Lab 1 - Fourier Series Analysis and Synthesis

Objectives

In this lab we will numerically study Fourier series (FS) analysis and synthesis. Periodic signals can be represented by (possibly infinite) sum of harmonically related complex exponentials given by the Fourier series representation. We will study

- Computing Fourier series coefficients numerically
- Fourier series reconstruction and approximation errors
- Gibbs Phenomenon
- Properties of Fourier series.

1.1 Finding Fourier series coefficients

For a periodic signal $x(t)$ Fourier series coefficients are determined by

$$a_k = \frac{1}{T}\int_{\langle T\rangle} x(t)\mathrm{e}^{-jk\omega_0 t}\, dt, \qquad k \in \mathbb{Z} \tag{1}$$

Write a **Matlab function** "fourierCoeff(t,xt,T,t1,t2,N)" to compute FS coefficients for a given periodic signal x(t). Your function should take inputs:

- t, symbolic variable for integration (see Hints below)
- xt, signal as a function of symbolic variable 't'
- T, time period of the signal
- t1, t2 are the left and right limits where the expression xt is valid (else zero otherwise)
- N, number of Fourier Series coefficients to compute

and should return:

- F, vector which contains the Fourier series coefficients { a_{-N}..., $a_{-1,}a_{0,}a_{1,}$... $a_{N,}$}.

a) Write a **Matlab script** file which calls "fourierCoeff" function for the sum of cosines signal $x(t) = 2\cos(2\pi t) + \cos(6\pi t)$, T = 1, and N = 5. Plot the FS coefficients. Verify your code by analytically computing the answer.

b) In the same script, repeat above for the periodic square wave with one period (T) given by

$$x(t) = \begin{cases} 1, & -\mathrm{T}_1 \le t \le \mathrm{T}_1 \\ 0, & \mathrm{T}_1 < |t| < T/2 \end{cases}$$

where N = 10, $\mathrm{T} = 1$ and $\mathrm{T} = 4\mathrm{T}_1$.

c) (Optional) Try it for any other periodic signal of your choice.

Hints:

1. Fundamental period (T) and fundamental frequency (w_0) are related as $w_0 = \frac{2\pi}{\mathrm{T}}$
2. Making use of symmetry, it is convenient to perform integration over $-T/2 \leq t \leq T/2$
3. Use MATLAB command ***int (expr, t, a, b)*** for numerical integration; **expr** represents the function to be integrated, **t** is a symbolic variable for the symbolic expression, and **a** and **b** are limits of the definite integral. You need to define **t** as symbolic variable using the command ***syms.*** Type 'doc int' in command prompt to get its documentation. An example is given below

   ```
   syms t;
   expr = t*(1+t);
   F = int (expr,t, [0 1]);
   ```
4. Avoid using i or j as loop variables. In Matlab they can be interpreted as the imaginary number iota i.e. square root of -1. Also, while using imaginary number iota in your code it is preferable to write it as 1i which is unambiguous, for example $\mathrm{e}^{j\frac{\pi}{3}}$ is exp(1i*pi/3).
5. Use different matlab files for the script and function. Use the following code templates to get started for this problem (copy-paste into matlab and edit):

Matlab Script	Matlab Function

% define relevant parameters
...
 	
% define relevant expressions
syms t;
xt = ... % sum of cosines wave
 	
% function call to find FS coefficients
F = fourierCoeff(t,xt,T,t1,t2,N);
 	
% plotting
FS_idx = -N:N;
figure; stem(FS_idx,F); grid on;
	function F = fourierCoeff(t,xt,T,t1,t2,N)
	% function to find FS coefficients
	
	% initialize
	w0 = ...
	F = zeros(2*N+1,1);
	
	% for-loop to find coefficents
	for nn = 1:2*N+1
	    F(nn) = ...
	end
	
	end

1.2 FS reconstruction and finite FS approximation error

For a signal $x(t)$ with Fourier series coefficients $\{a_k\}$, a partial Fourier sum (of order N) is given as

$$\hat{x}(t) = \sum_{k=-N}^{N} a_k \, \mathrm{e}^{jk\omega_0 t} \tag{2}$$

As the order N is increased (to infinity) the partial sum approaches the original signal $x(t)$.

Write a **Matlab function**, “partialfouriersum”, to compute a partial Fourier sum from a given vector of Fourier series coefficients { a_k }. Your function should take as input

- A, a 2*N+1 vector of Fourier Series coefficients { a_{-N}..., a_{-1},a_0,a_1,... a_N,}
- T, the period of the signal
- time_grid, a time vector (this is not a symbolic variable, example is given below)

and should return the vector y, which corresponds to the Fourier series reconstruction signal $y(t)$ obtained from computing the partial Fourier sum from -N to N. Note that we evaluate $y(t)$ at all the time points in the vector time_grid. Use the following template:

```
function y = partialfouriersum (A, T, time_grid)
% Compute N based on the length of A
y = zeros(size(time_grid));
for k = -N:N
    y = y + ...
end
end
```

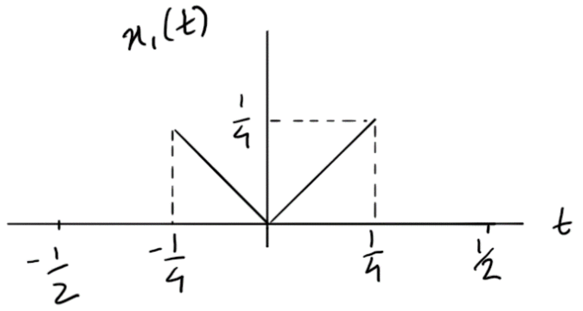
a) Write a **script** which takes the output of part (a) in 1.1 and reconstructs one period of the cosine wave, T = 1 and time_grid = -0.5:0.01:0.5. In reality, time is continuous, but for sake of plotting and analysis we consider a fine time grid (0.01 spacing) to imitate continuous time. You can increase spacing to 0.001 for a finer resolution (but longer computations).
b) Plot $x(t)$, original signal given in 1.1 (a) and the reconstructed signal in the same plot (use plot command and not stem).
c) For a fixed set of input parameters, from the outputs of the above function compute the following two types of error:
   1) The maximum absolute error (MAE) between original signal and reconstructed signal
   2) The root mean squared (RMS) error between original signal and reconstructed signal.
d) (Optional) Write a **Matlab script**, for the square wave in 1.1 (b), which computes these two errors as N is increased from 1 to 100 and plots them. What are your observations?
e) Commit all your codes GitHub at this point!!

1.3 Gibbs Phenomenon – revisit square wave

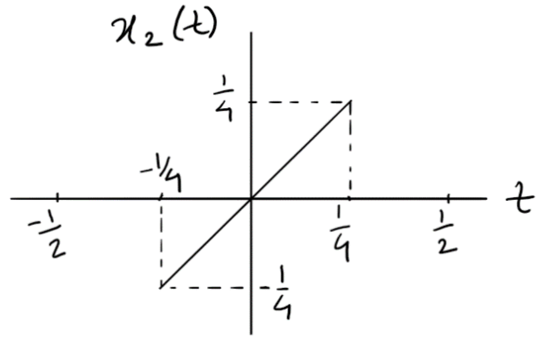
a) What are the Fourier Series coefficients { a_k } for a real, periodic square wave that has amplitude 1 in [-T_1, T_1] and period T? Note that we require T_1 < T/2.
b) Find these FS coefficients by calling the function “fourierCoeff”. Use T_1 = 0.1, T=1, N = 10*T. Plot the scaled coefficients Ta_k. What happens as T is increased to T = 10, T = 20, and more? Can you relate to the discussion in class as T goes to infinity?
c) Do partial Fourier sum reconstruction of square wave using the function “partialfouriersum” and plot it. Use T_1 = 0.1, T = 1, t = -0.5:0.01:0.5, N = 10. Repeat this as N is increased (N = 50,100,1000) and note your observations (Gibbs Phenomenon, Section 3.4 in book OWN).

1.4 Fourier series – more examples and symmetry properties

a) Team member 1 - write a Matlab script to find FS coefficients of $x_1(t)$ using “fourierCoeff”.
b) Team member 2 - write a Matlab script to find FS coefficients of $x_2(t)$ using “fourierCoeff”.
c) What do you expect the results to be? What symmetry properties hold for these examples?
d) Don’t forget to commit your codes and answers to GitHub!



Period $T = 1$


$x_2(t)$
$\frac{1}{4}$
$-1/4$
t
$-\frac{1}{2}$
$\frac{1}{4}$
$\frac{1}{2}$
$-\frac{1}{4}$