DSP Lab 09: The Discrete Fourier Transform

1. Objective

The students shall know how to use Matlab's `fft()` and `ifft()` functions for frequency analysis of discrete signals.

2. The Discrete Fourier Transform (DFT)

A discrete, periodic signal of period N:

$$x[n] = x[n+N]$$

can always be decomposed as a **sum of complex exponentials**:

$$x[n] = \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}, n = 0, 1, \dots N-1$$

The coefficients X_k can be found with:

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Compared to the Fourier series of continuous signals:

- the fundamental frequency is $f_0 = 1/N$
- there are only N terms, with frequencies $k \cdot f_0$:
  - $0, f_0, 2f_0, \dots (N-1)f_0$
- there are only N distinct coefficients X_k
- the N coefficients c_k can be chosen like $-\frac{N}{2} < k \leq \frac{N}{2}$ => the frequencies span the range $-1/2 \dots 1/2$

$$-\frac{1}{2} < f_k \leq \frac{1}{2}$$

$$-\pi < \omega_k \leq \pi$$

2.1 Basic properties of the DFT coefficients

1. Signal is **discrete** --> coefficients are **periodic** with period N

$$X_{k+N} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi(k+N)n/N} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

2. If signal is real $x[n] \in \mathbb{R}$, the coefficients are **even**:

   - $X_k^* = X_{-k}$
   - $|X_k| = |X_{-k}|$
   - $\angle X_k = \angle D_{-k}$

- Together with periodicity:
  - $|X_k| = |X_{-k}| = |X_{N-k}|$
  - $\angle X_k = -\angle X_{-k} = -\angle X_{N-k}$

2.2 Expressing as sum of sinusoids

The DFT indicates that a discrete, periodical signals can always be written as a sum of sinusoidal signals.

- Grouping terms with X_k and X_{-k} we get
$$x[n] = X_0 + 2\sum_{k=1}^{L} |X_k| cos(2\pi \frac{k}{N} + \angle X_k)$$
where $L = N/2$ or $L = (N-1)/2$ depending if N is even or odd
- Signal = DC value + a finite sum of sinusoids with frequencies kf_0
  - $|X_k|$ give the amplitudes (x 2)
  - $\angle X_k$ give the phases

3. Matlab functions

In Matlab, the DFT coefficients are computed with the function `fft()` . The inverse DFT is computed with `ifft()` .

Example:

In [ ]:
```
x = [0 1 2 3 4 5 6];
S = fft(x)l      % Compute the DFT
x2 = ifft(x)     % Compute back the signal
```

The DFT coefficients, like any Fourier transform, are **complex numbers** in general. Their modulus and phase can be obtained with `abs()` and `angle()` .

In [ ]:
```
% Plot the modulus and phase of the Fourier coefficients
S_mod = abs(S)
S_phase = angle(S)
plot(S_mod)
figure
plot(S_phase)
```

The coefficients are returned as a vector $[X_0, X_1, \dots X_{N-1}]$. They can be rearranged in the order $[X_{-N/2+1}, \dots X_0, \dots X_{N/2-1}]$ with the function `fftshift()` :

In [ ]:
```
figure
plot(fftshift(S_mod)
figure
plot(fftshift(S_phase))
```

3.1 Matlab subplots

A figure in Matlab can be split into multiple parts with `subplot(a, b, c)` . The function takes 3 arguments:

- a = number of rows of the split
- b = number of columns of the split
- c = the current part we are in.

Example:

In [ ]:
```
figure                  % Make new figure window
subplot(2, 1, 1)        % Split in 2 rows, 1 column. We are now in part 1 of the split
plot(S_mod)                  % Plot is displayed in the first part
subplot(2, 1, 2)        % Use same split in 2 rows, 1 column. But we move now to part 2 of the split
plot(S_mod)                  % Plot is displayed in the second part
```

4. Exercises

1. Generate a 100 samples long signal `x` defined as $x[n] = 0.7 \cos(2\pi f_1 n) + 1.2 \sin(2\pi f_2 n)$, with $f_1 = 0.05$ and $f_1 = 0.1$.

   a. Plot the signal in the top third of a figure (use `subplot()`).

   b. Compute the Fourier series coefficients with `fft()` and plot their magnitude in the middle third, and their phase in the lower third.

   c. Repeat the plot but do the FFT in N=1000 points (use `fft(x, N)` . What changes?

1. Repeat the above exercise for:

   a) a constant signal $x = 7$, 100 samples long

   b) an impulse of height 5: $x = [5, 0, 0, 0, 0, \ldots]$, 100 samples long in total

   c) The ECG signal loaded from the file `ECG_Signal.mat` . What is the dominant frequency here? Why?

   d) A random signal of length 1000

   e) A piece from the middle of the song `Kalimba.mp3` , 1024 samples long

1. Repeat Exercise 1 for a signal length of 93 samples. Why do additional frequency components appear in the spectrum?

1. Generate a 39 samples long **triangular** signal `x` defined as:

   - first 10 samples are zeros
   - next, `x` increases linearly from `x(10) = 0` up to `x(19) = 4` , then decreases linearly to `x(29) = 0` .
   - last 10 samples are 0

     a. Plot the signal in the top third of a figure, the magnitude of the DFT coefficients in the middle third, and their phase in the lower third.

     b. What is the amplitude of the **third harmonic component** in the signal's spectrum?

     c. Concatenate 50 zeros at the end of the signal and redo the exercise. What do you observe?

1. Take the Fourier series coefficients of the triangular signal before, and keep only the coefficients of the DC + first two sinusoidal components. Generate the signal from the Fourier coefficients with `ifft()` and plot it.

1. Generate two 10-long random signals `x` and `y` .

   a. Perform **linear convolution** with `conv()` .

   b. Perform **circular convolution** via the frequency domain, using `fft()` and `ifft()` .

   c. Perform **linear convolution** via the frequency domain using the `fft` in N points, with N larger then 19.

d. Which method of linear convolution is is faster, `conv()` or via `fft()` ? Use long signals (e.g. length 40000).

5. Final questions

1. How do you expect the amplitudes of the Fourier coefficients to be for:
   - a slow varying signal
   - a rapid varying signal