Construction of a window function for estimating the parameters of sinusoidal signals with non-harmonic frequencies

E A Altman, T V Vaseeva
Omsk State Transport University, 35, Marx ave., Omsk, 644046, Russia
tvaseeva@gmail.com

Abstract. Discrete Fourier Transform (DFT) allows you to determine the discrete spectrum of a signal. Due to the presence of its high-speed implementation, called Fast Fourier Transform (FFT), this transform is widely used in digital signal processing (DSP). Most DSP tasks that deal with analogy signal and spectrum adapt the DFT to find the signal spectrum between harmonics. One of the most commonly used ways of such adaptation – the use of window functions. The analysis of standard window functions (Kaiser, etc.) showed that their direct application to solving the problem of estimating the parameters (frequency, amplitude, and phase) of nonharmonic sinusoidal components of signals leads both to the need for additional corrections of the estimation results and to additional errors in determining the phase of the signal. The paper proposes a method that allows building a window function without the indicated drawbacks based on standard window functions. The essence of the method is to transform the standard window function so that its spectrum does not contain imaginary components, and the amplitude of the fundamental harmonic would be equal to 1. The results of modeling the proposed method on the example of the Kaiser window showed that the phase estimate of the nonharmonic components of the spectrum using the obtained window function, in contrast to the estimate using standard window functions, is not displaced.

1. Introduction
Spectral analysis using digital signal processing is widely used in solving various problems of science and technology. In addition to the known applications and advantages of spectral analysis and digital signal processing separately, their combination brings additional advantages associated with the discrete Fourier transform (DFT). This is, firstly, a computationally efficient procedure for fast Fourier transform (FFT), which significantly expanded the field of application of the DFT. Secondly, there are various properties of the DFT itself, associated primarily with the symmetry of this transformation. For example, the harmonics of the signal spectrum calculated with the help of DFT are not affected by the so-called window functions arising from the time limitation of the signal [1-4]. At the same time, if it is necessary to work with signals whose spectrum contains sinusoids with frequencies different from harmonic frequencies, there are difficulties in processing and analyzing such signals and the need to take more careful consideration of the digital nature of the signal – its time constraints, sampling and, in some cases, quantization.

In the problem considered in the article, it is necessary to estimate the parameters of the sinusoidal signal, the frequency of which does not coincide with the frequencies of harmonics or, what is the same, is not a multiple of the frequency of the fundamental harmonic.

The parameters of such a harmonic are not found with the help of DFT, and they cannot be found directly by applying to the sequence of the digital signal a formula constructed in the likeness of the formula for the analog case.
Such a problem is encountered in various fields of technology: when analyzing the spectra of signals in power supply networks, when evaluating the frequencies of the received signal in radio engineering, etc. The main method of its solution is to find the parameters of sinusoid by interpolating the values that are next to the signal harmonics obtained with the help of FFT.

The method of interpolating harmonic parameters gives good practical results, however, interpolation increases the error in estimating the parameters [5-8]. In our work, we will consider a method that allows you to directly calculate the required parameters. This method is of both theoretical interest as a reference method for estimating the parameters of sinusoids, and practical interest, making it possible to increase the accuracy of such an estimate.

2. Problem overview

Suppose we need to estimate the amplitude $A$ and phase $\varphi$ of a sinusoidal signal $s$ versus time $t$ at a frequency $\omega$:

$$s(t) = A \cdot \sin(\omega \cdot t + \varphi)$$

(1)

After sampling the signal $s(t)$, a time-limited sequence $(x_t)$ is obtained, where $t = 0...N - 1$. By analogy with the Fourier transform, we can write a formula for estimating the parameters of the harmonic:

$$y_p = \sum_{t=0}^{N-1} x_t \cdot e^{-j \frac{t \omega}{N}}$$

(2)

where $y_p$ – is a complex number whose trigonometric form shows the amplitude and phase of the sinusoid at frequency $p$. The frequency $p$ here is set in relative units, $p = 1$ corresponds to the frequency of the sinusoid, the period of which is equal in duration to $N$ samples of the signal $x_t$. This analogy would be valid only for frequencies with integer $p$. The calculation of these formulas for integer $p$ from 0 to $N - 1$ is the DFT.

For non-integer $p$, it is necessary to take into account the influence of the window function on the signal spectrum. By limiting the signal in time, we multiply it by some function, which is called a rectangular window, whose value is 1 for the time for which there are $x_t$ samples and is equal to 0 otherwise. After that, we perform further actions with the signal $s'(t)$.

$$s'(t) = s(t) \cdot a(t)$$

(3)

where $a(t)$ – is a window function.

As is known from mathematics, the Fourier transform ($F$) of the product of functions is equal to the convolution ($*$) of the Fourier transforms of these functions:

$$F(s(t) \cdot a(t)) = F(s(t)) \ast F(a(t)).$$

(4)

The Fourier transform of a signal containing one sinusoid will be the unit function times the amplitude of the sinusoid. A unit function is equal to one at a point equal to the frequency of the sinusoid and is equal to zero in other cases.

The convolution of a unit function with a Fourier transform of a window function would be a Fourier transform of a window function, shifted by a frequency at which the unit function takes a value equal to 1, i.e. to the frequency of the original sinusoid.

Consider the Fourier transform (spectrum) of a rectangular window function (Fig. 2).

As you can see, the spectrum of the rectangular window function takes on a value equal to zero at frequencies corresponding to neighboring harmonics. Due to this, the effect of the window function does not affect if the analysis is performed only for harmonics, i.e. for sinusoids which frequency is a multiple of the fundamental frequency.

The shape of the spectrum of the rectangular window function between the frequencies of the harmonics greatly complicates the analysis of the spectrum at such frequencies – this spectrum changes rapidly and slowly decays with distance from the maximum value.

To improve the properties of the spectrum of the window function, various window functions other than rectangular are used. For example, Fig. 3 and Fig. 4 show the Kaiser window and its spectrum.
As you can see from the figure, this window has better properties compared to a rectangular window. Its spectrum is smoother and most of its energy in the spectral region is concentrated in the main lobe (part of the spectrum between the frequencies adjacent to the central harmonics).

At the same time, as the results of modeling algorithms for estimating the parameters of a sinusoid show, such a window also does not allow using formula (2) for sinusoids with nonharmonic frequencies. To understand why this happens, consider the first few frequencies in the spectrum of this window. Let's execute the following Python program (see [9] for the complete program code for the article examples) (Fig. 5):

```python
import numpy as np
n_point = 1024
kaiser_win = np.kaiser(n_point, 10)
kaiser_spec = np.fft.fft(kaiser_win)
print("Regular Kaiser Window", kaiser_spec[0:5])
```

As a result, we get (Fig. 6):

```
Regular Kaiser Window [4.00130715e+02+0.00000000e+00j -2.54258492e+02 -7.80057732e-01j 5.60777969e+01+3.44093370e-01j -1.56357266e+00-1.43913489e-02j 4.65544597e-02+5.71337855e-04j]
```

We see that the elements of the window spectrum have small imaginary components. Even though they are much less than the amplitude, they nevertheless introduce an error in signal analysis, primarily in determining the phase of a sinusoid.
In addition, the amplitude of the zero component of the window spectrum is different from one. This circumstance does not introduce an error in the calculation, however, it requires additional scaling operations when analyzing the signal.

Consider the method of constructing a window function that allows, with the minimum possible error and the most convenient way, to find the parameters of sinusoids with non-harmonic frequencies.

3. The proposed method

The first task of constructing the required window is to obtain a window, the spectrum would be real. To obtain such a window, we will use the known properties of the spectrum of a real signal, as well as the symmetry of the direct and inverse Fourier transforms.

As a result of the discrete Fourier transform of the real signal, a complex conjugate spectrum is obtained. In this case, the center of symmetry will be shifted relative to the center of the spectrum - for a zero harmonic there is no harmonic symmetric, and the symmetry will be relative to the $\frac{N}{2} - 1$ harmonic, where $N$ is the number of samples in the signal (or the number of harmonics).

Based on the symmetry of the direct and inverse Fourier transforms, for the spectrum of the window function to be real, it must have this kind of symmetry. The easiest way is to do it - to remove from the original signal the sample symmetrical to zero, i.e. remove the last countdown of the window.

To obtain a window with the required number of points and a real spectrum, you can generate a window with some points by one more in the usual way, and then discard the last extra point. In Python, you can get such a window as follows (Fig. 7):

```python
symmetric_win = np.kaiser(n_point + 1, 10)[0:n_point]
symmetric_spec = np.fft.fft(symmetric_win)
print("Symmetrical window ", symmetric_spec[0:5])
```

![Figure 7. Spectrum analysis of a symmetrized window.](image)

The result of executing the above code will be (Fig. 8):

```
Symmetrical window [4.00521494e+02+0.j -2.54277648e+02+0.j
5.58915991e+01+0.j -1.53728661e+00+0.j 4.63682850e-02+0.j]
```

![Figure 8. The spectrum of a symmetrized window.](image)

As you can see, all imaginary components of the spectrum of the obtained window are zero. The same result is obtained if, instead of a function to generate a Kaiser window, functions for other window functions are used.

The problem with signal scaling when using window functions can be solved by normalizing the window so that the zero components of the spectrum of the window function are equal to one. Based on the definition of the discrete Fourier transform, the zero component is the sum of all samples of the window. For it to become equal to one, you need to divide all the samples of the window by their sum.

In Python, this can be done as follows (Fig. 9):

```python
normal_win = symmetric_win / np.sum(symmetric_win)
normal_spec = np.fft.fft(normal_win)
print("Normalized window ", normal_spec[0:5])
```

![Figure 9. Normalized Window Spectrum Analysis.](image)

As a result, we get a window with the following spectrum (Fig. 10):

```
Normalized window [1.00000000e+00+0.j
-6.34866423e-01+0.j 1.39547065e-01+0.j
-3.83821252e-03+0.j 1.15769780e-04+0.j]
```

![Figure 10. Normalized Window Spectrum.](image)

As a result, we obtained a method for forming a window function with the properties of its spectrum that interest us: it does not contain imaginary parts and its zero components are equal to one. In the same way, you can get window functions with specified properties based not only on the Kaiser window but also on any other standard windows.
We can use the obtained windows to estimate the parameters of sinusoidal signals of both harmonic and non-harmonic frequencies directly by formula (2).

4. Experimental results
Experimental verification of the proposed method for constructing the window function and estimating the parameters of the sinusoidal signal with its help was carried out as follows. The values of the amplitude, frequency, and phase of the sinusoid were randomly set. A signal with the specified parameters was generated, a window was superimposed on it. The formula (2) was used to estimate the amplitude and phase of the signal at a known frequency.

The source code of the simulation program is shown below:

```python
n_test = 1000
freq_s = np.random.rand(n_test) + 3
amp_s = np.random.rand(n_test)
phi_s = np.random.rand(n_test) * np.pi
eval = np.zeros(n_test, dtype=complex)
for i_test in range(n_test):
signal = amp_s[i_test] * np.cos([2 * np.pi * freq_s[i_test] * i / n_point - phi_s[i_test]
    for i in range(n_point)])
signal_windowed = signal * normal_win
 eval[i_test] = 2 * np.sum([signal_windowed[i] * np.exp(1j * freq_s[i_test] * i * 2 * np.pi / n_point)
    for i in range(n_point)])
print(max(np.abs(amp_s-np.abs(eval))))
print(max(np.abs(phi_s-np.angle(eval))))
```

![Figure 11](image1.png)

Figure 11. Experimental verification of the proposed method.

As a result of the verification, the following results were obtained, which testify to the correctness of the proposed method.

4.5567255301e-05
5.21748351825e-05

![Figure 12](image2.png)

Figure 12. Results of verification of the proposed method.

5. Results and conclusions
The proposed method of forming a window function well solves the problem posed – determining the parameters of sinusoidal signals with non-harmonic frequencies by direct calculation. It can also be used in the case when the frequency of the signal is not known in advance. In this case, you need to use the optimization procedure to find the maximum amplitude value in a given frequency range. The proposed modification of window functions is likely to be useful in solving other problems in which window functions are used.

6. References
[1] Altman E 2008 Optimization of the computational scheme of the fast Fourier transform Omsk Scientific Bulletin pp 1 (64) 149–151
[2] Ren Z, Wang B 2010 Estimation Algorithms of Harmonic Parameters Based on the FFT, 2010 Asia-Pacific Power and Energy Engineering Conference pp 1–4 DOI: 10.1109/APPEEC.2010.5448836.
[3] Altman E 2017 Construction of suboptimal models of FFT algorithms based on base 2 Information and telecommunication systems and technologies pp 409–411
[4] Altman E, Vaseeva T and Aleksandrov A 2019 Evaluation of the complexity of the FFT algorithm in modeling its operation *Innovations in information technologies, mechanical engineering and motor transport* pp 116–118

[5] Duda K 2011 DFT Interpolation Algorithm for Kaiser-Bessel, and Dolph–Chebyshev Windows *IEEE Transactions on Instrumentation and Measurement* vol 60, no. 3 pp 784–790 March DOI: 10.1109/TIM.2010.2046594.

[6] Wang Z, Li Y and Zhang H 2019 Harmonic Detection Method Based on Windowed Double-Spectrum Interpolation FFT *ICRIS* pp 425–427 DOI: 10.1109/ICRIS.2019.00111.

[7] Hsue W 2020 Eigenvectors of Ordinary, Generalized, Centered and Offset Discrete Fourier Transforms Based on Lookup Table Methods: Efficiency and Approximation Uses *IEEE Transactions on Signal Processing*, vol. 68 pp 1776–1791 DOI: 10.1109/TSP.2020.2976821.

[8] Vaseeva T, Altman E and Aleksandrov A 2020 Comparative study of harmonic and interharmonic methods for evaluating signals *Innovative projects and technologies in education, industry and transport*, pp 51–57

[9] Altman E 2021 Window normalizations. URL: https://github.com/AltmanEA/annex/tree/main/NormWin.