Understanding Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT).

Simulation and Digital Implementation by means of blocks, not C-code

Content

1 Introduction

The DFT and FFT are used mainly to obtain the amplitude of a harmonic component from a periodic signal containing several harmonics. For instance, it is desired to know the 180 Hz harmonic amplitude from a signal. Then, the DFT and FFT are the best choices to do that.

This report presents how to implement DFT and FFT by means of simplified blocks, which are *sine*, *cosine*, *sum*, *square*, etc. It is easy to find on the internet the DFT and FFT performed in C-code. Actually, the DFT and FFT are almost always performed in C-code mainly because its modularity. However, DFT and FFT in C-code are exhaustive for those who want to learn how the DFT and FFT work. Here, the objective is to present a step-by-step way to implement DFT and FFT. An 8-point DFT and FFT is presented using simplified blocks.

The simulation used in this report is freely available on https://sites.google.com/site/busarellosmartgrid/home

2 The Discrete Fourier Transform (DFT)

2.1. Defining the input signal

The input signal we want to compute the DFT is given by (1).

$$x_{input}(t) = A_1 \sin(2\pi 1000t) + A_2 \sin(2\pi 2000t + \beta) \qquad (1)$$

Where A_1 and A_2 are the amplitude of each harmonic.

The Fig. 1 presents how the input signal is simulated. In this case, A_1 = 1.125 and A_2 = 0.52.

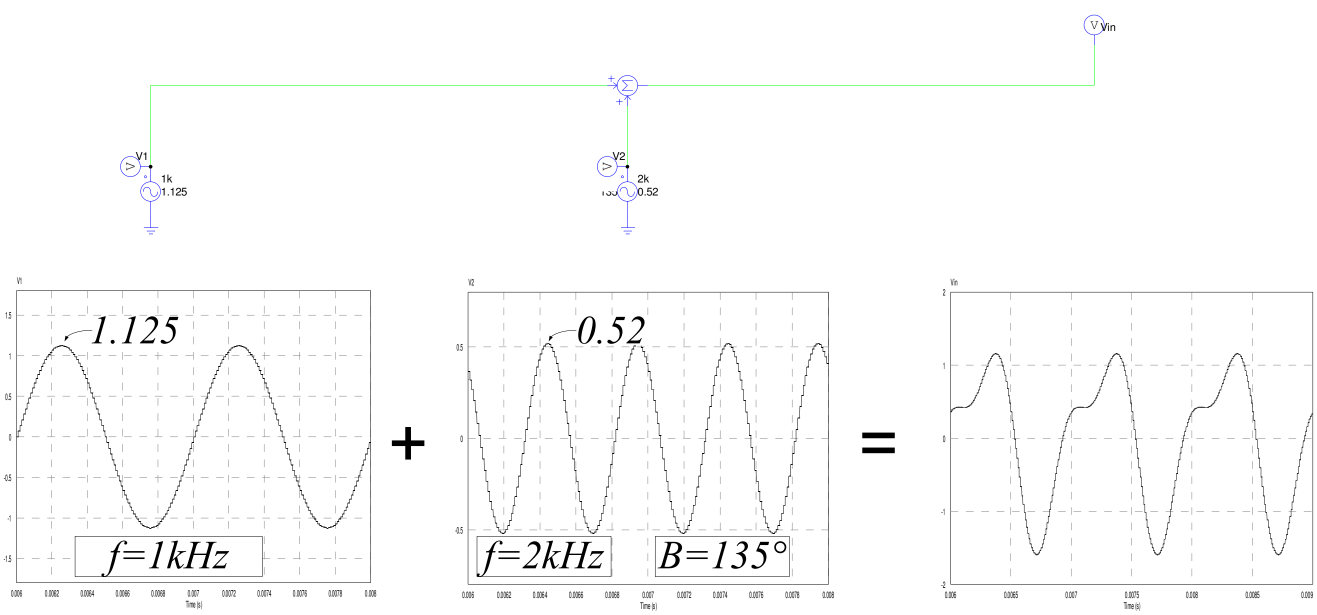

Fig. 1: How the input signal is simulated.

2.2. Simulating the DFT equation

Supposing we have just access to the Vin singnal (Vin(t) = x_{input}(t)) and we want to know what are the amplitude of the 1 kHz and 2 kHz components. This is what the DFT and FFT do.

Let's simulate the DFT equation, which is given by (2). Don't care about what are the terms *n*, *m* and *N* now.

$$X(m)=\sum_{n=0}^{N-1}x(n)e^{-j2\pi nm/N} \tag{2}$$

Let's see how to simulate it, but first equation (2) will be rewritten in a better visualized way. The Euler's relationship is given by (3).

$$e^{-j\theta}=\cos(\theta)-j\sin(\theta) \tag{3}$$

Therefore, equation (2) can be written as (4).

$$X(m)=\sum_{n=0}^{N-1}x(n)\left[\cos(2\pi nm/N)-j\sin(2\pi nm/N)\right] \tag{4}$$

The *j* term is still embarrassing us. So, we will split equation (4) into two parts, a real and an imaginary, but both with real values. The real part is given by (5).

$$X_{real}(m)=\sum_{n=0}^{N-1}x(n)\left[\cos(2\pi nm/N)\right] \tag{5}$$

The imaginary part is given by (6).

$$X_{img}(m)=\sum_{n=0}^{N-1}x(n)\left[(-1)\sin(2\pi nm/N)\right] \tag{6}$$

We clearly see that equations (5) and (6) don't have neither exponential (*e*) nor (*j*) terms. Both of them are real values. We will simulate these equations soon.

2.3. Understanding the *n*, *m* and *N* terms in DFT equation.

Let's see what are *n*, *m* and *N*.

N is the number of points the DFT will be performed. In this example, it will be 8 points.

$$\therefore N=8 \tag{7}$$

Therefore, we need 8 samples from our input signal. The sampling frequency is 8 kHz. The value *n* goes from 0 to 7.

$$\therefore\begin{cases}n=0\\n=1\\n=2\\n=3\\n=4\\n=5\\n=6\\n=7\end{cases} \tag{8}$$

x(n) in equation (4) is the sampled value. Let's see how to do that.

We are interesting in obtaining the amplitude of the 1 kHz and 2 kHz components. Therefore, the values *m* goes from 1 to 2.

$$\therefore \begin{cases} m = 1 \\ m = 2 \end{cases} \tag{9}$$

For each m, the n values go from 0 to 7, according to the following chart.

$$m = 1 \rightarrow \begin{cases} n = 0 \\ n = 1 \\ n = 2 \\ n = 3 \\ n = 4 \\ n = 5 \\ n = 6 \\ n = 7 \end{cases}$$

$$m = 2 \rightarrow \begin{cases} n = 0 \\ n = 1 \\ n = 2 \\ n = 3 \\ n = 4 \\ n = 5 \\ n = 6 \\ n = 7 \end{cases} \tag{10}$$

2.1. Sampling

Fig. 2 presents the input signal again and the 8-point sampled signal. We clearly see how bad a low sampling frequency is.

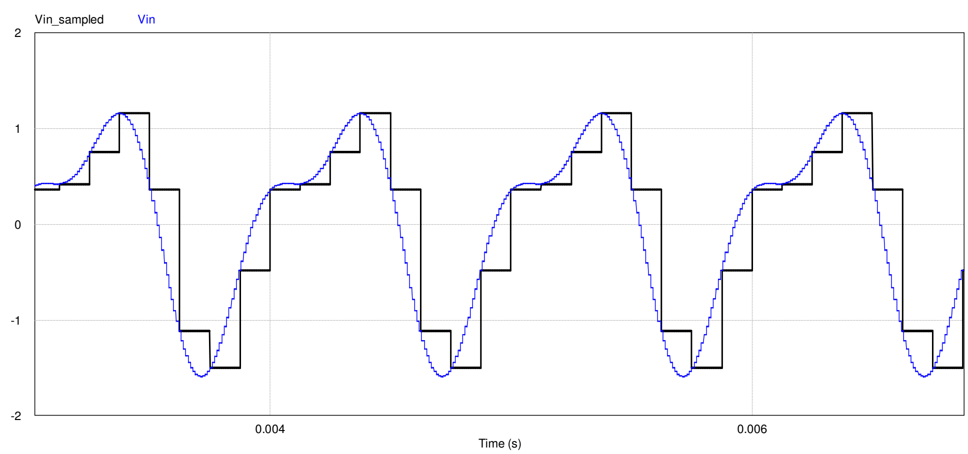

Fig. 2: The input signal and the 8-point sampled signal.

In order to compute equation (5) and (6) we need 8 values. Fig. 3 shows how to obtain 8 values in order to compute the DFT.

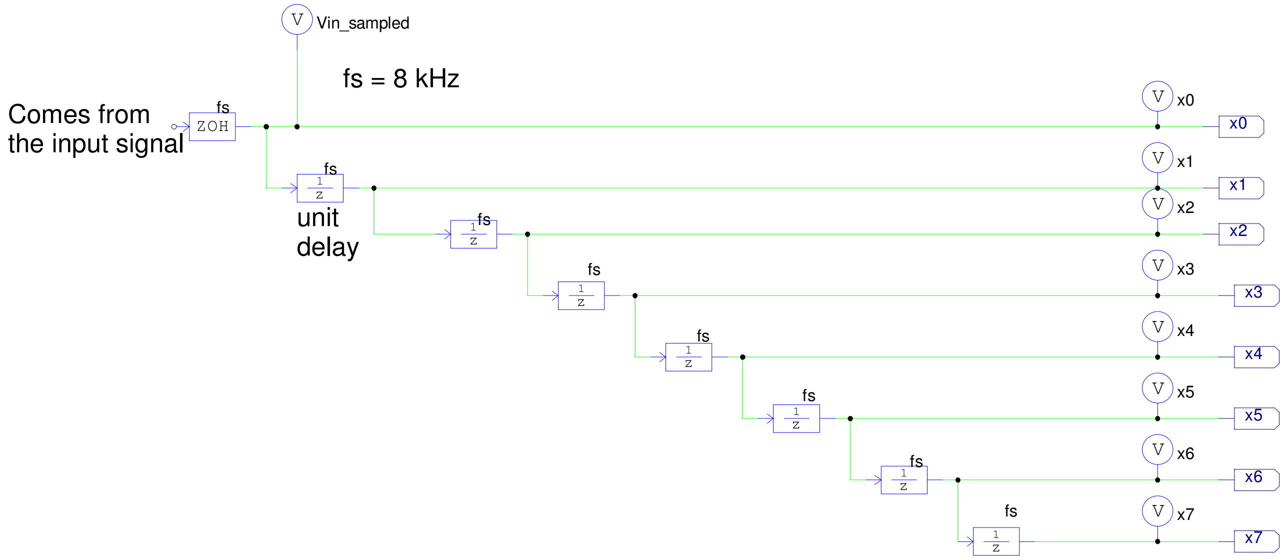

Fig. 3: How to obtain 8 values in order to compute the DFT.

We have 8 samples x(0), x(1), x(2), x(3), x(4), x(5), x(6) and x(7).

2.2. Computing the DFT for m = 1

Compute the DFT for m=1 means that we will calculate the amplitude values of the 1 kHz component which is hidden in our input signal. We have $m = 1$, $N = 8$ and n going from 0 to 7. With these values, equation (5) can be written as (11).

$$\begin{aligned}
X_{real}(m) &= \sum_{n=0}^{N-1} x(n)\left[\cos(2\pi nm/N)\right] = \\
X_{real}(1) &= \sum_{n=0}^{N-1} x(n)\left[\cos(2\pi n1/8)\right] = \\
X_{real}(1) &= x(0)\left[\cos(2\pi\cdot 0\cdot 1/8)\right] + x(1)\left[\cos(2\pi\cdot 1\cdot 1/8)\right] + x(2)\left[\cos(2\pi\cdot 2\cdot 1/8)\right] \\
&+ x(3)\left[\cos(2\pi\cdot 3\cdot 1/8)\right] + x(4)\left[\cos(2\pi\cdot 4\cdot 1/8)\right] + x(5)\left[\cos(2\pi\cdot 5\cdot 1/8)\right] \\
&+ x(6)\left[\cos(2\pi\cdot 6\cdot 1/8)\right] + x(7)\left[\cos(2\pi\cdot 7\cdot 1/8)\right]
\end{aligned} \tag{11}$$

Similarly, equation (6) can be written as

$$\begin{aligned}
X_{img}(m) &= \sum_{n=0}^{N-1} x(n)\left[(-1)\sin(2\pi nm/N)\right] = \\
X_{img}(1) &= \sum_{n=0}^{N-1} x(n)\left[(-1)\sin(2\pi n1/8)\right] = \\
X_{img}(1) &= x(0)\left[(-1)\sin(2\pi\cdot 0\cdot 1/8)\right] + x(1)\left[(-1)\sin(2\pi\cdot 1\cdot 1/8)\right] + x(2)\left[(-1)\sin(2\pi\cdot 2\cdot 1/8)\right] \\
&+ x(3)\left[(-1)\sin(2\pi\cdot 3\cdot 1/8)\right] + x(4)\left[(-1)\sin(2\pi\cdot 4\cdot 1/8)\right] + x(5)\left[(-1)\sin(2\pi\cdot 5\cdot 1/8)\right] \\
&+ x(6)\left[(-1)\sin(2\pi\cdot 6\cdot 1/8)\right] + x(7)\left[(-1)\sin(2\pi\cdot 7\cdot 1/8)\right]
\end{aligned} \tag{12}$$

Equations (11) and (12) are simulated according to the Fig. 4, which are also the equations (5) and (6).

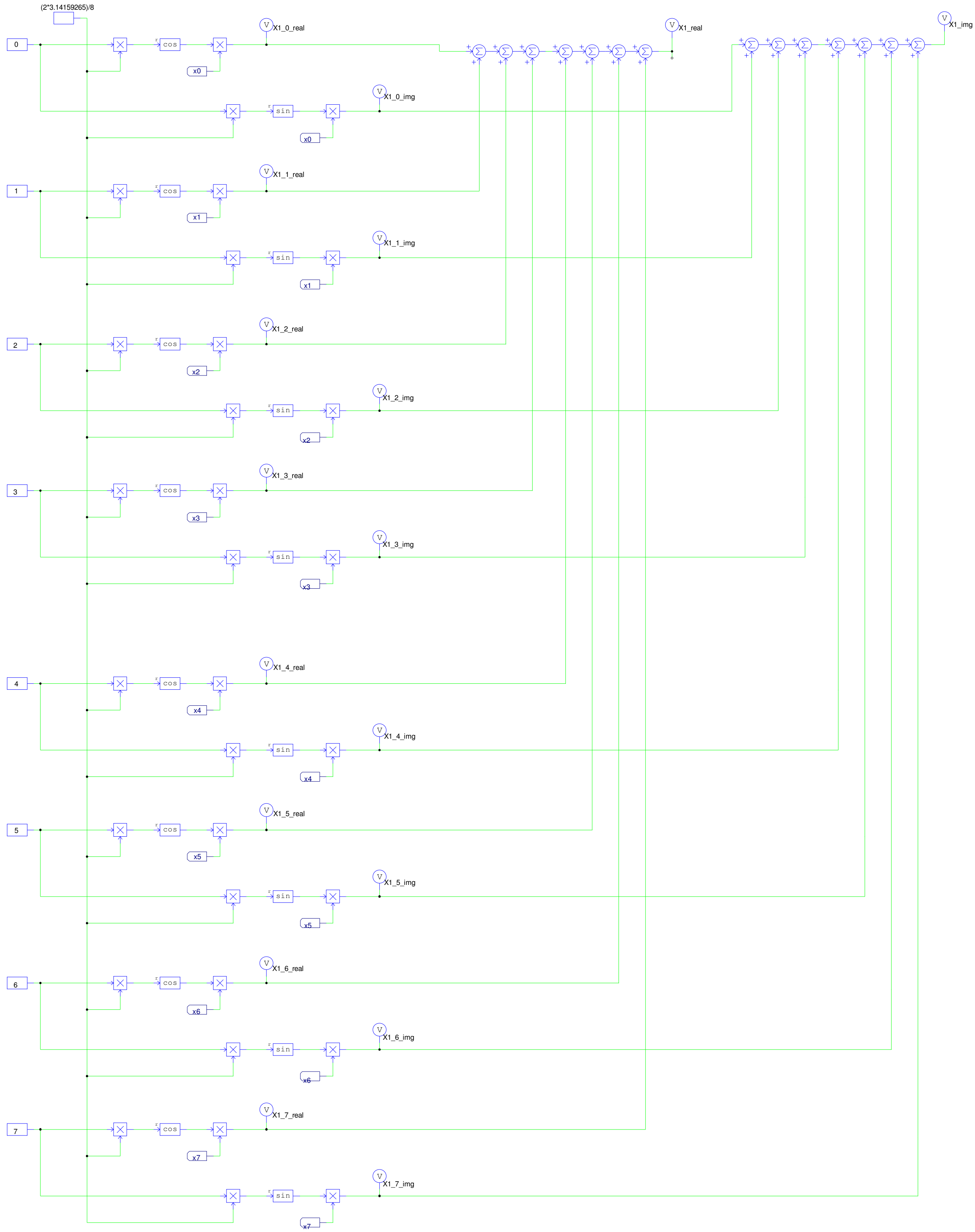

Fig. 4: Simulation of equations (5) and (6).

Now that we have the real and imaginary parts, the amplitude of 1 kHz components is given by (13).

$$X_{1magnitude} = \sqrt{X_{real}(1)^2 + X_{img}(1)^2} \tag{13}$$

The phase of the same component is given by (14).

$$X_{1phase} = \tan^{-1}\left(\frac{X_{img}(1)}{X_{real}(1)}\right) \qquad (14)$$

Fig. 5 presents how to simulate equations (13) and (14). The factor ¼ is inherent of the DFT, which is equal to N/2.

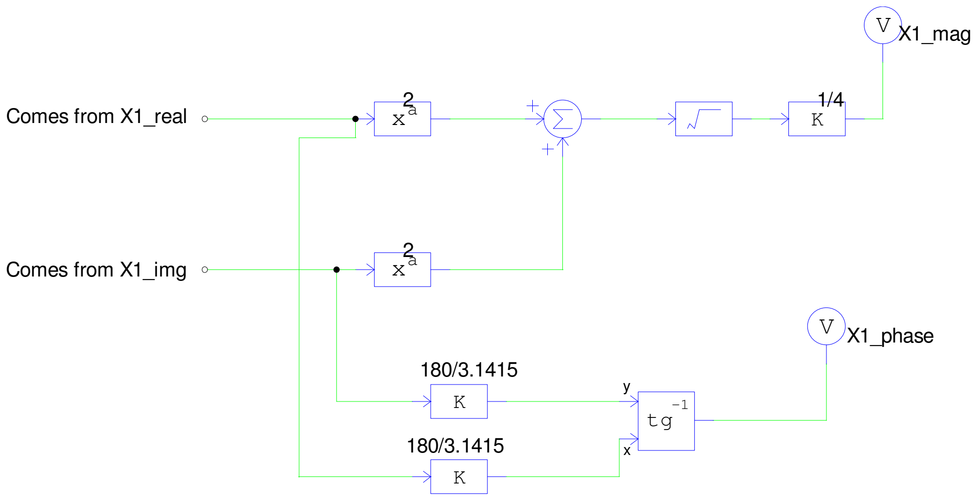

Fig. 5: how to simulate equations (13) and (14).

Fig. 6 presents the magnitude of the 1 kHz component calculated by means of the DFT. Its value is constant and very close to 1.125.

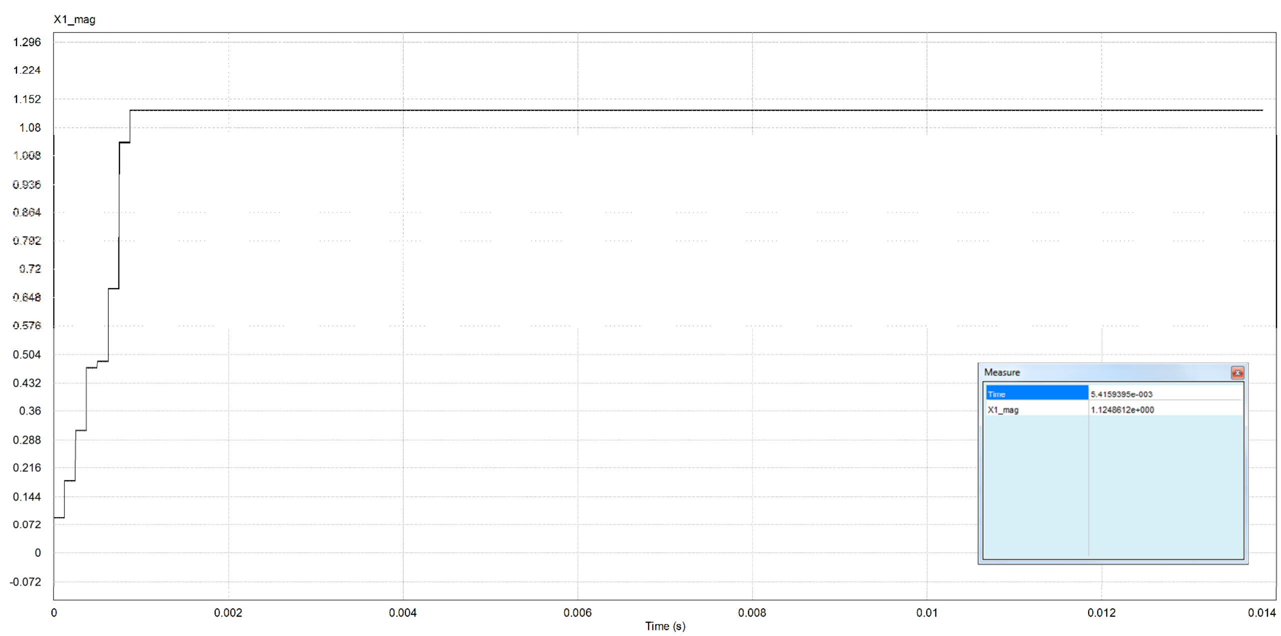

Fig. 6: Magnitude of the 1 kHz component calculated by means of the DFT.

Similar analysis is easily performed for m=2.

2.1. Computing the DFT for m =0

How about m= 0? Such value means the DC value of the input signal. By replacing m = 0 in equations (5) and (6), the results will be:

$$X_{real}(0) = \sum_{n=0}^{N-1} x(n) \tag{15}$$

And

$$X_{img}(0) = 0 \tag{16}$$

Therefore, in order to compute the DC value of an input signal, it is necessary just sum all the samples.

$$X_{0magnitude} = x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(5) + x(7) \tag{17}$$

The simulated circuit is shown in the following figure.

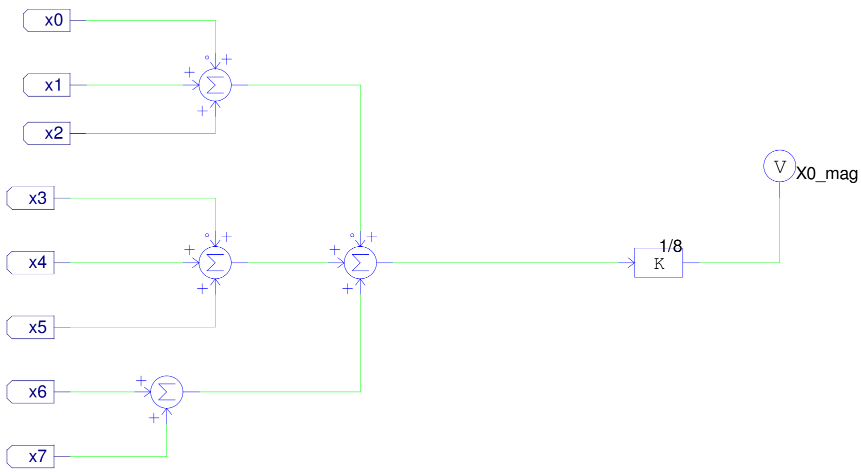

Fig. 7: How to compute the DC values of the input signal.

In our input signal the DC value is zero. But in order to verify the DFT efficacy, let's add 0.32V into the 1 kHz signal. Fig. 8 shows the new input signal and the DFT result for X(0). On the left is the new input signal and the average value of it, calculated by the simulator. On the right is the DFT result for X(0). Exactly the same.

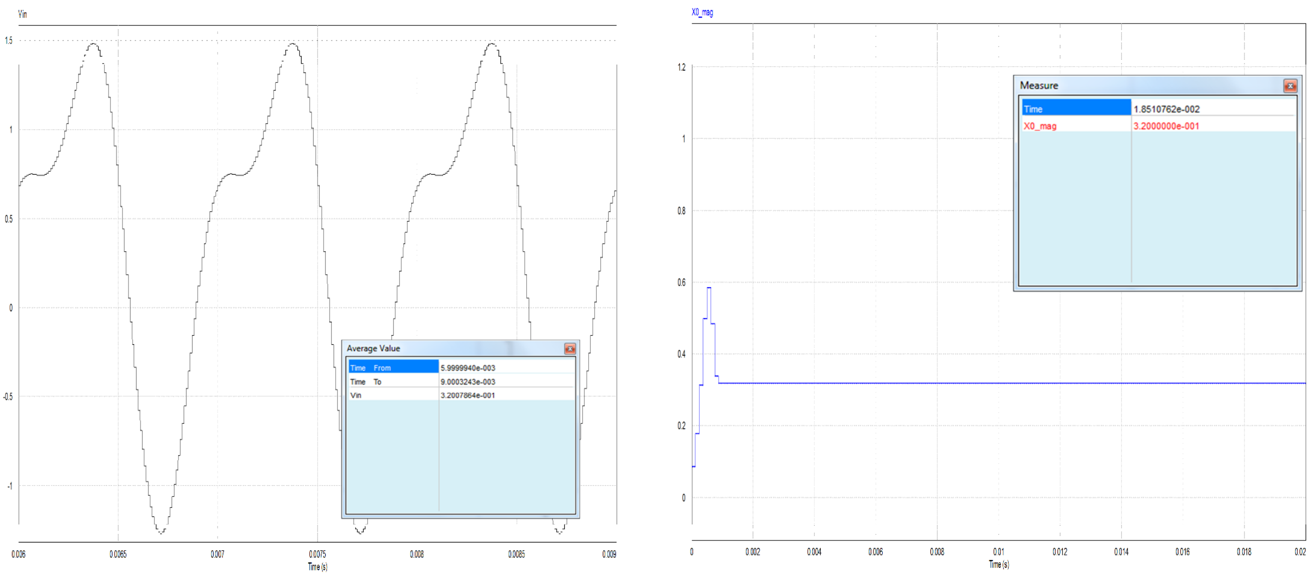

Fig. 8: New input signal and the DFT result for X(0).

3 The Fast Fourier Transform (FFT)

Solving *sines* and *cosine*s may be a burden for microcontrollers. As verified, the 8-point DFT makes use 8 *sines* and 8 *cosines*. Now imagine a 1024-point DFT. The Fast Fourier Transform appears as a solution to this inconvenience. The FFT is the same of DFT, but in a very simplified way. It is important to highlight that the FFT is the same thing of DFT, but performed in easiest way. The results are the same.

Here, the input signal is the same of the previous one.

3.1. Simulating the FFT equation

The FFT is given by (18).

$$X(m)=\sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nm}+W_{N}^{m}\sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nm} \tag{18}$$

Don't scare about it. Let's see part-by-part. Rewritten with distinguishing.

$$X(m)=\underbrace{\sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nm}}_{\text{for even values of } n}+\underbrace{W_{N}^{m}}_{\text{constant}}\underbrace{\sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nm}}_{\text{for odd values of } n} \tag{19}$$

The constant term is given by (20). The y and z are variable just to illustrate where to substitute them in the constant equation (20).

$$W_{z}^{y}=e^{-2j\pi y/z} \tag{20}$$

This constant is complex. So, we need to separate into two parts, real and imaginary, as follows (applying the Euler's relationship).

$$W_{z\_(real)}^{y}=\cos\left(\frac{2\pi y}{z}\right) \tag{21}$$

$$W_{\text{z\_(img)}}^{y} = -\sin\left(\frac{2\pi y}{z}\right) \tag{22}$$

Similar to the DFT methodology, for each interest *m*, the equation (19) must compute all values of *n*, from 0 to 7. For the sake of simplicity, let's write equation (19) as

$$X(m) = A(m) + W_N^m B(m) \tag{23}$$

Where

$$A(m) = \sum_{n=0}^{(N/2)-1} x(2n) W_{N/2}^{nm} \tag{24}$$

and

$$B(m) = \sum_{n=0}^{(N/2)-1} x(2n+1) W_{N/2}^{nm} \tag{25}$$

3.2. Computing the FFT equation for m = 1

For m = 1 and N = 8, equation (23) becomes

$$X(1) = A(1) + W_8^1 B(1) \tag{26}$$

equation (24) becomes:

$$A(1) = \sum_{n=0}^{3} x(2n) W_4^{n\cdot 1} \tag{27}$$

And equation (25) becomes:

$$B(1) = \sum_{n=0}^{3} x(2n+1) W_4^{n\cdot 1} \tag{28}$$

From equations (26), (27) and (28), the following constant must be computed, for real and imaginary parts.

$$\begin{cases} W_8^1 \\ W_4^0 \\ W_4^1 \\ W_4^2 \\ W_4^3 \end{cases} \tag{29}$$

The results are the following:

$$W^{1}_{8\_real} = \cos\left(\frac{2\pi \cdot 1}{8}\right) = 0.707 \tag{30}$$

$$W^{1}_{8\_\mathrm{img}} = -\sin\left(\frac{2\pi \cdot 1}{8}\right) = -0.707 \tag{31}$$

$$W^{0}_{4\_real} = \cos\left(\frac{2\pi \cdot 0}{4}\right) = 1 \tag{32}$$

$$W^{0}_{4\_\mathrm{img}} = -\sin\left(\frac{2\pi \cdot 0}{4}\right) = 0 \tag{33}$$

$$W^{1}_{4\_real} = \cos\left(\frac{2\pi \cdot 1}{4}\right) = 0 \tag{34}$$

$$W^{1}_{4\_\mathrm{img}} = -\sin\left(\frac{2\pi \cdot 1}{4}\right) = -1 \tag{35}$$

$$W^{2}_{4\_real} = \cos\left(\frac{2\pi \cdot 2}{4}\right) = -1 \tag{36}$$

$$W^{2}_{4\_\mathrm{img}} = -\sin\left(\frac{2\pi \cdot 2}{4}\right) = 0 \tag{37}$$

$$W^{3}_{4\_real} = \cos\left(\frac{2\pi \cdot 3}{4}\right) = 0 \tag{38}$$

$$W^{3}_{4\_\mathrm{img}} = -\sin\left(\frac{2\pi \cdot 3}{4}\right) = 1 \tag{39}$$

Fig. 9 presents the constant in the simulation file.

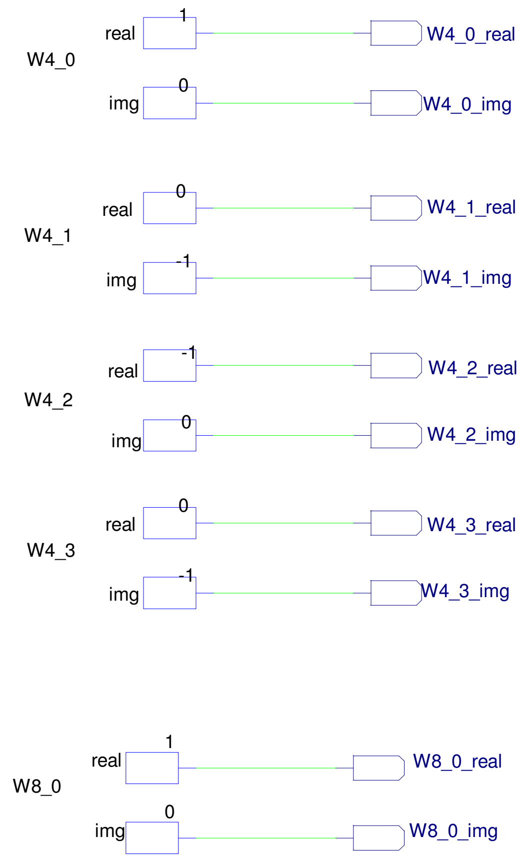

Fig. 9: The constants in the simulation file.

The equation (27) for real values is given by (40).

$$A(1)_{\text{real}} = \sum_{n=0}^{3} x(2n) W_4^{n \cdot 1} = x(0) W_{4\_real}^{0} + x(2) W_{4\_real}^{1} + x(4) W_{4\_real}^{2} + x(6) W_{4\_real}^{3} \tag{40}$$

Similarly, the equation (27) for imaginary values is given by (41).

$$A(1)_{\text{img}} = \sum_{n=0}^{3} x(2n) W_4^{n \cdot 1} = x(0) W_{4\_\text{img}}^{0} + x(2) W_{4\_\text{img}}^{1} + x(4) W_{4\_\text{img}}^{2} + x(6) W_{4\_\text{img}}^{3} \tag{41}$$

The equation (28) for real values is given by (42).

$$B(1)_{\text{real}} = \sum_{n=0}^{3} x(2n+1) W_4^{n \cdot 1} = x(1) W_{4\_\text{real}}^{0} + x(3) W_{4\_\text{real}}^{1} + x(5) W_{4\_\text{real}}^{2} + x(7) W_{4\_\text{real}}^{3} \tag{42}$$

Similarly, the equation (28) for imaginary values is given by (43).

$$B(1)_{\text{img}} = \sum_{n=0}^{3} x(2n+1) W_4^{n \cdot 1} = x(1) W_{4\_img}^{0} + x(3) W_{4\_img}^{1} + x(5) W_{4\_img}^{2} + x(7) W_{4\_img}^{3} \tag{43}$$

Fig. 10 presents how A(1) and B(1) both for real and imaginary parts are simulated.

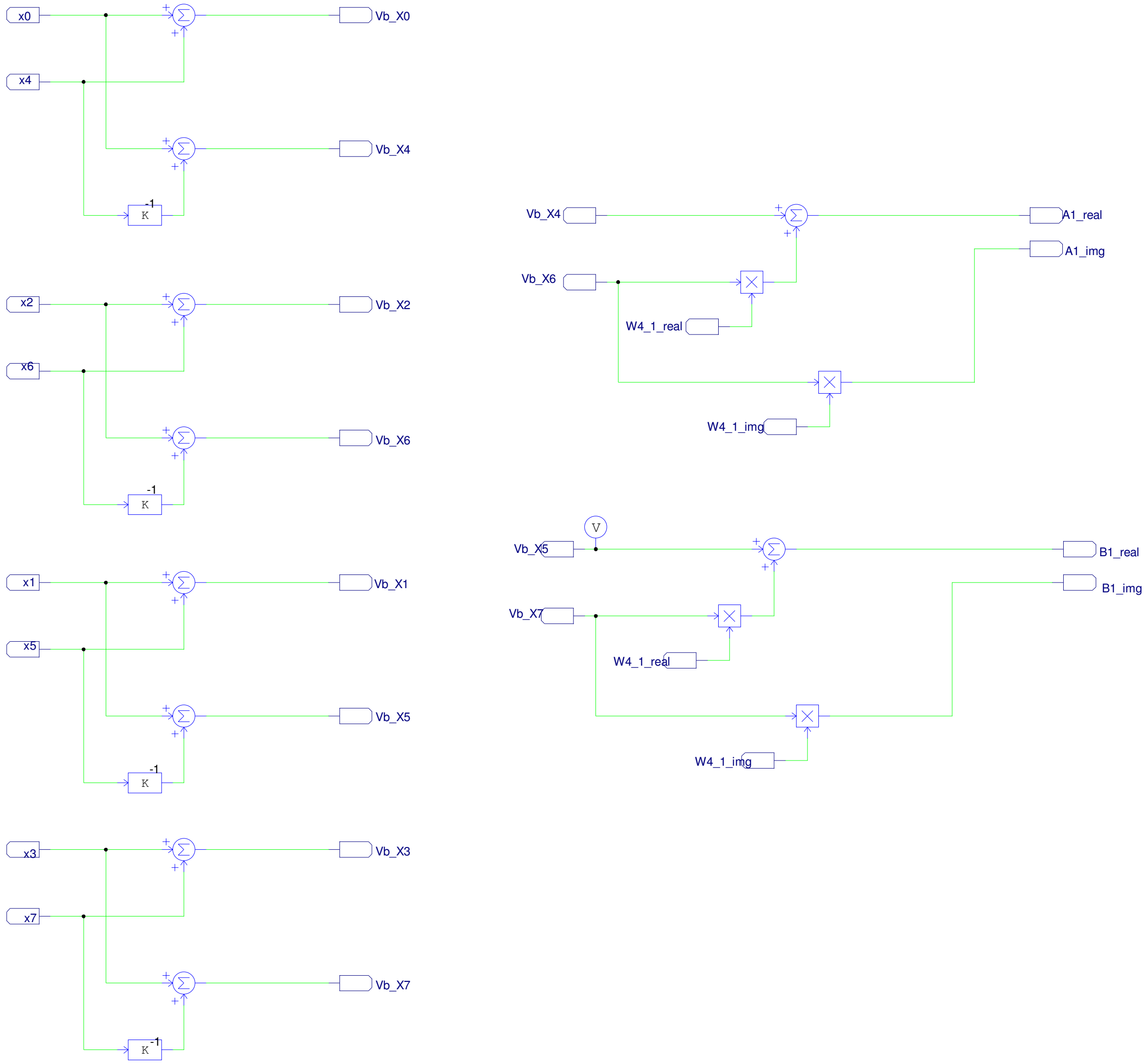

Fig. 10: How A(1) and B(1) both for real and imaginary parts are simulated.

Now we are ready to compute equation (23), rewrote here in (44) for simplicity.

$$X(m) = A(m) + W_N^m B(m) \tag{44}$$

But this equation is complex, so we need to write as it, given by (45).

$$X(m) = A(m)_{real} + jA(m)_{img} + \left(W_{N\_real}^{m} + jW_{N\_img}^{m}\right)\left(B(m)_{real} + jB(m)_{img}\right) \tag{45}$$

Which results in (46) and (47).

$$X(m)_{\mathrm{real}} = A(m)_{real} + W_{N\_real}^{m} B(m)_{real} - W_{N\_img}^{m} B(m)_{img} \tag{46}$$

$$X(m)_{\mathrm{img}} = A(m)_{img} + W_{N\_real}^{m} B(m)_{img} + W_{N\_img}^{m} B(m)_{real} \tag{47}$$

For m $=1$ and N$=$ 8, these equations results in (48) and (49).

$$X(1)_{\mathrm{real}} = A(1)_{real} + W_{8\_real}^{1} B(1)_{real} - W_{8\_img}^{1} B(1)_{img} \tag{48}$$

$$X(1)_{\text{img}} = A(1)_{img} + W_{8\_real}^{1} B(1)_{img} + W_{8\_img}^{1} B(1)_{real} \quad (49)$$

The magnitude of X(1) is given by

$$X_{1magnitude} = \sqrt{X_{real}(1)^2 + X_{img}(1)^2} \quad (50)$$

Fig. 11 presents how to calculate equations (48), (49) and (50).

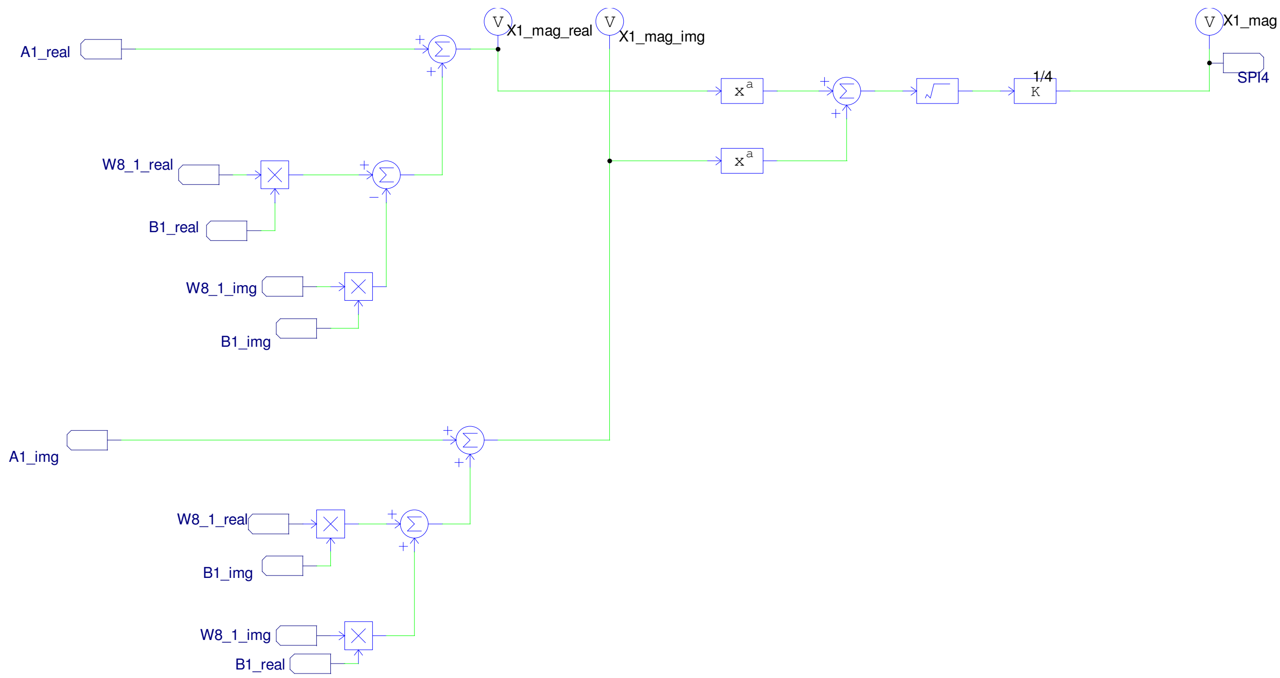

Fig. 11: How to calculate equations (48), (49) and (50).

Fig. 12 presents the X(1) magnitude. It is the same 1.125 found in the input signal.

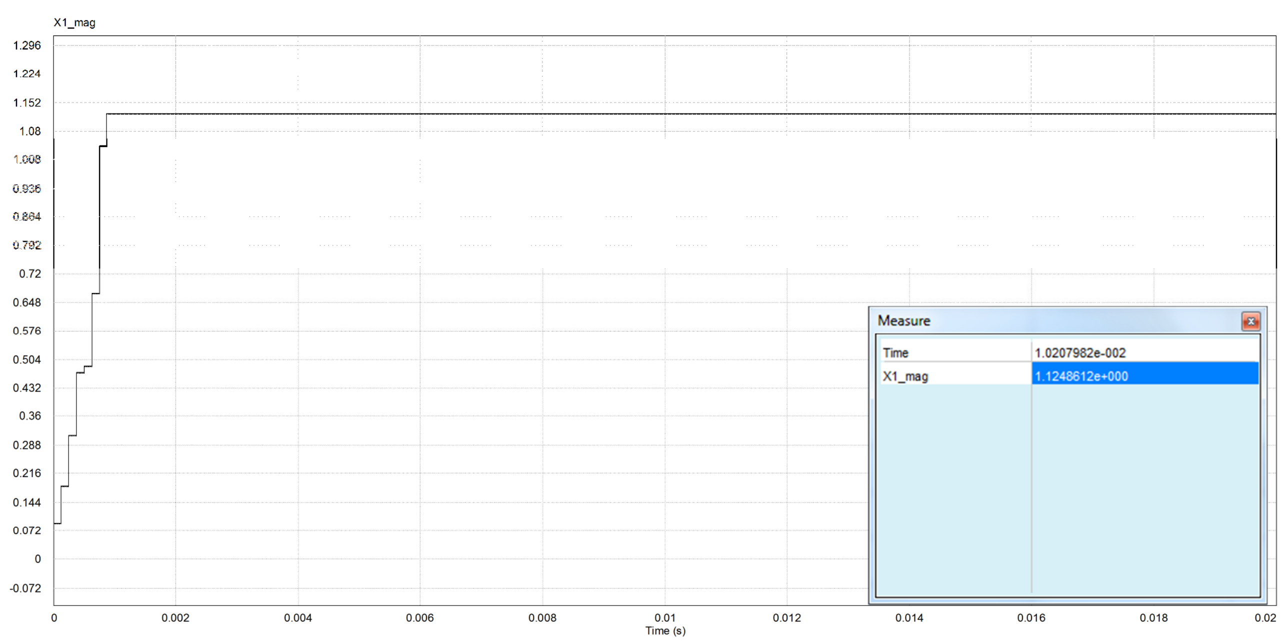

Fig. 12: X(1) magnitude.

4 Experimental results

The FFT was tested in the DSC 28335 microcontroller.

Fig. 13 presents the sine waves with 1kH and 2 kHz, as well as, the input signal. The frequencies and RMS values are also presented. The staircase visualization is due to the DAC used. The DAC is used in order to see a digital variable in the oscilloscope and it does not influence on the FFT analysis.

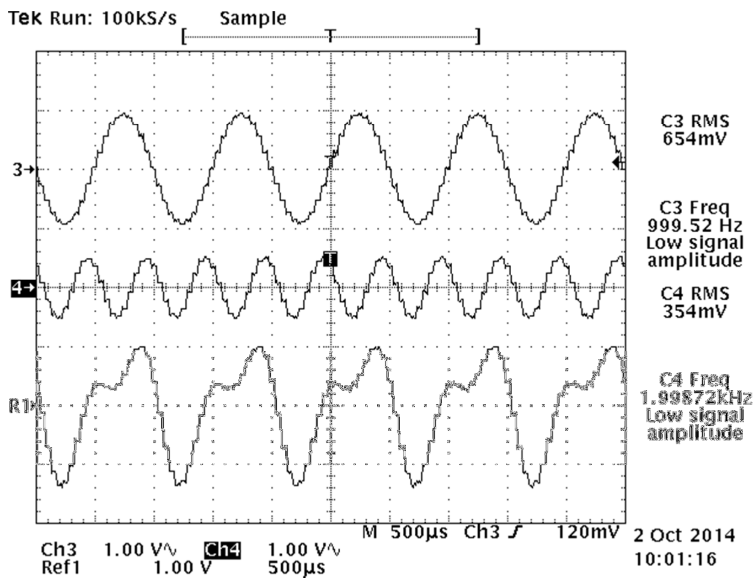

Fig. 13: The sine waves with 1kH and 2 kHz, as well as, the input signal.

Fig. 14 presents the magnitude of 1 kHz component from the input signal calculated by the FFT. Its average value is 654 mV, the same of the input signal 1 kHz component. (the value was divided by srqt(2) because the FFT gives the amplitude). The non-constant behavior is mainly due to the DAC.

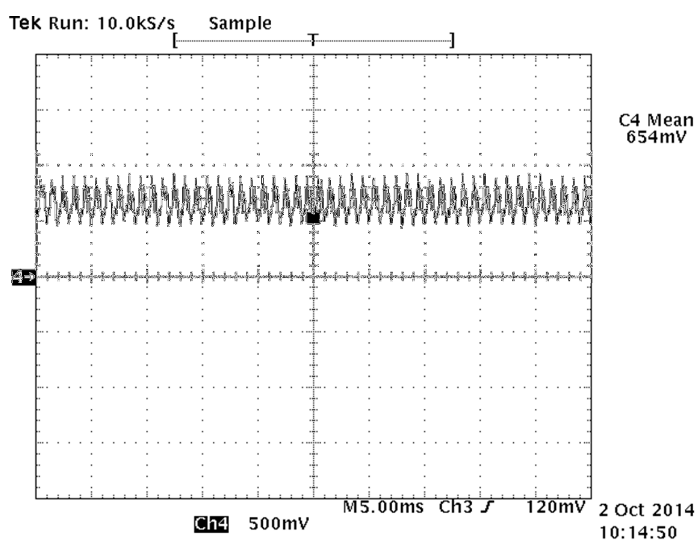

Fig. 14: The magnitude of 1 kHz component from the input signal calculated by the FFT

5 Final considerations

This report presented a simplified way to perform DFT and FFT. The objective was make the readers understand how the DFT and FFT work. Computing the DFT and FFT using 8 point gives poor results. The recommended is

512 or more points. Here, the phenomena like Leakage, Windows, Scalloping Loss, Zero Padding, etc were not took account. Reference [1] is one of the best books about Digital Signal Processing and which this report is based on.

6 References

[1] Understanding Digital Signal Processing (3rd Edition). Richard G. Lyons (Author)