

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.

Provide answers to all questions and your observations in the form of comments in the code.

1.1 Symbolic variables and Integration in MATLAB

- Define **t** as symbolic variable using the command ***syms***.
- Use the MATLAB built-in 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. An example is given below:

```
syms t;  
expr = t*sin(t);  
F = int (expr,t, [0 1]);
```

- 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 $e^{j\frac{\pi}{3}}$ is `exp(1i*pi/3)`.
- Fundamental period (**T**) and fundamental frequency (**ω_0**) are related as $\omega_0 = \frac{2\pi}{T}$

1.2 Finding Fourier series coefficients

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

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

- 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
 - **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) for the purpose of integration
- N, number of Fourier Series coefficients to compute

This function should return a vector F containing the Fourier series coefficients $\{a_{-N}, \dots, a_{-1}, a_0, a_1, \dots, a_N\}$. Use the following code templates to get started for this problem:

Matlab Script	Matlab Function
<pre>% 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;</pre>	<pre>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 coefficients for nn = 1:2*N+1 F(nn) = ... end end</pre>

b) 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 $N = 5$. Set $t1 = 0$, $t2 = T$. Choose Plot the FS coefficients. Verify your code by analytically computing the answer.

c) Repeat above for the periodic square wave with one period (T) given by

$$x(t) = \begin{cases} 1, & -T_1 \leq t \leq T_1 \\ 0, & T_1 < |t| < T/2 \end{cases}$$

where $N = 10$, $T = 1$ and $T = 4T_1$. Set $t1 = -T_1$, $t2 = T_1$ for integration.

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

1.3 FS reconstruction and finite FS approximation error

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

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

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 $\{a_{-N}, \dots, a_{-1}, a_0, a_1, \dots, a_N\}$
- 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
```

- Write a **script** which takes the output of part (b) in 1.2 and reconstructs one period of the cosine wave, $T = 1$ and $\text{time_grid} = -0.5:0.01:0.5$. 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).
- Plot $x(t)$, original signal given in 1.2 (b) and the reconstructed signal in the same plot (use plot command and not stem). Write code to compute the maximum absolute error between original signal and reconstructed signal.
- Write a **Matlab script**, for the square wave in 1.2 (c), which computes these two errors as N is increased from 1 to 100 and plots them. What are your observations?
- Commit all your codes GitHub at this point!!**

1.4 Gibbs Phenomenon – revisit square wave

- 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 $T_1 < T/2$.
- 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?
- 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 ($N = 50, 100, 200$) and note your observations (Gibbs Phenomenon, Section 3.4 in book OWN).