

TADEUSZ TOPÓR-KAMIŃSKI

POLITECHNIKA ŚLĄSKA

INSTYTUT METROLOGII I AUTOMATYKI ELEKTROTECHNICZNEJ

## Błędy własne algorytmu przetwarzania czasowego ciągu danych pomiarowych

### Streszczenie

W pracy przedstawiono metodę wyznaczania błędów własnych algorytmów służących do przetwarzania ciągów danych pomiarowych, głównie ciągów czasowych. Algorytm traktowany jest jako jedno z ogniw toru pomiarowego, a więc analiza jego właściwości metrologicznych przeprowadzana jest tak jak dla przetwornika pomiarowego. Zaproponowano ogólny model błędów własnych umożliwiający wyznaczanie ich rozkładów na wyjściu algorytmu. Dla przykładowego algorytmu DFT zaproponowano ogólny model błędów, a następnie wyznaczono rozkład wybranego błędu własnego.

### Abstract

The paper presents a method of calculating own error of an algorithm used for processing a measurement data time sequence. Algorithm is treated as one of the elements of measuring system then its uncertainty has been analysed in the same way as measuring converter. An general algorithm error model allowing determination of own error distributions has been proposed. An example of obtaining the histogram of chosen error for DFT algorithm has been described.

### WSTĘP

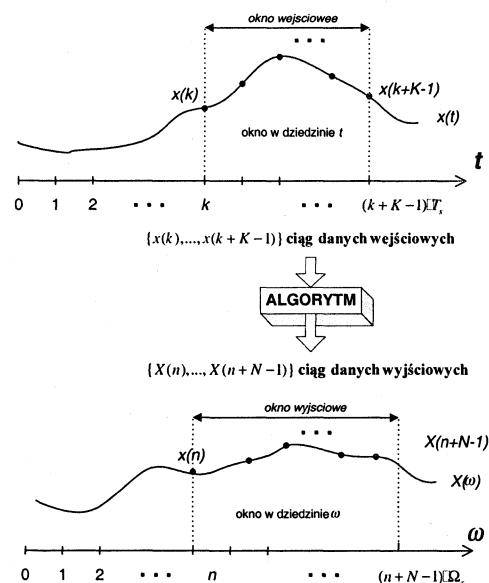
Współczesne pomiary wielkości zmiennych w czasie opierają się w większości na wykorzystaniu przetwarzania próbkującego [1]. Można wskazać wiele rozwiązań praktycznych realizujących ten cel, jednak w większości przypadków tor pomiarowy składa się z trzech grup elementów: układów analogowych, analogowo-cyfrowych oraz elementów realizujących przetwarzanie cyfrowe. Grupę elementów cyfrowych stanowią najczęściej różnego rodzaju algorytmy przetwarzania danych realizowane przez mikroprocesor.

Algorytm przetwarzania w ogólnym przypadku stanowi przepis na przekształcenie jednego ciągu danych pomiarowych na inny, jak to ogólnie pokazano na rys.1[2][3]. Ciąg danych wejściowych algorytmu złożony jest z pewnej ograniczonej liczby skwantowanych próbek wielkości  $x(t)$  wybranych z dowolnie licznego zbioru tych próbek za pomocą tzw. okna wejściowego. Algorytm przetwarza ciąg wejściowy na inny ciąg danych, nazywany ciągiem wyjściowym, reprezentujący jego wielkość wyjściową  $X(\omega)$  w ograniczonym zakresie zmienności  $\omega$ , czyli w oknie wyjściowym. Ogólnie można powiedzieć, że algorytm przetwarza dyskretną reprezentację

wielkości wejściowej  $x$  określonej w dziedzinie  $t$  na dyskretną wielkość  $X$  w dziedzinie  $\omega$ .

Zakładając, że wielkość analogowa  $x(t)$  jest próbkowana w chwilach  $t_k = kT_s$ , gdzie  $k$  jest numerem próbki,  $k = 0, 1, \dots, T_s$  jest okresem próbkowania w dziedzinie  $t$ ,  $T_s = \text{const.}$ , okno wejściowe wybiera z tego nieskończonego ciągu próbek pewien ciąg ograniczony składający się z próbek o numerach  $k, k+1, \dots, k+K-1$ , przy czym  $K$  jest liczbą próbek objętych tym oknem. Przyjmując dla prostoty, że okno zaczyna się od momentu rozpoczęcia pomiarów, czyli dla  $k = 0$ , ciąg wejściowy można zapisać jako  $\{x(k), k = 0, 1, \dots, K-1\}$ . Dziedziną wielkości wejściowej najczęściej jest czas. Do dalszych rozważań przyjmuje się czasową interpretację wejściowego ciągu danych.

Dla celów ogólnych rozważań zakłada się również, że wielkość wyjściowa algorytmu jest wielkością analogową.



Rys.1. Ilustracja działania algorytmu przetwarzania ciągu danych pomiarowych

W takim przypadku interpretacja działań algorytmu może być przeprowadzona w podobny sposób, jak operacji wykonywanych na wielkości wejściowej – mianowicie, algorytm dostarcza skończonego ciągu skwantowanych próbek swojej wielkości wyjściowej  $X(\omega)$ . Zakładając, że wielkość ta jest próbkowana w punktach  $\omega_n = n\Omega_s$ , gdzie  $n$  jest numerem próbki,  $n = 0, 1, \dots, \Omega_s$  jest okresem próbkowania w dziedzinie  $\omega$ ,  $\Omega_s = \text{const.}$ , ciąg wyjściowy można zapisać jako

$\{X(n), n = 0, 1, \dots, N-1\}$ . Również i w tym przypadku przyjęto dla prostoty, że początek okna ma miejsce w punkcie o numerze  $n = 0$ . W praktyce algorytmy wykonują działania na ocenach  $\hat{x}(k)$  wielkości wejściowej dokonując na wyjściu ocen  $\hat{X}(n)$  wielkości wyjściowej. W tej sytuacji algorytm przetwarzania może być ogólnie zapisany w postaci macierzowej jako

$$\hat{\mathbf{X}} = \mathbf{A}\hat{\mathbf{x}}, \quad (1)$$

gdzie

$$\hat{\mathbf{X}} = \begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{bmatrix}, \mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & \dots & a_{0, K-1} \\ a_{10} & a_{11} & \dots & a_{1, K-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1, 0} & a_{N-1, 1} & \dots & a_{N-1, K-1} \end{bmatrix}, \hat{\mathbf{x}} = \begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(K-1) \end{bmatrix}, \quad (2)$$

przy czym  $\hat{\mathbf{x}}$  jest wektorem danych wejściowych zawierający  $K$  elementów składających się na ciąg wejściowy, natomiast  $\hat{\mathbf{X}}$  jest wektorem danych wyjściowych o rozmiarze równym  $N$ . Macierz  $\mathbf{A}$  o rozmiarach  $N \times K$  zawiera współczynniki algorytmu. Dla większości algorytmów współczynniki te mają stałe wartości.

Dane wyjściowe niektórych algorytmów mają postać zespoloną. W takim przypadku algorytm stanowi właściwie parę algorytmów, z których jeden wyznacza wartości rzeczywiste, drugi urojone. Wyniki wyjściowe tworzą ciąg liczb zespolonych, który można przedstawić jako wektor

$$\hat{\mathbf{X}} = \hat{\mathbf{X}}_{\text{Re}} + j\hat{\mathbf{X}}_{\text{Im}}. \quad (3)$$

Istotnym wyróżnikiem algorytmu, ze względu na właściwości metrologiczne, jest liczba wyników uzyskiwanych na wyjściu po jednorazowym jego wykonaniu. I tak algorytm nazywa się jednopunktowym (prostym), jeżeli ciąg danych wejściowych przetwarzany jest na pojedynczy wynik wyjściowy. Algorytm wielopunktowy (złożony) przetwarza ten sam ciąg danych wejściowych na odpowiedni ciąg danych wyjściowych. Takim algorytmem jest dla przykładu algorytm dyskretnej transformaty Fouriera oznaczany symbolem DFT.

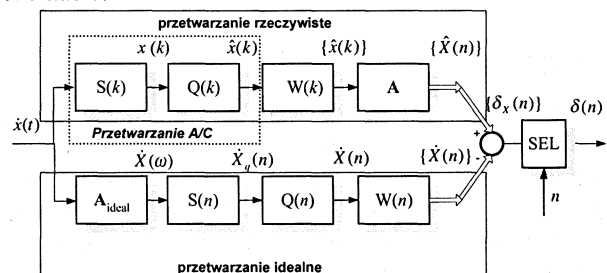
Z postaci zależności (1) i (2) wynika, że algorytm wielopunktowy można traktować jako równoległe połączenie algorytmów jednopunktowych o takiej samej strukturze, tzn. charakteryzujących się taką samą liczbą tego samego rodzaju współczynników (ich współczynniki różnią się jedynie wartościami). Zatem można uznać, że algorytm jednopunktowy dobrze reprezentuje różnego rodzaju algorytmy, a przy tym jego budowa jest stosunkowo prosta i dogodna dla celów analizy. Przedstawiając algorytm (1) w postaci  $N$  równań i biorąc następnie jedno z nich oraz pomijając indeks  $n$  współczynników, otrzymuje się prostszy zapis algorytmu

$$\hat{X}(n) = a_0\hat{x}(0) + a_1\hat{x}(1) + \dots + a_{K-1}\hat{x}(K-1), \quad (4)$$

czyli jako kombinacji liniowej stałych współczynników i ograniczonego ciągu wyników pomiaru reprezentującego wartości chwilowe wielkości mierzonej zmiennej w czasie.

## 2. BŁĘDY PRZETWARZANIA DANYCH POMIAROWYCH PRZEZ ALGORYTM

Liczbowym wyrazem dokładności wyniku pomiaru jest niepewność. Wiarygodna ocena niepewności wyniku wyjściowego algorytmu w postaci wyrażenia (4) - w złożonych warunkach, charakteryzujących się dużą liczbą źródeł błędów o różnym charakterze (statycznych, dynamicznych i losowych) i dużymi długościami przetwarzanych danych - wymaga uprzedniej identyfikacji wszystkich źródeł błędów. Oznacza to, że przed realizacją algorytmu muszą być znane rozkłady wszystkich błędów obarczających jego dane wejściowe, jak również rozkłady błędów wprowadzanych przez sam algorytm, a także ewentualne stopień skorelowania wymienionych błędów. Wynika stąd potrzeba użycia takiego sposobu definiowania błędów algorytmu, który umożliwia identyfikację poszczególnych źródeł tych błędów. W tym celu można posłużyć się schematem przedstawionym na rys.2 wynikającym bezpośrednio ze sposobu działania algorytmu przedstawionego na rys.1. Schemat ten opisuje wszystkie operacje mające bezpośredni związek z działaniem algorytmu, takie jak próbkowanie, kwantowanie próbek oraz wybór ciągu danych za pomocą okna. Pomija się wszystkie źródła błędów układów analogowych dostarczających sygnału na wejście układu próbkującego. Błędy te muszą być oczywiście znane, gdy chce się wyznaczyć całkowitą niepewność wyniku na wyjściu algorytmu, jednak sposób ich wyznaczania nie jest tutaj rozpatrywany. Przebieg wejściowy, oznaczony jako  $\hat{x}(t)$ , jest traktowany jako wzorcowy, czyli taki, którego przebieg znany jest z odpowiednio dużą dokładnością. Dzięki temu, dobierając odpowiednio warunki pracy algorytmu, można wyznaczać rozkłady poszczególnych źródeł błędów związanych z działaniami przedstawionymi na schemacie.



Rys.2. Model procesu wyznaczania błędów własnych algorytmu.

Model przedstawiony na rys.2 składa się z dwóch części. Górna, opisana jako przetwarzanie rzeczywiste, przedstawia sobie te działania, które muszą być wykonane, aby można było uzyskać ciąg wyników wyjściowych algorytmu. Pierwsze, oznaczone jako  $S(k)$ , polega na próbkowaniu przebiegu wejściowego  $\hat{x}(t)$  w równooddanych momentach o numerze  $k$ ,  $k = 0, 1, \dots$ . Kolejne, o symbolu  $Q(k)$ , polega na kwantowaniu uzyskanych w poprzednim ogniwie próbek. Wynik kwantowania  $\hat{x}(k)$  jest liczbą reprezentującą wartość prawdziwą wielkości mierzonej w chwili  $k$  z błędem będącym wypadkową błędów próbkowania i kwantowania. Trzecie działanie realizuje funkcję okna polegającą na wyborze  $K$  kolejnych skwantowanych próbek z nieskończonego ich ciągu (w rozpatrywanej sytuacji zakłada się, że jest to  $K$  początkowych próbek) oraz

zapamiętaniu ich, dzięki czemu uzyskuje się ich zbiór (ciąg) oznaczony jako  $\{\hat{x}(k)\}, k = 0, 1, \dots, K-1$ . Ostatnie działanie polega na przemnożeniu wyrazów ciągu wejściowego przez elementy kolejnych wierszy macierzy współczynników i zsumowaniu iloczynów w poszczególnych wierszach zgodnie z zależnością (2), w efekcie, czego uzyskuje się zbiór wyników wyjściowych oznaczony jako  $\{\hat{X}(n)\}$ .

Działania realizowane przez algorytm dodają specyficzne błędy do błędów poprzednich ogniw, czyli błędów próbkowania, kwantowania i błędów okna. Dolna część schematu modeluje te działania, które można nazwać przetwarzaniem idealnym, co oznacza, że wartość na wyjściu tego toru równa jest praktycznie wartości prawdziwej  $\dot{X}$  wielkości wyjściowej algorytmu (jej błąd jest pomijalnie mały). Na początku łańcucha działań występuje element nazywany algorytmem idealnym oznaczony symbolem  $A_{ideal}$ . Algorytm taki ma za zadanie przetworzyć wielkość  $\dot{x}(t)$  na wielkość  $\dot{X}(\omega)$  zgodnie z definicją zadania, do realizacji którego algorytm jest przeznaczony. Przykładowo, jeżeli rozpatrywany algorytm realizuje całkowanie numeryczne, to odpowiadający mu algorytm idealny wykonuje działania zgodnie z definicją całki oznaczonej. Z reguły algorytm idealny nie daje się zbudować, natomiast zawsze można wyznaczać wartości wielkości  $\dot{X}(\omega)$  na podstawie wyrażenia definicyjnego. Zatem algorytm idealny reprezentuje rozwiązywanie tego wyrażenia na przebiegu wzorcowym  $\dot{x}(t)$ . Kolejne działanie przetwarzania idealnego, oznaczone jako  $S(n)$ , polega na próbkowaniu przebiegu  $\dot{X}(\omega)$  w chwilach  $n$ . Następne ogniwo  $Q(n)$  nominalnie realizuje kwantowanie, ponieważ próbki wielkości wyjściowej  $\dot{X}(\omega)$  mają postać analogową, a na wyjściu toru powinna się pojawiać liczba o takiej samej postaci jak wynik działania algorytmu  $\hat{X}$ . Jednak kwantowanie to musi być znacznie dokładniejsze niż realizowane przez fizyczny przetwornik A/C zamodelowany w torze górnym przez element  $Q(x)$ , gdyż błąd kwantowania idealnego musi być pomijalnie mały. Ostatnie ogniwo o symbolu  $W(n)$  reprezentuje funkcje okna dokonując wyboru ciągu wyników wyjściowych  $\{\hat{X}(n)\}$  odpowiadających działaniom realizowanym przez rzeczywisty algorytm oznaczony jako  $A$ . Ostatecznie na wyjściu obu torów otrzymuje się wektor błędów algorytmu  $\{\delta_x(n)\}$ .

Opisany schemat z rys.2 ma przede wszystkim na celu stworzyć podstawę takiego sformalizowania opisu błędów przetwarzania, aby można było identyfikować wyodrębnione, pojedyncze źródła błędów i zidentyfikować ich rozkłady w postaci histogramu. W tym celu należy zastosować ogniwo oznaczone jako SEL, które dokonuje selekcji wybranego błędów z wektora błędów algorytmu. Tor przetwarzania idealnego dostarcza wyniku dokładnego. Natomiast dobierając warunki przetwarzania rzeczywistego w taki sposób, żeby dominowało tylko jedno źródło błędów, można uzyskać wartość tego błędów jako różnicę

$$\delta_x(n) = \hat{X}(n) - \dot{X}(n) \quad (5)$$

między wartością  $\hat{X}(n)$  wyznaczoną za pomocą algorytmu a wartością dokładną (prawdziwą)  $\dot{X}(n)$ . Powtarzając tego rodzaju eksperyment wielokrotnie uzyskuje się zbiór wartości określonego rodzaju błędów.

Z budowy modelu błędów przedstawionego na rys.2 wynika, że błędy przetwarzania danych przez algorytm (w skrócie błędy algorytmu) można podzielić na cztery grupy – błędy próbkowania, kwantowania, błędy okna oraz błędy realizacji operacji numerycznych przez algorytm, przy czym poprzez błędy próbkowania i kwantowania rozumie się błędy integralnie związane z algorytmem a wynikające z faktu, że algorytm operuje na dyskretnej reprezentacji przebiegu ciągłego. Wyznaczając wszystkie istotne błędy algorytmu i określając miejsce ich powstawania buduje się model błędów algorytmu [2][3][4]. To, ile i jakiego rodzaju błędy należy uwzględnić w tym modelu, zależy od specyfiki algorytmu. Model błędów algorytmu z rys.2 służy do identyfikacji źródeł błędów algorytmu. Innym zastosowaniem modelu jest systematyzacja opisu właściwości algorytmu ze względu na wprowadzane błędy.

Poniżej przedstawiono sposób zastosowania opracowanego modelu do identyfikacji źródeł błędów algorytmu DFT.

### 3. BŁĘDY ALGORYTMU DFT

Algorytm DFT służy do wyznaczania dyskretnych wartości transformaty Fouriera przebiegu czasowego w określonym przedziale częstotliwości  $f$ . Zatem algorytm idealny w tym przypadku ma postać funkcjonału całkowego opisującego przekształcenie Fouriera

$$\dot{X}(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt = A_{ideal}[\dot{x}(t)] = \text{Re}A_{ideal}[\dot{x}(t)] + j \text{Im}A_{ideal}[\dot{x}(t)] \quad (6)$$

Funkcjonał ten służy do dostarczania dokładnych wartości wielkości wyjściowej algorytmu dla znanych przebiegów wielkości wejściowej  $\dot{x}(t)$ . Są one porównywane z odpowiednimi wartościami uzyskiwanymi na wyjściu rzeczywistego algorytmu, którego błędy są identyfikowane. Różnica stanowi błąd zgodnie z wyrażeniem (5).

Algorytm DFT jest algorytmem wielopunktowym, a poszczególne wiersze jego macierzy współczynników, zgodnie z zależnością (3), przyjmują postać:

$$\text{Re}^n A = \left[ \cos \frac{2\pi \cdot n \cdot 0}{N}, \cos \frac{2\pi \cdot n \cdot 1}{N}, \dots, \cos \frac{2\pi \cdot n \cdot k}{N} \right] \quad (7)$$

$$\text{Im}^n A = \left[ \sin \frac{2\pi \cdot 0 \cdot n}{N}, \sin \frac{2\pi \cdot 1 \cdot n}{N}, \dots, \sin \frac{2\pi \cdot k \cdot n}{N} \right] \quad (8)$$

przy czym  $k = 0, 1, \dots, K-1$ , natomiast  $n$  jest numerem wiersza a zarazem numerem harmonicznej, gdzie  $n = 1, 2, \dots, N-1$ .

Zatem algorytm DFT może być przedstawiony jako równoległy zespół algorytmów „jednopunktowych” reprezentowanych przez poszczególne wiersze macierzy współczynników począwszy od wiersza pierwszego, operujących na takim samym ciągu danych wejściowych. Wielkość wyjściową algorytmu DFT otrzymuje się z pary równań, z których jedno służy do wyznaczania części rzeczywistej,

drugie – urojonej. Pojedynczy wynik na wyjściu algorytmu ma postać:

$$\hat{X}_{\text{Re}}(n) = \sum_{k=0}^{K-1} \hat{x}(t_k) \cos \frac{2\pi \cdot n \cdot k}{N} \quad (9)$$

$$\hat{X}_{\text{Im}}(n) = \sum_{k=0}^{K-1} \hat{x}(t_k) \sin \frac{2\pi \cdot n \cdot k}{N}, \quad (10)$$

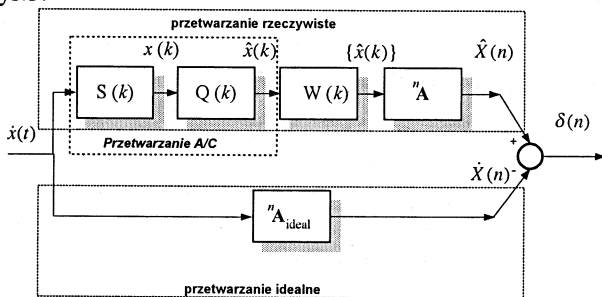
gdzie  $n$  częstotliwością  $n$  harmoniczną obliczanej przy użyciu  $n$  wiersza macierzy współczynników algorytmu.

Dla każdego z algorytmów jednopunktowych DFT, opisanych zależnościami (9) i (10) można wskazać algorytm idealny w postaci zależności:

$$\dot{X}_{\text{Re}}(n) = \int_{-\infty}^{+\infty} x(t) \cos(2\pi nt) dt = {}^n \mathbf{A}_{\text{ideal}}[\dot{x}(t)] \quad (11)$$

$$\dot{X}_{\text{Im}}(n) = \int_{-\infty}^{+\infty} x(t) \sin(2\pi nt) dt = {}^n \mathbf{A}_{\text{ideal}}[\dot{x}(t)]. \quad (12)$$

Dla tak opisanego jednopunktowego algorytmu DFT można zbudować uproszczony model błędów przedstawiony na rys.3.

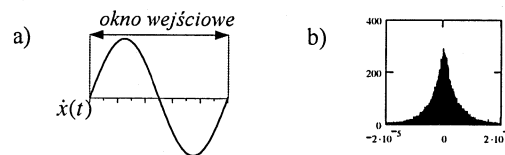


Rys.3. Model procesu wyznaczania błędów własnych jednopunktowego algorytmu DFT.

Model przedstawiony na rys.3 pozwala wyznaczyć rozkłady błędów dla poszczególnych algorytmów jednopunktowych DFT reprezentujących odpowiednie harmoniczne. Pojedyncza realizacja działań przedstawionych w powyższym modelu dostarcza pojedynczego wyniku wyjściowego dla wybranej harmonicznej, a wielokrotna realizacja działań dla kolejnych wierszy macierzy współczynników umożliwia wyznaczanie rozkładów błędów dla całego widma częstotliwościowego. Identyfikacja błędów przeprowadzana jest z reguły w sposób symulacyjny, co pozwala na taki dobór warunków pracy algorytmu, aby dominowały błędy z wybranego źródła. Symulacyjne badanie algorytmu niewiele różni się od badań fizycznych, ponieważ algorytm w obu sytuacjach realizowany jest przez procesor. Natomiast badanie symulacyjne pozwala na takie kształtowanie warunków pomiaru, które z reguły jest niemożliwe lub bardzo trudne do realizacji na rzeczywistych sygnałach. Poniżej opisano wykorzystanie modelu do wyznaczenia rozkładu błędów własnych algorytmu DFT.

Algorytm DFT cechuje duża liczba współczynników sięgająca kilku tysięcy. W takim przypadku istotne wartości może przyjmować błąd spowodowany realizacją operacji arytmetycznych na ciągu danych wejściowych. Główna przyczyna tego błędu tkwi w zaokrągleniu wyniku po każdej operacji do tylu miejsc znaczących, ile wynika z tzw. precyzji zapisu określonej przez liczbę bitów przeznaczonych

na zapis mantysy. Błąd ten można, zatem nazywać błędem zaokrąglenia. Standard IEEE 754 [5] dla binarnej arytmetyki zmiennoprzecinkowej określa cztery dokładności - liczb zmiennoprzecinkowych liczby pojedynczej precyzji, podwójnej precyzji i odpowiednio liczby pojedynczej precyzji rozszerzone oraz podwójnej precyzji rozszerzone. Aby wyznaczyć rozkład tego błędu należy tak dobrać warunki badań, aby ujawniał się tylko błąd zaokrąglenia. W tym celu można wybrać wielkość wejściową algorytmu o takim przebiegu, aby błąd próbkowania i błąd okna były równe zero. Warunki te spełnia przebieg sinusoidalny rys.4a o okresie równym szerokości okna wejściowego próbkowany równomiernie w taki sposób, aby błędy próbkowania były pomijalnie małe. Precyzję obliczeń algorytmu rzeczywistego ustalono na pojedynczą. Eksperyment przeprowadzono 100000 razy otrzymując rozkład błędów na wyjściu algorytmu jak na rys. 4b.



Rys.4. Sposób wyznaczania błędu zaokrąglenia algorytmu DFT, a) przebieg wejściowy, b) rozkład błędu na wyjściu algorytmu

#### 4. UWAGI KOŃCOWE

Zaproponowany model błędów własnych pozwala wyodrębnić pojedyncze źródła błędów, czyli określić rozkłady błędów dla każdego błędu cząstkowego. Znając funkcje gęstości dla poszczególnych błędów oraz zależności między nimi można stosując np. aparat matematyczny zwany redukcijną arytmetyką interwałową [3] wyznaczyć niepewność na wyjściu algorytmu. Eksperymenty symulacyjne przeprowadzono dla algorytmu jednopunktowego reprezentującego pierwszą harmoniczną jednak z badań przeprowadzonych przez autora wynika, że rezultaty można w prosty sposób przełożyć na kolejne harmoniczne.

#### 5. LITERATURA

1. Jakubiec J., Roj J.: *Pomiarowe przetwarzanie próbkujące*. Wydawnictwo Politechniki Śląskiej, Gliwice 2000
2. 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
3. Jakubiec J.: *Redukcyjna arytmetyka interwałowa w zastosowaniu do wyznaczania niepewności algorytmów przetwarzania danych pomiarowych*. Monografia. Wydawnictwo Politechniki Śląskiej, Gliwice 2002
4. Topór -Kamiński T.: *Definiowanie Niepewności Cząstkowych Algorytmów Przetwarzania Ciągów Danych Pomiarowych*. VIII Sympozjum Modelowanie i Symulacja Systemów Pomiarowych 21-25 września 1998r., Krynica Górská
5. *IEEE Standard 754 for Binary Floating-Point arithmetic*.

Artykuł recenzowany.