Digital Object Identifier (DOI) of the paper: 10.1109/TIM.2017.2729358.

Accurate Floating Point Argument Calculation for Sine Fitting Algorithms

Balázs Renczes
Budapest University of Technology and Economics,
Department of Measurement and Information Systems, Budapest, Hungary

***Abstract*—In this paper, accurate argument calculation for sine-fitting algorithms is investigated, assuming floatingpoint (FP) arithmetic. An easy-to-implement incremental calculation technique is suggested. In order to decrease error propagation, the algorithm is complemented with an advanced summation technique. Theoretical and numerical analyses on computational demand are performed to highlight that incremental argument calculation outperforms the method proposed in former research. Furthermore, an algorithm is implemented to mitigate the effect of imprecise representation of frequency on FP arithmetic. Monte Carlo analyses are carried out to demonstrate the accuracy of the suggested algorithms. Results show that phase information can be evaluated precisely even with single-precision FP arithmetic, applying incremental argument calculation. By this means, the cost of equipment that is needed to perform sine fitting can be reduced significantly. Finally, possible application areas are shown to demonstrate the applicability of the suggested solutions in the state-of-art measurement procedures.**

***Index Terms*— Analog-to-digital converter (ADC) testing, floating-point (FP) arithmetic, least-squares (LS) methods, numerical accuracy, parameter estimation, roundoff errors, sine fitting, single precision**

I. Introduction

Sine fitting algorithms are widely used in the field of measurement technology. They can be applied to measure the complex value of an impedance [1], or to characterize the quality of the power system [2]. In particular, there are application fields where it is of great importance to have an accurate sine wave estimator. Among these fields we can find the testing of analog-to-digital converters (ADCs) [3] and of digitizing waveform recorders [4].

In order to characterize a sine wave, the following description can be utilized:

$$y_k = A \cdot \cos(2\pi f t_k) + B \cdot \sin(2\pi f t_k) + \mathrm{C} \ , \ \ k = 1, \dots, N \tag{1}$$

where y_k is the *k*th sample in fitted sine wave, *A*, *B* and *C* are the cosinusoidal, sinusoidal and dc components, respectively. Furthermore, *f* is the frequency of the signal, t_k is the *k*th sampling time and *N* denotes the number of samples. In case of uniform (equidistant) sampling, time instants can be calculated as:

$$t_k = k/f_s, \qquad k = 1, \dots, N, \tag{2}$$

where f_s denotes the sampling frequency. In this paper, uniform sampling will be assumed, but results can be generalized to non-uniform sampling, as well. Let us introduce notation

$$\gamma_k = \frac{f}{f_s} k = f_{rel} \cdot k \,, \qquad k = 1, \dots, N \,, \tag{3}$$

where f_{rel} is the (to the sampling frequency) relative frequency. If the frequency of the signal is known, *A, B* and *C* are the parameters to be estimated, while if *f* is unknown, it also extends the parameter vector. The most widely used fitting criterion is the minimization of least squares (LS) errors [3]. The cost function (CF) of the LS fitting is:

$$\mathrm{CF_{LS}} = \sum_{k=1}^{N} (x_k - y_k)^2 \ , \tag{4}$$

where x_k is the *k*th value in the measured data set. Another possible fitting is based on the maximum likelihood criterion. This method maximizes the probability of observing the sampled data set [5].

The floating point (FP) implementation of sine fitting algorithms is wide-spread, due to the wide dynamic range an FP arithmetics can represent with (approximately) constant relative errors [8]. However, this advantageous property of the FP representation also implies that the larger the represented number, the larger the absolute value of the roundoff error of the representation. While in personal computers mostly double precision is applied, in digital signal processors (DSPs) and in field programmable gate arrays (FPGAs), where the power consumption and the cost of equipment are critical parameters, single precision is widely used, as well [9][10].

In [6], it was shown that using FP number representation, the evaluation of sine fitting algorithms are disturbed by roundoff errors with much larger amplitudes, than the resolution of the FP representation. In particular, it was pointed out that the error in the argument of sine and cosine functions, that is, the error in the calculated phase distorts the CF of the LS method considerably. As a result, the CF gets ragged, and optimization methods can be stuck in local minima [7]. Furthermore, as a result, the expected value of the CF increases, as well [6]. It is important to see that this phenomenon has an effect on every implementation of LS fitting, both in time and frequency domain. Namely, the CF of the fitting is disturbed as a result of the injection of roundoff errors by imprecise argument calculation.

The instantaneous phase equals to:

$$\varphi_k = 2\pi f t_k = 2\pi f_{rel} k = 2\pi \gamma_k \ , \ \ k = 1, \dots, N \ . \tag{5}$$

With increasing *k*, the absolute value of φ_k increases, as well, introducing a growing roundoff error $(\Delta\varphi)_k$ due to FP number representation. The expected value of the increase in $\mathrm{CF_{LS}}$ can be calculated as

$$E\{\Delta \mathrm{CF_{LS}}\} = \frac{\pi^2 R^2 J^2 eps^2 N}{18} \ , \tag{6}$$

where *eps* is the precision of the number representation, *R* is the amplitude of the signal $R = \sqrt{A^2 + B^2}$, and *J* is the number of sampled periods . For the proof, see Appendix. For single precision, $eps_s = 1.19 \cdot 10^{-7}$, while for double precision, $eps_d = 2.22 \cdot 10^{-16}$.

It is obvious that the more periods are sampled, the larger the effect of imprecise phase calculation. However, in practical situations, the error of double precision evaluation is insignificant. Nevertheless, as it was pointed out earlier in this section, in many applications, only single precision evaluation is available. In addition, if an algorithm yields precise results on limited precision platforms, then the power consumption and the cost of the needed equipment can be reduced significantly. Thus, it is reasonable to investigate, how the effect of imprecise argument calculation can be effectively mitigated using single precision arithmetic.

In [6], a splitting method was suggested for this purpose. The method divides floating point numbers into more parts and maps the resulting argument in $[-\pi;\pi)$, see Section II-A. As the absolute value of the phase cannot grow arbitrarily, the maximum absolute representation error is limited, and the effect of imprecise argument calculation can be mitigated significantly.

Since the effect of imprecise argument calculation has been revealed in [6], no alternative method to the splitting technique was published to decrease this phenomenon, neither in time, nor in frequency domain. However, from a point of computational demand, this method is ineffective. Namely, the operation of splitting requires a large number of operations, as it will be shown in Section II-E. The aim of this paper is to find an alternative solution that is sufficiently precise and can be evaluated much faster than the splitting method.

In Section II, an overview on the splitting technique will be given. Besides, an incremental calculation technique will be introduced. In order to decrease the effect of accumulating roundoff errors, this method will be complemented with an advanced summation technique. Accuracy will be demonstrated through simulation results. Furthermore, the computational burden compared to the splitting technique will be analyzed through theoretical and numerical analyses. In Section III, the case will be investigated when the frequency of the sinusoidal signal cannot be represented precisely in one floating point number. It will be pointed out that representing the frequency as two floating point numbers can keep the errors of argument calculation small. Results will be verified through simulations. Finally, in Section IV, two application areas will be shown to demonstrate the applicability of the suggested methods in the state-of-art measurement procedures.

II. Methods to ensure accurate argument calculation

In this section, different methods will be investigated to ensure that the arguments of sine and cosine functions can be evaluated precisely. First, an overview will be given on the algorithm proposed in [6], and the bottleneck of this method will be highlighted. Then, an alternative calculation method will be presented, based on incremental calculation. It will be shown that the pure incremental calculation is significantly disturbed by roundoff errors. Thus, the method will be complemented with an advanced summation technique. The effectiveness of the presented method will be demonstrated through simulations. Finally, the computational burden of both the splitting technique and the incremental argument calculation will be investigated by theoretical and numerical means.

A. The splitting technique

Based on software package "QDSP toolbox for MATLAB" [11], in [6], a splitting technique is proposed in order to calculate (5) accurately. The method divides f_{rel} into three parts, and k into two parts so that the sum of the parts yields the original values. After splitting, only a limited number of bits in the mantissa of each part contains information about the represented number. With single precision number representation, at most the first 11 bits in the mantissa differ from 0. By this means, the product of two parts can be represented without roundoff errors, since a single precision number has a 23-bit long mantissa. Formally:

$$f_{rel} = [f_1\ ,\ f_2\ ,\ f_3] \text{ and } k = [k_1\ ,\ k_2\ ]\ , \tag{7}$$

where f_1 is the first slice of f_{rel} and k_2 is the second slice of k. For instance, if $f_{rel} = 0.45$ on single precision, then we have:

$$f_1 = 0.44995 \quad f_2 = 4.8757 \cdot 10^{-5} \quad f_3 = 5.9605 \cdot 10^{-8}\ , \tag{8}$$

and their binary representations are:

$$f_1 = 1.1100110011 \cdot 2^{-2}\ \ f_2 = 1.1001100\,100 \cdot 2^{-15}\ \ f_3 = 1.0000000000 \cdot 2^{-24}\,, \tag{9}$$

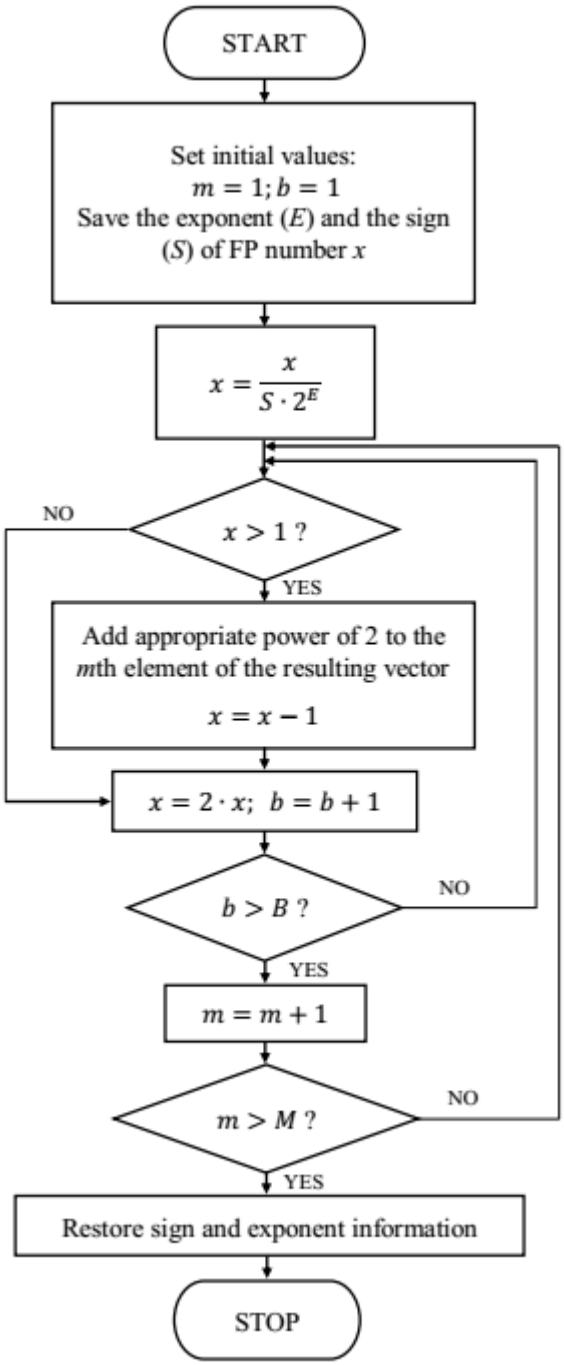

Figure 1. Flowchart of the splitting technique.

In order to evaluate (5), $f_{rel} \cdot k$ can be calculated with convolution [6]:

$$f_{rel} \cdot k = [f_1\,,\ f_2\,,\ f_3] * [k_1\,,\ k_2\,] = [f_1 k_1\,,\ f_1\,k_2 + f_2 k_1\,, f_2 k_2 + f_3 k_1\,,\ f_3 k_2] \ . \tag{10}$$

After this calculation, the fractional part of the four slices is to be calculated:

$$\langle f_{rel} \cdot k \rangle = [\langle f_1 k_1 \rangle\,, \quad \langle f_1\, k_2 + f_2 k_1 \rangle\,, \quad \langle f_2 k_2 + f_3 k_1 \rangle\,, \quad \langle f_3 k_2 \rangle]\,, \tag{11}$$

where $\langle \cdot \rangle$ denotes the fractional part after rounding to the nearest integer value. For instance, $\langle 3.4 \rangle = 0.4$ and $\langle 2.7 \rangle = -0.3$. The method is advantageous, since each slice is limited in $(-0.5; 0.5]$. Thus, their sum is also limited, and consequently it is much less influenced by roundoff errors – recall that with floating point representation, the larger the absolute value, the larger the possible representation error.

Since sine and cosine are periodic functions, the fractional part of $f_{rel}k$ contains all information that is needed to evaluate the phase information:

$$\varphi'_k = 2\pi \cdot (\langle f_1 k_1 \rangle + \langle f_1\, k_2 + f_2 k_1 \rangle + \langle f_2 k_2 + f_3 k_1 \rangle + \langle f_3 k_2 \rangle) \tag{12}$$

where φ'_k is the calculated phase information in $(-\pi; \pi]$. Due to periodicity:

$$\sin \varphi'_k = \sin \varphi_k \;\text{ and }\; \cos \varphi'_k = \cos \varphi_k. \tag{13}$$

The algorithm has a bottleneck from a computational point of view. Namely, the slices of f_{rel} and k are to be generated. According to MATLAB R2017a profiler, more than 80% of the computational time is spent with splittings. Thus, in the following, the algorithm of splitting will be investigated in detail.

The method is visualized in Fig. 1. To perform the splitting, first, the sign (S) and the exponent (E) of the FP number is calculated, and the number is normalized with these values between 1 and 2. After this normalization, the values of the bits are determined cyclically. In the first cycle, 1 is subtracted from the normalized number, and 1 is added to the first slice – each slice contains zeros at the beginning. The resulting FP number is shifted left by multiplying it by 2. If the result is greater than or equals to 1, then the second bit in the mantissa of the original number was 1. In this case, 0.5 has to be added to the first slice. If the result is smaller than one, then the second bit was 0. Thus, no operation is needed. By repeating this cycle, the original FP number is scanned bit by bit. In general, *M* slices are generated, each containing at most *B* bits. The first *B* bits are written in the first slice, the following *B* bits are written to the second one, and so on. At the end of the method, the signs and the exponents of the slices are restored by multiplying them by $S \cdot 2^E$.

The cycle has to be run $M \cdot B$ times. Although the slices of f_{rel} have to be calculated only once, the splitting technique is time consuming since every *k* has to be splitted individually. Thus, in the following, an alternative easy-to-implement method will be investigated that can calculate the phase information incrementally with less computational demand. Besides, a detailed analysis on the computational burden of both methods will be carried out in Section II-E.

B. Incremental argument calculation

The problem with the original argument calculation was that with increasing *k*, the absolute value of the phase also increased, implying an increasing roundoff error. This was compensated with the splitting technique: calculating the fractional parts mapped the absolute value of φ'_k in a limited range.

The idea behind incremental argument calculation is that instead of calculating the large value of $f_{rel} \cdot k$ with the splitting technique, then reducing it into a limited range, it is rational to avoid the increase in the arguments. Namely, it is possible that the phase information is calculated incrementally:

$$\gamma'_1 = \langle f_{rel} \rangle,\;\; \gamma'_{k+1} = \langle \gamma'_k + \gamma'_1 \rangle \text{ and } \varphi'_{k+1} = 2\pi\gamma'_{k+1}\;. \tag{14}$$

For example:

$$\gamma'_k = 0.45 \text{ and } f_{rel} = 0.15 \rightarrow \gamma'_{k+1} = \langle 0.45 + 0.15 \rangle = -0.4 \ . \quad (15)$$

First, the fractional part of $f_{rel} \cdot k$, that is, γ'_k is calculated incrementally, and then the result is multiplied by 2π. By this means, the absolute value of φ'_{k+1} is prevented from growing above π. Consequently, the roundoff error at the storage in a limited precision floating point number is limited, as well.

Applying fixed point number representation, this method is advantageous, since summations can be performed without roundoff errors. Furthermore, the calculation of the fractional part in unnecessary. Namely, it is performed inherently when overflow occurs. Contrarily, using FP arithmetic, summations are distorted by roundoff errors, and the calculation of the fractional part has to be evaluated.

To represent the numerical problem, phases are evaluated with the incremental argument calculation using single precision arithmetic. In order to have a benchmark for the error of the calculation, phase information is also evaluated using double precision that is assumed to be precise compared to single precision arithmetic. During the calculations, the following parameters are used:

$$f_{rel} = 2^{-8} + 2^{-26} \text{ and } N = 10^5 \ . \quad (16)$$

The frequency is set so that it can be represented without roundoff errors on single precision. Thus, the representation error of the relative frequency does not influence the results. The problem of imprecise relative frequency representation will be investigated in detail in Section III.

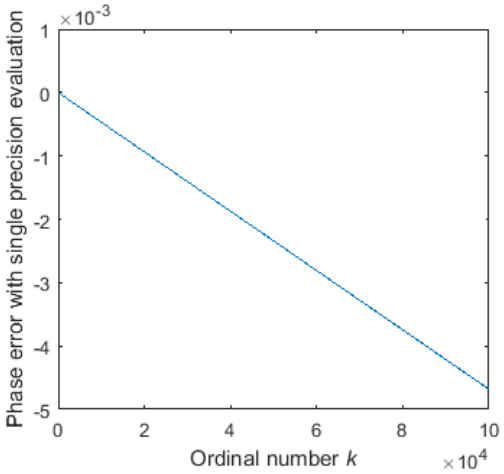

Figure 2. Phase error along the sample set with incremental phase calculation.

The error of single precision evaluation using incremental calculation can be seen in Fig. 2. The result is a drift: with increasing *k*, the phase error increases, as well. Phase error over time can be characterized by a straight line. The error grows to $-4.68 \cdot 10^{-3}$ at the end of the sample set.

The problem can be explained by fact that although the absolute value of γ'_k is limited, it can grow to 0.5. In the example,

$$\gamma'_{65} = \langle \gamma'_{64} + \gamma'_1 \rangle = \langle \gamma'_{64} + f_{rel} \rangle \quad . \quad (17)$$

The least significant digit (LSD) in the single precision mantissa of γ'_{64} equals to:

$$\text{LSD}(\gamma'_{64}) = \text{LSD}(2^{-2} + 2^{-20}) = 2^{-25} \ . \quad (18)$$

Since 2^{-26} in f_{rel} cannot be represented beside $\text{LSD}(\gamma'_{64})$, the error of the FP summation is $-2^{-26} = -1.49 \cdot 10^{-8}$. The problem with this method is that these small errors are accumulating during the summation [12], since γ'_{k+1} depends on γ'_k.

Consequently, the errors are not independent of each other, and they are propagating along the sample set. This problem can be solved by compensated summation [13].

C. Incremental argument calculation with compensated summation

The error of incremental phase calculation originates from γ'_k, the absolute value of which can grow to 0.5, while the term to be added, γ'_1 remains small compared to this value. Though the error of each summation step is small, the effect of these accumulating errors may be significant, as it was shown in Section II-B. In order to decrease this effect, compensated summation can be applied [13]. The flowchart of incremental argument calculation with compensated summation is shown in Fig. 3. Compensated summation adds the error of the previous addition to the next term. Let us denote the compensated term by *z*:

$$z = \gamma'_1 + e_k \tag{19}$$

where e_k is the roundoff error in the calculation of γ'_k. With the compensated term, γ'_{k+1} can be calculated as:

$$\gamma'_{k+1} = \gamma'_k + z \tag{20}$$

The error of the addition is:

$$e_{k+1} = (\gamma'_k - \gamma'_{k+1}) + z. \tag{21}$$

While each summation step is disturbed by roundoff errors, these errors do not accumulate along the record. Since γ'_1 has small absolute value, the error that could not be represented in the large FP value of γ'_{k+1}, becomes representable in *z*.

Though from (19) and (20), e_k is analytically 0, due to floating point additions, it assumes the roundoff error of the operations.

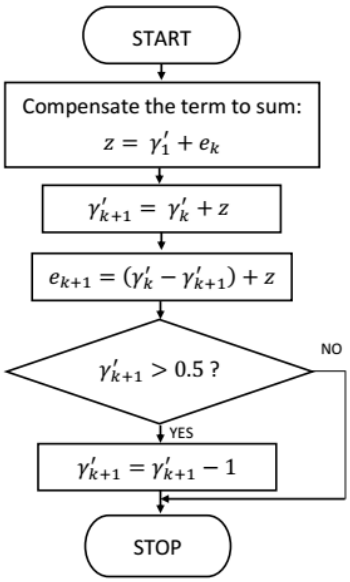

Figure 3. Flowchart of the compensated summation technique.

After performing the addition, the value of γ'_{k+1} has to be analyzed. If γ'_{k+1} grows beyond 0.5, 1 has to be subtracted in order to obtain the fractional part. This subtraction can be performed without roundoff error. For example, if $\gamma'_{k+1} = 0.625 = 0.1010_2 \cdot 2^0$, the result of subtraction from 1 can be stored without loss of accuracy (the order of the operands are changed for purpose of illustration):

$$\begin{array}{r|l} 1 & 0000 \\ - & 1010 \\ \hline & 0110 \end{array} \tag{22}$$

and $0.0110_2 \cdot 2^0 = 0.375$.

With the compensated summation, the error in the phases has been evaluated again for the example given in Section II-B. Results can be seen in Fig. 4. The maximum error in φ'_k is $3.00 \cdot 10^{-7}$. Since phase information φ'_k is in range $[-\pi; \pi)$, and the LSD in the mantissa of π is $2.38 \cdot 10^{-7}$, the maximum error is smaller than twice this least significant digit. The upper bound on the error in the sine function due to $\Delta\alpha$ is

$$|\sin(\alpha + \Delta\alpha) - \sin\alpha| \approx |\cos\alpha \cdot \Delta\alpha| \leq |\Delta\alpha| \,. \tag{23}$$

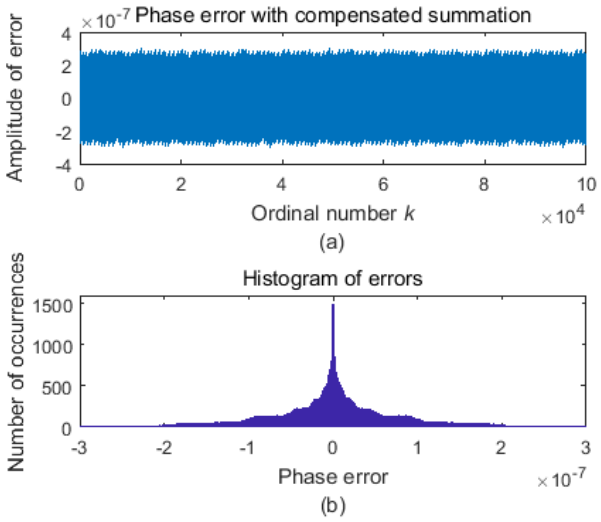

Figure 4. (a) Phase error along the sample set with compensated summation. (b) Histogram of the errors.

Similarly, the upper bound on the error in the cosine function is $\Delta\alpha$, as well. Thus, the error of sine and cosine calculations are also upper bounded by $3.00 \cdot 10^{-7}$. Furthermore, the histogram in Fig. 4b also shows that although the maximum error is $3.00 \cdot 10^{-7}$, the probability of lower errors is much larger – the standard deviation of the errors is $8.32 \cdot 10^{-8}$.

D. Simulation results

In this section, a Monte Carlo simulation will be executed to demonstrate the accuracy of the solution suggested in Section II-C.

To this aim, 10^5 different f_{rel} values were generated with uniform distribution in $[0; 0.5]$. The generated frequency was stored on single precision, and the stored frequency was assumed to be accurate in order to avoid the drift phenomenon that will be discussed in detail in Section III. Phase information was evaluated with $N = 10^5$. First, the maximum absolute error was determined. Results can be seen in Fig. 5a. The error is mapped in $[-\pi; \pi)$ due to the periodic property. For example, if the phase information of single and double precisions are $\varphi'_{k,single} = 0.9\pi$ and $\varphi'_{k,double} = -0.9\pi$, respectively, the absolute error is 0.2π, instead of 1.8π. Fig. 5a shows that the maximum error is always smaller than twice the LSD of the mantissa of π.

In order to further analyze the statistical properties of the evaluation errors, standard deviations values were also determined, see Fig. 5b. The standard deviation of the errors is in the order of magnitude of $eps_s = 1.19 \cdot 10^{-7}$, it is usually

between $7 \cdot 10^{-8}$ and $1.2 \cdot 10^{-7}$. Summing up the results, single precision evaluation with incremental argument calculation can be regarded as precise.

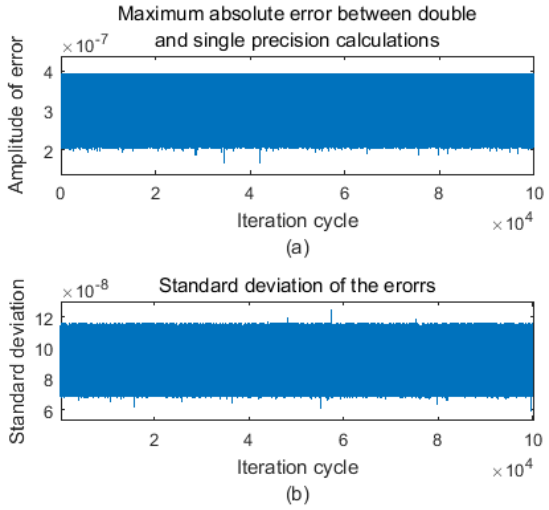

Figure 5. (a) Maximum absolute value and (b) standard deviation of the error in phase information, applying incremental argument calculation with compensated summation.

E. Computational demand

In this section, the computational demand of the splitting technique and the incremental argument calculation will be analyzed by means of theoretical analyses and numerical simulations.

Figs. 1 and 3 show the flowcharts of both methods. Both algorithms consist of simple steps, for instance, multiplication by 2, subtraction of 1, and of conditional statements. The splitting technique executes three or five floating-point operations (FLOPs) in a cycle, depending on the result of conditional statement $x > 1$. Besides, two conditional statements are evaluated. After the splitting, the convolution has to be evaluated. Its computational demand is eight FLOPs, see (10). To calculate the fractional parts, 4 conditional statements have to be evaluated (to decide whether the actual slice is greater than 0.5) and if needed, 4 FLOPs have to be executed to subtract 1. Finally, the slices can be added with 3 FLOPs. The computational demand is mostly determined by the splitting part, since the cycle has to be evaluated $M \cdot B = 22$ times. To sum up, the computational demand of the splitting technique is approximately $22 \cdot 5 = 110$ FLOPs and $22 \cdot 3 = 66$ conditional statement evaluations.

In the incremental phase calculation, 5 FLOPs and 1 conditional statement are needed to calculate the value of γ'_k. At the end of both methods, results have to be multiplied by 2π to get φ_k. This can be evaluated with 1 FLOP.

The advantage of the splitting technique is that γ'_k can be calculated independently of γ'_{k-1}, while in case of incremental argument calculation, the value γ'_k depends on the previous value γ'_{k-1}. Thus, with appropriate programming, for example, with parallel computing, the splitting technique can be fastened. However, it is definitely slower than the incremental phase calculation technique.

In order to show the numerical effectiveness of the incremental argument calculation over the splitting technique, the evaluation of phase information was performed with both algorithms for 1000 different f_{rel} values with $N = 10^5$ using MATLAB R2017a. While the splitting technique evaluation ran for 73.2 seconds, incremental phase calculation finished in 2.55 seconds. Thus, the proposed method is about 30 times faster than the splitting technique. As it was described in

Section II-A, in the splitting technique, more than 80% of the time is spent with splitting the frequency and the time instants. The exact value during the simulation was 86.7%. This means that the effective calculation of the convolution and of the fractional parts was performed in only 1/6 of the time. This explains the large difference between the run time of the algorithms.

III. Evaluation with increased frequency precision

A. *Error due to imprecise frequency representation*

Up to this point, it has been assumed that the frequency of the signal is known precisely. However, in practical situations, this assumption is not certainly fulfilled. In [14], it was shown that the root mean square error (e_{rms}) of the sine wave fitted in LS sense, due to inaccurate knowledge on frequency equals to:

$$e_{rms} = \sqrt{\frac{2}{3}} \cdot RJ\pi \frac{\Delta f}{f} \ . \tag{24}$$

The error can be explained by the phenomenon of drift. For purpose of illustration, an LS fitting with known frequency was performed on a sine wave with the following parameters:

$$A = 0.3, \qquad B = 0.4, \qquad C = 0.5, \qquad f = 57 \text{ Hz}, \qquad \frac{\Delta f}{f} = 10^{-6} \ . \tag{25}$$

Parameter $\Delta f / f$ means that the fitting assumed signal frequency to be f, while the real frequency value was $f + \Delta f$. Sampling parameters were:

$$f_s = 1 \text{ kHz} \ \ N = 1000 \ . \tag{26}$$

The error of fitting can be seen in Fig. 6. It shows that the LS error is minimized so that the fitting error is minimal in the middle of the data set. This way, the drift due to imprecise frequency knowledge is minimized, as well. However, the errors are growing from the middle to the edges of the data set.

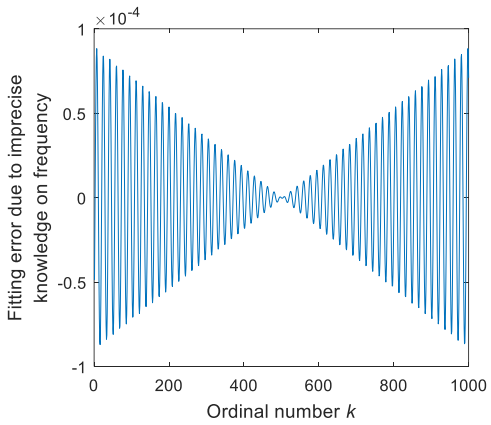

Figure 6. Fitting error due to imprecise knowledge on frequency.

The error is caused by not fulfilling the assumption that the frequency is precise. Imprecise knowledge on frequency can originate from two main sources: imprecise frequency estimation and finite precision number representation. In many cases, the initial frequency estimator is regarded as precise. Interpolated FFT (Fast Fourier Transform) methods can yield such estimators, applying different windows, for instance, rectangular window [15], Hanning window [16] or Blackman-

Harris window [17]. In [18], it was pointed out that using interpolated FFT, as initial frequency estimator, the four-parameter LS method that refines the frequency estimate does not outperform the three-parameter LS method (that estimates *A*, *B* and *C*) significantly, if $N > 512$. Nevertheless, if the initial frequency estimator is not precise enough for the purpose of fitting, it can be refined with iterative methods in time [3] or frequency domain [17]. Detailed investigation of imprecise frequency estimation exceeds the scope of this paper. In the following, the case will be investigated, when imprecise frequency information originates from finite mantissa length, focusing on single precision representation. Namely, double precision representation is so accurate that it does not introduce noticeable error in e_{rms}, according to (24). Contrarily, the accuracy of single precision representation may result in unexpectedly large errors. In worst case situation, the relative frequency error is $eps_s/2 = 5.96 \cdot 10^{-8}$. Depending on the number of sampled periods *J*, the user can decide whether this error is negligible or it has to be taken into consideration. In the following, a method will be presented to mitigate the error due to roundoff errors in the frequency, if this error source cannot be neglected.

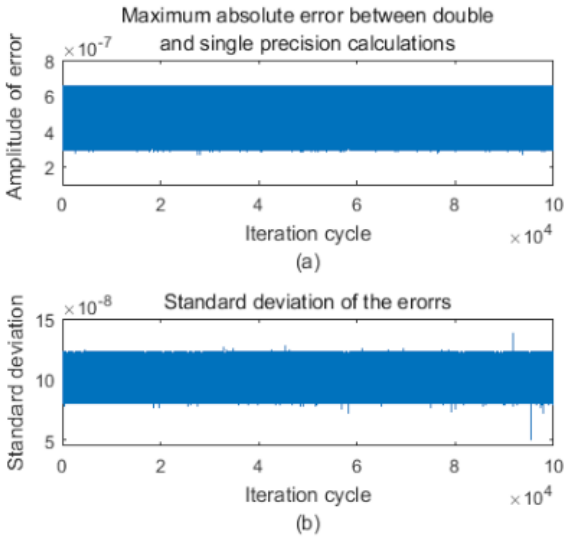

Figure 7. (a) Maximum absolute value and (b) standard deviation of the error in phase information with increased frequency precision, applying incremental argument calculation with compensated summation.

If the relative error of single precision number representation is not sufficiently small, frequency can be represented as the sum of two terms, as described in [6]:

$$f_{rel,prec} = \left[\text{single}(f_{rel}),\quad f_{rel,corr}\right] \ , \tag{27}$$

where $\text{single}(f_{rel})$ is the nearest representable single precision number and $f_{rel,corr}$ is the correction term due to roundoff. Such correction term can be obtained from the four-parameter LS fitting [3].

Certainly, the addition should not be performed on single precision. Namely, it would yield $\text{single}(f_{rel})$. Instead, similarly to the splitting technique in Section II-A, the frequency is stored in two single precision numbers. The value of $f_{rel,corr}$ is by about seven orders of magnitude smaller than that of $\text{single}(f_{rel})$. Since the length of single precision mantissa is 23, the mantissa that can be stored in two single precision numbers is 46-bit long, approaching the accuracy of double precision (53 bits). With these notations, phase information can be calculated by:

$$\varphi_k = 2\pi\left(\text{single}(f_{rel}) \cdot k + \left\langle f_{rel,corr} \cdot k \right\rangle\right) \tag{28}$$

Assuming that $\text{single}(f_{rel}) < 0.5$ holds, $|f_{rel,corr}| < 0.5 eps_s = 5.96 \cdot 10^{-8}$. It follows that $f_{rel,corr} \cdot k < 0.5$ holds for each k, provided that $N < 8 \cdot 10^6$, which is a reasonable assumption. Thus, $f_{rel,corr} k$ can be evaluated without either the splitting technique or the incremental calculation. For the calculation of the fractional part of $\text{single}(f_{rel}) \cdot k$, both the splitting technique and the incremental calculation can be applied. The result can be obtained by adding the two parts and multiplying it by 2π, as (28) shows.

B. Simulation results

In this section, the effectiveness of the evaluation with increased precision frequency will be demonstrated. To this aim, a Monte Carlo simulation with 10^5 different double precision f_{rel} values was run with uniform distribution in $[0; 0.5]$. Phase information was evaluated with $N = 10^5$ with double precision and with single precision, applying the method suggested in Section III-A. The fractional part of $\text{single}(f_{rel}) \cdot k$ was evaluated using incremental argument calculation. The maximum absolute values and standard deviations of the evaluation errors can be seen in Fig.7. In comparison to Fig. 5, both values are slightly larger. Nevertheless, the maximum absolute errors are still smaller than $3 \cdot \text{LSD}(\pi)$ and the standard deviations are still in the order of magnitude eps_s.

In conclusion, single precision phase evaluation is demonstrated to be precise if the exact frequency of the signal can only be represented on double precision, and this frequency is stored in two single precision numbers.

IV. Potential applications

In this section, two application areas will be shown where accurate argument calculation can be utilized: ADC testing and system identification.

In ADC testing, one of the most important results is the effective number of bits (ENOB). It can be defined by [3]:

$$\text{ENOB} = b - \log_2 \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - y_k)^2}}{\frac{Q}{\sqrt{12}}} = b - \log_2 \frac{\sqrt{\frac{1}{N} \cdot \text{CF}_{\text{LS}}}}{\frac{Q}{\sqrt{12}}}, \tag{29}$$

where b is the nominal bit number, and Q is the ideal code bin width of the converter. From a practical point of view, the ENOB value represents the number of bits that contains information on the signal at the input of the quantizer. For instance, if the mean square error of the conversion is twice as large as the mean square error of an ideal quantizer, the ENOB value drops by one.

Imprecise argument calculation influences the result of sine fitting. Its effect is especially significant for long records with single precision evaluation, see (6). To illustrate this effect, 100 noisy sine waves were generated with $N = 2^{16}$ samples, and the mean ENOB values were calculated in three different ways. In Evaluation 1, double precision arithmetic was applied. In Evaluation 2, single precision arithmetic without incremental argument calculation was used. Finally, in Evaluation 3, single precision evaluation was complemented with incremental argument calculation. In order to avoid the drift phenomenon, $f_{rel} = 2^{-5}$ was set so that it can be stored in one single precision number precisely. The nominal bit number of the converter was $b = 12$. Signal parameters were:

$$\text{A} = 0.4 \quad \text{B} = 0.3 \quad \text{C} = 0.5 \ . \tag{30}$$

The distribution of the additive noise was uniform between $-Q/2$ and $Q/2$. This models an ideal quantization. Thus, the real ENOB value was approximately 12. The result of the evaluation can be seen in Table 1. Though there is no

noticeable difference between double and single precision evaluations for $N = 1{,}000$, with increasing record length, single precision evaluation (without incremental argument calculation) yields inaccurate results. For $N = 50{,}000$, the difference between double and single precision evaluations is more than 0.5 bits. However, with incremental argument calculation, results can be evaluated accurately even using single precision arithmetic.

Record length	Eval. 1	Eval. 2	Eval. 3
1,000	12.00	12.00	12.00
10,000	12.00	11.97	12.00
20,000	12.00	11.89	12.00
50,000	12.00	11.43	12.00

Table 1 – Mean ENOB values for different record lengths with uniformly distributed noise (ideal quantization)

The simulation was repeated using additive white Gaussian noise (AWGN) with zero-mean and standard deviation $\sigma = Q$. Results are delineated in Table 2. This simulation shows that imprecise argument calculation affects the result of the evaluation much less, if the amplitude of the additive noise is increased.

The increase in the expected value of the CF due to imprecise argument calculation can be estimated by (6). If it is negligible compared to $\mathrm{CF}_{\mathrm{LS}}$, then it does not influence the result of the fitting, and the value of the ENOB considerably. Contrarily, if the noise level is low, and therefore $\mathrm{CF}_{\mathrm{LS}}$ is small, then the effect of imprecise argument calculation has to be mitigated. By this means, both the increase in the mean value and the raggedness of the CF can be decreased significantly [6][7].

Record length	Eval. 1	Eval. 2	Eval. 3
1,000	10.21	10.21	10.21
10,000	10.21	10.21	10.21
20,000	10.21	10.19	10.21
50,000	10.21	10.14	10.21

Table 2 – Mean ENOB values for different record lengths with AWGN

Another area where the effect of imprecise argument calculation should be considered is system identification. In frequency domain system identification, multi-sinusoidal excitation is widely used [20]:

$$u(t_k) = \sum_{n=1}^{M} R \cdot \sin(2\pi f n t_k + \phi_n) \ , \tag{31}$$

where $u(t_k)$ denotes the excitation signal at time instant t_k, ϕ_n is the initial phase of the *n*th harmonic component. The excitation signal consists of *M* sinusoidal components. If the excitation signal is generated digitally, and it is converted by a digital-to-analog converter (DAC), then the result is affected by imprecise argument calculation. In this case, the longer the record, the larger the error due to imprecise argument calculation. Furthermore, it is obvious that due to multiplication factor *n*, the magnitude of the error increases in the higher-order harmonics.

Although the effect of imprecise argument calculation is important to consider in case of single precision evaluation of long ($N > 10{,}000$) records, if *J* or *N* is decreased, or double precision evaluation is applied, this error source can be neglected, see (6). Thus, while in case of single precision evaluation, the user has to decide whether the effect of imprecise

argument calculation can be neglected or it should be mitigated, in case of double precision evaluation, this error source results in negligible disturbances.

CONCLUSIONS

In this paper, accurate argument calculation for sine fitting algorithms was investigated, assuming floating point arithmetic. The splitting technique was shown to have a bottleneck at the calculation of the slices. An easy-to-implement incremental argument calculation technique was suggested that maps the result in $[-\pi;\ \pi)$. It was shown that this method can result in a growing phase error due to the accumulation of roundoff errors. In order to obtain precise results, compensated summation was applied. This technique stores the roundoff error at each summation step, and compensates for it at the next step. Simulations showed that with this supplement, even single precision evaluation can be regarded as precise. Furthermore, theoretical and numerical analyses were carried out to highlight performance increase compared to the splitting technique. The analysis showed that incremental phase calculation can be evaluated about 30 times faster than with the splitting technique. Besides, a method was suggested to be able to calculate phase information precisely, even if the frequency of the sinusoidal waveform cannot be represented precisely in one floating point number. Results were verified through simulations. Finally, two possible application areas were shown to demonstrate the applicability of the suggested solutions in the state-of-art measurement procedures.

It was pointed out that double precision number representation is precise enough to neglect the effect of imprecise argument calculation and imprecise frequency representation. Contrarily, single precision evaluation can introduce noticeable errors. The magnitude of this error was estimated in former research, and the derivation was generalized in this paper. Consequently, during measurements, the user of sine fitting can decide whether the error sources has to be compensated in the actual measurements or they inject negligible errors compared to the measurement noise. Nevertheless, with the proposed methods, these errors can be mitigated significantly. By this means, the cost of equipment that is needed to evaluate sine fitting can be reduced, as well.

APPENDIX

The expected value of the increase in the least squares cost function ($\mathrm{CF_{LS}}$) is investigated. In [6], this value was determined for a purely sinusoidal waveform, where $B = 0$. In this section, a derivation is provided for a general sine wave, extending the ideas in [6].

The fitted sine wave due to the roundoff error of φ_k equals to:

$$\begin{aligned} & A\cos\{\varphi_k + (\Delta\varphi)_k\} + B\sin\{\varphi_k + (\Delta\varphi)_k\} + C \\ \approx\ & A\cos(\varphi_k) - A\sin(\varphi_k)\cdot(\Delta\varphi)_k + B\sin(\varphi_k) + B\cos(\varphi_k)\cdot(\Delta\varphi)_k + C \\ =\ & A\cos(\varphi_k) + B\sin(\varphi_k) + C + [B\cos(\varphi_k) - A\sin(\varphi_k)](\Delta\varphi)_k \end{aligned} \tag{32}$$

where $(\Delta\varphi)_k$ is the roundoff error in φ_k. Comparing this result to (1), the roundoff error in the fitted sine wave due to $(\Delta\varphi)_k$ is:

$$e_{phase,k} \approx [B\cos(\varphi_k) - A\sin(\varphi_k)](\Delta\varphi)_k\ . \tag{33}$$

The increase in the expected value of $\mathrm{CF_{LS}}$ equals to the expected value of the squared sum of $e_{phase,k}$ [6]:

$$E\{\Delta \mathrm{CF_{LS}}\} = E\left\{\sum_{k=1}^{N} e_{\mathrm{phase},k}^2\right\} . \tag{34}$$

From (33), we obtain:

$$\begin{aligned} E\left\{\sum_{k=1}^{N} e_{\mathrm{phase},k}^2\right\} &\approx E\left\{\sum_{k=1}^{N} ([B\cos(\varphi_k) - A\sin(\varphi_k)](\Delta\varphi)_k)^2\right\} \\ &= \sum_{k=1}^{N} [B^2\cos^2(\varphi_k) + A^2\sin^2(\varphi_k) - 2AB\sin(\varphi_k)\cos(\varphi_k)] \cdot E\{(\Delta\varphi_k)^2\} , \end{aligned} \tag{35}$$

The distribution of $(\Delta\varphi)_k$ can be regarded as independent uniform distribution in $[-\mathrm{LSD}\{(\Delta\varphi)_k\}/2; \mathrm{LSD}\{(\Delta\varphi)_k\}/2]$ [11]. Thus, its squared expected value equals to [11]:

$$E\{(\Delta\varphi_k)^2\} = \frac{\mathrm{LSD}\{(\Delta\varphi)_k\}^2}{12} \approx \varphi_k^2 \frac{eps^2}{12} , \tag{36}$$

where *eps* is the relative error of the floating point number representation. With this approximation:

$$\begin{aligned} E\{\Delta \mathrm{CF_{LS}}\} &\approx \sum_{k=1}^{N} [B^2\cos^2(\varphi_k) + A^2\sin^2(\varphi_k) - 2AB\sin(\varphi_k)\cos(\varphi_k)] \cdot \varphi_k^2 \frac{eps^2}{12} \\ &= \sum_{k=1}^{N} \left[B^2 \frac{1+\cos 2\varphi_k}{2} + A^2 \frac{1-\cos 2\varphi_k}{2} - AB\sin(2\varphi_k)\right] \cdot \varphi_k^2 \frac{eps^2}{12} \\ &= \sum_{k=1}^{N} \left[\frac{B^2+A^2}{2} + \cos(2\varphi_k)\frac{B^2-A^2}{2} - \sin(2\varphi_k)\,AB\right] \cdot \varphi_k^2 \frac{eps^2}{12} \\ &= \sum_{k=1}^{N} \left[\frac{R^2}{2} + \cos(2\varphi_k)\frac{B^2-A^2}{2} - \sin(2\varphi_k)\,AB\right] \cdot \varphi_k^2 \frac{eps^2}{12} . \end{aligned} \tag{37}$$

where R is the amplitude of the sine wave $R = \sqrt{A^2 + B^2}$. Formulas for squared cosine and sine values can be found in [19]. Assuming that N is large, and a high number of periods are sampled, the sum of cosinusoidal and sinusoidal terms can be neglected beside the sum of $R^2/2$. Thus, we get:

$$E\{\Delta \mathrm{CF_{LS}}\} \approx \sum_{k=1}^{N} \frac{R^2}{2} \cdot \varphi_k^2 \frac{eps^2}{12} = \sum_{k=1}^{N} \frac{R^2}{2} \cdot k^2\varphi_1^2 \frac{eps^2}{12} = \frac{R^2}{2} \cdot \varphi_1^2 \frac{eps^2}{12} \sum_{k=1}^{N} k^2 . \tag{38}$$

It is known that [19]:

$$\sum_{k=1}^{N} k^2 = \frac{N(N+1)(2N+1)}{6} \approx \frac{N^3}{3} . \tag{39}$$

Thus, the increase in the expected value of $\mathrm{CF_{LS}}$ is approximately:

$$E\{\Delta \mathrm{CF_{LS}}\} \approx \frac{R^2}{2} \cdot \varphi_1^2 \frac{eps^2}{12} \frac{N^3}{3} , \tag{40}$$

Considering that

$$\varphi_1 = 2\pi f_{rel} = 2\pi \frac{J}{N} , \tag{41}$$

the expected value of the increase in $\mathrm{CF_{LS}}$ is

$$E\{\Delta \mathrm{CF_{LS}}\} = \frac{\pi^2 R^2 J^2 eps^2 N}{18} \ . \tag{42}$$

ACKNOWLEDGEMENT

This work was partially supported by the Hungarian Research Fund (OTKA) under grant K–115820 and by the Pro Progressio Foundation.

REFERENCES

[1] T. Radil, P. Ramos, A. Cruz, "Impedance Measurement With Sine-Fitting Algorithms Implemented in a DSP Portable Device", *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1,pp. 197–204, Jan. 2008. doi: 10.1109/TIM.2007.908276

[2] T. Radil, P. Ramos, A. Cruz, "Detection and extraction of harmonic and non-harmonic power quality disturbances using sine fitting methods", *ICHQP 13th International Conference on Harmonics and Quality of Power*, Wollongong, Australia, 2008. doi: 10.1109/ICHQP.2008.4668813

[3] Standard IEEE-1241-2010, "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters", (2011) doi: 10.1109/IEEESTD.2011.5692956

[4] IEEE Standard-1057-2007, „Standard for Digitizing Waveform Recorders", 2007, doi: 10.1109/IEEESTD.2008.4494996

[5] L. Balogh, I. Kollár, A. Sárhegyi. "Maximum likelihood estimation of ADC parameters." Proceedings of *IEEE Instrumentation and Measurement Technology Conference (I2MTC),* pp. 24-29, Austin, USA, 2010, doi: 10.1109/IMTC.2010.5488286

[6] B. Renczes, I. Kollár, A. Moschitta, P. Carbone, "Numerical Optimization Problems of Sine Wave Fitting Algorithms in the Presence of Roundoff Errors", *IEEE Transactions on Instrumentation and Measurement,* vol. 65, no. 8., pp. 1785-1795, 2016, DOI: 10.1109/TIM.2016.2562218

[7] B. Renczes, I. Kollár, "Roundoff Errors in the Evaluation of the Cost Function in Sine Wave Based ADC Testing", *Proceedings of 20th IMEKO TC4 International Symposium and 18th International Workshop on ADC Modelling and Testing*, Benevento, Italy, Sep. 15-17, 2014. pp. 248-252, Paper 214.

[8] IEEE Standard-754-2008, „IEEE Standard for Floating-Point Arithmetic", 2008, DOI: 10.1109/IEEESTD.2008.4610935

[9] P. Ramos, T. Radil, F. Janeiro, "Implementation of sine-fitting algorithms in systems with 32-bit floating point representation", Measurement 45, pp. 155-163, 2012. doi:10.1016/j.measurement.2011.05.011

[10] A. Mehta, C. B. Bidhul, S. Joseph, P. Jayakrishnan, "Implementation of single precision floating point multiplier using Karatsuba algorithm", *IEEE International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*, pp. 254-256, 2013, doi: 10.1109/ICGCE.2013.6823439

[11] B. Widrow, I. Kollár, "Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications", *Cambridge University Press*, Cambridge, UK, 2008, doi: 10.1017/CBO9780511754661

[12] N. Higham, "The accuracy of floating point summation", *SIAM Journal on Scientific Computing*, vol. 14, no. 4: pp. 783–799, 1993. DOI: 10.1137/0914050

[13] W. Kahan, "Further remarks on reducing truncation errors", *Communications of the ACM*, vol. 8, no. 1: p. 40, 1965. DOI: 10.1145/363707.363723

[14] T. Bilau, T. Megyeri, A. Sárhegyi, J. Márkus, I. Kollár, „Four-parameter fitting of sine wave testing result: iteration and convergence", *Computer Standards & Interfaces*, vol. *26, no.* 1, pp. 51-56, 2004, DOI: 10.1016/S0920-5489(03)00062-X

[15] H. Renders, J. Schoukens, G. Vilain, „High-Accuracy Spectrum Analysis of Sampled Discrete Frequency Signals by Analytical Leakage Compensation", *IEEE Trans Instrum. Meas.*, Vol. 33, No. 4, pp. 287-292, Dec. 1984, DOI: 10.1109/TIM.1984.4315226

[16] T. Grandke, „ Interpolation Algorithms for Discrete Fourier Transforms of Weighted Signals", *IEEE Trans Instrum. Meas.*, Vol. 32, No. 2, pp. 350-355, Jun. 1983, DOI: 10.1109/TIM.1983.4315077

[17] V. Pálfi and I. Kollár, "Acceleration of the ADC Test With Sine-Wave Fit", *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no 5, pp. 880-888, 2013 DOI: 10.1109/TIM.2013.2243500

[18] D. Belega, D. Dallet, D. Petri „Performance comparison of the three-parameter and the fourparameter sine-fit algorithms", *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, 2011, DOI: 10.1109/IMTC.2011.5944010

[19] I. Gradshteyn, I. Ryzhik, „Table of integrals, series, and products, Fifth Edition", *Academic press*, London, UK, 1994

[20] R. Pintelon, J. Schoukens, "System Identification: A Frequency Domain Approach", *Wiley IEEE-Press*, Hoboken, NJ, 2012