

EEE - 321: Signals and Systems

Lab Assignment 2

Part 1

Consider a signal defined as

$$x_s(t) = \sum_{i=1}^M A_i e^{j\omega_i t}. \quad (1)$$

As you see, $x_s(t)$ is obtained as the superposition of M complex exponentials, where A_i and ω_i respectively denote the complex amplitude and angular frequency of the i th complex exponential.

In this part, you will write a Matlab program (in the form of a Matlab function) that computes the values of $x_s(t)$. Your function should look like

```
function [xs] = SUMCS(t,A,omega)
```

where

- **t**: $1 \times N$ vector that contains the time instants over which $x_s(t)$ is computed.
- **A**: $1 \times M$ complex-valued vector. i th element is A_i .
- **omega**: $1 \times M$ vector. i th element is ω_i .

While writing your function, make use of the **exp** function of Matlab. Use a single **for** loop over the index i of Eq. 1. Do not use any **for** loop over the time points. Your program does not need to show any error message if a wrong value is given for any of the inputs (for instance, if **A** is of size 1×5 but **omega** is of size 1×7). Simply assume that you will always run your function with correct inputs. Also, throughout all your programs, in order to avoid programming bugs, reserve the letter j for $\sqrt{-1}$, i.e., do not use it as the counter of any **for** loop or so.

Hint: Given **A** (or **omega**), you can use the **length** command of Matlab to determine its length. Type **help length** in Matlab to learn how it works.

After writing your function, take **t**=[0:0.001:1] and $n = \text{mod}(\text{your ID number}, 47) + 5$. Then, create a random $1 \times n$ complex amplitude array for **A**. The real and imaginary parts of the complex amplitude should be between 0 and 3. Also create another $1 \times n$ real frequency array for **omega**. The created frequency array should be **between 0 and π** . You can create these random arrays using the **rand** function of Matlab. Type **help rand** to learn how to use the **rand** function. (Note that Matlab's random number generator, and the other random number generators used in other computer programs and languages, are pseudo-random generators. That means, they do not produce completely random numbers. You can make a web search to learn more about this topic.)

Using your function, compute **xs** (which is complex valued). Then, extract the real and imaginary parts of **xs**. Plot the real and imaginary parts (versus **t**) separately.

Extract the magnitude and phase of **xs** and plot them separately. Increase **omega** range to **between 0 and 2π** and plot again. What difference do you observe? Try increasing range to **between 0 and 10π** . What might be the reason for seeing a different shape which has dips and peaks for 10π but not for π ? Do you think increasing time range would have the same effect?

Part 2

Consider a signal $x(t)$ which is periodic with T such that $x(t) = x(t + T)$ for all t . In the lectures, you learned that we can write

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{j\frac{2\pi kt}{T}} \quad (2)$$

where

$$X_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-j\frac{2\pi kt}{T}} dt. \quad (3)$$

Above, Eq. 2 is called the Fourier series expansion of $x(t)$, and the expansion coefficients X_k can be computed according to Eq. 3. As you see, in its Fourier series expansion, we write a signal $x(t)$ as the weighted superposition of complex sinusoidal signals $e^{j\frac{2\pi kt}{T}}$ where the weights are given by X_k .

Note that Eq. 2 contains an infinite number of complex sinusoids. In practical situations, if we wish to use Eq. 2 to compute $x(t)$, we can not add all the sinusoids, but instead we can only add a finite number of them. In this respect, consider the signal $\tilde{x}(t)$ which is defined as

$$\tilde{x}(t) = \sum_{k=-K}^K X_k e^{j\frac{2\pi kt}{T}} \quad (4)$$

where K is a positive integer. Note that if $K = \infty$, we would have $\tilde{x}(t) = x(t)$. When K is finite, $\tilde{x}(t)$ is only an approximation to $x(t)$. Now, suppose, within $t \in [-\frac{T}{2}, \frac{T}{2})$, $x(t)$ is given as

$$x(t) = \begin{cases} 1 - 2t^2 & \text{if } -\frac{W}{2} < t < \frac{W}{2}, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

where

$$W < T. \quad (6)$$

(Recall that in the beginning we assumed that $x(t)$ is periodic with T . The above equation describes $x(t)$ in only ONE period. It does not specify $x(t)$ FOR ALL t . Do not get confused on this point.)

Take $T = 2$, $W = 1.0$ and sketch (with your hand) $x(t)$ over $-1.5T < t < 1.5T$. Clearly indicate the signal amplitudes on the sketch.

Determine the Fourier series expansion coefficients X_k of $x(t)$ in Eq. 5 for generic values of T and W by evaluating the integral in Eq. 3 with your hand. (You may utilize the integration by parts while calculating the integral.) Simplify your result as much as possible.

Part 3

In this part, you will write a Matlab function which makes the Fourier synthesis for the signal given in Eq. 5. In other words, your code will compute $\tilde{x}(t)$ from the given parameters K , T , W for a given time interval. Your function should look like

```
function [xt] = FSWave(t,K,T,W)
```

where

- t denotes the time grid over which $\tilde{x}(t)$ is computed.
- xt denotes the values of $\tilde{x}(t)$ computed over t .
- K , T and W denote the parameters K , T and W that appear in Eq. 4 and Eq. 5.

While writing your code, use the function `SUMCS` that you developed in Part 1. Use only a single `for` loop over the index parameter k of Eq. 4. (Note that, within the code you need to compute X_k).

Let $D_{11} = \text{mod}(\text{your ID number}, 11)$ and $D_5 = \text{mod}(\text{your ID number}, 5)$.

Take $T=2$, $W=1$, $K=25+D_{11}$ and $t=[-5:0.001:5]$. Compute xt .

Extract the real and imaginary parts of xt and plot them versus t . Examine the maximum and minimum values that you see on the plots. What are these values for real and imaginary parts? What can you say when you compare them? Can you ignore the imaginary part when compared to the real part? Actually, since $x(t)$ in Eq. 5 is a real valued signal, we would expect the imaginary part to be zero. But then, why do you think that the imaginary part is not perfectly zero? (*Hint*: We know that $\sin(\frac{\pi}{6})$ is equal to 0.5. Then, $\sin(\frac{\pi}{6}) - 0.5$ must be equal to 0. Try computing $\sin(\frac{\pi}{6}) - 0.5$ in Matlab. What result do you get?)

Again take $T=2$, $W=1$ and $t=[-5:0.001:5]$ and produce five plots taking $K=2+D_5$, $K=8+D_5$, $K=17+D_5$, $K=50+D_5$, $K=100+D_5$. Now you know that xt is essentially real valued you can display only the real part of xt . What do you observe as K gets larger? In particular, what can you say about the success of $\tilde{x}(t)$ in approximating $x(t)$? To see clearly, first write a function that evaluates the periodic $x(t)$ of Eq. 5 over t (function $x(t)$ should cover $t=[-5:0.001:5]$ now, not just one period). Then, compute an error signal, which is the difference between $x(t)$ and $\tilde{x}(t)$, and observe what happens to its overall behavior and values at discontinuities as K increases. What might be the reason for the error signal to not drop at discontinuities as K increases? Do you observe some irregularities or oscillations within the neighborhood of the discontinuities in $\tilde{x}(t)$?

Part 4

In this part, you will make small changes in the function `FSWave` that you wrote in Part 3, and you will observe the resulting changes. Consider the following equation:

$$y(t) = \sum_{k=-K}^K Y_k e^{j\frac{2\pi kt}{T}} \quad (7)$$

- **Part a)** Suppose $Y_k = X_{-k}$. What is the required change in your `FSWave` function to compute $y(t)$ from X_k s. Take $T=2$, $W=1$, $K=25+D_{11}$ and $t=[-5:0.001:5]$. Run your code, and compute and plot $y(t)$ versus t (only the real part). What is the effect of the operation that we performed?

- **Part b)** Suppose $Y_k = X_k e^{-j\frac{2\pi kt_0}{T}}$ where t_0 is a new parameter. What is the required change in your **FSWave** function to compute $y(t)$ from X_k s. Take $T=2$, $W=1$, $K=25+D_{11}$, $\mathbf{t}=[-5:0.001:5]$ and $t_0 = 0.8$. Run your code, and compute and plot $y(t)$ versus t (only the real part). What is the effect of the operation that we performed?
- **Part c)** Suppose $Y_k = jk\frac{2\pi}{T}X_k$. What is the required change in your **FSWave** function to compute $y(t)$ from X_k s. Take $T=2$, $W=1$, $K=25+D_{11}$ and $\mathbf{t}=[-5:0.001:5]$. Run your code, and compute and plot $y(t)$ versus t (only the real part). What is the effect of the operation that we performed?
- **Part d)** Suppose

$$Y_k = \begin{cases} X_{K+1-k} & \text{if } k = 1, 2, \dots, K \\ X_k & \text{if } k = 0 \\ X_{-(K+1+k)} & \text{if } k = -K, -K+1, -K+2, \dots, -1 \end{cases} \quad (8)$$

What is the required change in your **FSWave** function to compute $y(t)$ from X_k s. Take $T=2$, $W=1$, $K=25+D_{11}$ and $\mathbf{t}=[-5:0.001:5]$. Run your code, and compute and plot $y(t)$ versus t (only the real part). What is the effect of the operation that we performed?