**Laboratory 2C:**

Fourier Series and Line Spectrum

In this laboratory you will experiment with non-ideal conditions to approximate the Fourier seires using the Fast-Fourier-Transformation (FFT).

Plus you will use the FFT to find out the encoding of telephone dialing system DTMF or “Touch-Tone”.

# Using the FFT in practical cases

So far the approximation we calculated with the fft() function was quite ideal, because we knew exactly the period of the incoming signal and we have adapted the time vector accordingly to match exactly one period of this time signal, plus using a fine resolution in the frequency domain. In practical cases this cannot be done, because the incoming signal is usually unknown, and you do not have the absolute control over the sampling frequency and length of the input buffer (e.g. on the oscilloscope).

What can we expect then in such “real” cases, and how can we optimise the calculated output?

1. Try out the code below and calculate for the different N values the achievable resolution in the frequency domain. Check with the pointer the frequency values for the first harmonic, and determine which ck coefficient does it corresponds to.

% PARAMETERS

N = 2^9; % number of points, try N=128, 512

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

Fs = 80e3; % sampling frequency

t = (1/Fs)\*aux;

f = (Fs/N)\*aux;

% FUNCTIONS

x\_t = 2\*square(2\*pi\*1.15e3\*t);

X\_f = (1/N)\*fft(x\_t);

% PLOTS

figure(1)

subplot(121),plot(t,x\_t),grid on

xlabel('t (s)')

subplot(122),plot(f,db(X\_f),'b',f,db(X\_f\_w),'r'),grid on

xlabel('f (Hz)')

1. Leave N=512, and add a second version of the spectrum calculated using a time window of the type Hamming (see code lines below). Superpose this second spectrum to your first plot and comment on the effect of this windowing in the spectrum calculation.   
   Hint: you can better observe the effect of the windowing by zooming in the first 10 harmonics.

window = hamming(N)';

X\_f\_w = (1/N)\*fft(x\_t.\*window);

1. Open, execute and analyse the Matlab script *sisy\_fft\_settings\_n\_effects.m*.   
   Identify the effects of the different settings. For example, how does the zero-padding works?

# Decoder for DTMF Telephone Ton Dialing

The DTMF or „Touch-Tone“ (dual-tone multi-frequency signaling) is a widespread dialing technics in analog telephony to transmit the dialed number over the network. The corresponding symbols and frequencies are given in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1209 Hz | 1336 Hz | 1477 Hz | 1633 Hz |
| 697 Hz | 1 | 2 | 3 | A |
| 770 Hz | 4 | 5 | 6 | B |
| 852 Hz | 7 | 8 | 9 | C |
| 941 Hz | \* | 0 | # | D |

1. Load in Matlab (use the command load() ) the data file *touchtoneX.mat (x for A, B, C)*, where a sequence of audio tones representing a dialed number, and other parameter (like sampling frequency) are saved.
2. Hear the audio sequence (use the command sound() ) and prepare a plot of the audio signal in the time domain.
3. Extend your script to process the audio sequence and find out the three dialed numbers.   
   Hints:

* Cut out nine intervals from the audio sequence representing each a digit from the dialed number, and use the fft() to analyse each interval.
* Prepare a plot of the spectrum and either manually or using the max function, find out the two frequencies contained in each interval and the corresponding symbol (in the DTMF table).