

Jerzy JAKUBIEC

### MODEL NIEPEWNOŚCI JAKO PODSTAWA OCENY DOKŁADNOŚCI ALGORYTMÓW PRZETWARZANIA POMIAROWEGO

**Streszczenie.** W artykule rozpatrywane są algorytmy przetwarzania pomiarowego, które dają się przedstawić w postaci kombinacji liniowej ciągu danych wejściowych i stałych współczynników. Opisano metodę budowy modelu niepewności algorytmu, która pozwala na uzyskiwanie analitycznych równań wiążących niepewność jego wyniku końcowego z niepewnościami danych wejściowych i niepewnościami wprowadzanymi przez sam algorytm, które nazywane są własnymi. Podstawą metody jest zdefiniowanie niepewności jako parametru zbioru wartości błęd. Wyodrębniono trzy rodzaje błędów przetwarzania: statyczne, dynamiczne i losowe, opisano sposób przenoszenia tych błędów z wejścia na wyjście algorytmu oraz zdefiniowano odpowiadające im niepewności. Przedstawiono ogólne postacie równań łączących wymienione rodzaje niepewności na wyjściu i wejściu modelu algorytmu, opisano także ogólnie sposób wyznaczania jego niepewności własnych. Zaproponowano regułę składania niepewności, która pozwala na obliczenie niepewności wypadkowej dla znanych niepewności cząstkowych. Sposób realizacji obliczeń zilustrowano przykładem wyznaczania niepewności końcowej algorytmu wygładzania danych pomiarowych. W końcowej części artykułu przedstawiono zastosowanie opisanej metody do budowy modelu niepewności łańcucha algorytmów.

### UNCERTAINTY MODEL AS A BASE OF ACCURACY EVALUATION OF MEASURING PROCESSING ALGORITHMS

**Summary.** The paper deals with measuring data processing algorithms which can be presented as a linear combination of an input data sequence and constant coefficients. The method of algorithm uncertainty model construction which allows to obtain analytical equations expressing relationship between the algorithm final result and the input data uncertainty as well as the algorithm uncertainty itself has been described. The base of this method is defining an uncertainty as a parameter of the error value set. It makes possible to create the uncertainty models independent from the number of the algorithm coefficients. There have been distinguished three kinds of processing errors, namely: static, dynamic and random ones. The uncertainties corresponding to them have been defined, too. The general equations linking the algorithm input and output uncertainties as well as the way of calculation of the algorithm specific uncertainties have been presented. There has been proposed the rule of uncertainty composition which enables to determine the resultant uncertainty for the given partial uncertainties. The way of calculation has been illustrated by the example of determining the final uncertainty of the data smoothing algorithm. Application of the described method to construction of the algorithm chain uncertainty model has been presented at the end of the paper.

## 1. WSTĘP

Obiektem rozważań w niniejszym artykule są algorytmy przetwarzania pomiarowego. Tego rodzaju algorytm spełnia taką samą funkcję, jak przetwornik pomiarowy, a zatem niezbędna jest ocena jego dokładności przetwarzania. Często algorytm jest ostatnim ogniwem w procesie przetwarzania i wówczas rezultaty jego realizacji są wynikami pomiaru, a zatem jego dokładność powinna być wyrażona za pomocą niepewności.

Dokładność algorytmu zależy zarówno od dokładności danych wejściowych, jak i od rodzaju operacji obliczeniowych wykonywanych przez algorytm. Wynika stąd wniosek, że punktem wyjścia do budowy procedury wyznaczania niepewności wyników wyjściowych algorytmu jest analiza jego struktury numerycznej.

Przetwarzania realizowane przez algorytm można przedstawić graficznie w sposób pokazany na rys.1. Ogólnie algorytm przekształca ciąg danych wejściowych  $\{x(k), \dots, x(k+K-1)\}$  na ciąg wyjściowy  $\{X(n), \dots, X(n+N-1)\}$ . Zakłada się, że ciąg wejściowy reprezentuje wielkość wejściową  $x$  zmieniającą się w sposób ciągły w dziedzinie  $t$ . Dane wejściowe są skwantowanymi próbkami wielkości  $x$  w pewnym ograniczonym obszarze dziedziny  $t$  nazywanym oknem wejściowym algorytmu. Symbolem  $k$  oznaczono bieżący wskaźnik początku okna wejściowego lokalizujący położenie okna w dowolnie liczonym ciągu próbek wielkości wejściowej,  $k = 0, 1, \dots$ . Zatem dane wejściowe algorytmu składają się z próbek  $x(k), \dots, x(k+K-1)$ , gdzie  $K$  jest liczbą próbek objętych oknem wejściowym.

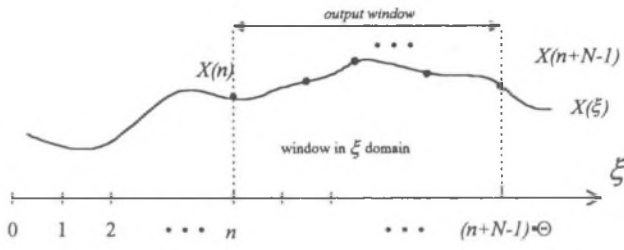
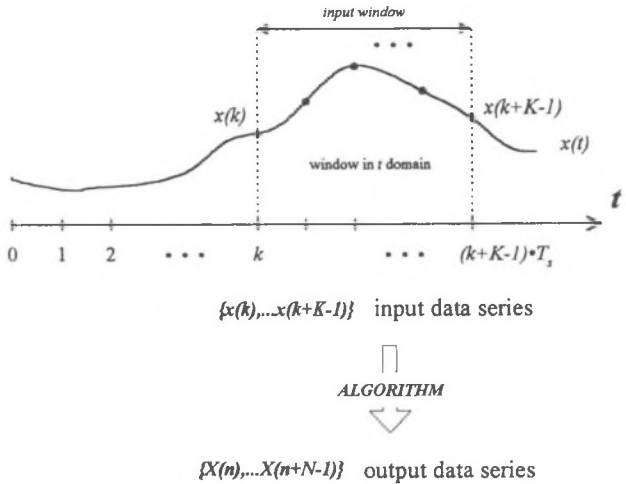
Przyjęto uproszczony zapis, zgodnie z którym  $x(k) = x(t_k)$ ,  $t_k = k \cdot T_s$ ,  $T_s$  jest okresem próbkowania w dziedzinie  $t$ .

## 1. INTRODUCTION

In this paper there are considered the measuring data processing algorithms. They perform the same functions as measuring transducers so it is necessary to evaluate the uncertainty of results delivered by these algorithms. The algorithm is often the last element of the processing and then its output data are the results of measurements. Hence, the algorithm accuracy should be expressed by the uncertainty.

This uncertainty depends both on the input data uncertainty and the kind of arithmetic operations made by the algorithm. Hence, it follows that the base of procedure construction of the algorithm output uncertainty calculation is its numerical structure analysis.

The processing performed by the algorithm is illustrated in Fig.1. In general, the algorithm transforms the input data sequence  $\{x(k), \dots, x(k+K-1)\}$  into the output data sequence  $\{X(n), \dots, X(n+N-1)\}$ . It is assumed that the input sequence represents the input quantity  $x$  varying continuously in time domain  $t$ . The input data are the quantified samples of  $x$  in a limited field of the domain  $t$  called the algorithm input window. The symbol  $k$  denotes a current index of the input window beginning, indicating the window position in an optionally numerous sequence of the input quantity samples,  $k = 0, 1, \dots$ . So, the algorithm input data consist of the samples  $x(k), \dots, x(k+K-1)$ , where  $K$  is the number of the samples included by the input window. To simplify the notation it has been assumed that  $x(k) = x(t_k)$ ,  $t_k = k \cdot T_s$ , where  $T_s$  is the sampling period in  $t$  domain.



Rys. 1. Graficzna interpretacja przetwarzania realizowanego przez algorytm

Fig. 1. Graphical interpretation of the algorithm operation

Ciąg danych wyjściowych może być również interpretowany jako dyskretna reprezentacja wielkości wyjściowej algorytmu zmieniającej się w dziedzinie  $\xi$ . Bieżący wskaźnik okna wyjściowego nosi symbol  $n$ ,  $n = 0, 1, \dots$ . Ciąg ten składa się z próbek  $X(n), \dots, X(n + N - 1)$ , gdzie  $X(n) = X(\xi_n)$ ,  $\xi_n = n \cdot \Theta$ ,  $\Theta$  jest okresem próbkowania w dziedzinie  $\xi$ .

Wielkość wejściowa zmienia się na ogół w dziedzinie czasu, może się również zmieniać przestrzennie [8]. Dziedzina wyjściowa jest często tożsama z dziedziną wejściową, występuje to zwykle dla wielkości

The output data sequence can be also interpreted as discrete representation of the algorithm output quantity varying in  $\xi$  domain. The output window current index is denoted by  $n$ ,  $n = 0, 1, \dots$ . This sequence consists of the samples  $X(n), \dots, X(n + N - 1)$ , where  $X(n) = X(\xi_n)$ ,  $\xi_n = n \cdot \Theta$ ,  $\Theta$  is the sampling period in  $\xi$  domain.

Generally, the input quantity varies in time domain but it can also vary in space [8]. The output domain is often identical to the input one, which usually occurs for time varying quantities. In such a case, a

zmiennych w czasie. W takim przypadku ogólnie zachodzi przesunięcie między początkiem okna wejściowego i wyjściowego oznaczone jako  $m$ . Przesunięcie to z reguły ma wartość stałą i zachodzi wówczas

$$n = k + m. \quad (1)$$

W sytuacji, gdy obie dziedziny są różne, algorytm realizuje operacje transformacji, przykładowo z dziedziny czasu do dziedziny pulsacji, jak to występuje w przypadku algorytmu dyskretnej transformaty Fouriera DFT. Ogólny podział algorytmów przetwarzania ze względu na realizowane zadania przedstawiono w pracy [7].

Zakłada się, że ciągi danych wejściowych i wyjściowych zawierają skończone liczby elementów. Algorytmy nierekurencyjne z założenia operują na ograniczonej liczbie danych. Natomiast algorytmy rekurencyjne dają się przedstawiać w opisany powyżej sposób, jak wykazano to w pracy [3].

Operacje arytmetyczne wykonywane przez algorytm przetwarzania dogodnie jest przedstawić w zapisie macierzowym. Ma on wówczas formę równania

$$\begin{bmatrix} X(n) \\ X(n+1) \\ \vdots \\ X(n+N-1) \end{bmatrix} = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,K-1} \\ a_{0,1} & a_{1,1} & \dots & a_{1,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,0} & a_{N-1,1} & \dots & a_{N-1,K-1} \end{bmatrix} \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+K-1) \end{bmatrix} \quad (2)$$

które w skrócie można zapisać jako

$$X = A \cdot x \quad (3)$$

Równanie to przedstawia związek między wektorem wejściowym  $x$ , którego elementy stanowią ciąg danych wejściowych, a wektorem wyjściowym  $X$  zawierającym wyniki realizacji algorytmu tworzące ciąg danych wyjściowych. Związek ten opisuje tablica  $A$  o rozmiarze  $N,K$ , gdzie  $K$  określa liczbę danych wejściowych,  $N$  jest liczbą danych wyjściowych algorytmu.

Dla algorytmów o stałej strukturze za-

phase shift between the beginnings of the input and output windows can appear. It is denoted by  $m$ . As a rule, this shift is constant. Then one can write

If the both domains are different the algorithm performs transformation, for instance from time domain to frequency domain as it takes place for the discrete Fourier transform (DFT) algorithm. The general division of the processing algorithms as far as the performed tasks are concerned, has been given in [7].

It is assumed that the input and output data sequences contain finite numbers of elements. Nonrecursive algorithms, as a rule, operate on the limited number of data, whereas recursive algorithms can be described in the mentioned above way as it is shown in the paper [3].

It is convenient to present the arithmetical operations performed by the processing algorithms in matrix notation. Then, the algorithm is expressed as

which can be written in short

The above equation presents the relation between the input vector  $x$  whose elements are the input data sequence and the output vector  $X$  containing the algorithm realization results being the output data sequence. This relation is described by the matrix  $A$  of  $N,K$  dimension, where  $K$  is the number of input data, and  $N$  is the number of output data.

For the algorithms with constant

równy rozmiar, jak i wartości współczynników tablicy  $A$  są stałe. Algorytmy takie można nazywać liniowymi, ponieważ stanowią zespół równań będących kombinacjami liniowymi danych wejściowych i stałych współczynników. Dla algorytmów nieliniowych lub adaptacyjnych mogą zachodzić zmiany liczby współczynników i ich wartości w zależności od aktualnego zbioru danych wejściowych. W przypadku algorytmów odtwarzania budowanych na zasadzie aproksymacji odcinkowo-liniowych nieliniowej charakterystyki przetwornika pomiarowego [4, 6] struktura algorytmu nie zmienia się w pewnych zakresach zmian zbioru danych wejściowych. Tego rodzaju algorytm można traktować jako zbiór algorytmów liniowych, z którego w danym momencie realizowany jest jeden algorytm wybierany w zależności od „punktu pracy” na charakterystyce przetwarzania [6].

Algorytm w postaci ogólnej (2) może być przedstawiony jako równoległy zespół algorytmów „jednopunktowych” [7], operujących na takim samym ciągu danych wejściowych. Realizacja każdego algorytmu jednopunktowego dostarcza pojedynczego wyniku. Zapis matematyczny tego rodzaju algorytmu stanowi równanie

$$X(n) = a_0 x(k) + a_1 x(k+1) + \dots + a_{K-1} x(k+K-1) \quad (4)$$

Można wskazać wiele algorytmów, które z zasady działają jako jednopunktowe. Takim algorytmem jest m.in. filtr wygładzający opisany w przykładzie w punkcie 5.

Przedstawianie algorytmów „wielopunktowych” w powyższy sposób upraszcza znacznie ich analizę bez uszczerbku dla ogólności rozważań. W dalszym ciągu artykułu rozpatrywane są właściwości metrologiczne algorytmów jednopunktowych liniowych. Jak wynika to z postaci równania (4), tego rodzaju algorytmy można określić jako kombinację liniową ciągu danych wejściowych i stałych współczynników.

Algorytm realizuje pomiar pośredni, a zatem można próbować ocenić niepewność

struktury, zarówno wymiary i wartości współczynników tablicy  $A$  są stałe. Algorytmy takie można nazywać liniowymi, ponieważ stanowią zespół równań będących kombinacjami liniowymi danych wejściowych i stałych współczynników. Dla algorytmów nieliniowych lub adaptacyjnych zmiany liczby współczynników i ich wartości w zależności od aktualnego zbioru danych wejściowych. W przypadku algorytmów odtwarzania budowanych na zasadzie aproksymacji odcinkowo-liniowych nieliniowej charakterystyki przetwornika pomiarowego [4, 6] struktura algorytmu nie zmienia się w pewnych zakresach zmian zbioru danych wejściowych. Tego rodzaju algorytm można traktować jako zbiór algorytmów liniowych, z którego w danym momencie realizowany jest jeden algorytm wybierany w zależności od „punktu pracy” na charakterystyce przetwarzania [6].

The general form of the algorithm (2) can be presented as a parallel set of „one-point” algorithms [7] operating on the same sequence of the input data. The performance of each one-point algorithm delivers the single result. The mathematical notation of such a kind of algorithm is the equation

One can point many algorithms which operate as one-point ones, as a rule. Such an algorithm is among others a smoothing filter described in Section 5.

The above way of presentation of „multi-point” algorithms simplifies considerably their analysis without loss of generality. Therefore, the further considerations deal with metrological properties of the linear one-point algorithm only. As it follows from the equation (4) such kind of algorithms can be determined as a linear combination of the input data sequence and constant coefficients.

The algorithm performs an indirect measurement, that is why its resultant uncertainty can be evaluated according to the rules proposed in [11]. Such a method is

jego wyniku zgodnie z regułami proponowanymi w [11]. Pojawiają się publikacje, w których stosuje się tego rodzaju podejście. Przykładem jest praca [1] dotycząca algorytmu FFT. Przedstawione w niej wyniki wskazują jednak, że taki sposób postępowania nie może być w pełni zaakceptowany. Jego podstawą jest przyjęcie, że wszystkie błędy cząstkowe danych wejściowych i błędy własne algorytmu mogą być traktowane jako losowe i niezależne od siebie. Jest to równoważne założeniu, że każda liczba wprowadzana na wejście algorytmu jest wynikiem pomiaru innej wielkości, niezależnej od wszystkich pozostałych. W rzeczywistości tak nie jest, bowiem dane wejściowe stanowią wyniki pomiaru tej samej wielkości. Zatem można wskazać takie rodzaje błędów, dla których występują deterministyczne związki między wartościami na wejściu i wyjściu algorytmu. Potraktowanie ich tak, jak błędy losowe może spowodować dużą niedokładność oszacowania niepewności w sytuacji, gdy ich udział w błędzie wypadkowym jest istotny. Pokazują to rezultaty eksperymentów opisanych w pracy [1]. W sytuacjach, w których dominują błędy losowe (błędy kwantowania, błędy zaokrągleń operacji arytmetycznych, itp.) zgodność wyników analitycznego wyznaczania niepewności z wynikami uzyskanymi na drodze symulacyjnej i eksperymentalnej jest dobra. Gdy tak nie jest, pojawiają się znaczne różnice.

Przedstawiona w artykule metoda wyznaczania niepewności wyników końcowych algorytmu wykorzystuje podejście innego rodzaju. Mianowicie, algorytmy są traktowane jako przetworniki błędów zachowujące się różnie dla różnych rodzajów błędów i wprowadzające błędy własne specyficzne dla rodzaju wykonywanych działań numerycznych. Przetwarzanie błędów opisywane jest w sposób deterministyczny lub probabilistyczny w zależności od rodzaju błędów. Natomiast w sytuacji, gdy występuje składanie błędów na wyjściu algorytmu, zakłada się, że wszystkie rodzaje błędów opisywane są w kategoriach probabilistycznych.

used by many authors, for example in [1] in which the FFT algorithm is considered. However, the results presented there show that this method cannot be entirely accepted. It assumes that all the input data partial uncertainties and the algorithm specific uncertainties can be treated as random ones and independent from each other. It is equivalent to the assumption that each number delivered to the algorithm input is the measurement result of the other independent quantity. However, the above condition is not satisfied because the input data are the measurement result of the **same quantity**. So, one can point such kinds of errors for which deterministic relations between the values at the algorithm input and output occur. Considering them as random errors can cause great inaccuracy of the uncertainty evaluation in case of their essential contribution to the resultant error, which was proved by the experimental results described in the paper [1]. If the random errors (quantization errors, arithmetical operations rounding errors and so on) are dominant the results of analytical calculation of the uncertainties are close to the simulation and experimental ones. If this kind of domination does not exist considerable differences appear.

The method of the algorithm final result uncertainty determination presented in the paper applies the other kind of approach. Namely, the algorithms are treated as error transducers processing different kinds of errors in a different way and introducing the errors typical for the performed numerical operations. The error processing is described in a deterministic or probabilistic way dependently on the type of the error. However, in case of composition of the errors at the algorithm output one assumes that all kinds of the errors are described in probabilistic categories.

Such an approach causes that the basic concept necessary to describe the algorithm properties is a **processing error** treated as a difference between the processing result and the output quantity true

Tego rodzaju podejście powoduje, że podstawowym pojęciem niezbędnym do opisu właściwości algorytmów jest **błąd przetwarzania** traktowany jak różnica między wynikiem przetwarzania a prawdziwą wartością wielkości przetworzonej. Wartości rzeczywistych błędów nie są znane, tym niemniej można badać właściwości algorytmów dla różnych rodzajów błędu w sposób symulacyjny, a wyniki tych badań przenosić odpowiednio na sytuacje pojawiające się w praktyce pomiarowej. Postępowanie takie jest szczególnie uzasadnione w przypadku algorytmów przetwarzania, bowiem procesor, który realizuje algorytm jest tego samego rodzaju urządzeniem, co procesor komputera, na którym przeprowadzana jest symulacja.

Powyższe rozważania pozwalają na przedstawienie zarysu koncepcji opisywanej metody wyznaczania niepewności algorytmu przetwarzania. Ma on strukturę liniową, a zatem możliwe jest stosowanie zasady superpozycji w odniesieniu do działań algorytmu na danych wejściowych i ich błędach. Wynika stąd wniosek, że można rozpatrywać działanie algorytmu w dwóch płaszczyznach. Jedną to wykonywanie operacji na danych wejściowych, czego efektem jest wynik wyjściowy. Operacje te odbywają się równocześnie na wartościach wielkości mierzonej i wartościach jej błędów, przy czym w trakcie tych działań algorytm wprowadza również błędy własne. W drugiej płaszczyźnie rozpatrywane są działania algorytmu wyłącznie na błędach, w wyniku czego uzyskuje się zbiór wartości błędu wyjściowego algorytmu. Wartość niepewności wyznacza się jako parametr zbioru wartości błędu wyjściowego. Tak uzyskana wartość jest przenoszona na płaszczyznę pierwszą jako niepewność wyniku wyjściowego algorytmu. Oznacza to, że **niepewność wielkości wyjściowej algorytmu** jest parametrem zbioru wartości błędu tej wielkości w określonych warunkach przetwarzania.

value. The true values of the errors are not known, nevertheless one can investigate the algorithm properties for different kinds of the error in a simulation way and the obtained results carry onto practical measurement situations. Such a procedure is especially good for the processing algorithms because the processor both realizes and simulates the algorithm.

The above considerations allow to present the idea of the described method of the processing algorithm uncertainty calculation. The algorithm is of a linear structure, so it is possible to use the principle of superposition for the algorithm operations on the input data and their errors. Hence, one can draw a conclusion that the algorithm performance can be considered in two fields. One of them is performing the operations on the input data which results in delivering the output data. These operations are made simultaneously on the measured quantity values and its errors during which the algorithm introduces its own errors. The other field deals with the algorithm operations on errors only which results in obtaining the algorithm output error value set. The uncertainty value is determined as a parameter of the output error value set. The value obtained in such a way is carried into the first field as the algorithm output result uncertainty. It means that the **algorithm output quantity uncertainty** is a parameter of the error value set of this quantity in the determined working conditions.

## 2. RODZAJE BŁĘDÓW PRZETWARZANIA PROGRAMOWEGO

Właściwości algorytmu dla określonego rodzaju błędu można wyznaczyć podając zależności między zbiorami wartości tego błędu na wejściu i wyjściu algorytmu. Wyniki badań algorytmów [2, 6] wskazują, że celowe jest wyodrębnienie trzech rodzajów błędów występujących w procesie przetwarzania. Są to:

- błędy statyczne,
- błędy dynamiczne,
- błędy losowe.

Kryterium powyższego podziału błędów oparte jest na sposobie zmian ich wartości w kolejnych momentach dyskretyzacji wielkości wejściowej wewnątrz okna pomiarowego. Zakres zmian tych błędów jest ograniczony przez ich dopuszczalne wartości. Oznacza to w przypadku błędów statycznych i dynamicznych, że błędy te są odpowiednio małe lub stanowią resztki nieusuniętych błędów systematycznych.

Przyjmuje się, że wartości błędu nazywanego statycznym są praktycznie stałe wewnątrz okna. Oznacza to, że jeżeli zachodzą zmiany wartości błędu statycznego w kolejnych chwilach dyskretyzacji, to zmiany te nie mają istotnego wpływu na wartość błędu wyjściowego algorytmu. Natomiast nie ogranicza się zmian tego błędu między kolejnymi oknami w przypadku wielokrotnej realizacji algorytmu, oczywiście, w zakresie wartości dopuszczalnych. W praktyce błędy te na ogół zmieniają swoje wartości między oknami stosunkowo wolno, jak przykładowo błędy temperaturowe.

W przypadku gdy błąd statyczny jest wprowadzany przez układ pomiarowy dostarczający dane na wejście algorytmu, zależność między błędem  $\delta_{st}$  wielkości wyjściowej algorytmu a błędem statycznym  $\delta_{s,x}$  zawartym w każdym wyniku wejściowym jest zgodnie z (4) następująca:

## 2. TYPES OF PROCESSING ERRORS

The results of investigations of the algorithm properties show (see [2][6]) that it is necessary to define three types of errors occurring during the processing. They are:

- static errors,
- dynamic errors,
- random errors.

The criterion of the above error definitions is based on the way of their value changes at the succeeding discretization instants of the input quantity inside the measuring window. The range of these errors changes is limited by their permissible values. It means that in case of the static and dynamic errors these errors are appropriately small or they are the remains of uncorrected systematic errors.

Therefore, one assumes that the source of static errors generates the error values which are practically constant inside the window. It means that if the static error values change at the succeeding discretization instants the changes do not essentially influence the algorithm output error value. However, these changes can be of any value but must be within a permissible range. In practice these errors vary relatively slowly between the windows as it takes place for the temperature errors.

In case when the static error is generated by the measuring system delivering data to the algorithm input the relationship between the algorithm output quantity error  $\delta_{st}$  and the static error  $\delta_{s,x}$  included in each input result is, according to (4), given by



$$\delta_{st}(n) = (a_0 + a_1 + \dots + a_{K-1}) \cdot \delta_{s,x}(k) = S \cdot \delta_{s,x}(k) \quad (5)$$

gdzie przyjęto

where it is assumed

$$\delta_{s,x}(k) = \delta_{s,x}(k+1) = \dots = \delta_{s,x}(k+K-1) \quad (6)$$

Z zależności tej wynika, że dla dowolnej realizacji algorytmu w chwili  $n$ , stosunek statycznego błędu wyjściowego do błędu generowanego przez statyczne źródło błędów wejściowych jest stały. Wartość tego stosunku równa jest sumie wszystkich współczynników algorytmu oznaczonej symbolem  $S$ .

Źródło błędów nazywane jest dynamicznym, gdy generowany przez nie błąd zmienia wartości w kolejnych punktach okna w taki sposób, że można określić deterministyczne związki między tymi wartościami, a wartość błędu wyjściowego zależy od przyrostów błędów. W celu wyznaczenia związku między błędem dynamicznym na wyjściu i wejściu algorytmu niezbędne jest przyjęcie opisu przebiegu wartości błędu w poszczególnych próbkach podawanych na jego wejście. Najdogodniej jest przyjąć przebieg sinusoidalnie zmiennej. Założmy, że przebieg błędu wielkości wejściowej algorytmu ma postać:

$$e(t) = E_m (\sin \omega t + \varphi) \quad (7)$$

gdzie  $E_m$  jest amplitudą,  $\omega$  - pulsacją przebiegu, a  $\varphi$  jest przesunięciem fazowym względem początku okna. Zbiór wartości błędu wejściowego składa się wówczas z wartości chwilowych przebiegu (7)

where  $E_m$  is a waveform amplitude,  $\omega$  - its frequency,  $\varphi$  - a phase shift related to the window beginning. Then the set of the input error values consists of instantaneous values of the waveform (7)

$$\delta_{s,x}(i) = E_m \sin(\omega T_s i + \varphi), \quad i = k, \dots, k+K-1 \quad (8)$$

gdzie  $T_s$  jest okresem próbkowania przebiegu wielkości wejściowej algorytmu. W takim przypadku błąd wyjściowy opisuje wyrażenie

where  $T_s$  is the sampling period of the algorithm input quantity waveform. In this case the output error is described by the expression

$$\delta_{st}(i) = E_m \sin(\omega T_s i + \varphi + \alpha), \quad (9)$$

gdzie

where

$$E_m = E_m \cdot |S(j\omega)|, \quad (10)$$

oraz

and

$$\alpha = \frac{\text{Im}|S(j\omega)|}{\text{Re}|S(j\omega)|}. \quad (11)$$

$S(j\omega)$  jest transmitancją zespoloną algorytmu, która na podstawie wyrażenia (4) ma postać:

$S(j\omega)$  is the algorithm complex transmittance determined from (4) and given by

$$S(j\omega) = \frac{X(j\omega)}{x(j\omega)} = [a_0 + a_1 \cdot e^{j\omega T_s} + \dots + a_{K-1} \cdot e^{j\omega T_s(K-1)}] \cdot e^{-j\omega T_s m}, \quad (12)$$

gdzie  $m$  przyjmuje wartości zgodnie z wyrażeniem (1).

where values of  $m$  are taken according to the expression (1).

Przedstawione powyżej zależności pozwalają na analityczne wyznaczanie wartości zbioru błęd dynamicznego na wyjściu algorytmu dla określonych właściwości źródła błędu wejściowego charakteryzowanych przez takie parametry, jak amplituda przebiegu błędu i jego pulsacja. Najprościej można to osiągnąć przez symulację powtarzając wielokrotnie obliczanie błędu wyjściowego dla losowo wybieranych wartości przesunięcia fazowego  $\varphi$ .

The presented above relationships allow to calculate analytically the dynamic error value set at the algorithm output for the known properties of the input error source characterized by such parameters as the error waveform amplitude and its frequency. It can be performed by simulation when repeating many times calculations of the error values for the random phase shift values  $\varphi$ . It is the simplest way of doing it.

Właściwości losowych źródeł błędów opisywane są w kategoriach probabilistycznych. Oznacza to w rozpatrywanej sytuacji, że wartości błędów pojawiających się w kolejnych punktach okna wejściowego są realizacjami tego samego procesu losowego o parametrach określonych przez zjawiska zachodzące w układzie dostarczającym dane na wejście algorytmu. Z zasady takim źródłem błędów jest proces kwantowania próbek wielkości bezpośrednio przetwarzanej przez algorytm. Innym czynnikiem losowym mogą być szumy generowane w układach przetwarzania A/C lub błędy losowe poprzedzających algorytmów.

Properties of the random error sources are described in probabilistic categories. In the considered situation it means that the error values in the successive points of the input window are realizations of the same random process with parameters determined by phenomena occurring in the circuit delivering data to the algorithm input. As a rule, quantization of the samples processed directly by the algorithm is such an error source. Noises generated in the chain delivering data or random errors of the previous algorithms can be the another random source.

Błąd losowy  $\delta_n(n)$  pojawiający się na wyjściu algorytmu w  $n$  kroku jego realizacji może być zatem przedstawiony, zgodnie z wyrażeniem (4), jako kombinacja liniowa

The random error  $\delta_n(n)$  occurring at  $n$  step of the algorithm output realization can be given, according to (4), as a linear combination of the random error and the algorithm coefficients values

realizacji losowego błędu wejściowego i współczynników algorytmu

$$\delta_{r,x}(n) = a_0 \delta_{r,x}(k) + a_1 \delta_{r,x}(k+1) + \dots + a_{K-1} \delta_{r,x}(k+K-1). \quad (13)$$

Wykonując algorytm wielokrotnie uzyskuje się zbiór błędów wyjściowych, którego wariancja  $\sigma_x^2$  związana jest z wariancją  $\sigma_x^2$  zbioru błędów wejściowych zależnością

$$\sigma_x^2 = a_0^2 \sigma_x^2 + a_1^2 \sigma_x^2 + \dots + a_{k-1}^2 \sigma_x^2 = A^2 \sigma_x^2 \quad (14)$$

uzyskaną przy założeniu, że wartości błędu wejściowego nie są ze sobą skorelowane. Oznaczając sumę kwadratów współczynników algorytmu jako

$$A^2 = a_0^2 + a_1^2 + \dots + a_{k-1}^2, \quad (15)$$

związek między wariancją błędu losowego na wyjściu i wejściu algorytmu przyjmuje postać

$$\sigma_x^2 = A^2 \sigma_x^2. \quad (16)$$

Współczynnik  $A^2$  stanowi istotną wielkość charakterystyczną algorytmu, ponieważ jest podstawą określania związków między zbiorem wartości błędu losowego na wejściu algorytmu i odpowiadającym mu zbiorem wartości losowego błędu wyjściowego. Biorąc pod uwagę ogólnie dużą liczbę współczynników algorytmu można przyjąć, że błąd na wyjściu ma rozkład normalny o wariancji opisanej wyrażeniem (16) niezależnym od rozkładu błędu wejściowego. Wynika to z centralnego twierdzenia granicznego, a dobrą ilustracją tej tezy stanowią wyniki badań algorytmu FFT przedstawione w pracy [10]. Uwzględniając ponadto fakt, że z reguły dysponuje się wiedzą o rozkładzie błędu wejściowego (najczęściej jest to rozkład jednostajny, ponieważ błędy losowe zdominowane są przez błąd kwantowania), w rozważanej sytuacji można określić jednoznacznie

Performing the algorithm many times one obtains the set of the output errors whose variance  $\sigma_x^2$  is connected with the variance  $\sigma_x^2$  of the input error set by the relationship

under assumption of the lack of correlation between the input error values. Denoting the square sum of the algorithm coefficients as follows

the relation between the random error variances at the algorithm output and input is given by

The coefficient  $A^2$  is the essential parameter of the algorithm because it is the base for determining relationships between the random error value set at the algorithm input and the corresponding set of the output random error values. Taking into account, in general, a great number of the algorithm coefficients one can assume that the output error distribution is gaussian with the variance described by the equation (16) independent from the input error distribution. It results from the central limit theorem which is illustrated by the results of FFT algorithm investigation presented in [10]. Moreover, taking into consideration the fact that the input error distribution is known as a rule (mostly it is uniform because the random errors are dominated by the quantization error) it is possible to determine the exact quantitative relationships between these sets of the

związki ilościowe między zbiorami wartości błędów losowych na wejściu i wyjściu algorytmu.

Prócz przenoszenia błędów z wejścia na wyjście każdy algorytm wprowadza błędy własne. Można wskazać wiele różnych mechanizmów powstawania tych błędów, ich specyfika zależy głównie od rodzaju algorytmu. Niezależnie od liczby i sposobu powstawania błędów własnych, celowe jest pogrupowanie ich w trzy rozpatrywane klasy błędów: statyczne, dynamiczne i losowe. Grupowanie takie jest niezbędne, gdy rozpatruje się łańcuch algorytmów, jak to ilustruje przykład przedstawiony w pracy [6].

Rozpatrując losowe błędy własne, algorytm można traktować jako generator liczb losowych powstających podczas realizacji każdej operacji arytmetycznej na skutek zaokrąglania jej wyniku. Wartości tych liczb zależą od rodzaju stosowanej arytmetyki (stało- lub zmiennoprzecinkowej) oraz długości słowa danych. Liczby te kumulują się dając na wyjściu algorytmu błąd losowy. Przykład analizy tego rodzaju błędów dla algorytmu FFT można znaleźć w pracy [1].

Analiza błędów własnych algorytmu o charakterze deterministycznym, tj. błędów statycznych i dynamicznych, wymaga przyjęcia jego modelu idealnego. Struktura numeryczna algorytmu, określona przez wartość i liczbę współczynników równania (4), stanowi z reguły dyskretne odwzorowanie operacji analogowych, jak to pokazano ogólnie w punkcie 1. Oznacza to, że algorytm spełnia swoje zadania w sposób przybliżony. Jednak dla każdego algorytmu można wskazać takie działania, których wykonanie może być traktowane jako realizacja algorytmu idealnego. Działania te wynikają z definicji zadania, dla którego wykonania algorytm został zbudowany. Algorytm idealny jest algorytmem dokładnym, a zatem nie wprowadza własnych błędów do procesu przetwarzania. Tego rodzaju algorytmy na ogół nie dają się budować, dostępny jest natomiast ich opis matematyczny. Opis ten nazywany jest tutaj modelem algorytmu idealnego i jest wykorzystywany jako źródło dokładnych wyni-

random error values at the algorithm input and output.

Besides transforming the errors from the input to the output each algorithm introduces its specific errors. One can point many different mechanisms of arising these errors. Their specificity depends mainly on the kind of the algorithm. Independently of the number and manner of specific error arising, it is useful to group them into three mentioned above categories: static, dynamic and random. Such grouping is necessary when considering the algorithm chain as it is illustrated by the example presented in [6].

When considering the random specific errors the algorithm can be treated as a generator of random numbers caused by rounding the results of each arithmetical operation. The values of these numbers depend on the kind of the used arithmetic (constant or floating point arithmetic) and the data word length. These numbers cumulate giving the random error at the algorithm output. The example of such a kind of the error analysis for FFT algorithm can be found in the paper [1].

The analysis of the algorithm specific errors of deterministic character, i.e. static and dynamic errors, requires assumption of its ideal model. The algorithm numerical structure determined by the value and number of the coefficients in the equation (4) is, as a rule, discrete mapping of the analog operations, as it is generally shown in Section 1. It means that the algorithm performs its tasks in an approximate way. However, for each algorithm it is possible to point such operations which can be treated as ideal ones. They result from the task definition for which the algorithm has been constructed. An ideal algorithm is an exact one, so it does not introduce any errors to the processing. Such algorithms are impossible to build but their mathematical description is available. In this paper this description is called an ideal algorithm model and it is used as a source of the exact output data of the considered real algorithm. For instance, the appropriate functional in the integral form is the ideal

ków wyjściowych rozpatrywanego algorytmu rzeczywistego. Przykładowo modelem idealnym algorytmu wyznaczania wartości średniej przebiegu jest odpowiedni funkcjonal w postaci całkowej. Wartość tego funkcjonau dla konkretnego przebiegu może być traktowana jako wynik realizacji algorytmu idealnego wykonującego swoje zadanie bezbłędnie.

Model idealny dla różnych rodzajów algorytmów przybiera na ogół różne postaci. Dla przykładu idealny model właściwości dynamicznych filtru wygładzającego opisanego w punkcie 5 ma postać:

$$S_{ideal}(j\omega) = 1. \quad (17)$$

Celem stosowania algorytmu wygładzania jest zmniejszenie błędów losowych danych do wartości akceptowalnych, co jednak nie powinno mieć wpływu na sam przebieg wielkości wejściowej. Sytuacja taka występowałaby, gdyby transmitancja filtru miała postać (17). W rzeczywistości transmitancja ta ma ogólną postać (12), a zatem użycie filtru spowoduje wprowadzenie przez niego błędu dynamicznego zależnego m. in. od różnicy transmitancji rzeczywistej (12) i idealnej (17).

### 3. MODEL BŁĘDÓW ALGORYTMU

Biorąc pod uwagę liniowe właściwości algorytmu, jego błąd wyjściowy  $\delta_x(n)$  może być przedstawiony jako suma wyodrębnionych błędów cząstkowych. Zakładając ogólnie, że błędy przetwarzania podzielono na trzy opisane rodzaje, a ponadto w każdej klasie można wyróżnić błędy własne i przeniesione, błąd wyjściowy może być zapisany w postaci sumy

$$\delta_x(n) = \delta_n(n) + \delta_{na}(n) + \delta_{di}(n) + \delta_{da}(n) + \delta_r(n) + \delta_{ra}(n), \quad (18)$$

model of the algorithm of computing the waveform mean value. This functional value for a given waveform can be treated as a result of realization of the ideal algorithm performing its task without any errors.

The ideal model for different kinds of algorithms has, in general, different forms. For example, the ideal model of the smoothing filter dynamic properties described in Section 5 is given by

The purpose of using the smoothing algorithm is to reduce the data random errors to the accepted level which, however, should not influence the input quantity waveform itself. Such a situation could occur if the filter transmittance was given by (17). But the general form of the transmittance is expressed by (12) so use of the filter causes introducing the dynamic error dependent, among others, on the difference between the real (12) and ideal (17) transmittances.

### 3. MODEL OF ALGORITHM ERRORS

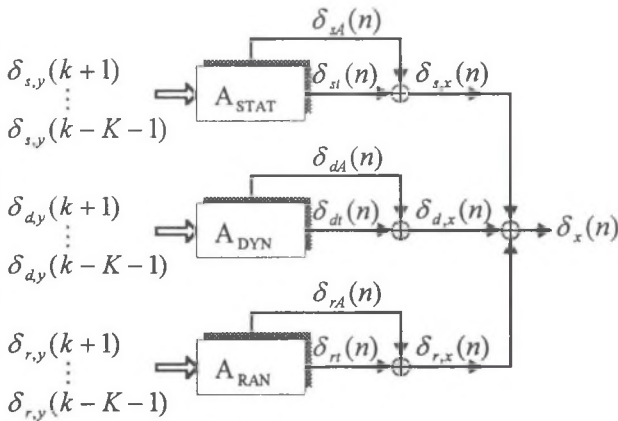
According to the considerations presented in Section 2 the algorithm output error  $\delta_x(n)$  can be presented as a sum of the separated partial errors. Assuming generally that the processing errors have been divided into three categories and, what is more, in each category one can distinguish the specific and transformed errors the output error can be written as the following sum

Uwzględniając sposób powstawania poszczególnych błędów cząstkowych, na podstawie równania (18) można uzyskać model błędów algorytmu przetwarzania pokazany na rys.2.  $A_{STAT}$ ,  $A_{DYN}$ ,  $A_{RAN}$  oznaczają odpowiednio właściwości statyczne, dynamiczne i losowe algorytmu. Mając do dyspozycji równania opisujące te właściwości oraz opis źródeł błędów wejściowych, można wyznaczać zbiory poszczególnych rodzajów błędów na wyjściu algorytmu, a także zbiór wypadkowy dla tych błędów.

Model błędów dostarcza pełnej informacji o błędach zarówno na wejściu, jak i na wyjściu algorytmu, a zatem pozwala na uzyskanie rozkładów wartości błędów w określonych warunkach realizacji algorytmu. Jest zatem bardzo przydatny w analizie metrologicznej algorytmu polegającej na wyznaczaniu poszczególnych źródeł błędów, rozpatrywaniu ich wpływu na błąd wypadkowy, a następnie na ewentualnym poszukiwaniu sposobu zmniejszania błędów dominujących. Drugie, równie istotne zastosowanie modelu błędów, polega na wykorzystaniu go w procesie syntezy niepewności wyniku końcowego algorytmu.

Taking into account the way of arising particular partial errors and basing on the equation (18) one can obtain the model of the processing algorithm errors shown in Fig.2.  $A_{STAT}$ ,  $A_{DYN}$ ,  $A_{RAN}$  denote the static, dynamic and random properties of the algorithm, respectively. Knowing the equations modelling these properties and the description of the input error sources it is possible to calculate the sets of particular kinds of the errors at the algorithm output as well as the resultant set.

The model of the errors gives full information on the errors both at the input and output of the algorithm, therefore it enables to obtain distributions of the error values in the determined conditions of the algorithm performance. So, it is very useful for the algorithm metrological analysis consisting in: determining the particular error sources, considering its influence on the resultant error and, next, searching for the way of reducing the dominant errors. The other application of the error model, as essential as the previous one, consists in using it for synthesis of the algorithm resultant uncertainty.



Rys.2. Model błędów algorytmu

Fig.2. Model of the algorithm errors

#### 4. NIEPEWNOŚĆ JAKO PARAMETR ZBIORU WARTOŚCI BŁĘDU

Zgodnie z modelem przedstawionym na rys.2 błąd na wyjściu algorytmu, dla każdego z opisanych rodzajów błędu, jest sumą błędu przeniesionego z wejścia i błędu własnego algorytmu. Zgodnie z przyjętym założeniem, że w tego rodzaju operacjach wszystkie zbiory cząstkowe są traktowane jako przypadkowe, dodawane wartości błędów pobierane są losowo z tych zbiorów zgodnie z odpowiednią funkcją gęstości prawdopodobieństwa. Takie postępowanie nazywane jest składaniem zbiorów wartości błędów. Zatem zbiór błędów na wyjściu algorytmu  $\{\delta_x\}$  powstaje jako wynik złożenia zbioru błędów przeniesionego z wejścia  $\{\delta_i\}$  i zbioru błędów własnego  $\{\delta_A\}$  algorytmu, co symbolizuje zapis

$$\{\delta_x\} = \{\delta_A\} + \{\delta_i\}. \quad (19)$$

Założmy, że niepewność  $\Delta_x$  zbioru wartości błędów  $\{\delta_x\}$  wielkości  $X$  definiowana jest jako parametr otrzymywany w rezultacie wykonania operacji  $M$  na tym zbiorze, co ogólnie można zapisać jako

$$\Delta_x = M[\{\delta_x\}]. \quad (20)$$

Uwzględniając zależność (19) wyrażenie (20) można zapisać w postaci

$$\Delta_x = M[\{\delta_A\} + \{\delta_i\}], \quad (21)$$

która oznacza, że wyznaczenie niepewności końcowej algorytmu polega najpierw na składaniu zbiorów cząstkowych, a następnie na określeniu niepewności zbioru wypadkowego zgodnie z przyjętą definicją reprezentowaną ogólnie przez wyrażenie (20). Są to w ogólnym przypadku działania złożone i dlatego dla celów praktyki pomiarowej dąży się do zastąpienia ich operacjami przybliżonymi, ale za to prostymi alge-

#### 4. UNCERTAINTY AS PARAMETER OF ERROR VALUE SET

According to the model shown in Fig.2 the error at the algorithm output, for each of the described kinds of the errors, is a sum of the error transmitted from the input and the algorithm specific error. The error values being summed are taken from the partial sets in a random way according to the appropriate probability density function. It follows from the assumption that all the sets are treated as random ones when summing the error sets, which is called the composition of the error value sets. Hence, the error set at the algorithm output  $\{\delta_x\}$  is the result of composition of the error set transmitted from the input  $\{\delta_i\}$  and the algorithm specific error set  $\{\delta_A\}$ , which is denoted by

Let us assume that the uncertainty  $\Delta_x$  of the error value set  $\{\delta_x\}$  of the quantity  $X$  is defined as a result of an operation  $M$  performed on this set, which can be generally written as

Taking into account (19) the expression (20) can be written in the form

which means that calculation of the algorithm resultant uncertainty consists in adding the partial error value sets and next determining the uncertainty of the sum of the sets in accordance with the assumed definition given by the equation (20). They are complex operation. So, in case of practical applications one attempts to replace them by approximate operations which are algebraically simple. Such ope-

braicznie. Operacje takie nazywa się składaniem niepewności. Zakładając, że została przyjęta reguła składania niepewności, wyrażenie (21) może być zastąpione uproszczoną formułą

$$\Delta_x = M[\{\delta_A\}] \otimes M[\{\delta_I\}] = \Delta_A \otimes \Delta_I, \quad (22)$$

która oznacza, że niepewność końcowa algorytmu jest w takim przypadku wyznaczona przez złożenie niepewności cząstkowych opisujących zgodnie z definicją składowe zbiory wartości błędów. W wyrażeniu tym użyto symbolu  $\otimes$  do oznaczenia operacji składania niepewności.

Założmy, że zbiór wartości błędów składa się z liczb rzeczywistych, które w ogólnym przypadku mogą przyjmować wartości dowolne. Dla każdego zbioru wskazywane są dwie wartości graniczne  $\delta_{\min}$  i  $\delta_{\max}$ , takie że prawdopodobieństwo pojawienia się wartości błędów  $\delta < \delta_{\min}$  lub  $\delta > \delta_{\max}$  jest tak małe, że w praktyce może być przyjęte jako równe zero. Zbiór obejmujący wartości  $\delta_{\min} \leq \delta \leq \delta_{\max}$  nazywany jest zbiorem zredukowanym wartości błędów. **Niepewność** jest definiowana jako połowa szerokości zbioru zredukowanego

$$\Delta = \frac{\delta_{\max} - \delta_{\min}}{2}. \quad (23)$$

Niepewność charakteryzująca zbiory wartości błędów statycznego, dynamicznego i losowego nazywana jest odpowiednio niepewnością statyczną, dynamiczną i losową.

Do dalszych rozważań przyjmuje się, że rozpatrywane zbiory zredukowane są symetryczne, tzn. zachodzi  $-\delta_{\min} = \delta_{\max}$ , co oznacza zarazem, że wartość średnia w zbiorze wartości błędów równa jest zero. Zakłada się ponadto, że wartość graniczna odpowiada poziomowi ufności 99,73%, co dla rozkładu normalnego daje wartość  $\delta_{\max} = 3\sigma$ ,  $\sigma$  jest odchyleniem standardowym rozkładu błędów.

Możliwe jest przyjęcie różnych kryteriów doboru wartości granicznych zbioru

rations are called the uncertainty composition. Using the assumed rule of uncertainty composition the expression (21) can be replaced by the simplified formula

which means that in such a case the algorithm resultant uncertainty is determined by composition of the partial uncertainties describing the error value sets according to the assumed definition. The symbol  $\otimes$  used in this expression denotes composition of the uncertainties.

Let us assume that the set of the error values consists of real numbers which, in general case, can take any values for each set. For each set one determines two boundary values  $\delta_{\min}$  and  $\delta_{\max}$  for which the probability of occurrence of the error values, such that  $\delta < \delta_{\min}$  or  $\delta > \delta_{\max}$ , is so small that it can be treated as equal to zero in practice. The set including the values  $\delta_{\min} \leq \delta \leq \delta_{\max}$  is called the reduced set of the error values. **The uncertainty** is defined as a half of the reduced set width

The uncertainty characterizing the sets of values of the static, dynamic and random errors is called the static, dynamic and random uncertainty, respectively.

For the further considerations it is assumed that the reduced sets are symmetric i.e.  $-\delta_{\min} = \delta_{\max}$ , which means that the mean value in the error set equals zero. Moreover, one assumes that the boundary value corresponds to the confidence level of 99,73%, which gives the value  $\delta_{\max} = 3\sigma$  for the gaussian distribution, where  $\sigma$  is the standard deviation of the error distribution.

It is possible to apply different criteria of the set boundary value choice to different kinds of the errors. When construc-



dla różnych rodzajów błędów. W przypadku gdy buduje się modele niepewności łącznie dla błędów statycznych, dynamicznych i losowych, należy tak dobrać wartości graniczne dla zbiorów tych błędów, aby odpowiednie zbiory zredukowane można było uznać za równoważne. Dla podanej powyżej zasady można przyjąć z dobrym przybliżeniem, że dla błędów statycznych i dynamicznych ich wartość graniczna stanowi maksymalną wartość błędu w przyjętych warunkach przetwarzania. Przykładowo, gdy wyznaczamy zbiór błędu dynamicznego dla sinusoidalnych zmian wielkości wejściowej wartość graniczna zbioru jest równa amplitudzie przebiegu błędu dynamicznego.

## 5. MODEL NIEPEWNOŚCI ALGORYTMU PRZETWARZANIA

Celem budowy modelu niepewności jest zastąpienie nim modelu błędu w zakresie związanym z wyznaczaniem niepewności wyniku końcowego algorytmu. Zamiast wykonywać złożone operacje na zbiorach wartości błędów dąży się do uzyskania prostych zależności analitycznych między niepewnością zbioru błędów wyjściowych a niepewnościami zbiorów błędów wejściowych i błędów własnych algorytmu. Uzyskanie takich związków pozwala na opracowanie procedur syntezy niepewności algorytmu na podstawie znajomości charakterystyk źródeł błędów wejściowych, ogólnego opisu wielkości mierzonej i właściwości samego algorytmu.

Można ogólnie powiedzieć, że model niepewności jest uproszczonym odwzorowaniem modelu błędu, które jest tworzone w celu uzyskania analitycznego zapisu procedury wyznaczania niepewności oceny wielkości mierzonej. Uproszczenia te wynikają z faktu, że zbiór błędu o dowolnym rozkładzie jest charakteryzowany przez jedną liczbę. Prowadzi to do powstawania różnic między niepewnością wyznaczaną z definicji na zbiorze błędu wyjściowego a niepewnością uzyskiwaną na podstawie

ting the uncertainty models being composition of all kinds of the errors, i.e. static, dynamic and random, one should select the boundary values for the sets of these errors so that the appropriate reduced sets can be considered as the equivalent ones. Further, it is assumed for the static and dynamic errors that the boundary value is the maximum value which can be determined in the real processing conditions. For example, determining the dynamic error for sinusoidal changes of the input quantity, the boundary error equals the amplitude of the dynamic error waveform.

## 5. MODEL OF THE ALGORITHM UNCERTAINTY

The purpose of the uncertainty model construction is to replace the error model for computing the algorithm resultant uncertainty. In spite of performing complex operations on the sets of the error values one attempts to get simple analytical relationships between the uncertainty of the output error set and the uncertainties of the sets of the input errors and the algorithm specific errors. Obtaining such relations allows to work out procedures of synthesis of the algorithm uncertainty basing on the knowledge of the characteristics of the input error sources, the general description of the measured quantity and the algorithm properties.

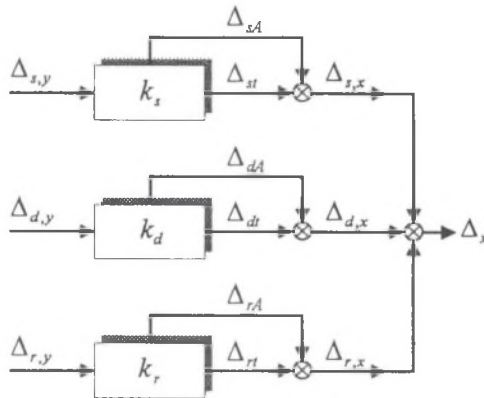
In general, one can say that the uncertainty model is a simplified mapping of the error model which is built in order to obtain the analytical description of the procedure of computing the measured quantity evaluation uncertainty. These simplifications result from the fact that the set of the error with any distribution is characterized by one number. It causes arising some differences between the uncertainty determined on the output error set from the definition and the uncertainty obtained

modelu. W tym sensie modele niepewności są modelami przybliżonymi. Skutki uproszczeń można określić porównując wyniki uzyskane w sposób symulacyjny na podstawie modelu błędów z wynikami obliczonymi na podstawie modelu niepewności.

Biorąc pod uwagę sposób budowy modelu niepewności jego struktura jest taka sama jak modelu błędów. W ogólnym przypadku może być ona przedstawiona w sposób pokazany na rys.3. Właściwości algorytmu są tu reprezentowane przez współczynniki  $k_s$ ,  $k_d$  i  $k_r$  określające stosunek między niepewnościami na wejściu i wyjściu algorytmu odpowiednio dla błędu statycznego, dynamicznego i losowego. Symbolem  $\otimes$  oznaczono operację składania niepewności.

from the model. In this sense, the uncertainty models are inaccurate (approximate). The simplification results can be evaluated by comparing the results obtained from simulation based on the error model with those calculated using the uncertainty model.

The uncertainty model structure is the same as that of the error model as far as the way of their construction is considered. The general structure of the algorithm uncertainty model is shown in Fig.3. In this figure the algorithm properties are represented by the coefficients  $k_s$ ,  $k_d$  and  $k_r$ , determining the ratio of the output to input algorithm uncertainties for the static, dynamic and random errors, respectively. The symbol  $\otimes$  denotes composition of the uncertainties.



Rys.3. Ogólna struktura modelu niepewności algorytmu

Fig.3. General structure of the algorithm uncertainty model

Błędy statyczne są przenoszone z wejścia na wyjście algorytmu ze współczynnikiem  $S$  zgodnie z wyrażeniem (5). Zatem również dla niepewności statycznych zachodzi związek

The static errors are transmitted from the algorithm input to its output with the multiplying coefficient  $S$  according to the formula (5). Hence, the following relationship is also valid for the static uncertainties

$$k_s = \frac{\Delta_{st}}{\Delta_{s,x}} = S. \quad (24)$$

Błędy dynamiczne o przebiegu sinusoidalnym mają na wyjściu amplitudę określoną wyrażeniem (10). Oznacza to, zgodnie z przyjętym sposobem wyznaczania niepewności dynamicznej, że dla tego rodzaju błędów współczynnik przeniesienia przyjmuje postać

$$k_d = \frac{\Delta_{dt}}{\Delta_{d,x}} = |S(j\omega)|. \quad (25)$$

Wariancję  $\sigma_x^2$  zbioru wyjściowego powstałego na skutek przeniesienia błędów losowych z wejścia na wyjście algorytmu określa zależność (16). Zakładając, że zbiór wartości błędu wyjściowego ma rozkład zbliżony do normalnego, jego niepewność określa zależność

$$\Delta_{\pi} = 3\sigma_x. \quad (26)$$

Dla znanego rozkładu wartości błędu losowego na wejściu algorytmu niepewność wyjściowa zgodnie z definicją może być określona jako

$$\Delta_{r,x} = k_{agran}\sigma_x, \quad (27)$$

gdzie  $k_{agran}$  jest współczynnikiem charakterystycznym dla typu rozkładu łączącym wariancję rozkładu błędu z jego niepewnością wyznaczoną dla poziomu ufności  $\alpha_{gran} = 99,73\%$ . Stosunek wyrażeń (26) i (27) stanowi współczynnik przenoszenia niepewności losowych. Uwzględniając wyrażenie (16) uzyskuje się

$$k_r = \frac{\Delta_{\pi}}{\Delta_{r,x}} = \frac{3A\sigma_x}{k_{agran}\sigma_x} = \frac{3A}{k_{agran}}. \quad (28)$$

Dla przykładu założmy, że błędy losowe na wejściu algorytmu zdominowane są przez błąd kwantowania, którego wartości mają rozkład jednostajny w przedziale od  $-\frac{q}{2}$  do  $+\frac{q}{2}$ , gdzie  $q$  jest wartością kwantu. Niepewność zbioru wartości błędu jest praktycznie w takim przypadku równa

The output amplitude of the sinusoidal dynamic errors is expressed by (10). It means that according to the assumed definition of the dynamic uncertainty the transformation coefficient for such kind of the errors is described by

The variance  $\sigma_x^2$  of the output set obtained as a result of transformation of the algorithm random errors from the input to the output is given by (16). Assuming that the distribution of the output error value set is approximately normal its uncertainty is

For the determined distribution of the random error values at the algorithm input the input uncertainty can be described by (according to the definition)

where  $k_{agran}$  is the coefficient dependent on the type of distribution connecting the error variance with its uncertainty calculated for the confidence level  $\alpha_{gran} = 99,73\%$ . The ratio of the formulae (26) and (27) is the coefficient of the random error transformation. Taking into account (16) one obtains

Let us assume, for example, that the random errors at the algorithm input are dominated by the quantization error whose values are of uniform distribution within the range from  $-\frac{q}{2}$  to  $+\frac{q}{2}$ , where  $q$  is the quantum value. In this case, the uncertainty of the error value set can be assumed

$\Delta_{r,x} = \frac{q}{2}$ , a wariancja rozkładu błędu wynosi  $\sigma_x^2 = \frac{q^2}{12}$ . Stąd

$$k_{\text{agran}} = \frac{\Delta_{r,x}}{\sigma_x} = \frac{\frac{q}{2}}{\frac{q}{\sqrt{12}}} = \sqrt{3}. \quad (29)$$

Po wprowadzeniu tej wartości do wyrażenia (28) otrzymuje się, że dla rozkładu jednostajnego błędu wejściowego współczynnik przenoszenia niepewności z wejścia na wyjście algorytmu ma postać

$$k_r = \sqrt{3} \cdot A. \quad (30)$$

Procedura składania niepewności określania działania arytmetyczne, jakie należy wykonać na niepewnościach cząstkowych w celu uzyskania niepewności wypadkowej. Przyjmijmy, że używana jest reguła w postaci proponowanej przez [11]. Stosując przyjęte w niniejszym artykule oznaczenia regułę tę dla dwóch składowych można zapisać jako

$$\Delta_{a,b} = \sqrt{\Delta_a^2 + \Delta_b^2 + 2\Delta_a\Delta_b\rho_{ab}}, \quad (31)$$

gdzie  $\Delta_{a,b}$  jest niepewnością wypadkową,  $\Delta_a$  i  $\Delta_b$  niepewnościami składowymi, a  $\rho_{ab}$  - współczynnikiem korelacji zbiorów cząstkowych. Dla przyjętej definicji niepewności zależność ta jest ścisła jedynie dla rozkładów normalnych zbiorów wartości błędów cząstkowych, w innych przypadkach daje wyniki przybliżone. Można korygować wartość niepewności wypadkowej m. in. poprzez przemnożenie jej przez współczynnik zależny od rozkładów błędów cząstkowych w podobny sposób, jak zrobiono to przykładowo dla błędu kwantowania. Celowość stosowania tego rodzaju współczynników zależy od wymaganej dokładności modelu niepewności. Porównania skuteczności różnych sposo-

to equal  $\Delta_{r,x} = \frac{q}{2}$ , whereas the variance of the error distribution is equal to  $\sigma_x^2 = \frac{q^2}{12}$ . Hence,

Having substituted this value to the expression (28) one obtains the following relationship describing the uncertainty transformation coefficient for the input error uniform distribution

The uncertainty composition procedure determines arithmetical operations which should be performed on the partial uncertainties in order to obtain the resultant uncertainty. For further considerations let us use the formula proposed by [11] which can be written for two components in the form

where  $\Delta_{a,b}$  is the resultant uncertainty,  $\Delta_a$  and  $\Delta_b$  are the partial uncertainties and  $\rho_{ab}$  - the correlation coefficient of the partial sets. For the assumed uncertainty definitions this relationship is strict only for the normal distributions of the partial error sets. In other cases the equation (31) gives approximate results. The resultant uncertainty value can be improved by, among others, multiplying it by the coefficient dependent on the partial error distributions, in a way similar to that used when determining the random uncertainty coefficient. The reason of applying this kind of coefficients depends on the required accuracy of the uncertainty model. Comparison of efficiency of the different

bów korygowania niepewności wypadkowej można dokonywać w sposób symulacyjny przy użyciu modelu błędów algorytmu.

W przypadku gdy nie zachodzi skorelowanie błędów, wyrażenie (31) przyjmuje prostszą postać

$$\Delta_{a,b} = \sqrt{\Delta_a^2 + \Delta_b^2}. \quad (32)$$

Tak realizowane składanie niepewności nazywane jest często „geometrycznym”. Stosowane jest również w sytuacji, gdy brak jest możliwości wyznaczenia współczynnika korelacji. Daje ono dobre rezultaty w praktyce pomiarowej, gdy występuje wiele źródeł błędów zarówno o charakterze układowym, jak i programowym [9].

Poniższy przykład ilustruje sposób postępowania przy budowie modelu niepewności algorytmu.

#### Przykład

Rozpatrzmy 5-punktowy filtr wygładzający opisany następującym równaniem

$$\hat{y}(n) = \frac{1}{8}\bar{y}(k) + \frac{1}{4}\bar{y}(k+1) + \frac{1}{4}\bar{y}(k+2) + \frac{1}{4}\bar{y}(k+3) + \frac{1}{8}\bar{y}(k+4), \quad (33)$$

gdzie  $\hat{y}(n)$  jest oceną w chwili o numerze  $n$  wielkości  $y$  poddawanej wygładzaniu,  $n = 0, 1, \dots$ ,  $\bar{y}(k), \dots, \bar{y}(k+4)$  są kolejnymi wynikami pomiaru tej wielkości obciążonymi jedynie błędami losowymi. Celem działania filtru jest zmniejszenie wartości błędu losowego wyników wyjściowych filtru w stosunku do błędu wartości wejściowych. Na ogół przyjmuje się, że  $n = k + 2$  i wówczas równanie (33) przyjmuje postać

$$\hat{y}(k) = \frac{1}{8}\bar{y}(k-2) + \frac{1}{4}\bar{y}(k-1) + \frac{1}{4}\bar{y}(k) + \frac{1}{4}\bar{y}(k+1) + \frac{1}{8}\bar{y}(k+2). \quad (34)$$

Założmy następnie, że właściwości filtru rozpatrywane są dla sinusoidalnie zmiennej wielkości przetwarzanej opisanego równaniem  $y(t) = \sin\left(\frac{2\pi}{T}t\right)$ , gdzie  $T$  jest okre-

kinds of the resultant uncertainty improvement can be made by simulation when using the algorithm error model.

In case of the uncorrelated errors the expression (31) simplifies to the form

Such a way of the uncertainty composition is often called „geometrical”. It is also used in situation when the correlation coefficient cannot be determined. It gives good results in measuring practice when there occur many error sources both of hardware and software types[9].

The following example illustrates the way of building the algorithm uncertainty model.

#### Example

Let us consider 5-point smoothing filter described by the equation

where  $\hat{y}(n)$  is the evaluation of the quantity  $y$  being smoothed at the instant  $n$ ,  $n = 0, 1, \dots$ ,  $\bar{y}(k), \dots, \bar{y}(k+4)$  are the successive results of measurement of this value loaded by the random errors only. The operation aim of the filter is to decrease the random error values of the filter output results in relation to the error of the input values. It is generally assumed that  $n = k + 2$ . Then the equation (33) can be written in the form

Next, let us assume that the filter properties for the sinusoidal input quantity given by the formula  $y(t) = \sin\left(\frac{2\pi}{T}t\right)$  are considered, where  $T$  is the period. The value

sem. Wartość wielkości  $y$  w chwili  $k$  opisuje wyrażenie

of the quantity  $y$  at the instant  $k$  is described by

$$y(k) = \sin\left(\frac{2\pi T_s}{T} k\right) \quad (35)$$

przy założeniu, że próbkowanie w kolejnych chwilach  $k$  jest realizowane co okres czasu  $T_s$ . Przyjmijmy następnie, że błędy losowe są wynikiem kwantowania próbek przez przetwornik A/C, a zatem każdą wartość podawaną na wejście algorytmu można ogólnie przedstawić jako

when assuming that sampling at the successive instant  $k$  is made with the period  $T_s$ . Next, let us assume that the random errors are the result of quantization of the samples by the A/D converter. So, each value delivered to the algorithm input can be generally expressed as

$$\bar{y}(k) = y(k) + \delta_q(k), \quad (36)$$

gdzie  $\delta_q(k)$  jest błędem kwantowania. Zakładając, że kwantowanie dokonywane jest przez 13-bitowy przetwornik A/C dostarczający danych w kodzie BCD, o zakresie przetwarzania od -1 do +1, niepewność zbioru wartości wejściowego błędu losowego jest stała i wynosi

where  $\delta_q(k)$  is the quantization error. Assuming that the quantization is made by a 13-bit A/D converter delivering data in the BCD code with the input range from +1 to -1, the uncertainty of the set of the input random error values is constant and equals

$$\Delta_q = \Delta_{r,\bar{y}} = \frac{q}{2} = \frac{1 - (-1)}{2 \cdot 2 \cdot 10^3} = 5 \cdot 10^{-4}. \quad (37)$$

Niepewność ta jest w modelu przenoszona z wejścia na wyjście zgodnie z zależnością

This uncertainty is transformed from the model input to the output according to the equation

$$\Delta_n = k_r \Delta_q, \quad (38)$$

w którym zgodnie z równaniem (30)

where, taking into account (30)

$$k_r = \sqrt{3} \sqrt{\left(\frac{1}{8}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{8}\right)^2} = 0,81. \quad (39)$$

Przyjmując, że błąd kwantowania jest jedynym błędem wprowadzanym przez układy przetwarzania A/C, to pozostałe źródła błędów związane są z właściwościami samego algorytmu. Załóżmy, że w rozpatrywanej sytuacji algorytm wprowadza jedynie błąd dynamiczny związany ze zmianami wielkości wejściowej. Zgodnie z przyjętymi definicjami błąd ten charakteryzuje niepewność opisana wyrażeniem

Assuming that the quantization error is the only error introduced by A/D converters the other error sources are connected with the properties of the algorithm itself. Let us assume that in the considered case the algorithm introduces only the dynamic error caused by the input quantity changes. According to the consideration in Section 3 this error is characterized by the following uncertainty

$$\Delta_{dA} = |S(j\omega) - S_{ideal}(j\omega)|, \quad (40)$$

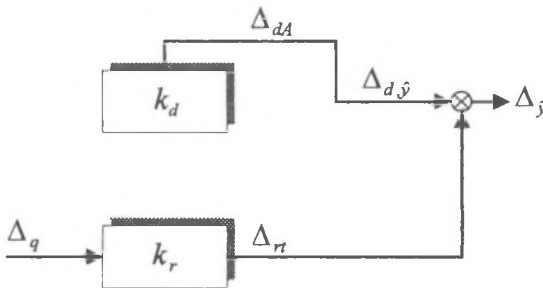
która po uwzględnieniu zależności (12), (17) i (34) po odpowiednich przekształceniach przyjmuje postać:

$$\Delta_{dA} = \left| \frac{1}{8} \cos(-2\omega T_s) + \frac{1}{4} \cos(-\omega T_s) + \frac{1}{4} + \frac{1}{4} \cos(\omega T_s) + \frac{1}{8} \cos(2\omega T_s) \right|. \quad (41)$$

Model niepewności filtru w postaci graficznej przedstawia rys.4. W modelu tym ujęte są dwa źródła błędów: kwantowania scharakteryzowane niepewnością  $\Delta_q$  o wartości stałej danej wyrażeniem (37) i błędu dynamicznego opisanego niepewnością  $\Delta_{dA}$  interpretowaną jako amplituda błędu własnego filtru. Zakładając, że błędy te nie są skorelowane, niepewność wyjściową opisuje zależność

$$\Delta_{\hat{y}} = \sqrt{(k_r \Delta_q)^2 + \Delta_{dA}^2}, \quad (42)$$

która dla rozpatrywanego przykładu stanowi zapis matematyczny modelu niepewności. Zapis ten pozwala na analityczne wyznaczanie niepewności wyjściowej na podstawie znanych wartości niepewności cząstkowych - wejściowych i własnych - algorytmu.



Rys.4. Model niepewności przykładowego filtru wygładzającego

Having taken into account the formulae (12), (17) and (34) and having rearranged the equation this equation can be written in the form

The filter uncertainty model is shown in Fig.4. It contains two error sources: the quantization one characterized by the uncertainty  $\Delta_q$  with the constant value given by (37) and the dynamic error source described by the uncertainty  $\Delta_{dA}$  treated as the amplitude of the filter specific error. Suppose that these errors are non-correlated. Then, the output uncertainty is described by the relation

which for the considered example is the mathematical notation of the uncertainty model. This notation enables to determine analytically the output uncertainty basing on the known values of the partial uncertainties - input and specific ones - of the algorithm.

Fig.4. Uncertainty model of the exemplary smoothing filter

W tabeli 1 zestawiono wyniki wyznaczenia niepewności wyjściowej algorytmu wygładzania zgodnie z zależnością (42) oraz w sposób symulacyjny. Symulację przeprowadzono dla sinusoidalnie zmiennej wielkości wejściowej algorytmu o amplitudzie równej 1, czyli przyjmującej wartości z całego zakresu przetwarzania. Dla określonej względnej wartości częstotliwości próbkowania  $f_s^o$ , równej co do wartości liczbie próbek przypadających na okres sinusoidy, wybierano w sposób losowy moment  $k$ , dla którego wyznaczano 5 kolejnych wartości  $y(k-2), \dots, y(k+2)$ . Wartości próbek poddawano następnie kwantowaniu przy założeniu, że wartość kwantu wynosi  $q = 0,001$ . Następnie wykonywano działania na skwantowanych próbkach zgodnie z postacią algorytmu (34). Od uzyskanej tym sposobem wartości oceny wielkości wyjściowej  $\hat{y}(k)$  odejmowano wartość prawdziwą  $y(k)$ . Tego rodzaju działania powtórzono 100 000 razy. Uzyskane wartości błędów zestawiono w histogram, na podstawie którego przy użyciu definicji niepewności wyznaczano niepewność opisaną w tabeli 1 jako  $\Delta_{\hat{y}_{sym}}$ .

The results of calculation of the output uncertainty of the smoothing algorithm from the relation (42) and by simulation are presented in Table 1. The simulation was made for the sinusoidal input quantity of the amplitude equal to 1, that is taking values within the whole input range. For the given relative value of the sampling frequency  $f_s^o$ , equal the number of the samples during the sinusoid period, the instant  $k$  was chosen in the random way. The five successive values  $y(k-2), \dots, y(k+2)$  were determined for this instant. The values of the samples were quantized next, assuming the quantum value to be  $q = 0,001$ . Further calculations were made on the quantized samples according to the algorithm (34). The true value  $y(k)$  was subtracted from the value of the output evaluation  $\hat{y}(k)$  obtained in this way. Such calculations were repeated 100000 times. The obtained error values were presented in the form of histogram which together with the uncertainty definition were the base for determining the uncertainty denoted as  $\Delta_{\hat{y}_{sym}}$  in Table 1.

Tabela 1

Zestawienie niepewności  $\Delta_{\hat{y}}$  wyznaczonej analitycznie z niepewnością  $\Delta_{\hat{y}_{sym}}$  wyznaczoną w sposób symulacyjny dla wybranych wartości względnej częstotliwości próbkowania  $f_s^o = \frac{T}{T_s}$

$f_s^o$	250	300	350	400	450	500
$\Delta_{\hat{y}} \cdot 10^{-4}$	6,2	5,2	4,7	4,5	4,3	4,2
$\Delta_{\hat{y}_{sym}} \cdot 10^{-4}$	7,4	6,2	5,3	4,9	4,8	4,8

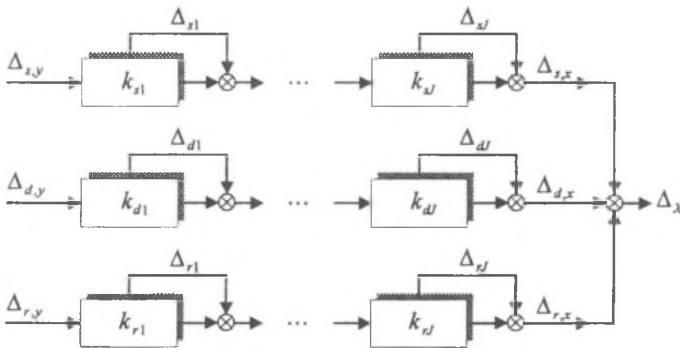
Table 1

The uncertainty calculated analytically  $\Delta_{\hat{y}}$  and the uncertainty determined by simulation  $\Delta_{\hat{y}_{sym}}$  for the chosen values of the relative sampling frequency  $f_s^o = \frac{T}{T_s}$



## 6. MODEL NIEPEWNOŚCI ŁAŃCUCHA ALGORYTMÓW

Uproszczenie działań prowadzących do wyznaczenia niepewności, które uzyskuje się wykorzystując model niepewności algorytmu, staje się szczególnie istotne w przypadku realizacji łańcucha algorytmów. Występuje to często w sytuacji, gdy stosuje się algorytmy odtwarzania łącznie z algorytmami filtracji, co ma na celu zmniejszenie wartości błędów losowych z reguły bardzo silnie wzmacnianych w trakcie odtwarzania [2]. Ponadto, gdy odtwarza się wielkości dynamiczne przetwarzane w układach nieliniowych, celowe jest stosowanie dekompozycji toru przetwarzania, aby uzyskać proste wersje algorytmów odtwarzania. Przykładem zastosowania modeli niepewności do wyznaczania niepewności odtwarzania z złożonych warunków opisywanych nieliniowymi, statycznymi i dynamicznymi równaniami jest praca [9]. Pokazuje ona m.in., że w warunkach, w których występuje wiele źródeł błędów o charakterze statycznym, dynamicznym i losowym, geometryczne składanie niepewności cząstkowych daje rezultaty akceptowalne w praktyce także wtedy, gdy nie znane są rozkłady błędów cząstkowych i nie są dostępne informacje o ich ewentualnym skorelowaniu.



Rys.5. Model niepewności łańcucha algorytmów

## 6. UNCERTAINTY MODEL OF THE ALGORITHM CHAIN

Simplification of calculations which is obtained when using the uncertainty model is particularly essential in case of realization of the algorithm chain. It often takes place if one applies the reconstruction algorithms together with the smoothing algorithms which aims at decreasing the random error values amplified strongly during the reconstruction [2]. Moreover, if one reconstructs dynamic quantities processed in nonlinear systems it is useful to decompose the processing chain in order to obtain the simple versions of the reconstruction algorithms. The doctor thesis [9] is an example of application of the uncertainty models to determining the reconstruction uncertainties in complex conditions described by nonlinear, static and dynamic equations. It shows, among others, that in the above situation geometrical composition of the partial uncertainties gives results acceptable in practice also when the distributions of the partial errors are unknown and the information about their correlation is not available.

Fig.5. Model of the algorithm chain uncertainties

Na rys.5 przedstawiono ogólny model niepewności łańcucha  $J$  algorytmów. Zakładając, że współczynniki przenoszenia poszczególnych algorytmów dla błędów statycznych, dynamicznych i losowych są oznaczone jako  $k_{sj}, k_{dj}, k_{rj}, j = 1, \dots, J$ , niepewności cząstkowe na wyjściu algorytmu powstałe na skutek przeniesienia odpowiednich niepewności z wejścia na wyjście mają postać

$$\begin{aligned}\Delta_{st} &= \left( \prod_{j=1}^J k_{sj} \right) \Delta_{s,x}, \\ \Delta_{dt} &= \left( \prod_{j=1}^J k_{dj} \right) \Delta_{d,x}, \\ \Delta_{rt} &= \left( \prod_{j=1}^J k_{rj} \right) \Delta_{r,x}.\end{aligned}\quad (43)$$

Powyższy zespół równań stanowi zapis reguły propagacji niepewności z wejścia na wyjście łańcucha algorytmów. Reguła ta dla propagacji niepewności własnej na wyjście dla algorytmu o numerze  $j = 1, \dots, J - 1$  przyjmuje postać

$$\begin{aligned}\Delta_{s(j)} &= \left( \prod_{i=j+1}^J k_{si} \right) \Delta_{sj}, \\ \Delta_{d(j)} &= \left( \prod_{i=j+1}^J k_{di} \right) \Delta_{dj}, \\ \Delta_{r(j)} &= \left( \prod_{i=j+1}^J k_{ri} \right) \Delta_{rj}.\end{aligned}\quad (44)$$

Powyższe zależności umożliwiają wyznaczanie niepewności cząstkowych na wyjściu modelu. Przyjmując regułę składania niepewności można uzyskać niepewności wypadkowe. Zakładając, że składanie wykonywane jest w sposób geometryczny,

The general model of the  $J$ -algorithm chain uncertainties is shown in Fig.5. Assuming that the transformation coefficients of the particular algorithms for the static, dynamic and random errors are denoted as  $k_{sj}, k_{dj}, k_{rj}, j = 1, \dots, J$ , the partial uncertainties at the algorithm output, arisen as a result of transformation of the appropriate uncertainties from the input to the output, are expressed by

The above set of equations represents the rule of the uncertainty propagation from the input to the output of the algorithm chain. This rule for the specific uncertainty propagation for the algorithm of the number  $j = 1, \dots, J - 1$  is given by

The above relations make possible to calculate the partial uncertainties at the model output. Defining the rule of the uncertainty composition one can get the resultant uncertainties. Assuming geometrical composition, the resultant uncertainty

niepewność wypadkową określają poniższe równania, odpowiednio dla niepewności statycznej, dynamicznej i losowej

$$\begin{aligned}\Delta_s^2 &= \Delta_{st}^2 + \sum_{j=1}^{J-1} \Delta_{s(j)}^2 + \Delta_{st}^2, \\ \Delta_d^2 &= \Delta_{dt}^2 + \sum_{j=1}^{J-1} \Delta_{d(j)}^2 + \Delta_{dt}^2, \\ \Delta_r^2 &= \Delta_{rt}^2 + \sum_{j=1}^{J-1} \Delta_{r(j)}^2 + \Delta_{rt}^2,\end{aligned}\quad (45)$$

Całkowita niepewność wyjściowa opisana jest zależnością

$$\Delta_x^2 = \Delta_s^2 + \Delta_d^2 + \Delta_r^2. \quad (46)$$

Powyższy model ma zastosowanie przede wszystkim do analitycznego wyznaczenia niepewności na wyjściu łańcucha algorytmów. Można go również używać do analizy metrologicznej. Odpowiednio modyfikując powyższe równania można wyznaczać niepewności w dowolnym punkcie łańcucha, co pozwala na rozpatrywanie wpływu różnych czynników na niepewność wypadkową. Tym samym model umożliwi poszukiwanie źródeł błędów najsilniej wpływających na tę niepewność i w konsekwencji środków do ich redukcji.

## 7. UWAGI KOŃCOWE

Właściwością każdego procesu pomiarowego jest uzyskiwanie wyników o ograniczonej dokładności, czego liczbowym wyrazem jest niepewność wyniku pomiaru. Tradycyjnie przyjmuje się, że niepewność opisuje zbiór wyników pomiaru charakteryzując zawarte w nich błędy, które traktowane są jako nie dające się oddzielić od wyników. W efekcie podstawą procedury wyznaczania niepewności jest statystyczna

is described by the following equations for the static, dynamic and random uncertainties, respectively

The resultant output uncertainty is expressed by

The above model is, first of all, applied to the analytical determination of the uncertainties at the algorithm chain output. It can also be used for metrological analysis. Modifying appropriately the above equations one can calculate the uncertainties at any point of the chain, which allows to consider the influence of different factors on the resultant uncertainty. Thus, the model enables to search the most essential error sources for this uncertainty and, as a consequence, the means for its reduction.

## 7. FINAL REMARKS

Each measuring process is inaccurate. Measure of its accuracy is the uncertainty of the final result. It is classically assumed that the uncertainty describes the set of the measuring results, thus characterizing the errors included in them which are treated as inseparable from the measuring results. Hence, the base of the procedure for determining the uncertainty is statistical processing of the set of the measuring data. In

obróbka zbioru wyników pomiaru. W przypadku algorytmu przetwarzania zastosowanie takiego podejścia jest z reguły bardzo trudne, ponieważ pojedyncza realizacja algorytmu dostarcza jednego wyniku pomiarowego. Powtórzenie działań na tych samych danych wejściowych skutkuje powtórzeniem wyniku, natomiast realizacja algorytmu na innym zbiorze danych daje inny wynik wyjściowy. Zatem w przypadku algorytmu uzyskanie zbioru danych wyjściowych, który mógłby być poddany obróbce statystycznej, jest w ogólnym przypadku trudne do zrealizowania.

Z powyższych względów w niniejszej pracy zastosowano inne podejście. Liniowe właściwości algorytmu umożliwiają oddzielne rozpatrywanie przetwarzania danych i ich błędów, a także wyodrębnianie różnych rodzajów błędów. Dzięki temu możliwe jest odrębne tworzenie opisów źródeł błędów dla poszczególnych ich rodzajów uwzględniając specyfikę działań realizowanych przez algorytm. Definiując niepewność jako parametr zbioru wartości błędów można wyznaczyć równania łączące niepewność zbioru na wyjściu algorytmu z niepewnościami zbiorów wejściowych i zbiorów błędów własnych. Powstały w ten sposób model niepewności cechuje prostotą, ponieważ nie zależy on od liczby wyrazów algorytmu. Istota modelu polega na analitycznym zapisie zależności łączących niepewności rozumiane jako parametry zbiorów wartości błędów.

Opisana w artykule metoda pozwala na uzyskiwanie modeli uproszczonych w sensie matematycznym, co wynika z zastosowanej procedury składania niepewności. Jednak w praktyce pomiarowej nie ma to istotnego znaczenia, bowiem wszystkie ścieżki wyznaczania niepewności cechuje relatywnie mała dokładność. Ważne jest natomiast, że w przypadku stosowania modeli niepewności możliwa jest weryfikacja skutków zastosowanych uproszczeń za pomocą badań symulacyjnych.

Przedstawiona metoda budowy modeli niepewności nie koliduje zasadniczo z propozycjami zawartymi w [11], można wręcz stwierdzić, że je w pewnym stopniu uzu-

case of the algorithm it is very difficult to do it, as a rule, because the algorithm single realization delivers one measuring result. Repeated performance of the algorithm on the same input data causes repetition of the result, whereas the algorithm realization on the other data set changes the processing conditions. So, in general case, obtaining the output data set on which statistical calculation could be made may be difficult to perform.

Because of that, the other approach to this subject is presented in the paper. Linear properties of the algorithm make possible to separate the error processing from the data processing, as well as to decompose the different kinds of errors. Owing to it, the separate creation of descriptions of the error sources for their particular kinds can be made when taking into account the specificity of calculations performed by the algorithm. Defining the uncertainty as a quantity characterizing the error value set it is possible to determine the equations connecting the uncertainty of the error set at the algorithm output with the uncertainties of the input sets and the specific error sets. The uncertainty model built in this way is simple because it does not depend on the number of the algorithm elements. The model essence consists in the analytical description of the relationships connecting the uncertainties treated as parameters of the error value sets.

The method described in the paper allows to get the simplified mathematical models. It is possible due to the applied procedure of the uncertainty composition. However, these simplifications are not of great importance in measuring practice because all the ways of determining the uncertainty are of relatively low accuracy. Possibility of the verification of the made simplification effects by simulation is the essential advantage of the use of the uncertainty models.

The presented method of the construction of the uncertainty models is not contradictory to the propositions described in [11] so, one may state that it is their complement to a certain degree. Adding

pełnia. Poszerzenie słownika tam zawartego o odpowiednie pojęcia pozwoliłoby na uzyskanie podstaw budowy modeli niepewności. Największą przeszkodą jest sama definicja niepewności odnosząca się w przypadku omawianej metody do zbioru wartości błędu, a tradycyjnie do zbioru wyników pomiaru. Sądzę, że pewne kroki w kierunku wprowadzenia do terminologii metrologicznej pojęcia modelu niepewności są bardzo wskazane. Modele takie mają bowiem cechę bardzo istotną dla współczesnej, silnie skomputeryzowanej techniki pomiarowej - pozwalają na algorytmizację procedur wyznaczania niepewności. Tym samym umożliwiają samookreślanie dokładności przez programowane przyrządy pomiarowe realizujące złożone programowo-sprzętowe procedury przetwarzania w zmiennych warunkach pomiarowych.

new terms to the vocabulary presented in it would allow to obtain a base for construction of the uncertainty models. The uncertainty definition in [11] referring to the measuring result set is the greatest difficulty in the way of building the uncertainty models by the method proposed in the paper which consists in calculations performed on the error value sets. I think, that it would be advantageous to introduce the term of the uncertainty model to metrological terminology. Such models have the property which is very important for modern computerized technique - they allow algorithmization of the uncertainty determination procedures. Thus, they enable self-determining the accuracy by programmed measuring instruments realizing complex software and hardware processing procedures in variable measuring conditions.

#### LITERATURA

#### REFERENCES

1. Betta G., Liguori C., Pietrosanto A.: Uncertainty Analysis in Fast Fourier Transform Algorithm. IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation and 3<sup>rd</sup> Workshop on ADC Modelling and Testing, Sep. 17-18, 1998 - Naples, Italy, pp.747-752.
2. Jakubiec J.: Bieżące programowe odtwarzanie wartości chwilowych dynamicznych przebiegów wejściowych nieliniowych przetworników pomiarowych. Real-Time Numerical Reconstruction of Instantaneous Values of Dynamic Input Quantities of Nonlinear Transducers (in Polish). Monografia. Zeszyty Naukowe Pol. Śl., ser. Elektryka z.111, Gliwice 1988.
3. Jakubiec J.: Metrologiczne własności pewnej postaci szeregu czasowego jako sposobu przedstawiania rekurencyjnych algorytmów przetwarzania danych pomiarowych. Metrological Properties of a Form of Time Series Used for Recurrent Algorithms Representation of Measured Data Processing (in Polish). Archiwum Elektrotechniki, t. XL, z. 157/158 - 3-4/1991, PWN Warszawa 1993, pp.723-735.
4. Jakubiec J.: Geometryczna metoda opisu przetwarzania w programowanym czujniku pomiarowym. Geometrical Method of Measuring Process Description in Programmable Sensor (in Polish). Metrologia i Systemy Pomiarowe. t. III, z. 3-4/1996, Wydawnictwo Naukowe PWN, Warszawa 1997, pp.225 - 249.
5. Jakubiec J., Roj J.: Metoda modelowania niepewności w torach pomiarowych o złożonej strukturze przetwarzania programowego. A method of Uncertainty Modelling for Measuring Chain with Complex Structure of Programmed Processing (in Polish). Mat. Konferencji „Systemy Pomiarowe w Badaniach Naukowych i w Przemysle” SP'98, Zielona Góra 1998, pp.65-80.
6. Jakubiec J., Konopka K.: Uncertainty Propagation Model of A/D Measuring Chain. IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation and 3<sup>rd</sup> Workshop on ADC Modelling and Testing, Sep. 17-18,1998 - Naples, pp. 831-836.

7. Jakubiec J., Topór-Kamiński T.: Uncertainty Modelling Method of Data Series Processing Algorithms. IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation and 3<sup>rd</sup> Workshop on ADC Modelling and Testing, Sep. 17-18, 1998 - Naples, Italy, pp.631-636.
8. Mesch F.: System Description of Imaging Methods - An Introduction. Proc. X-th IMEKO World Congress, Vol. 2, Praha 1985, pp. 49-56.
9. Roj J.: Pomiar stężenia mieszanki palnej za pomocą mostka pelistorowego zasilanego impulsowo. Measurement of Combustible Mixture Concentration by Means of Pellistor Bridge with Impulse Supply Voltage (in Polish). Rozprawa doktorska. Politechnika Śląska, Gliwice 1998.
10. Topór-Kamiński T.: Wpływ struktury numerycznej algorytmu na przenoszenie losowych błędów danych. Algorithm Numerical Structure Influence on Data Stochastic Error Transmission (in Polish). Mat. XXIX Międzyuczelnianej Konferencji Metrologów, Nałęczów 1997, s.562-570.
11. Guide to the Expression of Uncertainty in Measurement, ISO/IEC/OIML/BIPM, 1992,1995. Wyrażanie niepewności pomiaru. Przewodnik. GUM 1999.

Recenzent: Dr hab. inż. Janusz GAJDA

Wpłynęło do Redakcji dnia 2 czerwca 1999 r.