Accurate Frequency Estimation using DFT and Artificial Neural Networks

Andrea-Amalia Minda
Department of Engineering Science,
Babes-Bolyai University
Cluj-Napoca, Romania
andrea.minda@ubbcluj.ro

Daniela-Giorgiana Burtea
Doctoral School of Engineering, Babes-Bolyai University
Cluj-Napoca, Romania
daniela.burtea@stud.ubbcluj.ro

Gilbert-Rainer Gillich
Department of Engineering Science,
Babes-Bolyai University
Cluj-Napoca, Romania
gilbert.gillich@ubbcluj.ro

Cristian Tufisi
Department of Engineering Science,
Babes-Bolyai University
Cluj-Napoca, Romania
cristian.tufisi@ubbcluj.ro

Nicoleta Gillich
Department of Engineering Science,
Babes-Bolyai University
Cluj-Napoca, Romania
nicoleta.gillich@ubbcluj.ro

Zeno-Iosif Praisach
Department of Engineering Science,
Babes-Bolyai University
Cluj-Napoca, Romania
zeno.praisach@ubbcluj.ro

***Abstract*—In the case of using standard methods for estimating the frequencies of short signals, large errors can be obtained due to the large distance between the spectral lines. To solve the problem, we propose a learning-based approach to identifying frequencies at an inter-line position in the spectrum. The proposed approach uses data obtained by applying the DFT to a sinusoidal signal generated with known frequency and amplitude and of different lengths. In this way, the highest values of the amplitude in the spectrum are placed on a curve close in shape to that of the sinc function. To always have three points on the main lobe, we double the length of the signal by zero-padding. For each signal length, the amplitudes of the three points on the main lobe are determined, which is the input for training the ANN. The distance between the frequency indicated on the spectral line of the left point and the generated frequency is used as the target. The numerical results show that the proposed approach exceeds the accuracy of most current methods, the percentage errors resulting from the tests being in the order of thousands. By using normalized training data, the resulting network can estimate the correction time regardless of the value of the frequency or amplitude of the evaluated signal.**

Keywords—frequency estimation, zero padding, Discrete Fourier Transform, Artificial Neural Network, sinc function

I. INTRODUCTION

Accurate frequency estimation is a necessity in several engineering domains, such as communications [1], radar or sonar [2], detection of mechanical or structural faults [3]. It is very important also in several applications in the medical field, where the monitoring of the frequency changes of some biosignals acquired from the human body have a decisive role in diagnosis [4].

Standard frequency estimation methods are the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT). These methods work well for numerous applications but can fail if short signals with low frequency are analyzed because of the rough resulting frequency resolution. The large distance between two consecutive spectral lines makes the chance that the position of a spectral line matches the actual frequency to be small. A consequence is the so-called spectral leakage phenomenon.

Different methods to increase the accuracy of the frequency estimate are proposed in the literature. One of them is finding the maximum of the curve that crosses two or three points in the DFT spectrum found by interpolation [5-12]. An alternative interpolation method is zero-padding [13]. A comprehensive review is given in [14].

More advanced methods imply analyzing the spectra of the signal with different time lengths. The advantage of shortening the signal consists in changing the position of the spectral lines and thus obtaining denser lines in the spectrum by overlapping numerous spectra. The trim-to-fit strategy involves adjusting the length until the amplitude at one of the spectral lines presents a maximum. This means the leakage is minimal, and the real frequency is found [15]. Other interpolation methods imply selecting three magnitudes obtained from different spectra obtained by shortening the time length [16, 17].

Despite the offered advantages, actual methods are either not sufficiently accurate, imply high computational costs, or cannot be automated. The method proposed herein is simple and robust, ensures high accuracy, and requires no human intervention. It can be described as follows. First, we develop a behavioral model by finding the amplitudes of three points of the DFT of a zero-padded harmonic signal with different time lengths. In the meantime, we calculate the difference between the generated and the obtained frequency. The data is normalized and trained with an Artificial Neural Network (ANN), which is then used to correct the frequencies estimated for measured signals.

II. THEORETICAL BACKGROUND

Let us consider the discrete-time sinusoidal signal with the initial phase angle zero as a sampled sequence containing

$$a[n] = A\sin(2\pi f_R \frac{n}{f_S}) \quad n = 0...N \tag{1}$$

In the above equation, A is the amplitude, f_R is the frequency of the signal, f_S is the sampling frequency, N is the number of acquired samples, and n is the index of the samples.

When calculating the DFT, we obtain a set of N frequency-amplitude pairs. The frequencies are displayed on equidistantly distributed spectral lines. The distance between two spectral lines, known as frequency resolution, is

$$\Delta f = \frac{f_S}{N} = \frac{1}{t} \tag{2}$$

For each spectral line $k = 1\ldots N$, the DFT of the signal $a[n]$ calculates the amplitude:

$$A_k = \sum_{n=0}^{N-1} a[n] \cdot e^{-j(2\pi/N)nk} \quad (3)$$

where j^2=-1.

So, following the Euler formula, an N sample signal $a[n]$ can be decomposed into a set of cosine waves with the amplitudes contained in the real part $Re A_k$ and a set of sine waves with the amplitudes in the imaginary part $Im A_k$.

$$Re\, A_k = \sum_{n=0}^{N-1} a[n] \cos\left(\frac{2\pi}{N} nk\right) \quad (4)$$

$$Im\, A_k = -\sum_{n=0}^{N-1} a[n] \sin\left(\frac{2\pi}{N} nk\right) \quad (5)$$

Hence, the amplitude displayed at the k-th spectral line of the DFT is calculated as

$$|A_k| = \sqrt{(Re\, A_k)^2 + (Im\, A_k)^2} \quad (6)$$

If the signal length is a multiple of the signal period T, all amplitudes in the spectrum are zero except the spectral line which fits the real frequency f_R of the signal. Else, the signal cannot be constructed with one sinusoid, and therefore more sinusoids with different amplitudes are necessary for its construction. In consequence, the spectrum will contain more lines with non-zero amplitudes. An example is shown in Fig. 1, where the red squares indicate the amplitudes A_{k-1}, A_k, and A_{k+1} located on the spectral lines k-1, k, and k+1, respectively. With k we denoted the spectral line that is closest to the inter-line position on which the real frequency f_R should be displayed. The amplitude of the sinusoidal signal is 1.

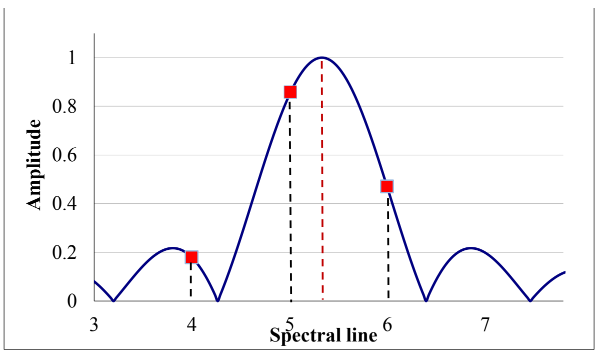

Fig. 1. The three points in the DFT spectrum of the sine signal

As it can be observed in Fig. 1, the calculated amplitudes approximately belong to a *sinc* function that has the maximum amplitude equal to the signal amplitude. Small differences occur because the *sinc* function is obtained after neglecting small quantities [18]. This function has a main lobe extended on an interval equal to twice the frequency resolution and lateral lobes extended on intervals equal to the frequency resolution. There are two spectral lines in the main lobe, and one line in each lateral lobe.

Since the proposed method involves the use of amplitudes belonging to the main lobe, we must ensure that the points involved meet this condition. To fulfill this condition, we lengthen the signal by zero-padding and so we obtain a double number of points, all belonging to the same *sinc* function. The dots are represented by green squares in Fig. 2. It can be observed that half of the green squares overlap the red squares.

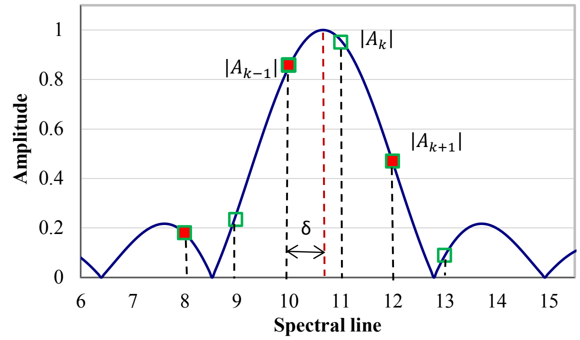

Fig. 2. The denser points in the DFT spectrum of the zero-padded sine signal

Now, we know the maximizer A_k in Fig. 2 and its two neighbors A_{k-1} and A_{k+1} belong to the main lobe. We can also calculate the distance between the first point, namely the spectral line k-1, and the spectral inter-line where the real frequency f_R is located. We call this distance correction term and denote it δ. The relation between the real frequency f_R and the frequency estimated at line k-1 is given by

$$f_R = (k-1+\delta)\Delta f = k_R \Delta f \quad (7)$$

We can simplify (7) by Δf, and thus obtain

$$k_R = k - 1 + \delta \quad (8)$$

Hence, the correction term result as

$$\delta = \frac{f_R - f_{k-1}}{\Delta f} \quad (9)$$

In (9), the term f_{k-1} represents the frequency estimated at the spectral line k-1, thus that of the frequency that has the amplitude A_{k-1}.

III. Creating the Database to Train the ANN

The correction term δ can be obtained by simulation involving a signal with known frequency f_R and amplitude A. To this aim, we generated a signal with A=1 and f_R=5 Hz, involving N=2149 samples by f_S=1000 Hz. This signal is zero-padded to double its length and consequently the number of spectral lines.

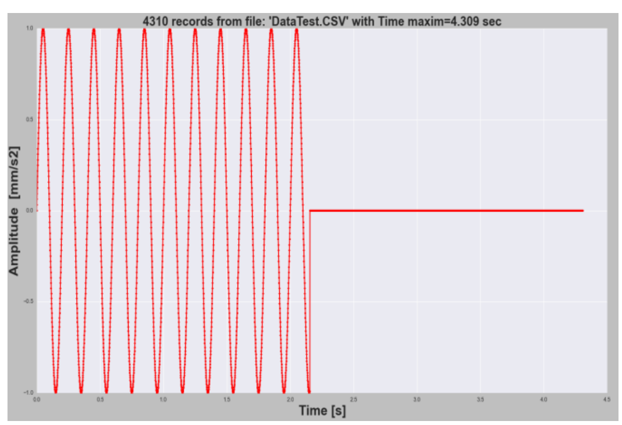

Fig. 3. The generated sine signal after zero-padding

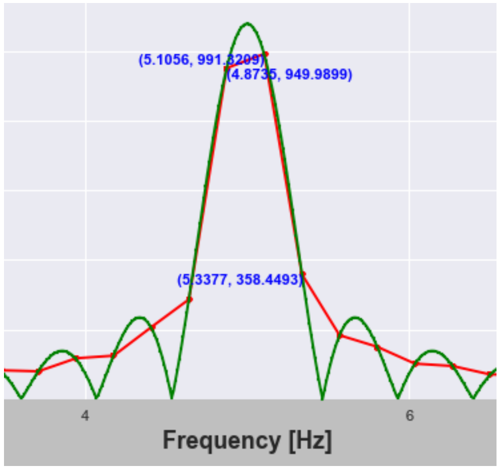

Fig. 4. A zoom on the main lobe of the DFT spectrum

Fig. 3 shows the zero-padded signal that was generated using a Python application designed to perform rapid calculus. The DFT is calculated, and we extract the biggest amplitudes that belong to the main lobe, namely A_{k-1}, A_k, and A_{k+1}, and the frequencies f_{k-1}, f_k, and f_{k+1}. A zoom on the main lobe as it is displayed by the application is shown in Fig. 4. We can observe the three points and the values of the frequencies and DFT amplitudes. To ensure the reproducibility of the tests, we provide the database generated for training [19].

From the original signal, we extract two samples at a time until we ensure the signal has lost an entire period T. The shortest signal contains therefore 2051 samples. This means we have generated all possible scenarios. Fig. 5 shows the amplitudes of the three points versus the number of samples contained in the signal.

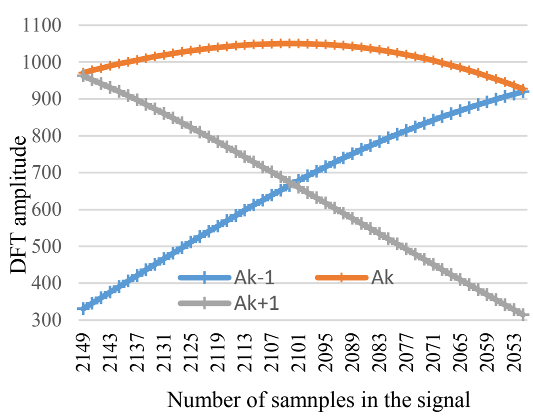

Fig. 5. A zoom on the main lobe of the DFT spectrum

Now, we have all the necessary data to calculate the INPUT and TARGET values. The values considered as INPUT consist of 120 normalized amplitude values A_{k-1}/A_k, A_k/A_k, and A_{k+1}/A_k. We perform normalization to achieve generality since by normalization we eliminate the signal amplitude.

The values that are taken as TARGET consist of the correction terms calculated with (9), knowing that the real frequency is f_R=5 Hz. Examples of INPUT and TARGET data are given in Table 1, while the complete set of data is provided in [19].

TABLE I. EXAMPLES OF INPUT AND TARGET DATA

N (samples)	INPUT			TARGET
	A_{k-1}	A_k	A_{k+1}	δ (-)
2149	0.34056317	1	0.99144261	1.485
2147	0.35362327	1	0.97444708	1.465
2145	0.36667106	1	0.95761995	1.445
2101	0.64594329	1	0.62834404	1.005
2099	0.65834327	1	0.61533858	0.985
2055	0.958326898	1	0.361511871	0.545
2053	0.97470059	1	0.350551739	0.525
2051	0.991461997	1	0.339541093	0.505

IV. THE TRAINING OF THE ANN

Artificial Neural Networks (ANN) are powerful information processing systems, composed of simple processing units, interconnected, and acting in parallel. Usually, ANNs are trained for performing complicated tasks while achieving high accuracy. In paper [20] the authors present the precision of a feedforward backpropagation network that is used for detecting and locating transverse cracks. The advantages of using ANNs in damage detection applications are presented in the paper [21] where the authors achieved a high-frequency estimation by employing an ANN using the normalized amplitudes as input data and compared its performance to other approximation functions. In more recent studies [22] the denoising of measurement data is successfully achieved by processing the acquired signal using a convolutional neural network. The efficiency of a feedforward backpropagation network for the determination of the natural frequencies of a Split Ring Resonator is presented in [23]. The ANN developed in this paper is a feedforward network, trained using a backpropagation algorithm. Its architecture is presented in Fig. 6, and the metadata parameters are shown in Table II.

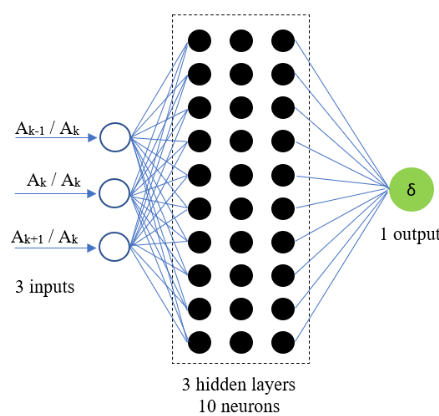

Fig. 6. The architecture of the ANN

Due to the non-linearity of the data used for training and for using the ANN in the future with larger amounts of data, the number of hidden layers is set to three and the number of hidden neurons N_h for each layer is determined considering the number of samples N_s, the number of input N_i and output neurons N_o, and a scaling factor $\alpha=3$.

$$N_h = \frac{N_s}{(\alpha \cdot (N_i + N_o))} \tag{10}$$

The accuracy of the parameters used are confirmed by the cross-validation process, for which, 15% of the training data is used. For testing the ANN, 15% of data is used.

TABLE II. NEURAL NETWORK STRUCTURE

Type of the network	Feed-forward backpropagation
Activation function	Hyperbolic Tangent
Performance validation	Mean Square Error
Training algorithm	Bayesian Regularisation

After the training stage of the network, its precision can be evaluated by checking the regression plots shown in Fig. 7. These plots illustrate how close the output of the ANN is to the actual target values. The error versus epoch for the training, validation, and test performances is shown in Fig. 8.

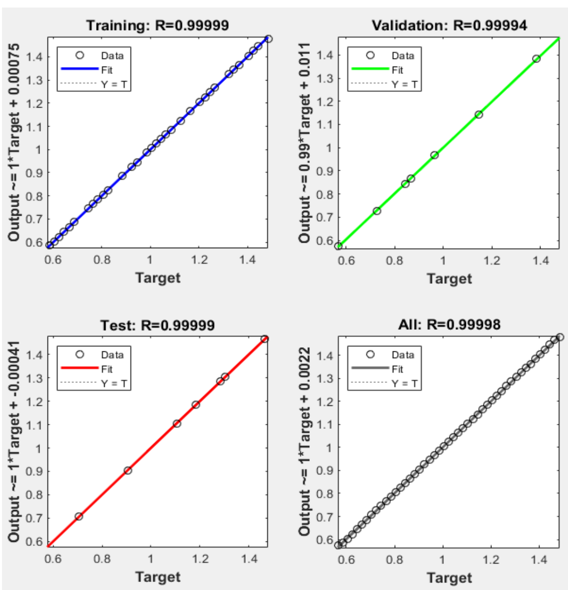

Fig. 7. ANN network training regression

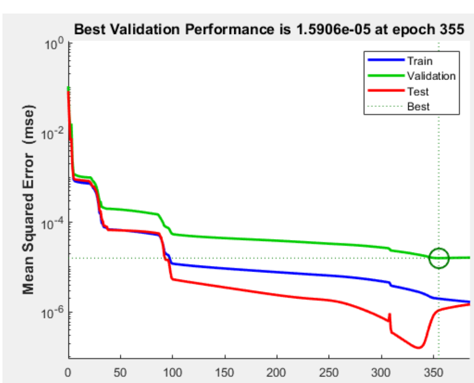

Fig. 8. ANN network best validation performance

V. TESTING THE ACCURACY OF THE PROPOSED METHOD

Several simulations involving the PyFEST application [16] were performed for short time-signals. The signals are generated with the amplitudes A=1 and A=1.5 and the frequency f_O=5 Hz. Tables III and V present the normalized values of the amplitudes in the DFT achieved for different signal lengths. The number of samples defining the signal length is in the range used for training. Tables IV and VI present the normalized values of the amplitudes for signals having the length outside the training interval. Note that the normalized DFT amplitudes are similar for the two considered signal amplitudes, so we can train the network for a single signal amplitude without loss of generality.

By using this normalized training data, the network can estimate the correction term δ, whose values are independent of the signal amplitude. Thus, the value for obtained frequency can be calculated, using the equation

$$f_O = f_{k-1} + \delta \Delta f \tag{11}$$

The achieved values for the estimated frequencies are shown in the last column of tables III, IV, V and VI.

TABLE III. RESULTS OBTAINED FOR THE TEST DATA INVOLVING THE SIGNAL WITH A=1 AND THE LENGTH IN THE TRAINING INTERVAL

N (samples)	Normalized amplitudes			Correction term	Obtained frequency
	A_{k-1}	A_k	A_{k+1}	δ (-)	f_O (Hz)
2149	0.340563	1	0.9914426	1.4669	4.9957877
2137	0.418670	1	0.8919170	1.3704	5.0012637
2119	0.533746	1	0.7533009	1.1819	4.9992683
2107	0.608741	1	0.6684116	1.064	4.9997626
2095	0.683195	1	0.5898408	0.946	5.0002387
2083	0.758982	1	0.517154	0.8277	5.0006482
2067	0.86712	1	0.426993	0.6629	4.9994918
2059	0.92666	1	0.383337	0.5839	4.9997328

TABLE IV. RESULTS OBTAINED FOR THE TEST DATA INVOLVING THE SIGNAL WITH A=1 AND THE LENGTH OUTSIDE THE TRAINING INTERVAL

N (samples)	Normalized amplitudes			Correction term	Obtained frequency
	A_{k-1}	A_k	A_{k+1}	δ (-)	f_O (Hz)
2043	0.37993	1	0.9409282	1.4289	5.000954
2031	0.45791	1	0.8440793	1.3056	5.000147
2017	0.54711	1	0.7382367	1.1621	4.999280
2005	0.62197	1	0.6542038	1.0437	4.999675
1991	0.70887	1	0.5642647	0.9054	5.000100
1979	0.78545	1	0.4934039	0.7864	5.000353
1965	0.88174	1	0.4157072	0.6417	4.999160
1953	0.97470	1	0.3505204	0.5397	5.003764

TABLE V. RESULTS OBTAINED FOR THE TEST DATA INVOLVING THE SIGNAL AMPLITUDE A=1.5 AND THE LENGTH IN THE TRAINING INTERVAL

N (samples)	**Normalized amplitudes**			**Correction term**	**Obtained frequency**
	A_{k-1}	A_k	A_{k+1}	**δ (-)**	f_0 **(Hz)**
2149	0.340563	1	0.991443	1.4669	4.99578
2137	0.418671	1	0.891917	1.3704	5.001264
2119	0.533746	1	0.753301	1.1819	4.999268
2107	0.608741	1	0.668412	1.064	4.999763
2095	0.683196	1	0.589841	0.946	5.000239
2083	0.758983	1	0.517154	0.8277	5.000648
2067	0.867124	1	0.426993	0.6629	4.999492
2059	0.926664	1	0.383337	0.5839	4.999733

TABLE VI. RESULTS OBTAINED FOR THE TEST DATA INVOLVING THE SIGNAL AMPLITUDE A=1.5 AND THE LENGTH OUTSIDE THE TRAINING INTERVAL

N (samples)	**Normalized amplitudes**			**Correction term**	**Obtained frequency**
	A_{k-1}	A_k	A_{k+1}	**δ (-)**	f_0 **(Hz)**
2043	0.379934	1	0.940928	1.4289	5.000955
2031	0.457914	1	0.844079	1.3056	5.000147
2017	0.547110	1	0.738236	1.1621	4.999280
2005	0.621974	1	0.654203	1.0437	4.999675
1991	0.708871	1	0.564264	0.9054	5.00010
1979	0.785455	1	0.493403	0.7864	5.000353
1965	0.881740	1	0.415707	0.6417	4.99916
1953	0.974709	1	0.350520	0.5397	5.003764

One can observe from tables III÷VI that the proposed approach allows estimating the frequencies accurately, the errors being between 0.01% and 0.04%.

VI. CONCLUSION

In this paper, we propose a method to accurately estimate the frequencies of a signal involving the DFT and ANN. To train the network, the method requests a few data that can be obtained by determining the DFT amplitudes of a signal with a known frequency. This data is used as input data. In addition, we determine the distance between the generated frequency and the obtained frequency. This data is used as the target in the training process. It was shown that, by normalizing the amplitudes in the DFT and the error in estimating the frequency, we can use the trained network to find the correction term for signals with any amplitude and frequency.

For testing the method, we used signals with a known frequency. This permitted calculating the difference between the generated frequency and the obtained frequency. The results confirmed the accuracy of the method, the errors being less than 0.04%. These excellent results were obtained both for signals with the length within the range of lengths used for training as well as outside this range.

REFERENCES

[1] C. Johnson, W. Sethares, and A. Klein, Software Receiver Design: Build Your Own Digital Communications System in Five Easy Steps, 2011.

[2] W. Knight, R. Pridham, and S. Kay, “Digital signal processing for sonar”, Proceedings of the IEEE, vol. 69, no. 11, pp. 1451–1506, November 1981.

[3] G. R. Gillich, I. C. Mituletu, Z. I. Praisach, I. Negru, and M. Tufoi, “Method to Enhance the Frequency Readability for Detecting Incipient Structural Damage”, IJST-T Mech. Eng., vol, 41, no. 3, pp.233-242, September 2017.

[4] C. Park and D. Lee, “Classification of Respiratory States Using Spectrogram with Convolutional Neural Network”, Appl. Sci., vol. 12, pp. 1895, February 2022.

[5] B. G. Quinn, “Estimating Frequency by Interpolation Using Fourier Coefficients”, IEEE T. Signal Proces, vol. 42, pp. 1264-1268, May 1994.

[6] E. Jacobsen and P. Kootsookos, “Fast, accurate frequency estimators”, IEEE Signal Proc. Mag., vol. 24, no. 3, pp. 123-125, May 2007.

[7] C. Candan, “A method for fine resolution frequency estimation from three DFT samples”, IEEE Signal Proc. Let., vol. 18, no. 6, pp. 351-354, June 2011.

[8] E. Aboutanios and B. Mulgrew, “Iterative frequency estimation by interpolation on Fourier coefficients”, IEEE T. Signal Proces., vol. 53, no. 4, pp. 1237-1242, May 2005.

[9] T. Grandke, “Interpolation Algorithms for Discrete Fourier Transforms of Weighted Signals”, IEEE T. Instrum. Meas., vol. 32, pp. 350-355, June 1983.

[10] V. K. Jain, W. L. Collins, and D. C. Davis, “High-Accuracy Analog Measurements via Interpolated FFT”, IEEE T. Instrum. Meas., vol. 28, pp. 113-122, June 1979.

[11] K. Ding, C. Zheng, and Z. Yang, “Frequency Estimation Accuracy Analysis and Improvement of Energy Barycenter Correction Method for Discrete Spectrum”, J. Mech. Eng., vol. 46, no. 5, pp. 43-48, 2010

[12] P. Voglewede, “Parabola approximation for peak determination”, Global DSP Magazine, vol. 3, no. 5, pp. 13-17, May 2004.

[13] J. Z. Xiang, S. Qing, and C. Wei, “A novel single tone frequency estimation by interpolation using DFT samples with zero-padding”, Proc. IEEE ICSP, Chengdu, China, pp. 277-281, March 2017.

[14] A. A. Minda, C. I. Barbinita, and G. R. Gillich, “A Review of Interpolation Methods Used for Frequency Estimation”, Rom. J. Acoust. Vib., vol. 17, no. 1, pp. 21-26, June 2020.

[15] G. R. Gillich, I. C. Mituletu, I. Negru, M. Tufoi, V. Iancu, and F. Muntean, “A Method to Enhance Frequency Readability for Early Damage Detection”, J. Vib. Eng. Technol., vol. 3, no. 5, pp. 637-652, October 2015.

[16] D. Nedelcu and G. R. Gillich “A structural health monitoring Python code to detect small changes in frequencies”, Mech. Syst. Signal Pr. Vol. 147, art. 107087, January 2021.

[17] I. C. Mituletu, G. R. Gillich, and N. M. M. Maia, “A method for an accurate estimation of natural frequencies using swept-sine acoustic excitation”, Mech. Syst. Signal Pr., vol. 116, pp. 693-709, February 2019.

[18] A. A. Minda and G. R. Gillich, “Sinc function based interpolation method to accurate evaluate the natural frequencies”, Analele Universitatii Eftimie Murgu Resita, vol. 24, no. 1, pp. 211-218, October 2017.

[19] G. R. Gillich and C. Tufisi, “DFT-ANN database”, Mendeley Data, V1, 2022, doi: 10.17632/vpzvgdtjk2.1mendeley.

[20] M. M. Jazini, M. Khoshakhlagh, and N. Masoumi, "A Novel Combination of Neural Networks and FFT for Frequency Estimation of SAW Resonators' Responses," 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), pp. 1-5, December 2019.

[21] Y. Jiang, H. Li, and M. Rangaswamy, “Deep Learning Denoising Based Line Spectral Estimation”, IEEE Sig. Proc. Let., vol. 26, no. 11, November 2019.

[22] L. Pauly, A. S. P. Priyaa, and A. Pradeep, “SRR Frequency Estimator Using Artificial Neural Network”, NATCON, pp.305-308, February 2014.

[23] W. S. Seo and S. H. Kang, “A novel frequency estimation algorithm based on DFT and second derivative”, J Int. Counc. Elec. Eng., vol. 7, no. 1, pp. 69-75, May.