Metrol. Meas. Syst., Vol. 27 (2020) No. 3, pp. 465–472
DOI: 10.24425/mms.2020.132782

METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.wat.edu.pl

FAST FOUR-POINT ESTIMATORS OF SINUSOIDAL SIGNAL PARAMETERS – NUMERICAL OPTIMISATIONS FOR EMBEDDED MEASURING SYSTEMS

Jarosław Zygarlicki[1], Małgorzata Zygarlicka[1], Janusz Mroczka[2]

1) *Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Prószkowska 76, 45-758 Opole, Poland* (✉ *j.zygarlicki@po.opole.pl*, +48 77 449 8074, *m.zygarlicka@po.opole.pl*)
2) *Wroclaw University of Technology, Chair of Electronic and Photonic Metrology, Bolesława Prusa 53/55, 50-317 Wrocław, Poland* (*janusz.mroczka@pwr.wroc.pl*)

Abstract

The paper presents an algorithm for determining parameters of single sinusoidal components contained in the analyzed digital signal with the use of a small number of mathematical operations. The proposed algorithm can be applied, among others, in measuring devices to monitor basic parameters of electric energy quality as well as in devices used to determine the phasor in the power system. The proposed simplification of the algorithm for determining the sinusoidal components of the analyzed signal allows to use it in embedded devices with low computing power, which translates into lower cost of construction of devices of this type, while maintaining full functionality of the measuring system. The article contains a mathematical argument, which leads to the proposed algorithm, then the optimization of the number of performed mathematical operations is presented. The last part of the paper includes information about performed mathematical operations and presents exemplary times of execution of the algorithm for simple embedded devices.

Keywords: signal processing, harmonics, measurements.

1. Introduction

Analysis of digital measurement signals based on the estimation of parameters of its sinusoidal components is a very extensive subject. In measurement and monitoring systems, Fourier transform implementations have been used for many years to determine the parameters of the estimated components. This field, among other aspects, includes analyses aimed at estimating a single sinusoidal component with the highest possible accuracy and short calculation time.

In recent years, the rapidly growing market for inverter systems for the production of electricity from renewable energy sources has made the estimation of voltage and current phasors in the grid increasingly important. Some modifications of the Fourier transforms can also be used in this application area [1–4]. Modifications of Prony's method also bring good results [5–8].

Methods of estimating the parameters of single sinusoidal components in the time domain constitute a separate class of methods. The methods are the three- and four-point estimators proposed by Vizireanu [9, 10]. Works concerning these methods and their modifications were also conducted by Sienkowski *et al.*, and published in [11, 12]. Works concerning related iteration methods are being carried out as well [13].

The proposition of the method presented in the article is to a certain extent based on the considerations presented in papers concerning four-point methods [9].

This article presents an optimized method for determining the parameters of single sinusoidal components of the analyzed digital signal to be used it in simple systems to analyze and monitor the parameters of signals from various sources.

The priority when working on the proposed solution was to obtain the simplest possible method with a low number of computational operations, using basic mathematical operations on real data types, while maintaining high accuracy of estimation of parameters of single components of the signal. The article deliberately does not present an advanced analysis of the accuracy of estimation of parameters and the examples of using this method in specific applications. This is quite an extensive subject, which will be addressed in further publications.

The article consists of four sections. Section 1 provides the introduction to the article. Section 2 includes the proposed method and modifications leading to the simplification of calculations. Section 3 presents the results of analyses of the contribution of particular mathematical operations to the method and the results of measurements of the time of algorithm execution in real embedded systems. Section 3 also contains very basic (illustrative) measurements of the method accuracy for the sample signals. Section 4 is the summary of the article.

2. Description of the proposed method

The proposed method consists of two main stages. The amplitudes are determined in the first stage and frequencies and initial phases of single sinusoidal components of the analysed digital signal are determined in the second stage.

2.1. Determination of amplitudes

In the first stage of calculations, in the proposed method based on samples $x_1, \ldots, x_4$ of the analyzed signal, the equations modelling the analyzed signal with the use of a sinusoid are formed:

$$\begin{aligned} x_1 &= A \cdot \sin(\omega \cdot t_1 + \phi), \qquad & x_2 &= A \cdot \sin(\omega \cdot t_2 + \phi), \\ x_3 &= A \cdot \sin(\omega \cdot t_3 + \phi), \qquad & x_4 &= A \cdot \sin(\omega \cdot t_4 + \phi), \end{aligned} \tag{1}$$

where: ω – pulsation, A – estimated amplitude, $t_1, \ldots, t_4$ – sampling time for 1 . . . 4 of the sample to be analysed. For x_1 and x_2 after simple transformations, we can write down:

$$\omega \cdot t_1 + \phi = \arcsin\left(\frac{x_1}{A}\right), \tag{2}$$

$$\omega \cdot t_2 + \phi = \arcsin\left(\frac{x_2}{A}\right). \tag{3}$$

Subtracting equations (3) and (2) by sides and substituting: $\Delta t = t_2 - t_1$, we get:

$$\omega \cdot \Delta t = \arcsin\left(\frac{x_2}{A}\right) - \arcsin\left(\frac{x_1}{A}\right) \tag{4}$$

and, analogously, for samples x_3 and x_4 we define $\Delta t = t_4 - t_3$ and get:

$$\omega \cdot \Delta t = \arcsin\left(\frac{x_4}{A}\right) - \arcsin\left(\frac{x_3}{A}\right). \tag{5}$$

Substituting equation (4) for (5) we can write down:

$$\arcsin\left(\frac{x_2}{A}\right) - \arcsin\left(\frac{x_1}{A}\right) = \arcsin\left(\frac{x_4}{A}\right) - \arcsin\left(\frac{x_3}{A}\right). \tag{6}$$

This is an equation with one unknown which, after the transformation, can be used to determine the amplitude of a single sinusoidal component to be searched for. For transformations, a program enabling automated transformations of equations was used. Substituting the following: $x_1 = a$, $x_2 = b$, $x_3 = c$, $x_4 = d$ and by entering the following commands into the program workbook:

```
Given
asin(b/A) - asin(a/A) = asin(d/A) - asin(c/A)
Find(A) ->
```

a solution has been obtained which in its general form can be presented in the form of a dependency:

$$A = -2\frac{\sqrt{m_1 \cdot m_2}}{m_1} \tag{7}$$

where:

$$m_1 = \mathrm{a}^4 + \mathrm{b}^4 + \mathrm{c}^4 + \mathrm{d}^4 - 2\left(\mathrm{a}^2\mathrm{b}^2 + \mathrm{a}^2\mathrm{c}^2 + \mathrm{a}^2\mathrm{d}^2 + \mathrm{b}^2\mathrm{c}^2 + \mathrm{b}^2\mathrm{d}^2 + \mathrm{c}^2\mathrm{d}^2\right) + 8\mathrm{abcd} \tag{8}$$

and:

$$m_2 = \mathrm{a}^3\mathrm{bcd} + \mathrm{ab}^3\mathrm{cd} + \mathrm{abc}^3\mathrm{d} + \mathrm{abcd}^3 - \left(\mathrm{a}^2\mathrm{b}^2\mathrm{c}^2 + \mathrm{a}^2\mathrm{b}^2\mathrm{d}^2 + \mathrm{a}^2\mathrm{c}^2\mathrm{d}^2 + \mathrm{b}^2\mathrm{c}^2\mathrm{d}^2\right). \tag{9}$$

In the form of a code in Matlab language equations (7), (8), (9) can be written as follows:

```
m1 = - 2*d^2*b^2 + b^4 - 2*c^2*b^2 + c^4 + a^4 - 2*d^2*a^2 + 8*c*d*a*b
     - 2*a^2*b^2 - 2*c^2*a^2 + d^4 - 2*c^2*d^2;
m2 = - ^2*b^2*c^2 + c*a^3*b*d - c^2*d^2*a^2 + c^3*a*b*d - d^2*b^2*a^2
     + c*a*b^3*d - a^2*b^2*c^2 + c*d^3*a*b;
A = - 2*sqrt(m1*m2)/m1.
```

The next step is to optimize the number of mathematical operations at the expense of the memory size needed to store intermediate results. By grouping similar operations and storing them in temporary variables, we get:

```
a2 = a*a;  b2 = b*b;  c2 = c*c;  d2 = d*d;
a3 = a2*a;  b3 = b2*b;  c3 = c2*c;  d3 = d2*d;
a4 = a3*a;  b4 = b3*b;  c4 = c3*c;  d4 = d3*d;
ad = a*d;  bc = b*c;  abcd = ad*bc;
c2d2 = c2*d2;  b2c2d2 = b2*c2d2;
c2_d2 = c2 + d2;  b2_c2_d2 = b2 + c2_d2;
b2c2_d2 = b2*c2_d2;  a2b2_c2_d2 = a2*b2_c2_d2;
b3c = b3*c;  c3b = c3*b;  d3a = d3*a;  a3d = a3*d;
m1 = a4 + b4 + c4 + d4 - 2*(a2b2_c2_d2 + b2c2_d2 + c2d2) + 8*abcd;
m2 = ad*(b3c + c3b) + bc*(a3d + d3a) - a2*(b2c2_d2 + c2d2) - b2c2d2;
A = - 2*sqrt(m1*m2)/m1.
```

To sum up, for the calculation of the first stage of the proposed method, the following operations are required after optimization: **30** multiplications, **15** additions/subtractions, **1** division, and additionally **1** root extraction. Importantly, all the operations are performed on real data types.

It is worth noting that for the case where $a = 0$, the calculations are simplified considerably. It can be used in some applications of the method, for example in devices with hardware detection of the passage of the analyzed signal through zero.

If $a = 0$ then:

$$m_1 = b^4 + c^4 + d^4 - 2\left(b^2c^2 + b^2d^2 + c^2d^2\right) \tag{10}$$

and:

$$m_2 = -b^2c^2d^2 \tag{11}$$

and then:

```
m1 = b4 + c4 + d4 - 2*(b2c2_d2 + c2d2);
m2 = - b2c2d2;
A = - 2*sqrt(m1*m2)/m1.
```

For this case ($a = 0$) for the calculation of the first stage of the proposed method, the following operations are required after optimization: **15** multiplications, **5** additions **1** division, and additionally **1** root extraction.

2.2. Determination of frequencies and initial phases

With the amplitude determined in point 2.1, it is possible to determine the missing parameters of frequency and initial phase of the component being searched for. Subtracting equations (3) and (2) manually after simple transformations, assuming that the sampling period of the signal is $T_s = t_2 - t_1$ and $\omega = 2\pi f$ dependence on the frequency of the component being searched will be expressed by the following formula:

$$f = \frac{\arcsin\left(x_2/A\right) - \arcsin\left(x_1/A\right)}{2\pi T_s} \tag{12}$$

and the initial phase is determined from the dependence (2) where after the transformations we obtain:

$$\phi = \arcsin\left(\frac{x_1}{A}\right) - 2 \cdot \pi \cdot f \cdot t_1 \,. \tag{13}$$

In the form of a code in Matlab after simplifying equation (12) and (13), substituting for: $x_1 = a$, $x_2 = b$ we can write down:

```
k1 = asin(a/A);
k2 = asin(b/A);
dk = k2 - k1;
pi2 = 2*pi; %constant
pi2Ts = pi2*Ts; %constant
f = dk/pi2Ts;
fi = k1 - pi2*f*t1;
```

To sum up, for the calculation of the second stage of the proposed method, the following operations are required after optimisation: 2 multiplications, 2 additions/subtractions, 3 divisions, and 2 additional arcus sinus operations. In this case also all operations are performed on real data types.

Metrol. Meas. Syst., Vol. 27 (2020), No. 3, pp. 465–472
DOI: 10.24425/mms.2020.132782

2.3. Summary of the number of operations

A summary of operations required by the method is given in Table 1 and Fig. 1. Clearly for both the full method and the reduced method ($a = 0$) multiplying and adding operations dominate. In total, these operations constitute 87% for the full version of the method and 78% for the reduced version of the method respectively.

Table 1. A summary of the number of operations of the method (on the left: full method, on the right: reduced method: $a = 0$).

Operation	**Number of operations I stage**	**Number of operations II stage**	**Operations in total**
*	30	2	32
+	15	2	17
/	1	3	4
$\sqrt{\ }$	1	0	1
arcsin	0	2	2
total:	47	9	56

Operation	**Number of operations I stage (a = 0)**	**Number of operations II stage**	**Operations in total**
*	15	2	17
+	5	2	7
/	1	3	4
$\sqrt{\ }$	1	0	1
arcsin	0	2	2
total:	22	9	31

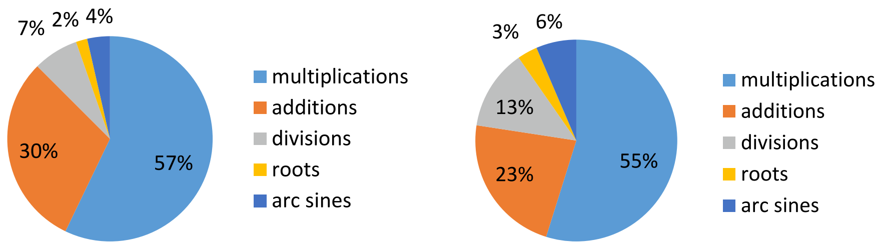

Fig. 1. Comparison of the number of individual mathematical operations in the proposed method (on the left: full method, on the right: reduced method: $a = 0$).

3. Method implementation and measurements

3.1. Analysis of computation times

The presented method has been implemented in four microprocessors of different architecture: from simple RISC systems through ARM systems to specialized DSP signal processors. These were the following systems:

1. 16-Bit RISC on the example of the system: MSP430G2553 (clock: 16 MHz) [14],
2. ARM Cortex M0 (32-bit), on the example: Nu-LB-NUC140 (clock: 50 MHz) [15],
3. ARM Cortex-M4F (32-bit), on the example: LM4F120H5QR (clock: 80 MHz) [16],
4. DSP Texas Instruments (32-bit), system: TMS320C28027 (clock: 60 MHz) [17].

The method was implemented in C language and calculations in all systems were performed on double data (8 bytes). The summary of the calculation times of the full and simplified versions of the method is presented in Table 2.

Figure 2 shows the percentage proportions of execution times of particular types of instructions for an exemplary processor with ARM Cortex M0 architecture.

Table 2. Comparison of computation times for different processor systems (on the left: full method, on the right: reduced method: a = 0).

Processor	Clock [MHz]	Computation time [μs]
16-Bit RISC	16	2945.6
ARM Cortex M0	50	266.0
ARM Cortex M4F	80	169.7
DSP Texas Instr.	60	217.3

Processor	Clock [MHz]	Computation time [μs]
16-Bit RISC	16	1993.3
ARM Cortex M0	50	180.0
ARM Cortex M4F	80	114.9
DSP Texas Instr.	60	147.1

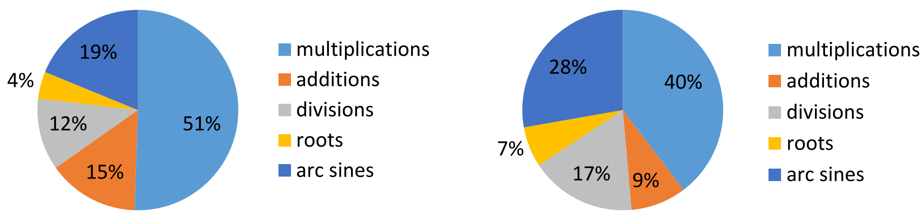

Fig. 2. Comparison of execution times of individual instructions in proposed method (computed for ARM Cortex M0).

Hardware implementations of the method allow to determine the total time share of individual mathematical operations. It is still clearly visible that for both the full method and the reduced method (a = 0) additions and multiplications dominate. However, their percentage share is decreasing. This results from the fact that hardware multipliers and summators, which significantly accelerate basic mathematical operations, appear in the analyzed processor systems. Together, the multiplication and addition operations represent 66% for the full version of the method and 49% for the reduced version of the method, respectively.

3.2. Basic analysis of method estimation accuracy

Figure 3 below presents examples of basic characteristics for the accuracy of estimation of amplitude and frequency parameters for the presented method. These characteristics were determined for a simulated signal (with a frequency f = 50 Hz, with a sampling frequency of

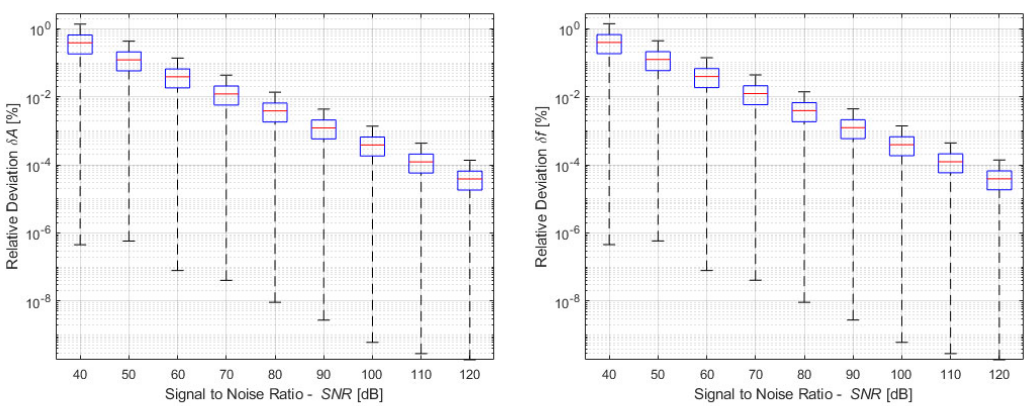

Fig. 3. Accuracy of estimation of amplitude and frequency parameters for the proposed method for sample signals.

www.czasopisma.pan.pl PAN www.journals.pan.pl
Metrol. Meas. Syst., Vol. 27 (2020), No. 3, pp. 465–472
DOI: 10.24425/mms.2020.132782

f_s = 1800 Hz and a random initial phase). White noise with variable SNR (*Signal to Noise Ratio*) was added to the simulated signal. The tests were carried out 1,000,000 times for a given SNR level and the characteristics show the statistical distribution of measurements. A more detailed analysis should be made after determining the specific application of the method and with the purpose to determine the parameters for the design of the initial filter at the input of the method.

4. Conclusions

The method presented in the article can be used in a wide range of embedded measuring devices for estimation of parameters of single sinusoidal components of a signal. The performed optimization of the number of mathematical operations of the method allows to obtain short calculation times of the algorithm.

Thanks to the small number of mathematical operations, the proposed solution can be used in devices with a low computing power, without compromising the accuracy of calculations of the estimated components.

Using only 4 samples of the analysed signal in the method translates into very short time windows of the analysis, which allows for correct determination of estimated parameters for fast-changing signals with a slight delay. It is of great practical importance in many areas of application (*e.g.*: protection systems against exceeding specific parameters of the monitored object). On the other hand, with such a small number of samples being processed, it is very important to prepare the analyzed signal before the process of component estimation. In many applications it is necessary to use a band-pass filter [18]. Selection of this filter for a specific implementation of an algorithm for real signal analysis will be the subject of further research.

References

[1] Borkowski, J., Kania, D., Mroczka, J. (2014). Influence of A/D quantization in an interpolated DFT based system of power control with a small delay. *Metrology and Measurement Systems*, 21(3), 423–432.

[2] Borkowski, J., Kania, D. (2016). Interpolated-DFT-based fast and accurate amplitude and phase estimation for the control of power. *Metrology and Measurement Systems*, 23(1), 13–26.

[3] Yu, C., Huang, Y., Jiang, J. (2010). A Full- and Half-Cycle DFT-based technique for fault current filtering. *Proceedings of the 2010 IEEE International Conference on Industrial Technology (ICIT)*, Vina del Mar, Chile, 859–864.

[4] Wen, H., Teng, Z., Wang, Y., Zeng, B., Hu, X. (2011). Simple interpolated FFT algorithm based on minimize sidelobe windows for power-harmonic analysis. *IEEE Transactions on Power Electronics*, 26(9), 2570–2579.

[5] Zygarlicki, J., Mroczka, J. (2012). Prony method used for testing harmonics and interharmonics of electric power signals. *Metrology and Measurement Systems*, 19(4), 659–672.

[6] Zygarlicki, J., Zygarlicka, M., Mroczka, J., Latawiec, K. (2010). A reduced Prony's method in power quality analysis – parameters selection. *IEEE Transactions on Power Delivery*, 25(2), 979–986.

[7] Zygarlicki, J. (2017). Fast second order original Prony's method for embedded measuring systems. *Metrology and Measurement Systems*, 24(4), 721–728.

[8] Zahlay, F.D., Rama Rao, K.S. (2012). Neuro-Prony and Taguchi's methodology based adaptive autoreclosure scheme for electric transmission systems. *IEEE Transactions on Power Delivery*, 27(2), 575–582.

[9] Vizireanu, D.N. (2011). A simple and precise real-time four point single sinusoid signals instantaneous frequency estimation method for portable DSP based instrumentation. *Measurement*, 44(2), 500–502.

[10] Vizireanu, D.N., Preda, R.O. (2013). Is "five-point" estimation better than "three-point" estimation? *Measurement*, 46(1), 840–842.

[11] Sienkowski, S. (2016). A method of m-point sinusoidal signal amplitude estimation. *Measurement Science Review*, 16(5), 244–253.

[12] Sienkowski, S., Krajewski, M. (2018). Simple, fast and accurate four-point estimators of sinusoidal signal frequency. *Metrology and Measurement Systems*, 25(2), 359–376.

[13] Pan, X., Zhao, H., Zou, W., Zhou, Y., Ma, J., Wang, J., Hu, F. (2016). Frequency estimation of discrete time signals based on fast iterative algorithm. *Measurement*, 82, 461–465.

[14] Texas Instruments. (2011). *MSP430G2x53, MSP430G2x13: Mixed signal microcontroller*. [Datasheet SLAS735J]. http://www.ti.com/lit/ds/slas735j/slas735j.pdf (accessed on Dec. 2019).

[15] Nuvoton. (2012). *NuMicro™Family NUC140 Data Sheet*. [Datasheet, v3.02]. https://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf (accessed on Dec. 2019).

[16] Texas Instruments. (2014). *Tiva™TM4C1233H6PM Microcontroller*. [Datasheet, v15842.2741]. http://www.ti.com/lit/ds/symlink/tm4c1233h6pm.pdf (accessed on Dec. 2019).

[17] Texas Instruments. (2008). *TMS320F2802x Microcontrollers*. [Datasheet, SPRS523N]. http://www.ti.com/lit/ds/symlink/tms320f28027.pdf (accessed on Dec. 2019).

[18] Kumaresan, R., Feng, Y. (1991). FIR prefiltering improves Prony's method. *IEEE Transactions Signal Processing*, 39(3), 736–741.