**Homework 2**



**Louis Lin**

**January 17, 2022**

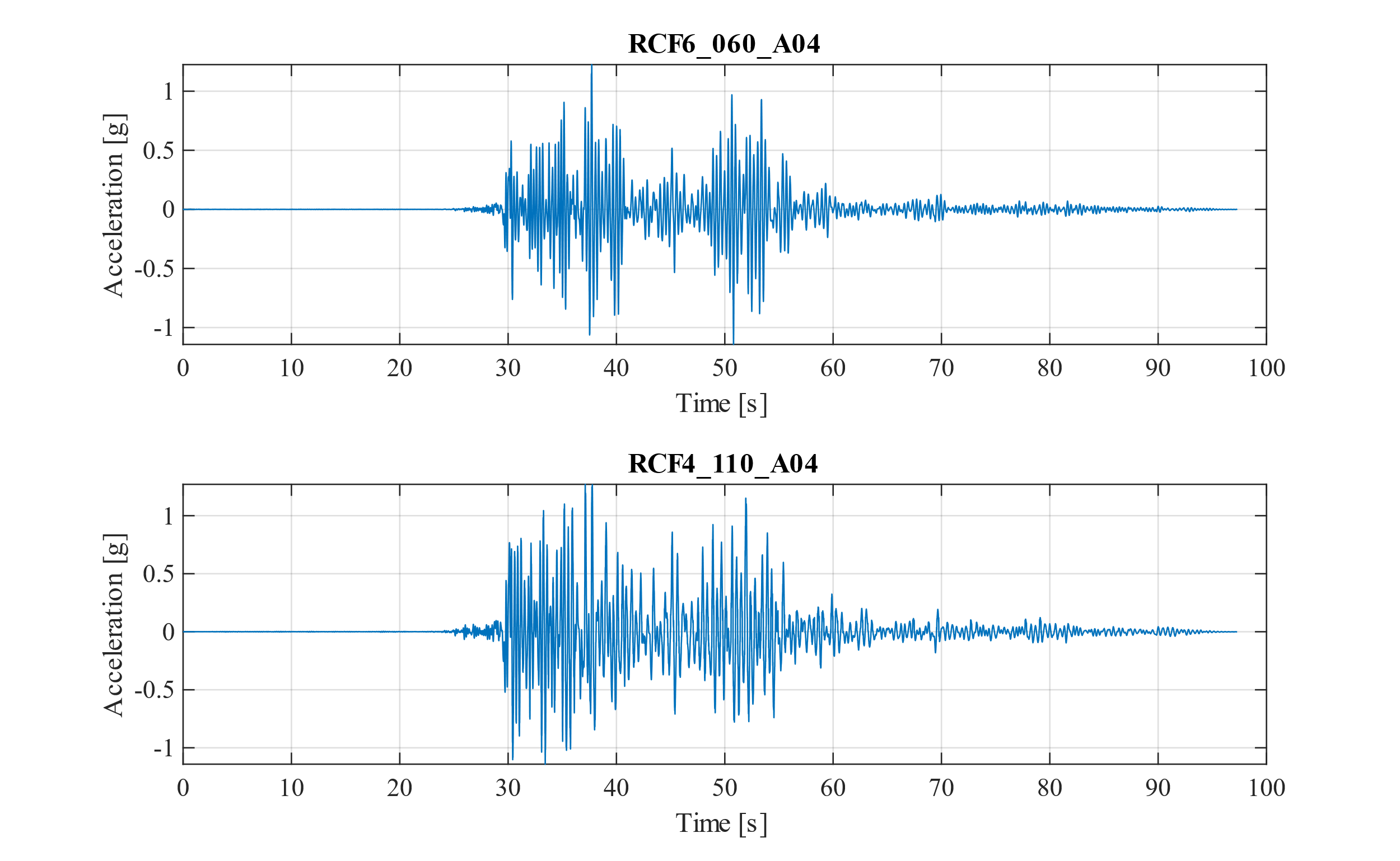
**University of California, San Diego**

**Prof. Chin-Hsiung Loh**

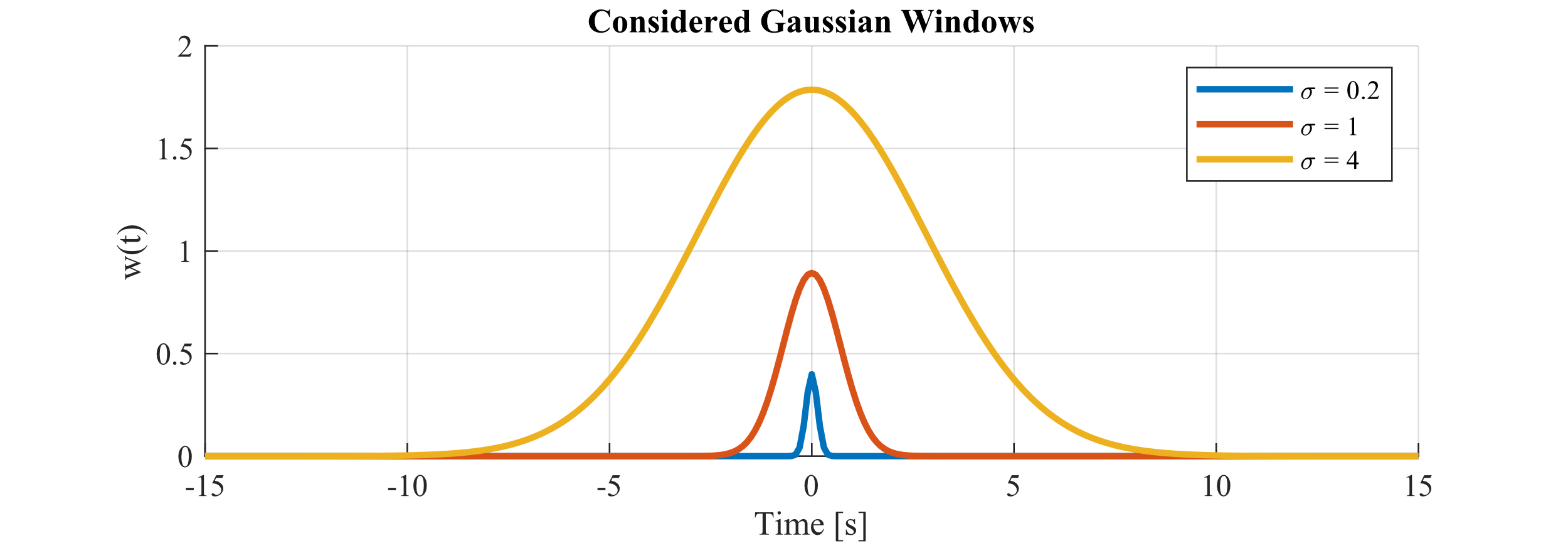
**SE 267A Signal Processing and Spectral Analysis**

**Problem 1**

The data comes from seismic testing of a reinforced concrete frame for 4 different configurations. Input and output data were provided at a sampling rate of 200Hz. The physical spectrums of the output data for the 4th and 6th floors of the test are given for various Gaussian windowing functions of the spectrum.

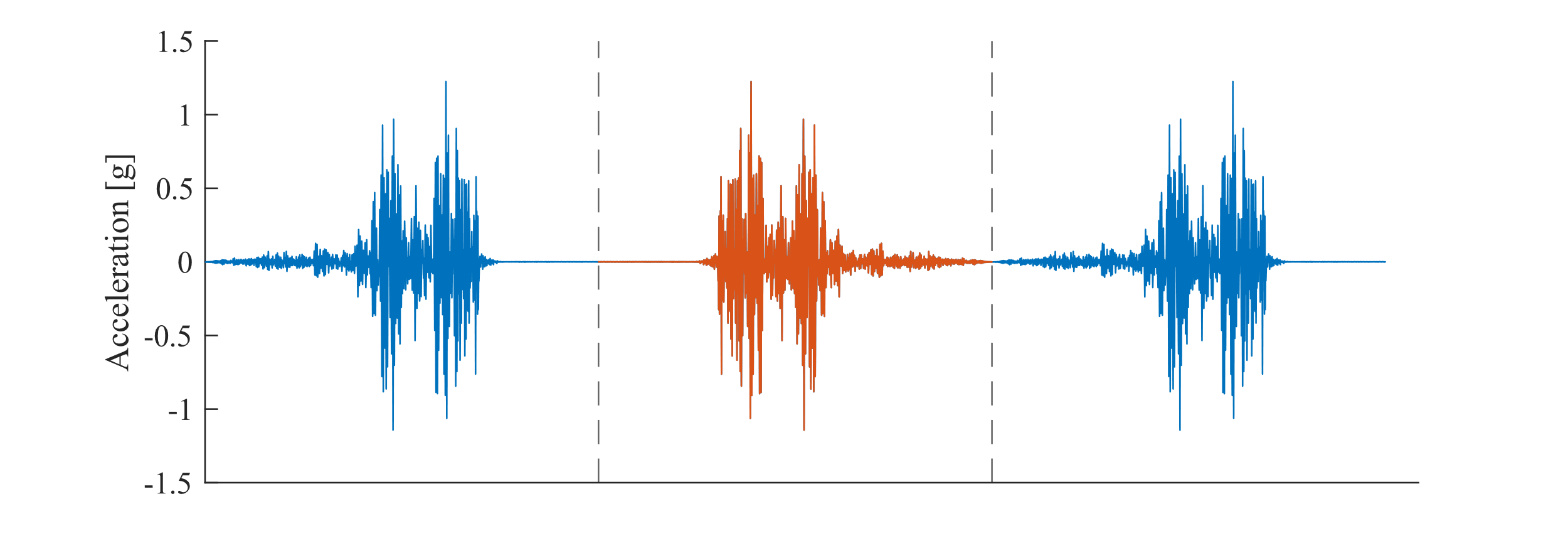


Windowing, or filtering the data, in the time domain gives a smoother spectrum when transforming the signal into the frequency domain. This was explored in this homework by using a Gaussian window with various variances, . The different windows are shown below and is formulated as



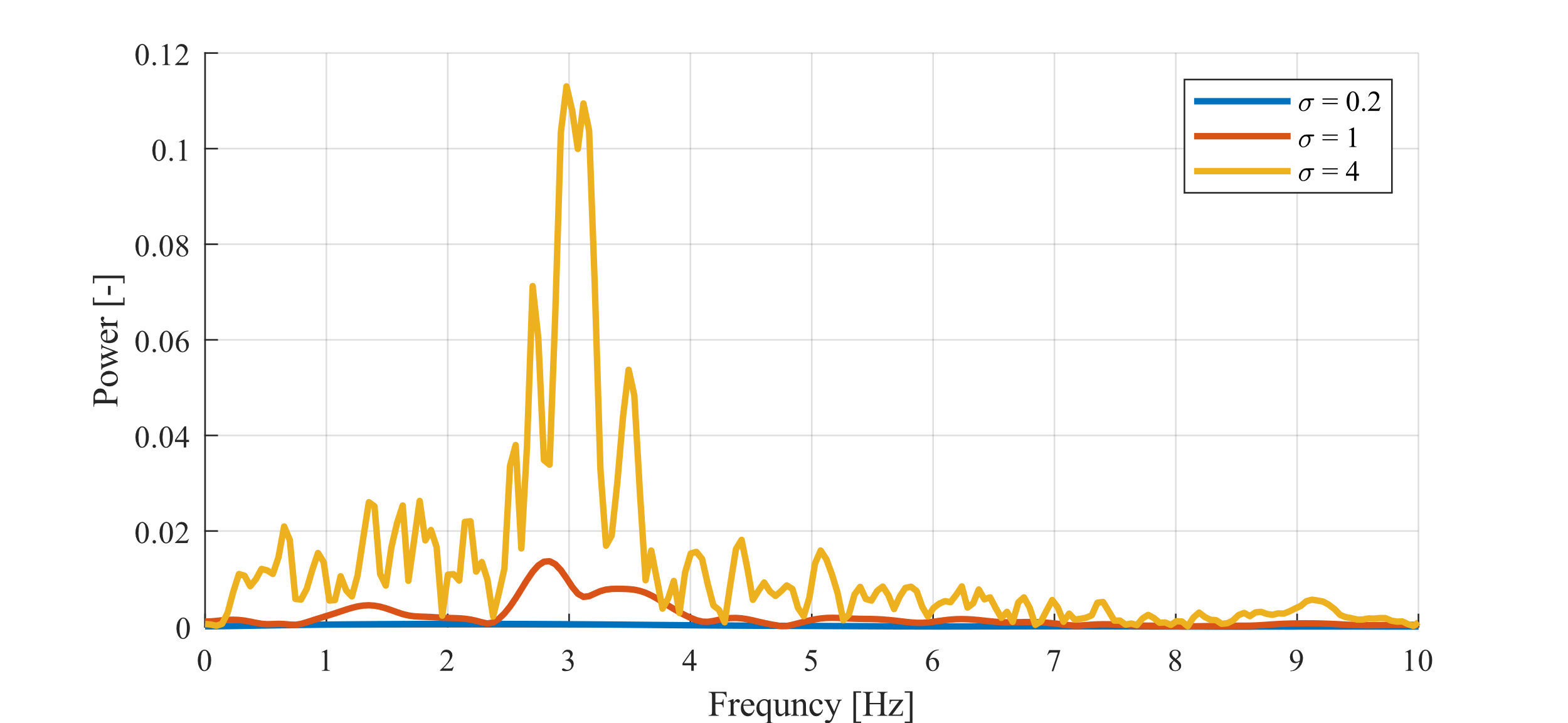
The time window considered was [-10,10] considering the case had wider spread. The spctrogram can be seen on the next page.

In order to mitigate edge effects and preserve the characteristics of the signal, mirrored signals were introduced at the beginning and ends of the signal. Thus, the window could be applied for the entire length of the signal.



The spectrogram represents the fast Fourier transform for the entire length of the signal, with a slice being taken for every time stamp in the signal.

For a larger variance, there are more defined peaks for the spectrogram. This is because more of the signal is being averaged and smoothed by the frequency hence the peaks are more prominent. For the case where was smaller, the spread in the spectrogram is wider and shallower, as shown by the color bar. This is because the gaussian window is smaller and thus the frequency content of the original signal remains largely unchanged which is a larger spread. The smaller variance of the Gaussian window does not capture enough data in its window.



A picture containing graphical user interface

Description automatically generated

A picture containing timeline

Description automatically generated

|  |
| --- |
| %% Load Data  clear; clc;  data = readtable("Homework-2 data set-RCF-Four Specimen Test Data.xlsx");  %% Plot Input  figure(1); clf; clc;  nexttile  X = data.Time;  Y = data.RCF6\_060\_A04;  plot(X, Y )  xlabel("Time [s]")  ylabel("Acceleration [g]")  grid  title("RCF6\_060\_A04",Interpreter="none")  nexttile  X = data.Time;  Y = data.RCF4\_110\_A04;  plot(X, Y )  xlabel("Time [s]")  ylabel("Acceleration [g]")  grid  title("RCF4\_110\_A04",Interpreter="none")  print\_figure(1, '.\', "Signals",4)  %% Plot the windowing function  figure(2); clf; clc;  hold on;  t = -15:0.1:15;  for sigma = [0.2, 1.0 4.0]  wFunction = sqrt(2/sqrt(2\*pi) \* sigma) .\* exp(-t.^2 /sigma^2);  plot(t, wFunction,"DisplayName","\sigma = "+sigma,LineWidth=2);  end  title("Considered Gaussian Windows")  xlabel("Time [s]")  % ylabel("$$(\frac{2}{\sqrt{(2\*\pi}\sigma}^{0.5})\cdot e^{\frac{-t.^2} {\sigma^2}}$$", Interpreter="latex")  ylabel("w(t)")  legend()  grid on  print\_figure(2, '.\', "Windows",2.25)  %% Perform windowning for various sigma into the windowing function  figure(3); clf; clc;  tiledlayout(3,1,"TileSpacing","compact","Padding","compact")  % Get inputs  name = "RCF6\_060\_A04";  % name = "RCF4\_110\_A04";  signal = data.(name);  npt = length(signal);  fs = 200;  dt = 1/fs;  % Prepare weighing function  t = -10:dt:10; % Time domain windowing length  for sigma = [0.2, 1.0, 4.0] ; % Variance  wFunction = sqrt(2/sqrt(2\*pi) \* sigma) .\* exp(-t.^2 /sigma^2); % weighing function  wNpt = length(wFunction); % length of weighing function  % Calculate the Spectrogram  physical\_spectrum = zeros(wNpt, npt); % Spectrogram  mirroredSignal = [flip(signal); signal; flip(signal)]; % Accounts for the boundary  for ii = npt+1: 2\*npt -1  weightedF = wFunction' .\* mirroredSignal(ii: ii + wNpt-1);  physical\_spectrum(:,ii - npt) = abs(fft(weightedF))./wNpt\*2;  end  % Plotting the Results  nexttile()  X = data.Time; % Plot for the entire length of the signal  fps = fs\*((1:wNpt)'-1)/wNpt; % Frequency Spectrum Domain (Y-direction)  freqBand = [0, 10]; % Frequency band of interest  idx = (fps >= freqBand(1) & fps <= freqBand(2)); % Plot only relevant portion of spectrogram  Y = fps(idx); % Gets the proper frequencies for plot (SAVES TIME)  mag = physical\_spectrum(idx,:); % Gets FFT Magnitude  surf(X,Y,mag,'LineStyle','none','FaceLighting','phong');  view(0,90),colormap('jet'),ylabel('Frequency (Hz)');  xlim([0,90]);ylim(freqBand);title("\sigma = "+sigma);  colorbar;set(gcf,'Position',[50,50,1024,768])  xticks([])  end  xticks()  xlabel('Time (sec)')  sgtitle(["Synthetic Time Series Physical Spectrum",name],fontname="times",Interpreter="none")  % Print Picture  h = findobj('Type','axes');  set(h, 'FontName', 'Times');  set(gcf, 'Position', [8 5 6.5 4.25]\*100,"Color",'w' );  print(name,'-djpeg')  %%  figure(4); clf; clc;  hold on;  plot(mirroredSignal);  xticks([]);  xline(npt,'--')  xline(2\*npt,'--')  plot(npt+1:2\*npt, signal)  ylabel("Acceleration [g]");  print\_figure(4, '.\', "Mirrored Signal",2.25)  %%  figure(5); clf; clc;  hold on;  % Get inputs  name = "RCF6\_060\_A04";  signal = data.(name);  npt = length(signal);  fs = 200;  dt = 1/fs;  % Prepare weighing function  wNpt = 4096; % I want this many points in the weighin function  t = -(wNpt/fs/2) - 1:dt: (wNpt/fs/2); % Time domain windowing length  for sigma = [0.2, 1.0, 4.0] % Variance  wFunction = sqrt(2/sqrt(2\*pi) \* sigma) .\* exp(-t.^2 /sigma^2); % weighing function  wNpt = length(wFunction); % length of weighing function    % Calculate the Spectrogram  ii = 6000; % Somewhere in the middle of the signal w/o edge issues  weightedF = wFunction' .\*signal(ii: ii + wNpt-1);  x = weightedF;  N = wNpt;  y = fft(x,N);  mag = abs(y)\* 2/N;  f = (0:N-1)\*fs/N;  id = (1:N/2); % First half  % mag = 20\*log10(mag);  plot(f(id), mag(id),"linewidth",2,DisplayName="\sigma = "+ sigma);  end  legend()  grid  xlim([0,10]);  xlabel("Frequncy [Hz]");  ylabel("Power [-]");  print\_figure(5, '.\', "Weighed Spectrum",3) |