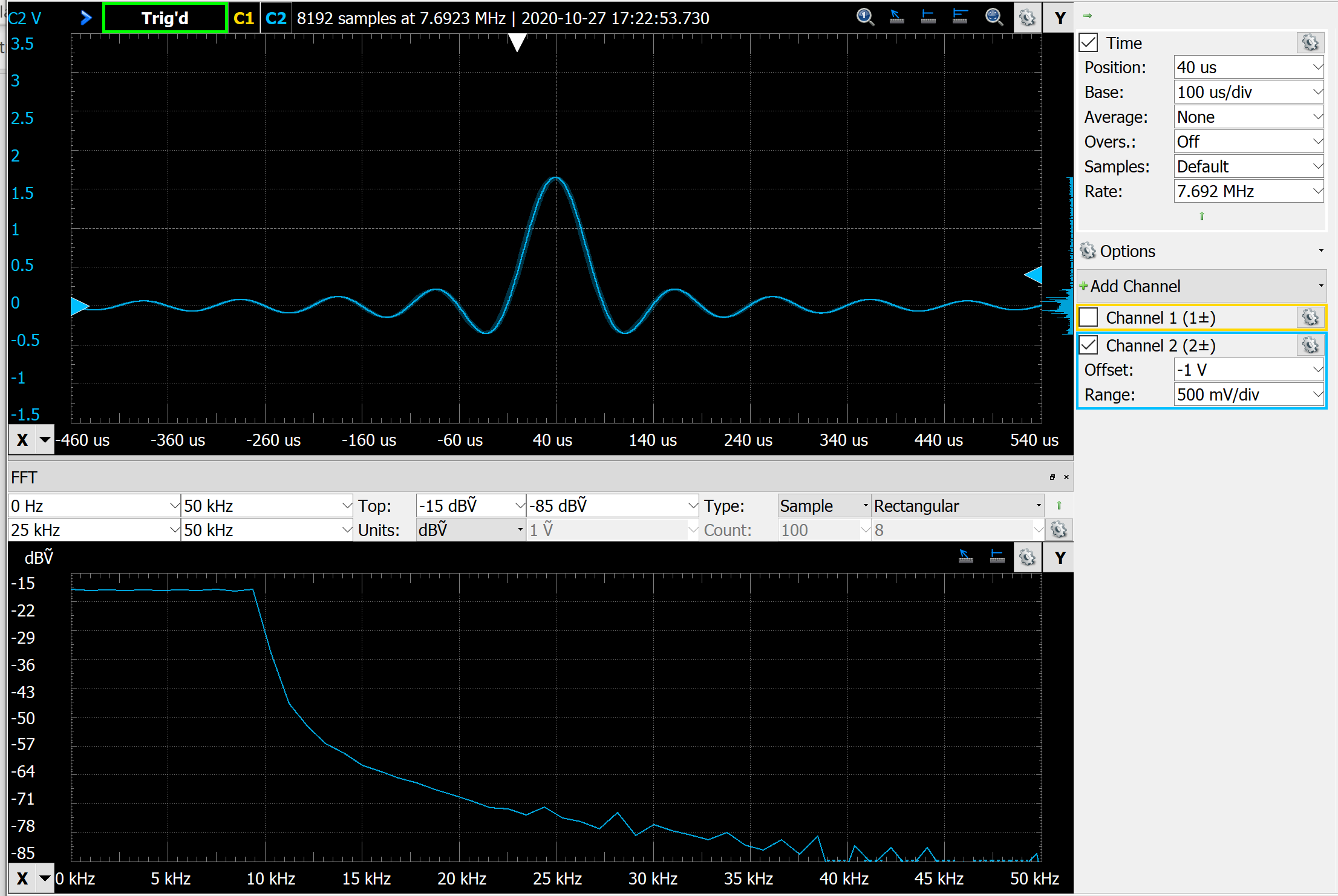
==============================================================

Snapshot-G : lobe-width=50us => bandwidth = 1/50us=20kHz = [-10k, +10kHz]

In the spectrum measurement we see the positive frequencies, therefore 0-10kHz.

Obs.: with the current selected window for FFT, we see a bit wider BW. Change the window of FFT to see its influence.





**Exercise 3 : Time-Bandwidth Product**

Several possibilities. Liked a lot this one below done by students:

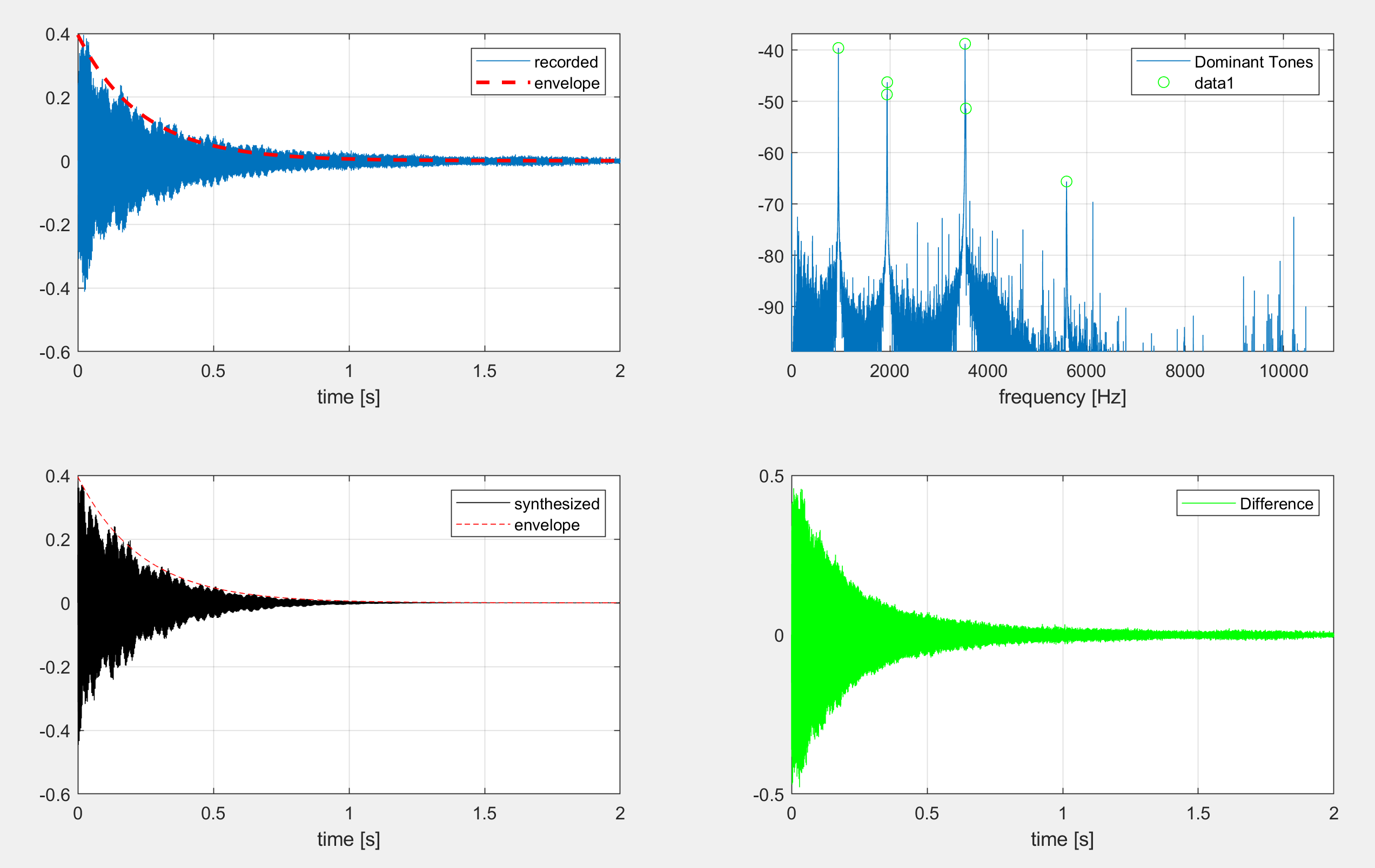
Computergenerierter Alternativtext:
WaveForms (Sine_simple) 
Workspace Settings Window Help 
Welcome 
Help 
File Control Edit Window 
Scope 1 
e Wavegen 1 
estop All 
Channel 1 (Wl) 
Run 
Channels 
[Z] Enable Custom 
No synchronization 
Idle: 
Stop 
Outpu 
0 ms 
Offset 
0.2 ms 
Channel 2 (W2) 
e Stop [Z] Enable Custom 
Edit 
New Import 
Frequency: 
Idle: 
Running 
Offset 
Edit 
New Import 
Frequency: 
cu... 
cu... 
1 kHz 
Sample Rate: 
4.096 MHz 
Amplitude: 
Offset: 
ov 
Phase: 
0.4 ms 
0.6 ms 
0.8 ms 
2.5 
1.5 
0.5 
-0.5 
-1.5 
-2.5 
1 ms 
cu... 
cu... 
1 kHz 
Sample Rate: 
4.096 MHz 
Amplitude: 
Offset: 
ov 
Phase: 
Outpu 
0 ms 
0.2 ms 
0.4 ms 
0.6 ms 
0.8 ms 
Manual Trigger Discovery2 SN:210321A29679 USB 
x 
2.5 
1.5 
0.5 
-0.5 
-1.5 
-2.5 
1 ms 
Status: OK 
Suchbegriff hier eingeben 
19:31 
A 4)) DEU 
28.10.2020 

Computergenerierter Alternativtext:
WaveForms (Sine_simple) 
Workspace Settings Window Help 
Welcome 
Help 
File Control View Window 
Export +XY +XYZ 3D +Zoom 
x 
Scope 1 
FFT Spectrogram 
Mode: 
Wavegen 1 
Spectrogram 3D Histogram Persistence 
Data Measurements 
Channel 1 
Source: 
Logging 
1.4 ms 
Audio X Cursors 
Condition: 
2.4 ms 
Y Cursors Notes 
Rising 
3.4 ms 
Digital 
Measurements 
Single 
2.5 
1.5 
0.5 
-0.5 
-1.5 
-2.5 
-4.6 ms 
Run 
Stop 
-3.6 ms 
10.4 kHz 
Repeated 
O 
Auto 
-0.6 ms 
Level : 
4.4 ms 
ov 
C2 8192 samples at 800 kHz | 2020-10-28 
illil 
Z] Time 
Position: 
Base: 
Average: 
Overs. : 
Samples: 
Rate: 
Options 
400 us 
1 ms/div 
None 
Off 
Default 
800 kHz 
+ Add Channel 
Z] Channel 1 (1±) 
Offset: 
Range: 
ov 
500 mV/div 
v Channel 2 (2±) 
Stop: 
Span: 
-2.6 ms 
95 kHz 
94 kHz 
19.8 kHz 
-1.6 ms 
Top: 
Units: 
29.2 kHz 
0.4 ms 
Bottom: 
Reference : 
48 kHz 
5.4 ms 
95 kHz 
Offset: 
Range: 
ov 
500 mV/div 
Start: 
Center: 
dBV 
52 
32 
12 
-28 
-48 
-68 
-88 
-108 
-128 
-148 
1 kHz 
48 kHz 
1 kHz 
52 dBV 
dBV 
38.6 kHz 
-148 dBV 
IV 
57.4 kHz 
Type: 
Count: 
Sample 
100 
Window: 
Beta: 
66.8 kHz 
76.2 kHz 
Blackman-Harris 
8 
85.6 kHz 
Manual Trigger Discovery2 SN:210321A29679 USB 
DEU 
Status: OK 
19:32 
Suchbegriff hier eingeben 
28.10.2020 

**Exercise 3** *Tchin-Tchin: Synthesis of a Glass-Sound.*

Observation:

* Major tones and relative amplitudes are the main characteristics perceived by human hearing. Relative phase difference of major tones not relevant, and this explains why point to point difference between synthesized and original sound can be so big, although both sounds perceived as similar.
* Interesting: adding some low-pass noise, makes synthesized sound quite difficult to distinguish from original one.



% SiSy1 Prak3A - Aufgabe Klang Synthese

% =================================

clear all, close all, clc;

% \_\_1\_\_Load and hear to reference sound

load Glas

sound(y,Fs)

% \_\_2\_\_Check loaded variables and declare aux / time / freq vectors

% Fs [Hz]

% y [vector with amplitude values over time]

Ts = 1/Fs; % seconds

N = length(y); % number of points in registered sample

aux = 0:1:N-1; % index vector

t = Ts\*aux; % time vector

f = (Fs/N)\*aux; % freq vector

% \_\_3\_\_Plot registered sound in time

figure(1)

subplot(221), plot(t,y), grid on, hold on;

xlabel('time [s]')

% Model envelope as decaying exponential

max(y)\*0.37

tau = 0.235; % check where curve decayed to 0.37\*initial value (approx exp(-t/tau) )

x\_env = max(y)\*exp(-t/tau);

figure(1)

subplot(221), plot(t,x\_env,'r--','LineWidth',2), hold off

legend({'recorded','envelope'})

% \_\_4\_\_Analyse registered sound in frequency

Y = (1/N)\*fft(y); % fast-fourier-transform

Y\_dB = 20\*log10(abs(Y)); % amplitude in dB

% \_\_5\_\_Plot registered sound in frequency

figure(1)

subplot(222), plot(f,Y\_dB), grid on, hold on;

pause(2)

axis([0 Fs/2 max(Y\_dB)-60 max(Y\_dB)+2])

xlabel('frequency [Hz]')

legend('Dominant Tones')

%%

% Synthesize sound with: (sum of sinus) \* (envelope curve)

% Look for first 6 dominant peaks, within max-40dB

[pksdB,locs] = findpeaks(Y\_dB(1:N/2),'NPeaks',6,'MinPeakHeight',max(Y\_dB)-30,'MinPeakDistance',10,'MinPeakProminence',2);

figure(1)

subplot(222), plot(f(locs),pksdB,'go'), hold off;

f(locs) % check frequency value for peaks selected

AmpdBr = pksdB-pksdB(1) ; % take amplitude values relative to 1st peak

AmpLir=10.^(AmpdBr/20); % calculate amplitude in linear scale

% Sum up sinus-signals (each with respective frequency & amplitude)

x=zeros(1,N);

for id=1:1:length(locs)

x = x + AmpLir(id)\*sin(2\*pi\*f(locs(id))\*t);

end

x = x/max(x) ; % normalize for max(amplitude)=1

% Shape the sum of sinus with envelope curve

x\_synth = 1.15\*(x\_env.\*x);

% tuned amp (with 1.15) to better match volume-impression of recording

figure(1)

subplot(223), plot(t,x\_synth,'k', t,x\_env,'r--'), grid on

xlabel('time [s]')

legend({'synthesized','envelope'})

% Play Synthesised Sound and Replay original for comparison

sound(x\_synth,Fs);

pause(2)

figure(1)

subplot(224), plot(t, (y-x\_synth),'g'), grid on

xlabel('time [s]')

legend('Difference')

pause(2), figure(2)

plot(t,y,'b',t,x\_synth,'k',t,(y-x\_synth),'g--'),grid on

xlabel('time [s]'), legend({'recorded','synthesized','difference'})

pause(2), axis([0 0.01 -0.4 0.4])

%% Add band-limited noise (background)

x\_spr = x\_synth + 0.02\*lowpass(rand(1,N)-0.5,0.4);

sound(x\_spr,Fs);

pause(4)

sound(y,Fs);

pause(2), figure(2)

plot(t,y,'b',t,x\_synth,'k',t,(y-x\_synth),'g--',t,x\_spr,'m-.'),grid on

xlabel('time [s]'), legend({'recorded','synthesized','difference','synth+noise'})

pause(2), axis([0 0.01 -0.4 0.4])