XIX IMEKO World Congress
Fundamental and Applied Metrology
September 6–11, 2009, Lisbon, Portugal

IDENTIFICATION OF MEASUREMENT DATA PROCESSING ALGORITHM COEFFICIENTS PRESENTED ON SELECTED FORM OF FFT ALGORITHM

Krzysztof Konopka[1], Tadeusz Topór-Kamiński[2]

[1] Institute of Measurement Science, Electronics and Control, Silesian University of Technology, Gliwice, Poland, krzysztof.konopka@polsl.pl
[2] Institute of Measurement Science, Electronics and Control, Silesian University of Technology, Gliwice, Poland, tadeusz.topor-kaminski@polsl.pl

Abstract – Nowadays measurement data coming from all kinds of measurement systems is usually processed by algorithms. These algorithms are often delivered to the user as complex program and their numerical structure is not known. Therefore also influence of algorithm on an accuracy of processed data is not known. Coefficient matrix of algorithm represents its numerical operations and it can be a basis to algorithm accuracy evaluation. The paper presents a method how to identify this coefficient matrix. As an example this method is used to identify an FFT algorithm implemented in LabVIEW.

Keywords: algorithm uncertainty, algorithm matrix form, algorithm identification

1. INTRODUCTION

Measurement systems and instruments use many different algorithms for calculations on measured quantity values. Generally the measurement data processing algorithm is a mathematical formula, which converts one data sequence into another [1]. Next section presents how the algorithm operations can be described in a matrix form. Matrix form of algorithm facilitate analysis of its metrological properties. Knowledge of algorithm coefficient matrix helps in determining how this algorithm transfers errors from input to the output. The coefficients of the matrix can be calculated using algebraic form of the algorithm if it is known. However in many cases algorithms are implemented in different software environments and their coefficients calculations are not overt. The paper describes an identification method of these coefficients, and the algorithm can be described just as a black box with inputs and outputs. This method enables quick, current calculations of algorithm coefficients in particular measurement situation and subsequently using this coefficients to calculate current uncertainty of measurement values.

2. MATRIX FORM OF ALGORITHM

Generally it can be assumed that the algorithm converts quantity $x(t)$, continuous in t domain, to another quantity $X(\omega)$, continuous in ω domain [2].

In practice algorithm operates on series of discrete samples of input quantity. The output values of algorithm have also discrete form. Number of samples the algorithm operates on is limited (input window). Also output quantity has limited representation (output window). Considerations in the paper are limited to rectangular input window. It means that algorithm input data consist of input quantity samples $x(0), x(1), \ldots, x(K-1)$, where K means the number of input window samples. On the output of the algorithm one gets N element algorithm output data series $\{X(n)\} := \{X(0), X(1), \ldots, X(N-1)\}$ which can be interpreted as the discrete representation of algorithm output value X, continuous in ω domain

In the situation presented one can describe algorithm data processing as a matrix equation [1][3]

$$\begin{bmatrix} X(0) \\ X(1) \\ . \\ . \\ X(N-1) \end{bmatrix} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,K-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,K-1} \\ & & . & \\ & & . & \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,K-1} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ . \\ . \\ x(K-1) \end{bmatrix}, \quad (1)$$

where $a_{0,0,}$, $a_{0,K-1}$, $a_{N-1,K-1}$ are the algorithm coefficients. Algorithm input and output data series are presented in (1) as vectors. When one denotes this vectors respectively as **x** and **X,** the data processing algorithm represented by equation (1) can be written as

$$\mathbf{X} = \mathbf{A}\mathbf{x}. \quad (2)$$

Complex output data.

Output data of some algorithms, for example DFT, are presented in complex form. In such case data processing can be described by two algorithms, where one determines real values and the second imaginary ones. Such algorithms can be presented as two matrix equations

ISBN 978-963-88410-0-1

$$\mathbf{X}_{\mathrm{Re}} = \mathbf{A}_{\mathrm{Re}}\mathbf{x}, \qquad (3)$$

$$\mathbf{X}_{\mathrm{Im}} = \mathbf{A}_{\mathrm{Im}}\mathbf{x}. \qquad (4)$$

Output data of these algorithms form complex number series, which can be presented as vector

$$\mathbf{X} = \mathbf{X}_{\mathrm{Re}} + j\mathbf{X}_{\mathrm{Im}}. \qquad (5)$$

Single and multipoint algorithms.

One can indicate a group of algorithms which produce a single output value – for example filtration algorithms, average or rms calculating algorithms etc. This kind of algorithm is called *singlepoint* algorithm.

In another case, when one run of algorithm produces a set of values, such algorithm is called *multipoint* algorithm [1][3].

Multipoint algorithm can be presented as a set of parallel singlepoint algorithms which operate on the same input values series [1]. Every singlepoint algorithm is represented by corresponding coefficient vector ${}^{0}\mathbf{A}, {}^{1}\mathbf{A} \ldots {}^{N\text{-}1}\mathbf{A}$ which is a single row of coefficient matrix $\mathbf{A}$ in (1). Execution of every singlepoint algorithm generates one value, which can be described for n-th row of coefficient matrix $\mathbf{A}$ as

$$X(n) = \left[a_{n,0}\ a_{n,1}\ \ldots\ a_{n,K-1}\right]\left[x(0)\ x(1)\ \ldots\ x(K-1)\right]^{\mathrm{T}} = {}^{n}\mathbf{A}\,\mathbf{x}^{\mathrm{T}}, \qquad (6)$$

where ${}^{n}\mathbf{A} = \left[a_{n,0}\ a_{n,1}\ \ldots\ a_{n,K-1}\right]$ is a coefficient vector of singlepoint algorithm represented by n-th row of matrix **A,** $a_{n,0}\ a_{n,1}\ \ldots\ a_{n,K-1}$ are constant coefficients of singlepoint algorithm, $\mathbf{x}^{\mathrm{T}} = \left[x(0)\ x(1) \ldots x(K-1)\right]^{\mathrm{T}}$ is the algorithm input values vector.

Dependence (6) can be also written in other form

$$X(n) = a_{n,0}x(0) + a_{n,1}x(1) + \ldots + a_{n,K-1}x(K-1) = \sum_{k=0}^{K-1} a_{n,k}x(k). \qquad (7)$$

Dependence (7) is a general singlepoint algorithm processing equation, which form the base for further metrological analysis, where it is assumed, that multipoint algorithm is considered as set of independent singlepoint algorithms [1]. It means that output data from multipoint algorithm is considered as a set of independent numbers.

Considering multipoint algorithm as a set of singlepoint ones considerably simplifies its metrological analysis with no influence on the generality of considerations. That is because usually analysis of singlepoint algorithm gives information about structure and many properties of the whole multipoint algorithm [1][3].

In further considerations it is assumed, that algorithm input values form a series of instantaneous measurement results of changing in time quantity. Moreover it is assumed that algorithm coefficients values are constant and independent from both input values series $\{x(k)\}$ and output values series $\{X(n)\}$.

3. IDENTYFICATION OF ALGORITHM COEFFICIENTS

As mentioned in introduction, to determine metrological properties of algorithm one needs to identify its coefficient matrix values. This identification can be achieved by giving to the algorithm input appropriate test series. An example of coefficient identification method is presented on FFT algorithm, which is a transformation algorithm. Specifically there is identified FFT implementation in LabVIEW environment.

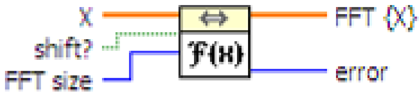

Fig. 1. Icon representing FFT algorithm in LabVIEW.

FFT is a generally known algorithm. It transforms data from time domain to frequency domain. General equation of discrete Fourier transform has a form:

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{-j2\pi nk/N}, \text{for } k = 0, 1, \ldots, N\text{-}1. \qquad (8)$$

In general case this algorithm processes series of N complex input values into series of N complex output values.

Considering matrix form of algorithm (1) one can notice that if the input vector has form $\left[x(0)\ x(1)\ \ldots\ x(N-1)\right]^{\mathrm{T}} = \left[1\ 0\ \ 0 \ldots 0\right]^{\mathrm{T}}$, than output vector will have values equal to values of coefficients from the first

Table 1. Exemplary FFT coefficient matrix **A** for N=10.

0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i	0,1 +0 i
0,1 +0 i	0,081 -0,059 i	0,031 -0,095 i	-0,031 -0,095 i	-0,081 -0,059 i	-0,1 +0 i	-0,081 +0,059 i	-0,031 +0,095 i	0,031 +0,095 i	0,081 +0,059 i
0,1 +0 i	0,031 -0,095 i	-0,081 -0,059 i	-0,081 +0,059 i	0,031 +0,095 i	0,1 +0 i	0,031 -0,095 i	-0,081 -0,059 i	-0,081 +0,059 i	0,031 +0,095 i
0,1 +0 i	-0,031 -0,095 i	-0,081 +0,059 i	0,081 +0,059 i	0,031 -0,095 i	-0,1 +0 i	0,031 +0,095 i	0,081 -0,059 i	-0,081 -0,059 i	-0,031 +0,095 i
0,1 +0 i	-0,081 -0,059 i	0,031 +0,095 i	0,031 -0,095 i	-0,081 +0,059 i	0,1 +0 i	-0,081 -0,059 i	0,031 +0,095 i	0,031 -0,095 i	-0,081 +0,059 i
0,1 +0 i	-0,1 +0 i	0,1 +0 i	-0,1 +0 i	0,1 +0 i	-0,1 +0 i	0,1 +0 i	-0,1 +0 i	0,1 +0 i	-0,1 +0 i
0,1 +0 i	-0,081 +0,059 i	0,031 -0,095 i	0,031 +0,095 i	-0,081 -0,059 i	0,1 +0 i	-0,081 +0,059 i	0,031 -0,095 i	0,031 +0,095 i	-0,081 -0,059 i
0,1 +0 i	-0,031 +0,095 i	-0,081 -0,059 i	0,081 -0,059 i	0,031 +0,095 i	-0,1 +0 i	0,031 -0,095 i	0,081 +0,059 i	-0,081 +0,059 i	-0,031 -0,095 i
0,1 +0 i	0,031 +0,095 i	-0,081 +0,059 i	-0,081 -0,059 i	0,031 -0,095 i	0,1 +0 i	0,031 +0,095 i	-0,081 +0,059 i	-0,081 -0,059 i	0,031 -0,095 i
0,1 +0 i	0,081 +0,059 i	0,031 +0,095 i	-0,031 +0,095 i	-0,081 +0,059 i	-0,1 +0 i	-0,081 -0,059 i	-0,031 -0,095 i	0,031 -0,095 i	0,081 -0,059 i

column of matrix **A**. If one shifts '1' in input vector to the successive positions one gets successive columns of coefficient martix **A**. Generally test series for λ column determining has a form:

$$v_\lambda = \begin{cases} 1 & \text{for} \quad k = \lambda \\ 0 & \text{for} \quad k \neq \lambda \end{cases} \tag{9}$$

where λ = 0, 1, ..., *N*-1 is a number of test series and *k* = 0, 1, ... , *N*-1 is a number of term in series.

Repeating the operation *N* times one obtains the whole coefficient matrix **A**. Exemplary matrix for *N*=10 is presented in table 1.

The matrix presented in table 1 is only an example and *N* value is selected for presentation. Values of coefficients for another *N* values can be calculated currently depending on measurement situation. Coefficients values depend only on the length of the input vector. Presented algorithm coefficients identification method enables to build a subroutine which calculates coefficients values currently as required.

4. EXEMPLARY APPLICATION OF IDENTIFIED ALGORITHM COEFFICIENTS

Coefficient of matrix **A** calculated in section 3 are used to determine propagation of uncertainty caused by random errors through FFT algorithm. Specific data processing in algorithms causes, that it is convenient to consider uncertainty as a parameter of error values set, even if this error is only theoretical and impossible to determine, but it facilitates processing analysis and later transition to description by uncertainties [1][4][5]. Knowledge about error propagation enables to determine algorithm output uncertainty basing on input uncertainties and on input errors distribution shapes.

Therefore algorithm uncertainty model is based on error model presented in fig. 2. In this model algorithm transfers errors from the input to the output with appropriate coefficient and also introduces its own errors.

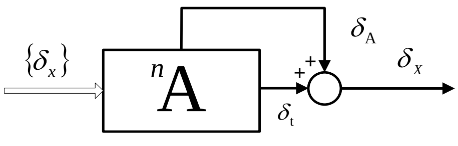

Fig. 2. General algorithm error model. $\{\delta_x\}$ is an algorithm input data error sequence, δ_t is an error resulting from data errors transfer from input to output, δ_A is an error added by algorithm.

Errors can be divided into three categories regarding the way they are processed by algorithms – static, dynamic and random errors [1][4][5][9]. Uncertainty can be considered as a measure characterizing error set [1][5][8], therefore also uncertainties can be categorized as static, dynamic and random.

This section considers propagation of random errors, it means errors which can be considered in probabilistic categories. Basing on [1], the dependence between random error variances on the input and the output of the algorithm is:

$$\sigma_X^2 = A^2 \sigma_x^2 . \tag{10}$$

where *A* is a square root of the sum of squared singlepoint algorithm coefficients $A = \sqrt{\sum_{k=0}^{K-1} a_k^2}$. Therefore when data contains random errors, uncertainty propagation can be calculated basing on appropriate variances.

When random errors are concerned one meets usually one of two situations. First - when quantization errors dominate and then input error distribution shape can be approximated by uniform distribution [10]. Second one is when noise dominates and error distribution shape is normal.

Assuming, that algorithm has more than three coefficients and the coefficient values do not differ from each other excessively one can assume, that probability density function of algorithm output error $g(\delta_X)$ has approximately normal distribution, no matter if input errors have normal or uniform distribution. This results from central limit theorem [6][7]. Uncertainty with confidence level $\alpha = 0{,}95$ calculated using definition [1][8] for normal distribution, and presented as multiplicity of standard deviation, is [7]:

$$\mathrm{U}(X) = 1{,}96 \cdot \sigma_X . \tag{11}$$

Taking into consideration expression (10), equation (11) can be noted as

$$\mathrm{U}(X) = 1{,}96 \cdot A \cdot \sigma_x . \tag{12}$$

Uncertainty with confidence level $\alpha = 0{,}95$ calculated for random error which probability density function is uniform is [7]

$$\mathrm{U_q}(x) = 1{,}65 \cdot \sigma_x . \tag{13}$$

Determining proportion of output uncertainty (12) and input uncertainty (13), one obtains coefficient k_q which specifies propagation of uncertainty caused by quantization error from algorithm input to the output:

$$k_\mathrm{q} = \frac{\mathrm{U}(X)}{\mathrm{U_q}(x)} = \frac{1{,}96 \cdot A \cdot \sigma_x}{1{,}65 \cdot \sigma_x} = 1{,}19 \cdot A . \tag{14}$$

Therefore coefficient k_q value is directly proportional to value *A*, which is an important parameter of every algorithm. Basing on *A* one can specify quantitatively uncertainty propagation through algorithm.

In case when noise dominates, propagation coefficient can be written as

$$k_\mathrm{r} = \frac{\mathrm{U}(X)}{\mathrm{U_r}(x)} = \frac{1{,}96 \cdot A \cdot \sigma_x}{1{,}96 \cdot \sigma_x} = A . \tag{15}$$

Presented considerations show, that when **A** matrix coefficients are identified and the parameters of input uncertainty sources are known, then output uncertainty can be calculated.

Example I: Input values of algorithm contain quantization error of 12-bit AD converter. Converter input voltage range is $U_{IN} = -1, \ldots +1$ V, so quantum value is $q = \frac{1-(-1)}{2^{12}} = 4{,}883 \cdot 10^{-4}$ V. Uncertainty on the input of algorithm is $U_q(x) = 0.95 \cdot \frac{1}{2} \cdot 4.883 \cdot 10^{-4}\,V = 2.32 \cdot 10^{-4}\,V$, and the standard deviation $\sigma_{q,x} = U_q(x)/1{,}65 = 1{,}41 \cdot 10^{-4}$ V. Assuming that AD conversion results are processed by FFT algorithm described in section 3 and its input window is 1024, value $A_{Re,1}$ for the first row of coefficient matrix **A** is $A_{Re,1}$ = 0,0220971. When quantization error dominates, then uncertainty of the first element of the output vector is calculated as follows:

$$U_q\left(X(1)_{Re}\right) = 1{,}96 \cdot A_{Re,1} \cdot \sigma_x = 6{,}09 \cdot 10^{-6}. \qquad (17)$$

The same calculations can be made for other algorithm output values.

Example II: Errors in input values of algorithm are dominated by gaussian noise. Standard deviation of the noise is for example $\sigma_{r,x} = 5 \cdot 10^{-3}$ V. Input uncertainty is then $U_r(x) = 1{,}96 \cdot 5 \cdot 10^{-3}\,V = 9{,}8 \cdot 10^{-3}$ V. Uncertainty of the first element of FFT output vector is calculated as:

$$U_r\left(X(1)_{Re}\right) = 1{,}96 \cdot A_{Re,1} \cdot \sigma_x = 2{,}17 \cdot 10^{-4}. \qquad (18)$$

Experiments.

Results (17) and (18) were checked experimentally. Experiments can be done in two ways presented on fig. 3.

Procedure presented on fig. 3a) replicates normal functioning of measurement data processing. Standard signal is processed by algorithm twice – with added random errors having known distribution, and without these errors. Difference of output values (in presented examples it is the

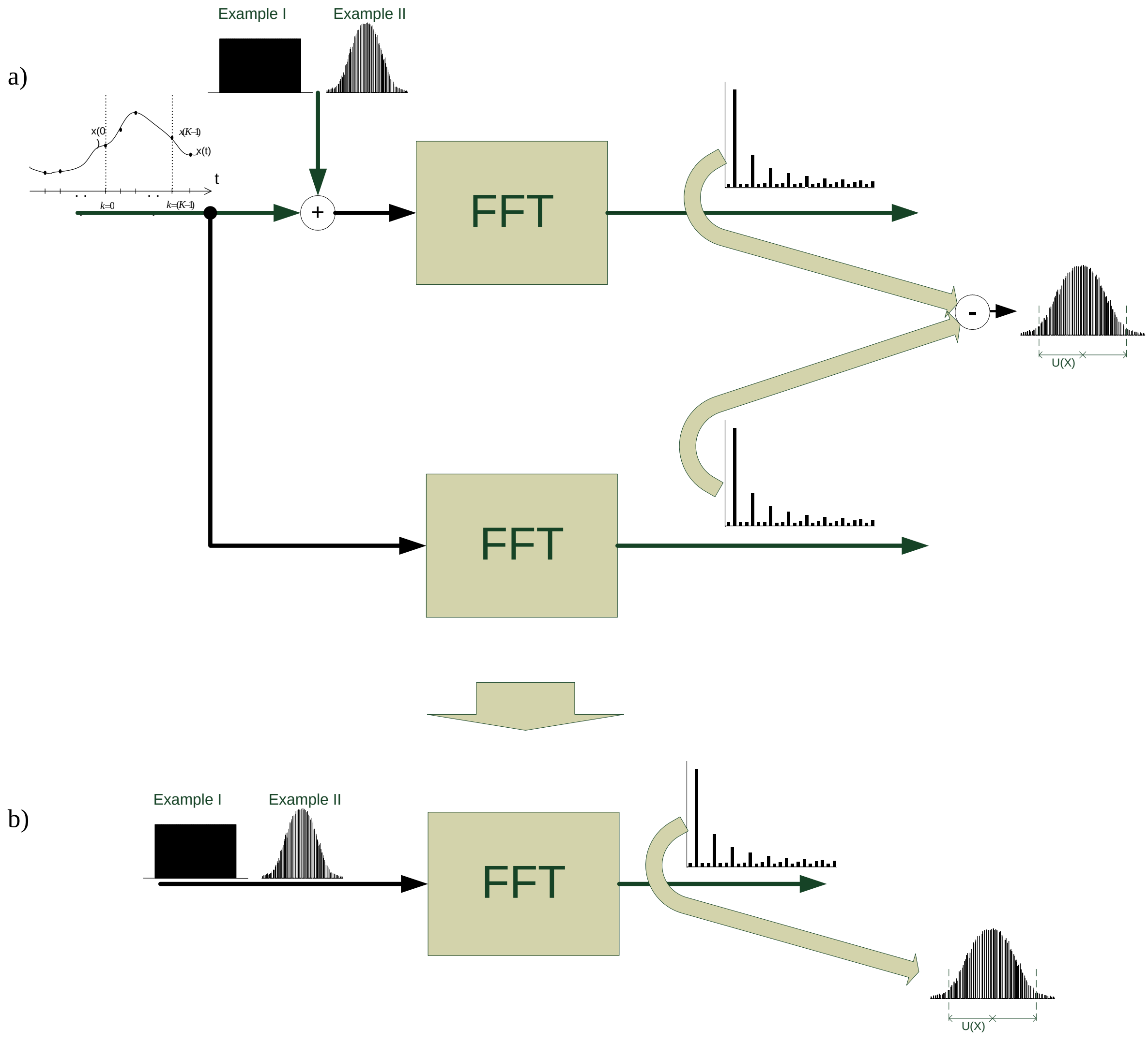

Fig. 3. Scheme of experimental determination of FFT output first element uncertainty. a) Input values contain random errors. Output values are compared with results without random errors. Histogram of their difference is a base for uncertainty evaluation. Algorithm own errors are eliminated; b) Pure noise on algorithm input. Output contains both transferred input errors and errors inserted by algorithm When algorithm own errors are small, both procedures give the same results.

difference between the first elements of the output vectors) is an output error. Basing on set of these errors standard deviation and uncertainty can be calculated.

In second procedure presented on fig. 3b) the input quantity of FFT is pure noise of desired distribution (uniform or normal). Then basing on the output value histogram one can determine the standard deviation and uncertainty.

Experiments presented in the paper were made using the second procedure (fig. 3b). Results for errors which have uniform distribution and presented earlier parameters (range $\pm 2{,}4415 \cdot 10^{-4}$) are: standard deviation of output quantity $\sigma_{q,X_{Re}} = 3{,}11 \cdot 10^{-6}$ and uncertainty of the first element of algorithm output vector $U_q\left(X(1)_{Re}\right) = 6{,}10 \cdot 10^{-6}$. Results for errors which have normal distribution and standard deviation $\sigma_{r,x} = 5 \cdot 10^{-3}\,V$ are: standard deviation of output quantity $\sigma_{r,X_{Re}} = 1{,}11 \cdot 10^{-4}$ and uncertainty of the first element of algorithm output vector $U_q\left(X(1)_{Re}\right) = 2{,}17 \cdot 10^{-4}$.

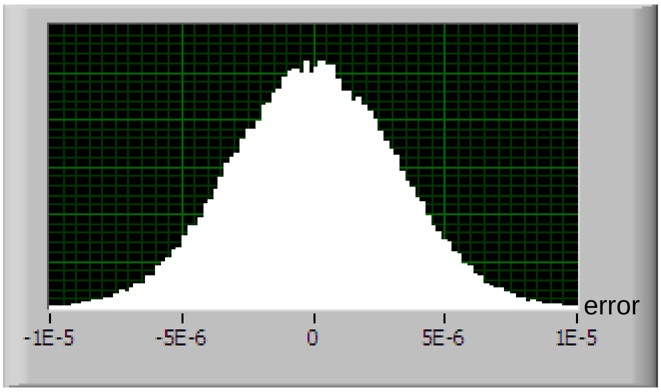

Fig. 4. The error histogram of real part of output FFT vector first element (window of 1024 samples). In the input data uniformly distributed errors dominate. Errors are in range $\pm 2{,}4415 \cdot 10^{-4}$.

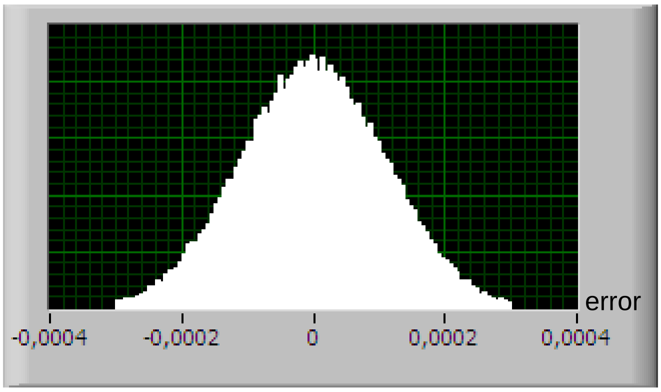

Fig. 5. The error histogram of real part of output FFT vector first element (window of 1024 samples). In the input data errors having normal distribution dominate. Standard deviation is $\sigma_{r,x} = 5 \cdot 10^{-3}\,V$.

5. CONCLUSIONS

The paper presents the identification method of data processing algorithm coefficient matrix. This method enables easy determination of matrix coefficients, even when algorithm structure is not known. Knowledge about the length of input and output vector is sufficient. Presented method can be implemented as a subroutine which automatically calculates current values of coefficient matrix **A** depending on changing operating conditions.

Basing on determined matrix coefficients one can determine propagation of errors of different kinds through the algorithm. Further analysis determines also uncertainty propagation through algorithm. Nowadays measurement data processing algorithms are generally used, therefore analysis of uncertainty propagation through algorithms is important, although often omitted. When algorithms influence in error budget and in uncertainty budget is omitted, it can cause false estimation of measurement accuracy, as different kinds of errors and uncertainties can be considerably amplified or attenuated. Presented in the paper algorithm coefficients identification method can considerably facilitate uncertainty determining process and the error estimation.

REFERENCES

[1] J. Jakubiec, *Application of reductive interval arithmetic to uncertainty evaluation of measurement data processing algorithms*, Monograph. Wydawnictwo Politechniki Śląskiej, Gliwice 2002.

[2] J. Jakubiec, T. Topór-Kaminski, *Uncertainty modelling method of data series processing algorithms*. IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation and 3rd Workshop on ADC Modelling and Testing, September 17-18, Naples, Italy 1998.

[3] J. Jakubiec, *Uncertainty Model as a Base of Accuracy Evaluation of Measuring Processing Algorithms*. Zeszyty Naukowe Pol. Śl., Elektryka Z. 169, Gliwice 2000, pp.7-36.

[4] J. Jakubiec, K. Konopka, *Identification method of error sources of A/D measuring chain*, IEEE IMTC 2003 – Instrumentation and Measurement Technology Conference Vail, CO, USA, 20-22 May 2003..

[5] J. Jakubiec, K. Konopka, *Coherence coefficients as uncertainty parameters of error value set*, IMEKO TC7 Symposium „Measurement Science of the Information Era", Krakow 2002, pp. 76-81.

[6] Papoulis A.: Prawdopodobieństwo, zmienne losowe i procesy stochastyczne. WNT, Warszawa, 1972. Probability, Random Variables, and Stochastic Processes. McGraw-Hill, Inc., 1965.

[7] Guide to the Expression of Uncertainty in Measurement, ISO/IEC/OIML/BIPM, 1992, 1995.

[8] J. Jakubiec, *Probabilistic error description as a base of uncertainty definition of a single measurement result*. Pomiary Automatyka Kontrola PAK, nº. 02/2007, pp. 04-07.

[9] J. Jakubiec, *Probabilistic and deterministic description of errors in measuring system*. Pomiary Automatyka Kontrola PAK, nº. 06/2006, pp. 08-10.

[10] J. Jakubiec, *Random properties of quantization error in measuring system*. Pomiary Automatyka Kontrola PAK, nº. 01/2008, pp. 08-11.