Mathematical Fundamentals for Electrochemical Energy Storage Systems

Exercise 8

Task 1: Fourier Transform

In this exercise a simple example of a real life application of Fourier transformation will be shown. We will create a signal, add noise to it, and use Fourier transformation to identify and remove the noise.

a) Create a sine wave of frequency 2 Hz and for a duration of 5 seconds. Use matplotlib to plot this sine wave. Assume that the signal is sampled at a frequency of 44100 Hz.
b) b. Create a sine wave of frequency 400 Hz. Assume that the sampling rate and the duration are same. Save this data as a "nice tone". Now create a "noise tone" of frequency 4000 Hz and multiply this by 0.3 to reduce its power. Finally, add the two signals together to create a "mixed tone" and plot it.
c) Now we will save the data as an audio file. To do so, we need to first normalize it by dividing the mixed tone by its maximum value and multiplying it by 32767. Convert this number to int16 data type.
d) Save this data as a wave file using the write function that is available in scipy.io.wavefile Now it is time to use fourier transformation on the generated audio. The best way to implement this in python is using the fast fourier transform algorithm, which is a method to apply fourier transform on discrete signals. The fft and fftfreq functions from the scipy.fft library are useful here.
e) Apply the two functions above to calculate the frequency components of the normalized tone. Plot the frequency components to identify the nice tone and the noise tone.
f) Remove the noise tone in the frequency spectrum by setting the power of the noise component to zero
g) Use the ifft function to convert the signal back to time domain. Plot the filtered sine wave and hear the new tone without the noise.