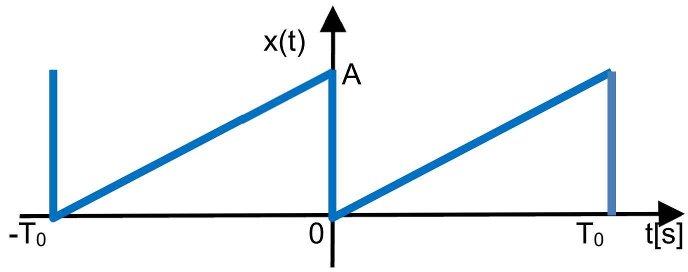
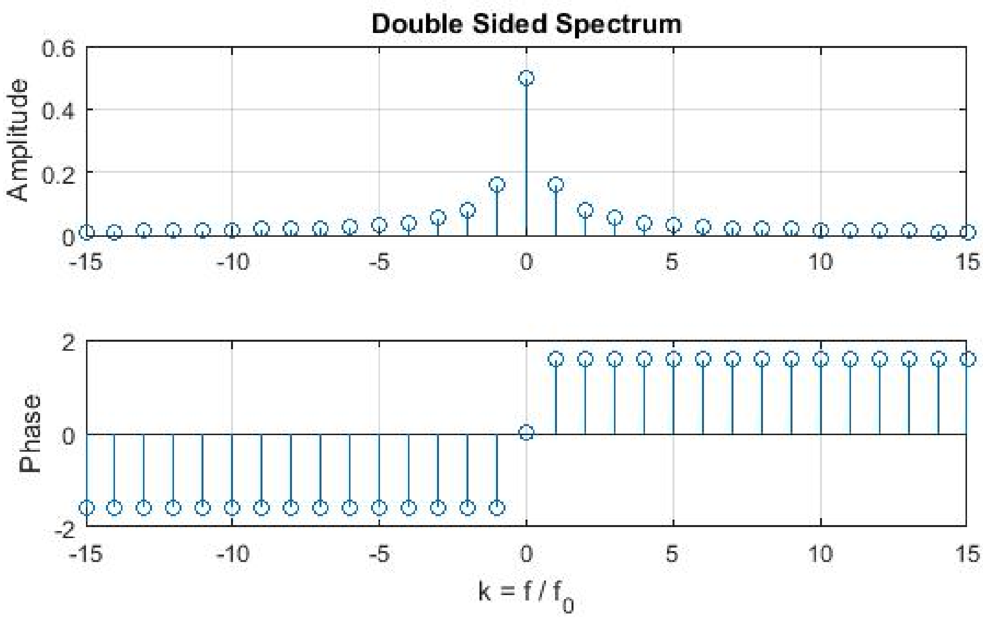
Laboratory 2B:

Fourier Series and Line Spectrum

In this laboratory you practice the concept of the Fourier Series and the corresponding Line Spectrum, which describes periodic signals in the frequency domain. We use the Fourier Series notation with the complex coefficients c_k and the associated Double Sided Spectrum.

Let us consider the example below:

Time Domain	
Transformation Fourier Series with Complex Coefficients	**Fourier Analysis: from time to frequency domain** (find out the harmonics composing a signal) $$c_k = \frac{1}{T_0} \int_{T_0} x(t) \cdot e^{-jk\omega_0 t} dt$$
	Fourier Synthesis: from frequency to time domain (sum up the harmonics components building up a signal) $$x(t) = \sum_{k=-\infty}^{+\infty} c_k \cdot e^{jk\omega_0 t}$$
Frequency Domain Line Spectrum Amp(c_k) Phase(c_k)	

The c_k coefficients for the example above (periodic sawtooth signal), are calculated step-by-step in the list of exercises **sisy-en-material\EXERCISES\SiSy_exer2_fouseri.docx** (exercise-7), and are equal to:

$$c_k = \frac{jA}{k2\pi} \qquad for \quad k \neq 0$$

$$c_0 = \frac{A}{2} \qquad for \quad k = 0$$

Calculating the c_k coefficients for x(t), means doing the **Fourier Analysis** of x(t) and determining the corresponding Double Sided Spectrum of x(t).

Now, in order to check the expression that you calculated for the c_k coefficients, you can carry out the **Fourier Synthesis**, and build up the time signal, by adding up its harmonic components. Let us do that in a Matlab script.

```
% Define Constants and time vector
T0 = 2; w0 = 2*pi/T0; A = 1;
t = -2*T0:T0/100:2*T0;

Kmax = 5;   % Max number of harmonics
c0 = A/2;   % DC-Offset (average value)

% since c0 has a different expression than the other ck, will take c0
% outside of the loop, and combine with the initialisation of x(t)
x_t = c0*ones(1,length(t));   % initialise x(t) with DC-content

for k = -Kmax:1:Kmax
    if k ~= 0
      ck = j*A/(k*2*pi);
      x_t = x_t + ck*exp(j*k*w0*t);
    end
end

figure(), plot(t,x_t), grid on
xlabel('t [s] \rightarrow'),ylabel('x(t)')
title('Periodic Synthesised Time Function')

disp('Because of warning, check how big mag(imag(x_t))')
max(imag(x_t))
```

The for-loop above implements the sum of the **Fourier Synthesis** equation. This loop could also be implemented as:

$$x(t) = c_0 + \sum_{k=1}^{\infty}\left[c_k \cdot \exp(+jk\omega_0 t) + c_{-k} \cdot \exp(-jk\omega_0 t)\right] = \sum_{k=-\infty}^{\infty}\left[c_k \cdot \exp(jk\omega_0 t)\right]$$

…

```
for k = 1:Kmax
    ck = j*A/(k*2*pi);
    x_t = x_t + ck*exp(j*k*w0*t) + conj(ck)*exp(-j*k*w0*t) ;
end
```

…

Or also like this, using the relationship of the c_k coefficients with the A_k and φ_k coefficients of the single sided spectrum.

…

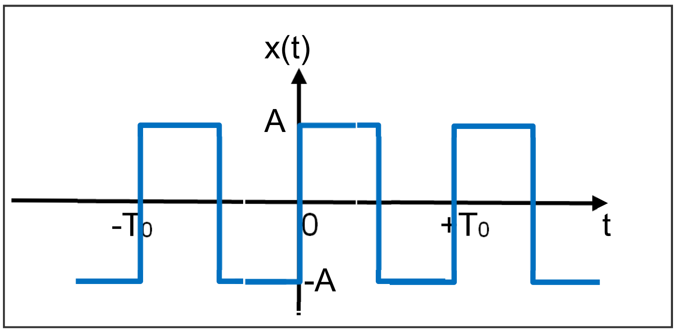
```
for k = 1:Kmax
    ck = j*A/(k*2*pi);
    x_t = x_t + 2*abs(ck)*cos(k*w0*t + phase(ck));
end
```

…

Now it is your turn to experiment the Fourier Analysis and the Fourier Synthesis with another periodic signal. Follow the instructions below.

1. Fourier Series Coefficients: calculate and plot

(a) Calculate the complex Fourier coefficients of an odd periodic square with amplitude varying between +A and -A, period T_0 and duty cycle 50%.



Please use the c_k integral definition for your calculation:

$$c_k = \frac{1}{T_0} \int_{T_0} x(t) \cdot e^{-jk\omega_0 t} \, dt$$

(b) Do the values of c_k depend on T_0 ?
Is it possible to have a simpler expression for all odd, and all even c_k coefficients?

(c) Check what are the numerical values of c_k for k within [-5 ; +5] .

k	-5	-4	-3	-2	-1	0	1	2	3	4	5
c_k											

(d) Are the non-zero c_k coefficients pure real or pure complex numbers? What does this mean compared to the real coefficients a_k and b_k ?

(e) Check the values of your c_k coefficient with a Fourier Synthesis in a Matlab script.

(f) Prepare a plot of the corresponding Double Sided Spectrum in Matlab. You can use k (index of harmonics) or frequency in Hz for the horizontal axis.

Hint: you can define your f vector as f = k*f_0 , where k is the index vector for the c_k coefficients and f_0 is the fundamental frequency of the periodic signal x(t).

2. Numerical Approximation for Fourier Series with FFT

(a) Start a new script in Matlab, and insert the code shown below:

```
% SiSy : LAB-2B Exercise-2
% ==============================
clear all, close all, clc;

% DEFINE PARAMETERS & VECTORS
T0 = 2;                 % period of the signal in time domain

N = 2^8;                % number of points for the FFT
aux = 0:1:N-1;          % auxiliary index vector

% Select the sampling period, to have exactly N-points/period (ideal situation)
tstep = 1*T0/N;         % resolution in the time domain
t = tstep*aux;          % time vector

Fs = (1/tstep);         % sampling frequency
fstep = Fs/N;           % resolution in the frequency domain
f = fstep*aux;          % frequency vector

% DEFINE TIME-FUNCTION AND CALCULATE NUMERICAL APPROX OF SPECTRUM
x_t = square(2*pi*t/T0,50); % periodic square with 50% duty cycle and period T0
c_k = (1/N)*fft(x_t);   % normalised FFT = approx(ck coefficients)

scrsz = get(groot,'ScreenSize');
figure('Position',[1 0.4*scrsz(4) 0.8*scrsz(3) 0.5*scrsz(4)])

subplot(221),plot(t,x_t),grid on,xlabel('t (sec)'),ylabel('x(t)')
subplot(222),stem(aux,abs(c_k)),grid on,xlabel('k'),ylabel('abs(c_k)')
subplot(224),stem(f,abs(c_k)),grid on,xlabel('f [Hz]'),ylabel('abs(c_k)')

pause(2)
subplot(221),ylim([-1.2 1.2]),title('Time Domain')
subplot(222),xlim([0 20]),title('Freq Domain')
subplot(224),xlim([0 20*fstep])
```

(b) Execute the code and analyse the following points:

i) How is the periodic square function x(t) defined?
ii) With how many points per period?
iii) Which function is used to calculate the numerical approximation of the ck coefficients?
iv) Which ck coefficients are visible after the zoom (with xlim() function) ?
v) What is plotted: amplitude-part or phase-part of the spectrum ?

(c) Vary the duty cycle of the periodic square x(t) (e.g. 50%, 25%, 20%, 10%). What happens with the corresponding spectra? Find out what are the zero-crossings of the amplitude spectrum for the different values of the duty cycle. For example for:

Duty cycle	50% or $\frac{\tau}{T_0} = \frac{1}{2}$	25% or $\frac{\tau}{T_0} = \frac{1}{4}$	20%	10%
Zero-crossings $c_k = 0$ for …	k = 2 ; 4 ; 6 ; 8 … or k = 2*n			

(d) Change x(t) for a periodic ramp (use in Matlab the function sawtooth()) . Which changes can you observe in the spectrum?

(e) Let us observe now the measurement of an amplitude spectrum with the FFT function of the oscilloscope. Start by measuring the spectrum of a sine function with 2Vpp and f_0=1kHz. Generate your sine signal with the Function Generator (FuGe). Which curve do you expect to see in this measurement?

(f) Change now the FuGe settings to generate a periodic square with different duty cycles (as in your Matlab code of part-c). Adjust the horizontal scale such that you can observe at least the first 20 harmonics (and maximum 50 harmonics).

(g) Can you imagine why the spectrum displayed by the oscilloscope vary significantly (specially the “noise” part) when you change the horizontal scale? Change the window type from rectangular to Hanning, which changes do you observe in the spectrum?

(h) Open, execute and read the Matlab script *sisy_fft_ideal_vs_non_ideal.m* . Which of these non-ideal effects do you think can have an influence on your measurement of part (g) ? In Lab2C we will continue to investigate the FFT, and its usage in non-ideal situations (practical cases).