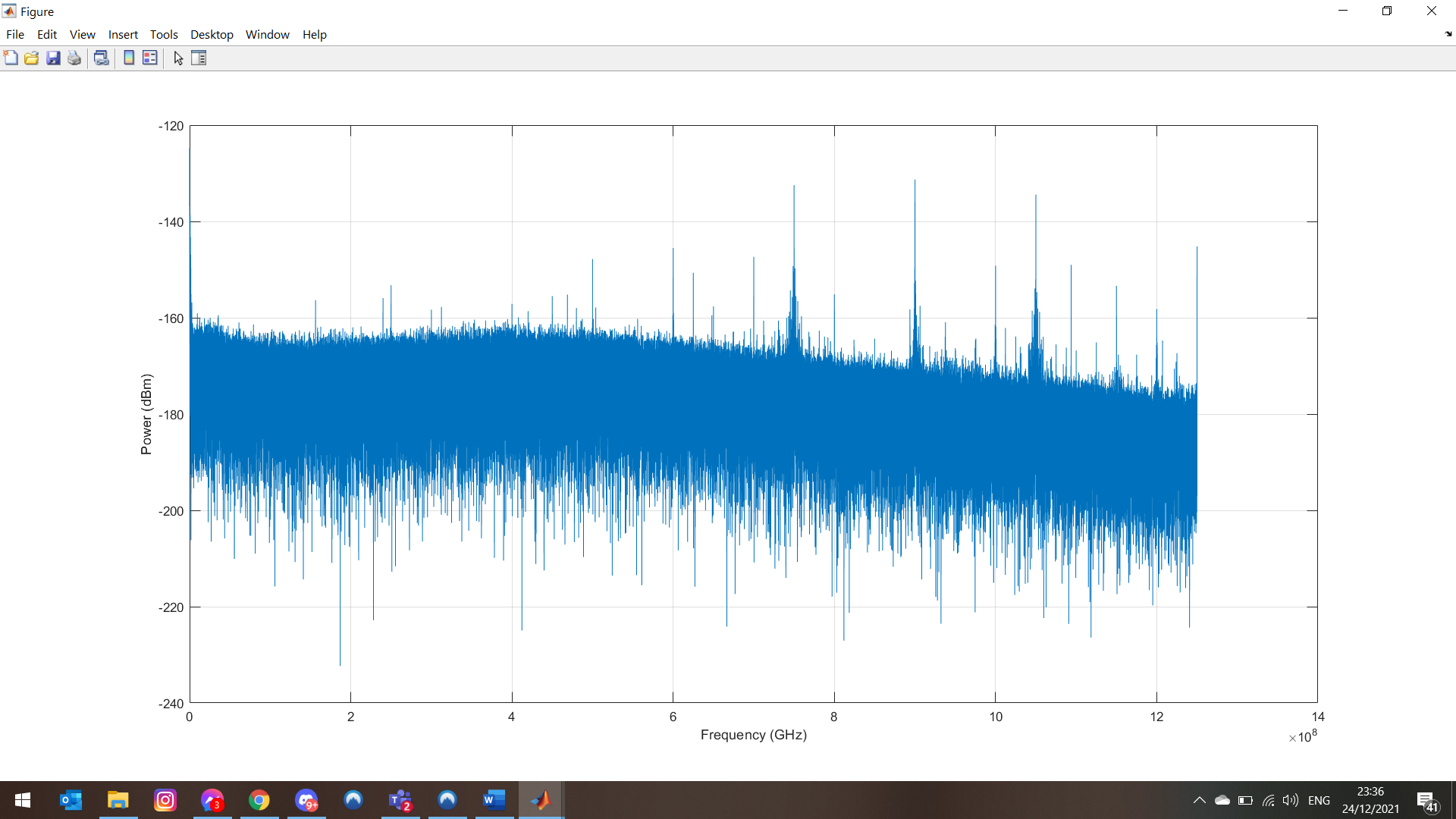
Lab report

For task one we used the following code:

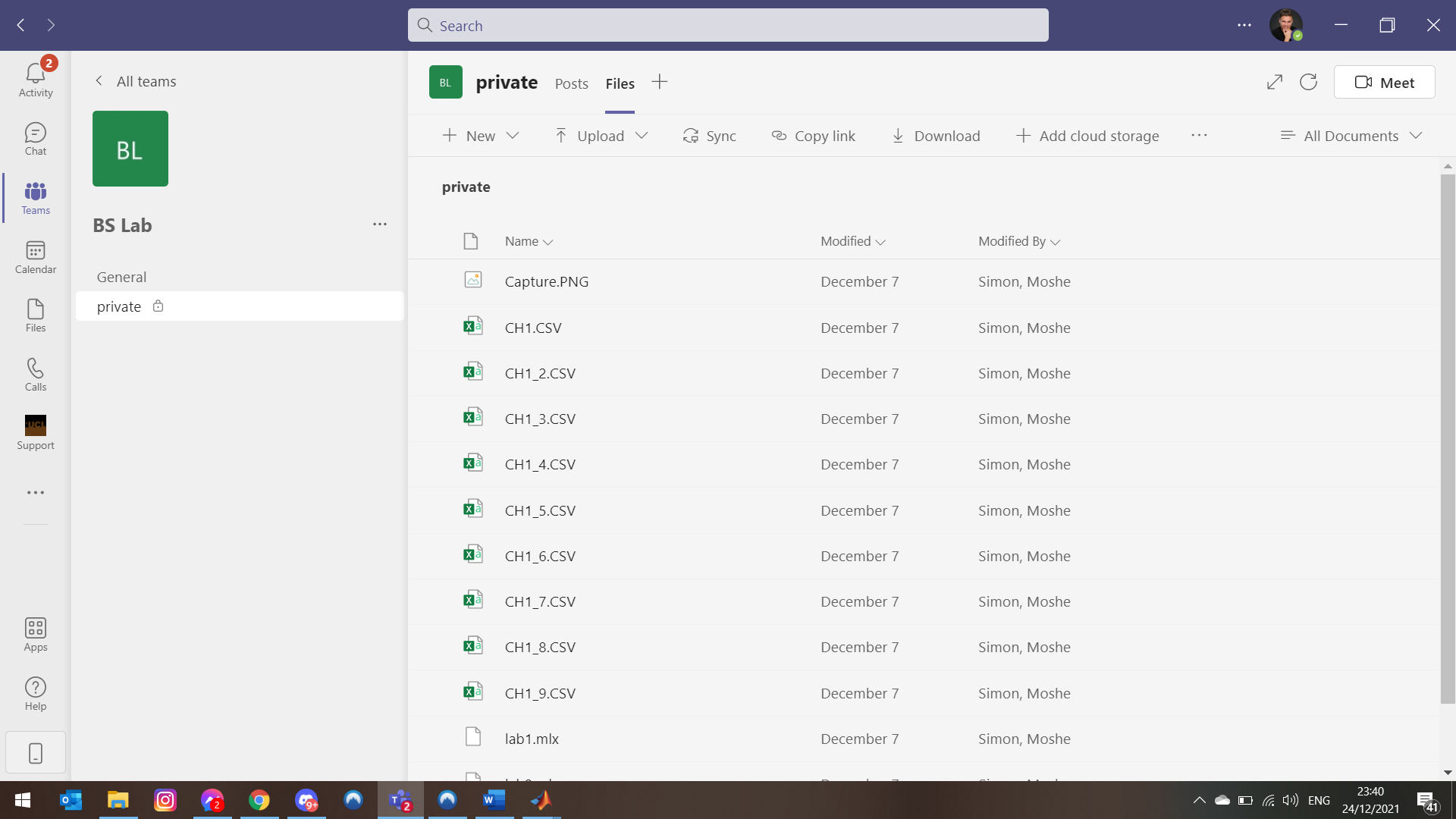
|  |
| --- |
| clc; clear; close all;format long;  % data path - change this to your own data path (suggest using your own onedrive)  %datapath = 'testdata\SNR\';  % read data and calculate  waveform\_filename = 'CH1.CSV';  %waveform\_filename = strcat(datapath,waveform\_filename);  data = csvread(waveform\_filename,1,0);  T = data(:,1); % time [s]  Waveform = data(:,2) - mean(data(:,2)); % amplitude [V]  % get parameters (fill in the following lines)  fs = 2.5\*10^9; % sampling rate  dt = 1/fs; % sampling period  N = length(data); % number of samples  df = fs/N; % bandwidth of each frequency bin  %% Calcualte and plot PSD  % frequencies correspond to the samples in frequency domain  %f = [0:N/2-1 -N/2:-1].' \* (fs / N);  f = 0:fs/N:fs/2;  % calculate the Fourier transform of the complex envelope  Ampf = fft(Waveform);  % Then calculate the single-side band PSD (W/Hz)  xdft = Ampf(1:N/2+1);  psdx = (1/(fs\*N)) \* abs(xdft).^2;  psdx(2:end-1) = 2\*psdx(2:end-1);  % ---- plot single-side band PSD in dBm ----  psd\_dBm = 10\*log10(psdx)  plot(f(2:end),psd\_dBm(2:end));  grid on; xlabel('Frequency (GHz)'); ylabel('Power (dBm)');  %axis([0 f(N/2)/1e6 -200 -80]);  % --- total noise power ---  total\_noise\_power = 10\*log10(sum(psdx\*df)/1e-3);  fprintf('The noise power over the Nyquist bandwidth is %.2f dBm .\n',total\_noise\_power); |

We did not connect the oscilloscope to anything as we were interested in the noise power. We imported the oscilloscope reading data from the computer on the right hand side as the oscilloscopes are only connected to the PCs on the right hand side.

Using the data obtained from the oscilloscope we calculated the noise power over the Nyquist bandwidth to be -49.27 dBm. We also plotted the readings and received the following graph:



For the second task we connected the oscilloscope to the function generator. We generated a signal with different frequencies and saved them into separate excel files as can be seen below:



We tried to solve task 2 with the following code

|  |
| --- |
| clc; clear; close all;format long;  % data path - change this to your own data path (suggest use your own onedrive)  %datapath = 'testdata\SNR\';  %% SNR characterisation (use your own script, NOT the MATLAB build-in functions)  bandwidth = 100e3; % signal tone bandwidth [Hz]  Vpp = 2:9; % [mV]  SNR = zeros(size(Vpp));  for i = 1:numel(Vpp)  % load waveform  %waveform\_filename = 'CH1task2.CSV';  waveform\_filename = sprintf('CH1\_%d.CSV', Vpp(i));  %waveform\_filename = strcat(datapath,waveform\_filename);  data = csvread(waveform\_filename,1,0);  % get waveform info  T = data(:,1); % time [s]  Waveform = data(:,2) - mean(data(:,2)); % amplitude with AC blocked [V]  fs = 2.5\*10^9; % sampling rate [Hz]  dt = 1/fs; % sampling period [s]  N = length(data); % number of samples  df = fs/N; % bandwidth of each frequency bin  f = 0:fs/N:fs/2;  % calculate the Fourier transform of the complex envelope  Ampf = fft(Waveform);  % Then calculate the single-side band PSD (W/Hz)  xdft = Ampf(1:N/2+1);  psd = (1/(fs\*N)) \* abs(xdft).^2;  psd(2:end-1) = 2\*psd(2:end-1);  % find the signal frequency  [peakPSD, index] = max(psd);  % signal centre frequency frequency  f\_sig = (index-1) \* df;  % signal psd  psd\_sig = psd(f>f\_sig-bandwidth/2 & f<f\_sig+bandwidth/2);  psd\_sig\_dBm = 10\*log10(psd\_sig);  % noise psd  psd\_noise = psd(f<f\_sig-bandwidth/2 | f>f\_sig+bandwidth/2);  psd\_noise\_dBm = 10\*log10(psd\_noise);  %f\_sig = zeros(3);  f\_noise = zeros(length(f)-3);  sig=1;  noise = 1;    for j = 1:length(f)  if ((f(j)>(f\_sig-bandwidth/2)) && (f(j)<(f\_sig+bandwidth/2))  f\_sig(sig) = f(i);  sig =sig+1;  end  if ((f(j)<(f\_sig-bandwidth/2)) || (f(j)>(f\_sig+bandwidth/2)))  f\_noise(noise) = f(j);  noise =noise+1;  end  end  % show signal and noise separately with 'dBm' as y-axis  figure;  psd\_dBm = 10\*log10(psd);  total\_power = 10\*log10(sum(psd\*df)/1e-3);  total\_noise\_power = 10\*log10(sum(psd\_noise\*df)/1e-3);  ratio = total\_power/total\_noise\_power  hold on  plot(f(2:end),psd\_sig\_dBm(2:end))  plot(f(3:end),psd\_noise\_dBm(2:end))  % plot signal PSD  % plot noise PSD  xlabel('Frequency (MHz)'); ylabel('PSD (dBm)');  drawnow  % SNR calculation  sig\_power = sum(psd\_sig)\*df; % signal power  noise\_power = sum(psd\_noise)\*df; % noise power  SNR(i) = 10\*log10(sum(sig\_power)/sum(noise\_power)); % [dB]  end  % --- plot SNR vs Vpp ---  figure; plot(Vpp,SNR,'-s');  xlabel('peak-to-peak voltage (mV)');  ylabel('SNR (dB)'); |

Unfortunately, we did not succeed to obtain a plot for the plot and the lab session ended before we could resolve our issue.