EBU5375 Signals and Systems: Analysis and synthesis equations in Matlab

Dr Jesús Requena Carrión

Queen Mary University of London

Background

Given a periodic discrete-time signal $x_N[n]$ of period N:

1. Its **fundamental frequency** is $\Omega_0 = \frac{2\pi}{N}$.
2. According to the **synthesis equation**, $x_N[n]$ can be expressed as the sum of N harmonically related complex exponentials of frequencies $\Omega_k = k\frac{2\pi}{N}$:

$$x_N[n] = \sum_{k=\langle N \rangle} a_k e^{jk\Omega_0 n}$$

3. The Fourier coefficients a_k can be determined by using the **analysis equation** as

$$a_k = \frac{1}{N} \sum_{n=\langle N \rangle} x_N[n] e^{-jk\Omega_0 n}$$

Background

In order to obtain the Fourier series decomposition of a periodic signal $x_N[n]$ of period N, we need to:

1. Identify the **fundamental frequency** Ω_0.
2. Determine the N **harmonic frequencies** $\Omega_k = k\Omega_0$.
3. Obtain the **Fourier coefficients** a_k.

Background

Determining the fundamental frequency Ω_0 and its harmonics Ω_k is **very easy**.

The Fourier coefficients a_k can be obtained analytically. For instance, for the periodic square wave of period N defined within one period centred around $n = 0$ as:

$$x_N[n] = \begin{cases} 1, & |n| \leq N_1 \\ 0, & \text{otherwise} \end{cases}$$

we can obtain its Fourier coefficients are:

$$a_k = \frac{1}{N} \sum_{n=-N_1}^{N_1} e^{-jk\Omega_0 n} = \begin{cases} \frac{2N_1+1}{N}, & k = 0 \\ \frac{1}{N} \frac{\sin[2\pi k(N_1+1/2)/N]}{sin(\pi k/N)}, & k \neq 0 \end{cases}$$

Background

You will have noticed that the analysis equation in DT:

$$a_k = \frac{1}{N} \sum_{n=\langle N \rangle} x_N[n] e^{-jk\Omega_0 n}$$

is essentially a mathematical operation in which we:

1. Calculate N products of complex numbers.
2. Add N complex numbers.

Computers are very good at doing additions and multiplications so, **why don't we let computers calculate the Fourier coefficients for us?**

Objectives of the lab

In this lab, we will **obtain numerically** the Fourier coefficients of a periodic discrete-time signal and then we will synthesise the signal by using its complex exponential components.

We will:

1. Define a square wave and identify its fundamental frequency.
2. Obtain numerically the Fourier coefficients a_k.
3. Plot the coefficients a_k.
4. Synthesise the periodic square wave as a Fourier series.
5. Lowpass filter the signal.

Step 1: Definition of the periodic DT signal

In this lab, we will work with the periodic square wave $x[n]$ with period $N = 21$ defined as:

$$x[n] = \begin{cases} 1, & |n| \leq N_1 \\ 0, & N_1 < |n| \leq 10 \end{cases}$$

Your job now is to

- Identify its fundamental frequency Ω_0 and its harmonics Ω_k.
- Draw on a piece of paper three periods of $x[n]$. We will assume $N_1 = 2$

Step 2: Obtaining the Fourier coefficients

The following lines of code calculate the Fourier coefficients a_k of $x[n]$:

```
n_s = -10;
n_e = 10;
n = n_s:1:n_e; % Definition of the time vector
x = zeros(size(n));
x(9:13)=1; % Definition of x[n]
N = 21; % Period of x[n]
omega_0=2*pi/N; % Fundamental frequency of x[n]

a_k=zeros(1,N);
for k=0:20 % This loop calculates the Fourier coefficients a_k
a_k(k+1)=(1/N)*sum(x.*exp(-k*1i*omega_0*n));
end
```

Your job now is to

- Plot the signal $x[n]$ in the time interval $-10 \leq n \leq 10$.
- Identify the analysis equation in the previous lines of code.

Step 3: Plotting the Fourier coefficients

The following lines of code plot the magnitude of the coefficients a_k

```
figure
stem([0:10,-10:-1],abs(a_k)) % plots a_k agains n
xlabel('k') % adds text below the X-axis
ylabel('a_k') % adds text beside the Y-axis
```

Your job now is to

- Identify and justify the shape of the magnitude of the coefficients a_k.
- Understand how we plot a_k against k by using `stem`.

Step 4: Synthesising $x[n]$

The following lines of code synthesise 3 periods of $x[n]$ by using the Synthesis equation:

```
n_s = -31;
n_e = 31;
n = n_s:1:n_e; % New time vector
x_syn = zeros(size(n));
k=0:20;
for m=-31:31 % Synthesis of x
x_syn(m+32)=sum(a_k.*exp(k*1i*omega_0*m));
end
```

Your job now is to

- Plot the signal $x[n]$ in the time interval $-31 \leq n \leq 31$.
- Identify the synthesis equation in the previous lines of code.

Step 4: Lowpass filtering $x[n]$

In the following lines of code, we filter out the frequencies $|\Omega| > 2\pi/21$ of $x[n]$, producing the signal $y[n]$ with Fourier coefficients b_k:

```
b_k=zeros(size(a_k));
b_k(1)=a_k(1);
b_k(2)=a_k(2);
b_k(21)=a_k(21);
for m=-31:31 % Synthesis of x
y(m+32)=sum(b_k.*exp(k*1i*omega_0*m));
end

figure
stem(n,y) % plots a_k agains n
xlabel('n') % adds text below the X-axis
ylabel('y') % adds text beside the Y-axis
axis tight
```

Your job now is to

- Plot the signal $y[n]$ in the time interval $-31 \leq n \leq 31$.
- How have we implemented the filter $x[n]$?

Step 5: Changes in the duty cycle

The duty cycle of the proposed discrete-time periodic signal is $\rho = (2N_1 + 1)/21$. For instance, for $N_1 = 2$, the duty cycle is $\rho = 5/21$.

Your job now is to

- Plot the $x[n]$ for $\rho = 1/21, 3/21, 11/21, 21/21$.
- Plot the Fourier coefficients a_k for $\rho = 1/21, 3/21, 11/21, 21/21$. Discuss your results.