Prof. Dr.-Ing. Gangolf Hirtz
Dipl.-Ing. Julia Richter

WS 2015/2016

Digital Systems

Exercise E01

Exercise 1 Sampling

A 12 bit analog to digital converter has a nominal input range of [-10V..10V]. The digital value is interpreted as a positive integer value.

(a) What is the integer code for the incoming voltage of -3V?

(b) What is the binary code for the incoming voltage of -3V?

(c) What is the incoming voltage calculated from that integer code?

(d) What is the range of incoming voltages that would receive the same code?

Solution

(a) As $-10V$ would be represented by code 0 and $10V$ would be represented by code $2^{12}-1$, we can create the following equation:

$$\begin{aligned} I(V_{in}) &= \frac{V_{in} - V_{ADCmin}}{V_{ADCmax} - V_{ADCmin}} \times (2^n - 1) \\ &= \frac{-3+10}{20} \times 4095 = 1433.25 \approx 1433 \end{aligned} \tag{1}$$

(b) 12 bit binary code for 1433: $0101\ 1001\ 1001_b$

(c) Inverting (a) we get:

$$\begin{aligned} V_{est} &= I(V_{est}) \times \frac{V_{ADCmax} - V_{ADCmin}}{(2^n - 1)} + V_{ADCmin} \\ &= -3.001221V \end{aligned} \tag{2}$$

(d) Maximum quantization error is half the quantization interval (LSB).

$$LSB = \frac{2\hat{U}}{2^n - 1} \tag{3}$$

$$Q_{err,max} = \pm\frac{1}{2}LSB = \frac{\hat{U}}{2^n - 1} \tag{4}$$

So the voltage range for the integer code 1433 is

$$-3.001221V \pm 0.00244V. \tag{5}$$

Exercise 2 Discrete Fourier Transform: Real and imaginary part

Figure 1 shows a complex signal consisting of cosine and sine components.

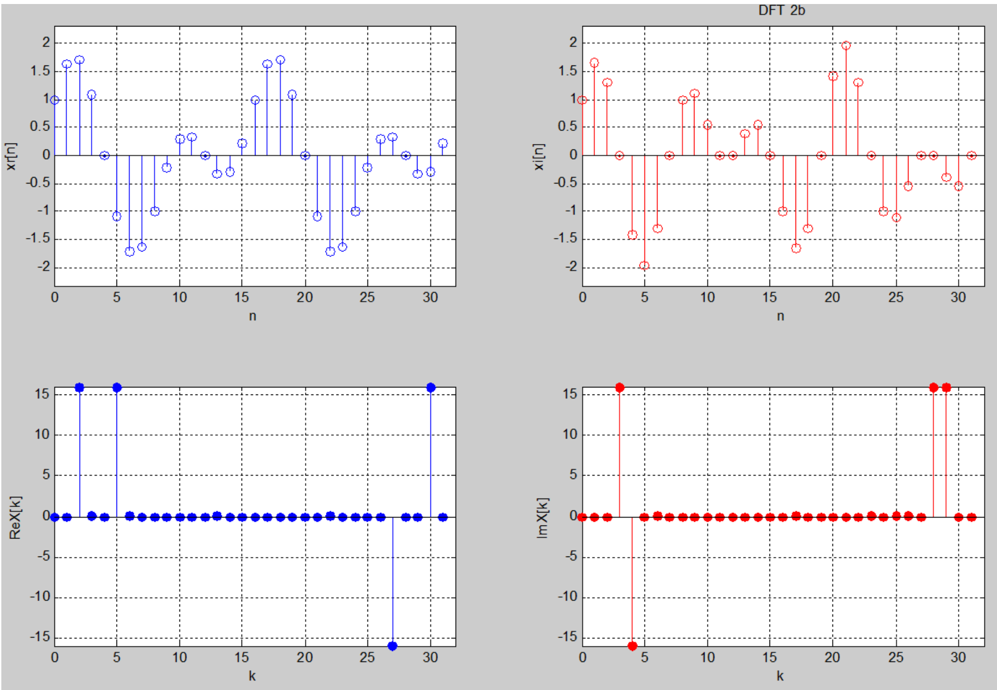

Figure 1: Signal x[n] and corresponding DFT spectrum

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \cdot 2\pi \cdot n \cdot \frac{k}{N}} = \sum_{n=0}^{N-1} x[n] \cdot cos(2\pi \cdot n \cdot \frac{k}{N}) - j \sum_{n=0}^{N-1} x[n] \cdot sin(2\pi \cdot n \cdot \frac{k}{N})$$
$$\text{with } k = 0, 1, \ldots N-1 \tag{6}$$

The following statements are valid:

- The similarity of the real part of x[n] with a cosine is represented by the DFT as real-part of X[k] (axis symmetry).

- The similarity of the real part of x[n] with a sine is represented by the DFT as imaginary part of X[k] (point symmetry).
- The similarity of the imaginary part of x[n] with a cosine is represented by the DFT as imaginary part of X[k] (axis symmetry).
- The similarity of the imaginary part of x[n] with a sine is represented by the DFT as real part of X[k] (point symmetry).

Consider the DFT coefficients in the bottom images of Figure 1 and answer the following questions:

(a) Which of the peak pairs represent a cosine component, which a sine component? Do they belong to the real or the imaginary part of the signal?

(b) Determine the frequencies the respective cosine and sine functions.

(c) In how many DFT coefficients does the DFT transformation of the given signal x[n] result?

Solution

As there are 32 samples of x[n], the DFT transformation results also in 32 coefficients.

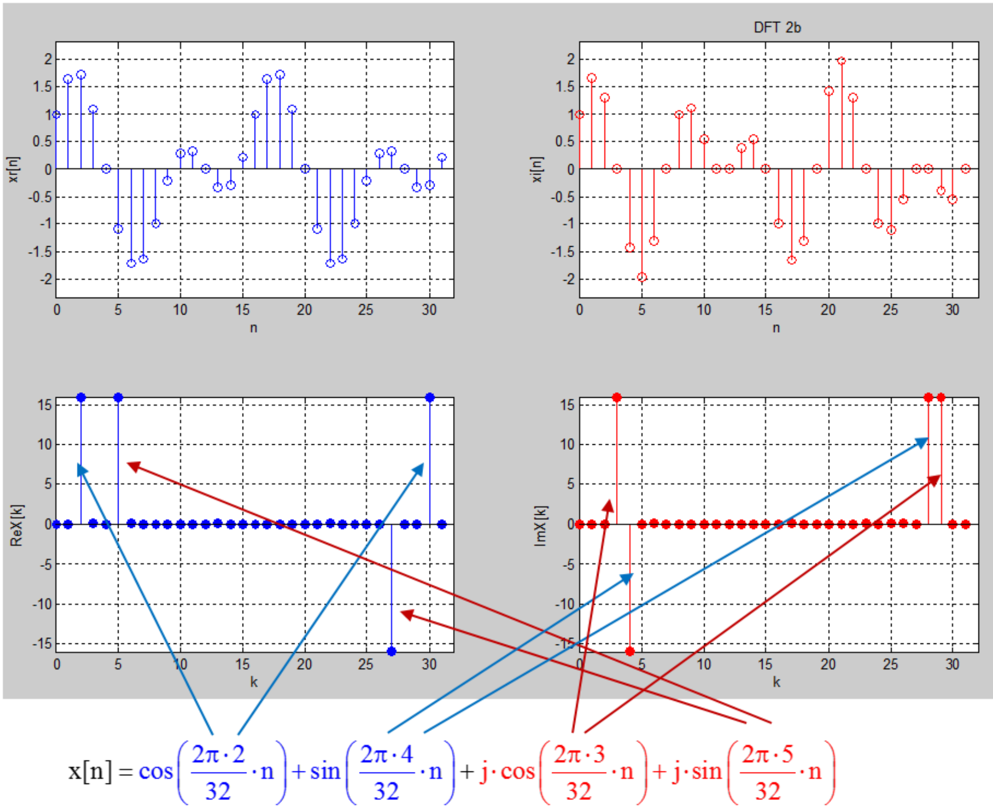

Figure 2: Representation of the signal in the DFT spectrum

Exercise 3 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) $X[k]$ of a discrete signal $x[n]$ is given by:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \cdot 2\pi \cdot n \cdot \frac{k}{N}} \text{ with } k = 0, 1, \ldots N-1 \tag{7}$$

(a) What is the difference between FFT and DFT?

(b) Given the following input signal, which was sampled with a sampling frequency $f_s = 8$ kHz for a period of 1 ms.

$$x(t) = 0.5 + \cos(2\pi \cdot 1 \text{ kHz} \cdot t) + j \cdot \cos(2\pi \cdot 3 \text{ kHz} \cdot t) \tag{8}$$

Calculate the coefficients of the DFT (real and imaginary part). Draw the spectrum.

Solution

1. Fast Fourier Transform (FFT) is a collective term for different methods of calculating the DFT. It produces the same result, but it is much more efficient. It employs different strategies to increase the computational effectiveness. The basic idea is to break up a transform of length N into two transforms of length N/2. This reduces the order of the computational effort from

$$\mathcal{O}(N^2) \tag{9}$$

for the DFT, to

$$\mathcal{O}(N \log_2 N) \tag{10}$$

for the FFT. Due to this principle the FFT can only be applied for sample numbers which are powers of 2, typically N=1024.

Example
$N = 10^9$, calculating one sample takes a computation time of 1 ns (1 GHz):

$$\begin{aligned} T_{DFT} &= 1 \text{ ns} \cdot 10^{9^2} = 10^{18} \text{ ns} \approx 31.7 \text{ years} \\ T_{FFT} &= 1 \text{ ns} \cdot 10^9 \log_2(10^9) = 2.9897 \cdot 10^{10} \text{ ns} \approx 29.99 \text{ s} \end{aligned} \tag{11}$$

The original idea was already proposed in 1805 by Gauss but was not recognized in that time. In modern time it was presented in a paper by Colley and Tukey in 1965 in a time when coputational effectiveness became more important. Common implementations of FFT are part of Matlab and OpenCV. A widely used open source library for C is FFTW (http://www.fftw.org/).

2. At first, we have to calculate the time values, where the signal was sampled:

$$dt = \frac{1}{f_s} = 0.125 \text{ ms} \tag{12}$$

$$t = 0 \text{ ms}, 0.125 \text{ ms}, 0.250 \text{ ms}, 0.375 \text{ ms}, 0.500 \text{ ms}, 0.625 \text{ ms}, 0.750 \text{ ms}, 0.875 \text{ ms} \tag{13}$$

Now, for each time step we can calculate the sampled value using Equation 8:

n	$x[n]$
0	$1.5000 + 1.0000j$
1	$1.2071 - 0.7071j$
2	$0.5000 - 0.0000j$
3	$-0.2071 + 0.7071j$
4	$-0.5000 - 1.0000j$
5	$-0.2071 + 0.7071j$
6	$0.5000 + 0.0000j$
7	$1.2071 - 0.7071j$

Using the equation

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \cdot 2\pi \cdot n \cdot \frac{k}{N}} \text{ with } k = 0, 1, \ldots N-1 \tag{14}$$

we get the Fourier coefficients.

Example ($k = 1$)
For the complex e-Term we get the values:

n	$e^{-j \cdot 2\pi \cdot n \cdot \frac{1}{N}}$
0	1.0000
1	$0.7071 - 0.7071j$
2	$0.0000 - 1.0000j$
3	$-0.7071 - 0.7071j$
4	$-1.0000 - 0.0000j$
5	$-0.7071 + 0.7071j$
6	$-0.0000 + 1.0000j$
7	$0.7071 + 0.7071j$

Multiplying each $x[n]$ and the according e-Term gives:

n	$x[n]$	$e^{-j \cdot 2\pi \cdot n \cdot \frac{1}{N}}$	$x[n] \cdot e^{\cdots}$
0	$1.5000 + 1.0000j$	1.0000	$1.5000 + 1.0000j$
1	$1.2071 - 0.7071j$	$0.7071 - 0.7071j$	$0.3536 - 1.3536j$
2	$0.5000 - 0.0000j$	$0.0000 - 1.0000j$	$-0.0000 - 0.5000j$
3	$-0.2071 + 0.7071j$	$-0.7071 - 0.7071j$	$0.6464 - 0.3536j$
4	$-0.5000 - 1.0000j$	$-1.0000 - 0.0000j$	$0.5000 + 1.0000j$
5	$-0.2071 + 0.7071j$	$-0.7071 + 0.7071j$	$-0.3536 - 0.6464j$
6	$0.5000 + 0.0000j$	$-0.0000 + 1.0000j$	$-0.0000 + 0.5000j$
7	$1.2071 - 0.7071j$	$0.7071 + 0.7071j$	$1.3536 + 0.3536j$
		$\sum$	$4.0000 - 0.0000j$

Figure 3 visualizes this values as complex spectrum.

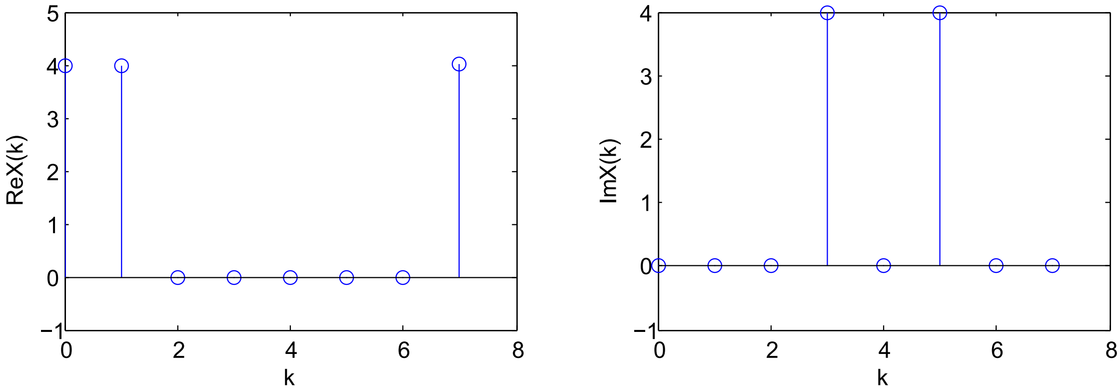

Figure 3: Complex spectrum