

*Piotr WARDA*  
*Politechnika Lubelska*  
*Wydział Elektrotechniki i Informatyki*  
*Katedra Automatyki i Metrologii*

## **METODA WAGOWEGO DOBORU PRZESUNIĘCIA FAZOWEGO DO SYNCHRONIZACJI DANYCH POMIAROWYCH**

Przesunięcie fazowe pomiędzy przebiegami jest powodowane przez różne czynniki. Może ono być zarówno informacją przydatną jak i zbędną. Wartość przesunięcia pomiędzy przebiegami jest jednym z czynników, który ma istotny wpływ na możliwość prawidłowego porównania wartości chwilowych przebiegów. Praca prezentuje ideę i implementację algorytmu wagowej synchronizacji danych reprezentujących przebiegi testowy i odtworzony. Zsynchronizowane przebiegi dają możliwość porównania wartości chwilowych, oceny różnic i wyznaczenia eksperymentalnego błędu przetwarzania informacji w systemie. Ze względu na oferowane możliwości przetwarzania danych do realizacji przedstawionej metody wykorzystano środowisko programistyczne LabVIEW. Przedstawiono algorytm pracy aplikacji, zamieszczono przykładowe wyniki.

## **METHOD OF WEIGHTED SELECTION OF THE PHASE SHIFT FOR SYNCHRONIZATION OF MEASUREMENT DATA**

The phase shift between waveforms is caused by various reasons. It can be both useful and redundant information. The phase shift value between the signals is one of the reasons that has a significant effect on the correct comparison of the instantaneous signals values. The article presents the idea and implementation of an algorithm for weighted synchronization of data representing test and measured signal. Synchronized waveforms give the opportunity to compare instantaneous values, evaluate differences and determine an experimental error in information processing in the system. Due to the offered data processing capabilities, the LabVIEW programming environment was used to implement the presented method. The LabVIEW environment was used for implementation. The application algorithm and examples of work results are presented.

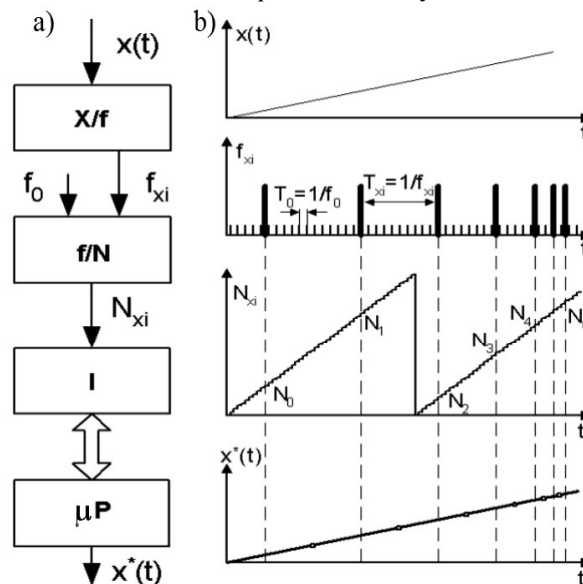
### **1. WSTĘP**

Porównanie przebiegu testowego i odtworzonego z danych przekazanych przez system pomiarowy to zagadnienie często spotykane w analizie różnych zjawisk i obiektów fizycznych [1]. Niniejsza praca prezentuje przykład

realizacji algorytmu wagowej synchronizacji przebiegów, wykorzystany podczas wyznaczania krzywej błędu przetwarzania w torze pomiarowym z częstotliwościowym nośnikiem informacji [2, 3], zawierającym przetwornik „częstotliwość-kod” [4]. Przebieg testowy można z zadowalającą precyzją wygenerować wykorzystując generator arbitralny o dużej dokładności [5]. Utrudnieniem jest uzyskiwanie danych w nierównomiernych odstępach czasu [6], co dodatkowo komplikuje porównanie wyników odtworzenia z sygnałem testowym.

## 2. WPŁYW PRZESUNIĘCIA FAZOWEGO NA BŁĄD PRZETWARZANIA

Struktura toru pomiarowego z częstotliwościowym nośnikiem informacji [7] w podstawowym zestawieniu została pokazana na rysunku 1.



Rys. 1. Tor pomiarowy, a - schemat, b - przebiegi w wybranych punktach toru  
Fig. 1. Measurement channel, a - diagram, b - waveforms at selected points in the channel

Układ zawiera przetwornik „wielkość fizyczna - częstotliwość” (X/f) [8, 9], przetwornik „częstotliwość-kod” (f/N) połączony wybranym interfejsem (I) z systemem mikroprocesorowym ( $\mu P$ ) pozwalającym na analizę danych,

odtworzenie wejściowej wielkości fizycznej  $x(t)$ , wizualizację przebiegów i archiwizację wyników.

Przetwarzanie wielkości fizycznej na częstotliwość zazwyczaj odbywa się w układzie całkującym, który po osiągnięciu wyznaczonej wartości całki generuje impuls wyjściowy [4]. Im większa amplituda wejściowego przebiegu przetwornika X/f tym częściej generowane są impulsy na wyjściu przetwornika.

Przetwornik „częstotliwość - kod” realizuje cyfrową metodę pomiaru okresu [10]. Liczy, występujące z częstotliwością  $f_0$  (rys. 1), impulsy generatora zegarowego pomiędzy impulsami granicznymi przedziału międzyimpulsowego  $T_{xi}$ . Liczba  $N_{xi}$ , będąca różnicą wartości stanów licznika przetwornika f/N na początku i końcu przedziału  $T_{xi}$ , reprezentuje długość mierzonego przedziału. Wartości  $N_{xi}$  są przekazywane do nadrzędnego systemu mikroprocesorowego mP, który wylicza wartości chwilowe wejściowej wielkości  $x(t)$ .

Jednym z zagadnień, które należy rozwiązać podczas badania prezentowanego toru jest ocena rzeczywistego błędu przetwarzania  $\delta_p$ , wprowadzanego przez przetwornik „częstotliwość - kod” [3]:

$$\delta_p = \left( \frac{T_g}{T_{xi}} + \left( 1 - \frac{\sin \pi T_{xi} F}{\pi T_{xi} F} \right) \frac{1}{\frac{f_0}{f_m} + 1} \right) 100\%, \quad (1)$$

gdzie:  $T_g$  - okres generatora zegarowego przetwornika f/N,  $F$  - częstotliwość sygnału wielkości fizycznej,  $f_0$  - składowa stała sygnału częstotliwościowego,  $f_m$  - amplituda sygnału częstotliwościowego.

Zależność na błąd została wyznaczona po przyjęciu sinusoidalnego przebiegu testowego opisanego wzorem:

$$x(t) = X_0 + X_m \sin(\omega t), \quad (2)$$

gdzie:  $X_0$  - składowa stała,  $X_m$  - amplituda,  $\omega$  - pulsacja przebiegu. Założono, że wielkość  $x(t)$  nie będzie przyjmować wartości ujemnych, tzn. będzie spełniona relacja  $X_0 \geq X_m$ , przy czym amplituda i składowa stała będą wartościami większymi od zera. Podczas wyznaczania zależności (1) pominięto błędy o znikomych wartościach (błędy bramkowania i generatora zegarowego). Przyjęto idealne przetwarzanie w przetworniku X/f oraz przetwarzanie danych w trybie *offline*, co skutkuje analizą błędu z pominięciem przesunięcia fazowego. Wyrażenie (1) zawiera dwa, mające największy i istotny wpływ na finalną

wartość błędu  $\delta_p$  składniki. Są to: błąd kwantowania wpływający na wartość  $\delta_p$  przy wysokich częstotliwościach i błąd uśredniania zwiększający wartość  $\delta_p$  dla częstotliwości niskich [10].

Dane do eksperymentalnego wyznaczenia krzywej błędu przetwarzania zebrano w torze pomiarowym podobnym do przedstawionego na rysunku 1. Jediną różnicą było zastąpienie przetwornika X/f generatorem programowalnym wytwarzającym sygnał o zmiennej częstotliwości [11]. Przed eksperymentem ustalono, że generator ma dostateczną stabilność wytwarzanego przebiegu [5]. W konsekwencji pojawiła się możliwość obliczania wartości przebiegu testowego z formuły matematycznej. Przyjmując, że zależność pomiędzy wielkością wejściową  $x(t)$  a częstotliwością  $f_x(t)$  sygnału wyjściowego przetwornika X/f opisuje wzór:

$$f_x(t) = Sx(t), \quad (3)$$

gdzie  $S$  to czułość przetwornika X/f, częstotliwość sygnału wyjściowego przetwornika X/f jest generowana według zależności:

$$f_x(t) = f_0 + f_m \sin(2\pi Ft), \quad (4)$$

gdzie:  $f_0$  - składowa stała przebiegu,  $f_m$  - amplituda przebiegu,  $2\pi F$  - pulsacja przebiegu. Uwzględniając przyjęte dla przebiegu testowego (2) założenia przyjmuje się, że przetwornik X/f może wytwarzać częstotliwość zawierającą się w zakresie od zera do sumy składowej stałej i amplitudy sygnału (4). System mikroprocesorowy  $\mu P$  wyznacza częstotliwość sygnału wejściowego przetwornika f/N jako:

$$f_x^*(t) = \frac{1}{T_{xi}} = \frac{f_0}{N_i + N_{i_{max}} P - N_{i-1}}, \quad (5)$$

gdzie:  $N_i$  oraz  $N_{i-1}$  - stany licznika przetwornika "częstotliwość - kod" w chwilach czasowych odpowiednio  $t_i$  oraz  $t_{i-1}$ ,  $N_{max}$  - stan maksymalny licznika,

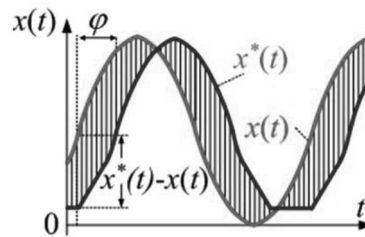
$P$  - liczba przepelnień licznika w czasie pomiaru każdego z przedziałów międzyimpulsowych  $T_{xi}$ . W idealnym przypadku odtworzone wartości chwilowe przebiegu częstotliwościowego będzie opisywała zależność:

$$f_x^*(t) = f_0 + f_m \sin(2\pi Ft + \varphi). \quad (6)$$

Po uwzględnieniu zależności (3) i przeprowadzeniu obliczeń otrzymuje się z systemu  $\mu P$  odtworzone wartości chwilowe przebiegu wejściowego przetwornika X/f, które można opisać zależnością:

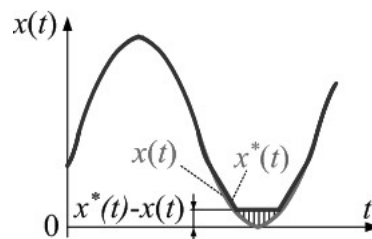
$$x^*(t) = X_0 + X_m \sin(\omega t + \varphi). \quad (7)$$

Niestety przebieg  $x^*(t)$  po odtworzeniu będzie opóźniony względem przebiegu zadanego  $x(t)$  na wejściu przetwornika X/f [11]. Relacja pomiędzy przebiegami została w przybliżony sposób pokazana na rysunku 2.



Rys. 2. Przykładowa relacja pomiędzy przebiegiem wejściowym  $x(t)$  i odtworzonym  $x^*(t)$   
Fig. 2. Example relation between of the input signal  $x(t)$  and the reconstructed signal  $x^*(t)$

Przebieg zadany oznaczono kolorem czerwonym, a odtworzony - niebieskim. Oczywiste jest, że przeważający udział w wytworzeniu zaprezentowanego błędu ma przesunięcie pomiędzy przebiegami. Zniekształcenie przebiegu odtworzonego nie jest przypadkowe. Aplikacja do synchronizacji danych z eksperymentu jest wykorzystywana do badania błędu przetwarzania w torze pomiarowym z częstotliwościowym nośnikiem informacji. Odkształcenie widoczne w otoczeniu minimum sinusoidy prezentuje wpływ występującego w opisywanym torze błędu uśredniania. Powierzchnię pomiędzy przebiegami zaznaczono czarnymi liniami w celu zobrazowania różnicy pomiędzy wartościami chwilowymi przebiegów dla kolejnych chwil czasowych.



Rys. 3. Prezentacja przebiegów  $x(t)$  i  $x^*(t)$  po synchronizacji  
Fig. 3. Presentation of  $x(t)$  and  $x^*(t)$  signals after synchronization

Na rysunku 3 pokazano, jak wygląda relacja pomiędzy przebiegami po eliminacji przesunięcia fazowego. Widać, że różnice są zdecydowanie mniejsze i reprezentują błąd wprowadzany podczas przetwarzania w systemie pomiarowym. Pozostał tylko błąd widoczny w otoczeniu minimum sinusoidy. Łatwo wysnuć wniosek, że pominięcie przesunięcia fazowego pozwala na prezentację zbliżonej do rzeczywistej wartości błędu przetwarzania wprowadzanego przez badany układ pomiarowy. Stąd w rozważaniach, w których nie jest istotne przesunięcie fazowe, w dokładnej analizie błędu przetwarzania należy przesunięcie zminimalizować.

### 3. DOBÓR WSPÓŁCZYNNIKÓW KOREKCYJNYCH FAZY

Przebieg testowy można dość precyzyjnie wygenerować z użyciem programowalnego generatora o zadowalających i znanych parametrach [5]. Znajomość parametrów zadanego przebiegu umożliwia wyliczenie wartości chwilowej częstotliwości przebiegu wejściowego przetwornika "częstotliwość-kod" dla dowolnej chwili czasowej. Wystarczy wykorzystać do tego celu przekazane

do generatora programowalnego parametry i zależność matematyczną opisującą zadany przebieg testowy (4). Do rozwiązania pozostaje zagadnienie, jak przeprowadzić eliminację przesunięcia fazowego pomiędzy przebiegiem wytworzonym przez generator sygnału testowego a przebiegiem odtworzonym z zestawu nieregularnie w czasie pobranych wartości liczbowych reprezentujących kolejne czasy międzyimpulsowe  $T_{xi}$ ?

Jednym z rozwiązań może być krokowa zmiana przesunięcia fazowego z ciągłą oceną błędu przetwarzania [11]. Biorąc pod uwagę, że pomiar każdego czasu międzyimpulsowego  $T_{xi}$  jest niezależny od poprzedzającego uznano, że do oceny dokładności przetwarzania zostanie zastosowany błąd bezwzględny, będący różnicą pomiędzy wartościami przebiegu odtworzonego i przebiegu testowego dla chwil czasowych, w których zostały umieszczone (rys.1) odtworzone wartości przebiegu  $x^*(t)$ :

$$\Delta_{\Sigma}(t) = x^*(t) - x(t). \quad (8)$$

Wartość błędu  $\Delta_{\Sigma}$  (rys.2) jest zależna głównie od błędu kwantowania, uśredniania i od przesunięcia fazowego pomiędzy przebiegami. Błąd wynikający z przesunięcia fazowego, przyjmując odtwarzanie przebiegu

w trybie *offline*, można uznać za czynnik dodatkowy. Należy wobec powyższego zminimalizować przesunięcie fazowe. Przyjęto założenie, że wyszukanie minimum błędu (8) pozwoli na uzyskanie wartości reprezentującej głównie błędy kwantowania i uśredniania. Na rysunku 2 można zauważyć, że wyszukanie pojedynczego minimum błędu  $\Delta_{\Sigma}$  nie gwarantuje jednak uzyskania minimum przesunięcia fazowego. Istotne jest, aby błąd przetwarzania (8) miał jak najmniejszą wartość w całym przedziale czasowym reprezentowanym przez zebrane dane pomiarowe. W konsekwencji przyjęto, że wartości błędu  $\Delta_{\Sigma}$  będą wyliczane dla wszystkich wartości liczbowych zebranych podczas pomiaru. Wyniki wraz z przesunięciami fazowymi wykorzystanymi w danym cyklu analizy będą zestawiane w tablicy. Po zakończeniu obliczeń, tablica będzie przeszukiwana pod kątem odnalezienia przesunięcia fazowego wywołującego minimalną wartość różnicy:

$$\min(\Delta_{\Sigma}(t)) = \max(f_{xi}(t) - f_x(t)) - \min(f_{xi}(t) - f_x(t)), \quad (9)$$

czyli wyszukiwane będzie minimum różnicy maksimum i minimum, różnic pomiędzy wartościami chwilowymi przebiegu odtworzonego i zadanego. W ten sposób uzyska się, w przybliżeniu, relację pomiędzy przebiegami pokazaną na rysunku 3. Zaprezentowane kryterium wyszukiwania minimum zastosowano w praktyce [11]. W efekcie, po minimalizacji przesunięcia fazowego, uzyskano krzywą błędów z eksperymentu odpowiadającą teoretycznej krzywej błędów dla badanego toru pomiarowego [3]. Autor ma świadomość, że przedstawione kryterium nie jest typowe. Zostało przyjęte wyłącznie z powodu skuteczności realizacji procesu minimalizacji przesunięcia fazowego, pozwalającego na uzyskanie błędu z eksperymentu o przebiegu zgodnym z rozważaniami teoretycznymi. W dalszych pracach nad algorytmem planowane jest, w celu porównania z zaproponowanym kryterium (9), zastosowanie jako kryterium jakości dopasowania minimum błędu średniokwadratowego.

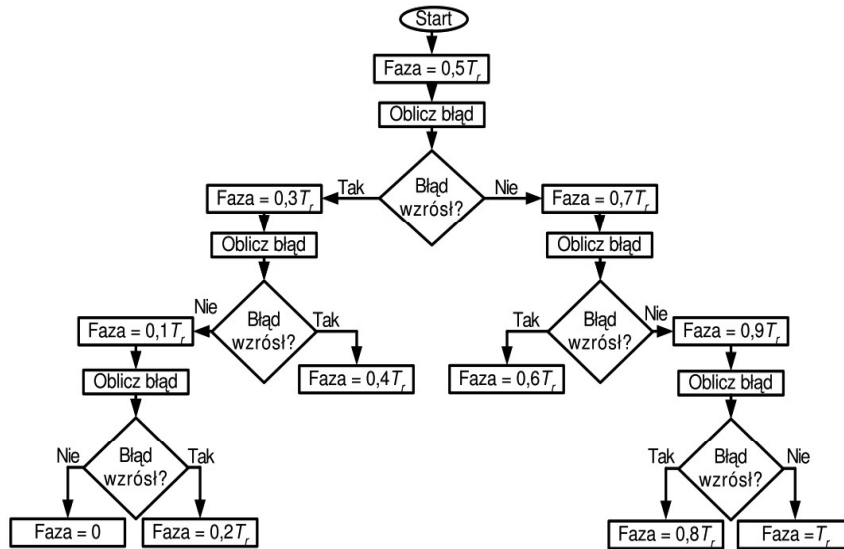
Pewnym problemem z praktycznego punktu widzenia jest szybkość porównania przebiegu zadanego z odtworzonym. Można dobierać fazę zmieniając kolejne cyfry w liczbie reprezentującej przesunięcie fazowe [11]. Nie jest to rozwiązanie optymalne. Wymaga od komputera dość czasochłonnych obliczeń. Z praktycznych prób wynika, że do oceny błędu przetwarzania reprezentacja fazy powinna mieć minimum 5-6 cyfr znaczących. Przy niewielkiej liczbie wartości liczbowych reprezentujących zmieniającą się częstotliwość przebiegu wejściowego przetwornika "częstotliwość-kod" synchronizacja jest realizowana stosunkowo szybko. O wiele większym

problemem jest eliminacja przesunięcia w przypadku przebiegu reprezentowanego przez dużą liczbę wartości  $N_{xi}$ . Dla przykładu, badanie dokładności przetwarzania przebiegu opisanego zestawem miliona wartości liczbowych trwało około 12 godzin przy zastosowaniu komputera z procesorem Intel i5-6300U taktowanego zegarem 2,4 GHz i 8 GB RAM.

Praca algorytmu wagowego doboru współczynników korygujących przesunięcie fazowe jest, co do zasady, bardzo zbliżona do pracy przetwornika analogowo-cyfrowego realizującego metodę kompensacyjną. Przy przetwarzaniu napięcia, po cyklu przetwarzania, przybliżony wynik reprezentuje iloczyn liczby w kodzie binarnym i napięcia odniesienia. Przy doborze przesunięcia fazowego natomiast należy zwrócić uwagę na problem przedziału czasu w którym były zbierane dane. Może nim być pełny okres, jego część czy wiele okresów sygnału wejściowego przetwornika  $X/f$ . Wprowadzenie okresu jako odniesienia, w przypadku danych reprezentujących tylko część okresu, mogłoby wydłużyć proces minimalizacji przesunięcia fazowego. Ze względu na nieznaną na początku zakres zmiany przesunięcia fazowego, który dodatkowo zmienia się w trakcie realizacji procesu minimalizacji, za dogodniejsze uznano zastosowanie wagowego podziału aktualnie rozpatrywanego przedziału czasu. W wyniku, dla danego etapu obliczeń, otrzymuje się wówczas iloczyn współczynnika wagowego i długości rozpatrywanego przedziału czasu. Oczywiście można zmodyfikować algorytm minimalizacji fazy tak, aby pracował w sposób bardziej zbliżony do pracy wspomnianego przetwornika analogowo-cyfrowego. Należałoby wówczas przyjąć okres jako czas odniesienia i zmodyfikować metodę przetwarzania danych.

Przy zmianie każdej pozycji dziesiętnej z krokiem 1 należy dla każdego miejsca dziesiętnego wykonać dziesięć przetworzeń całego zestawu danych. Dla sześciu cyfr, w skrajnym przypadku, będzie wymagane przeliczenie sześćdziesięciokrotne.





Rys. 4. Schemat doboru współczynników korekcyjnych metodą wagową  
 Fig. 4. Diagram of the selection of correction coefficients using the weight method

W celu przyspieszenia procesu synchronizacji powstała autorska idea nowej, wagowej zmiany przesunięcia. Dobór współczynników odbywa się wówczas poprzez zestawianie liczb reprezentujących dane miejsce dziesiętne podobnie jak odważniki na wadze, w zależności od uzyskanego efektu porównania następuje skokowa zmiana do kolejnej wartości przesunięcia fazowego. Kolejność zmian dla pojedynczego miejsca dziesiętnego w liczbie reprezentującej przesunięcie fazowe i możliwe warianty doboru liczb pokazuje algorytm na rysunku 4. W celu porządkowym długość trwania analizowanego przedziału czasu oznaczono symbolem  $T_r$ . Współczynniki korekcyjne dobierane są w zależności od wyliczonych wartości minimum  $\Delta_\Sigma$  (9). Jak widać każde miejsce dziesiętne będzie analizowane w maksymalnie pięciu krokach, zamiast dziesięciu przy analizie z krokiem 1.

Po przeprowadzeniu analizy wyników dla danej serii ponownie wyznacza się wartość minimalną  $\Delta_\Sigma$  (9), dobiera czas  $T_r$ , zachowuje fazę i błąd minimalny do kolejnego cyklu obliczeń. Liczba cykli pracy algorytmu zależy od sposobu określenia kryterium kończącego pracę aplikacji, może to być długość czasu  $T_r$  lub zadowalającej wartości minimum  $\Delta_\Sigma$  (9).

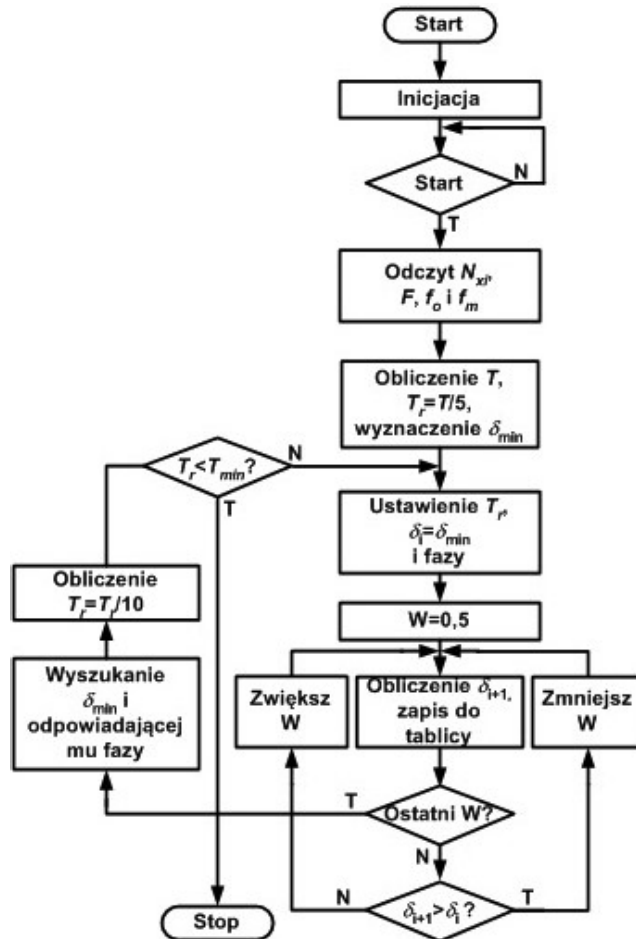
#### 4. ALGORYTM PRACY APLIKACJI

Algorytm pracy programu realizującego wagowy dobór współczynników pokazano na rysunku 5. Praca aplikacji jest realizowana w trzech etapach.

Wstępnie dokonywana jest inicjacja aplikacji i ustawiane wartości początkowych zmiennych. Program oczekuje na wprowadzenie danych niezbędnych do rozpoczęcia procesu synchronizacji. Po wprowadzeniu danych i uruchomieniu następuje odczyt wskazanego pliku z wartościami liczbowymi  $N_{xi}$  reprezentującymi kolejne czasy między impulsowe  $T_{xi}$ . Dodatkowo z nazwy pliku odczytywane są również parametry sygnału testowego: amplituda, częstotliwość i okres sygnału częstotliwościowego.

W celu przyspieszenia osiągnięcia rezultatu, minimalizacja przesunięcia fazowego rozpoczyna się od wstępnego określenia przedziału czasu, w którym należy szukać możliwości minimalizacji przesunięcia fazowego pomiędzy przebiegami. Inicjacyjnie oblicza się wartości minimum  $\Delta_{\Sigma}$  co 1/5 okresu przebiegu (6) i wyszukuje wartość minimalną z serii  $\Delta_{\Sigma}$ . Długość przedziału czasu w którym wystąpiło wyszukanie minimum przyjmuje się jako startowy do realizacji procesu dalszej minimalizacji błędu. Analogicznie przesunięcie fazowe, dla którego wykryto minimum  $\Delta_{\Sigma}$  (9) jest przyjmowane jako wartość odniesienia w dalszej pracy algorytmu eliminującego przesunięcie fazowe.

Kolejny etap to wagowy dobór zmiany przesunięcia fazowego dla aktualnie ustawionego przedziału czasu  $T_r$  według podanego wcześniej algorytmu zmiany współczynników (rys.4).



Rys. 5. Algorytm pracy aplikacji minimalizujący przesunięcie fazowe między sygnałami  
 Fig. 5. Application work algorithm that eliminates phase shift between signals

Dobór rozpoczyna się od ustawienia współczynnika korekcyjnego  $W$  na wartość 0,5 po czym następuje obliczenie błędu (8), zapis wartości do tablicy i kontrola, czy jest to analiza błędu dla ostatniego współczynnika korekcyjnego  $W$  dla danej serii czy należy kontynuować obliczenia. W razie potrzeby doboru kolejnego współczynnika korekcyjnego, następuje kontrola relacji między błędami uzyskanymi w dwu ostatnich cyklach pracy programu i stosownie do wyniku porównania dobór współczynnika o mniejszej lub większej wartości wagowej. Jeżeli był to, według rysunku 4, ostatni współczynnik w serii, program przechodzi do analizy zebranych danych. W ostatniej fazie przetwarzania jest wyszukiwana minimalna wartość  $\Delta_{\Sigma}$  (9) w pozyskanej serii

danych dla ustawionego przesunięcia fazowego. Po wyszukaniu minimum następuje zapamiętanie wyszukanej, minimalnej wartości różnicy  $\Delta_\Sigma$  jako wartości odniesienia do uruchomienia obliczeń w kolejnej serii, ustawienie fazy dla minimum jako startowej w kolejnym cyklu obliczeń, wartość przedziału  $T_r$  zostaje zmieniona na 0,1 poprzedniej wartości  $T_r$ . Powodem takiego doboru długości nowego przedziału czasu jest fakt, że współczynniki wagowe dzielą przedział czasu zawsze na dziesięć podprzedziałów, choć wyliczenia są przeprowadzane wyłącznie w obszarze malejącej wartości  $\Delta_\Sigma$ .

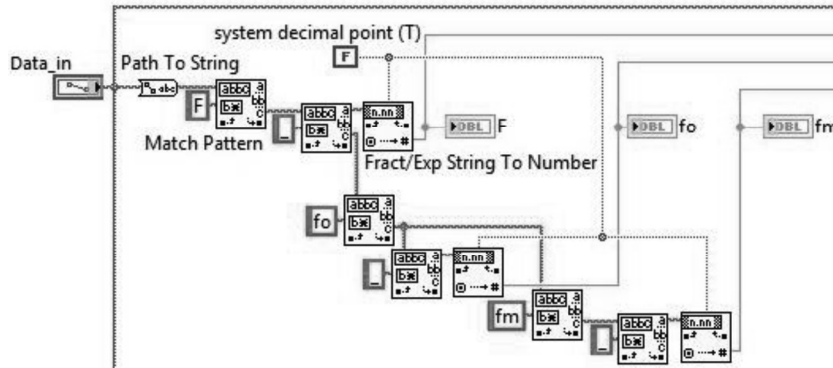
W przedstawionym algorytmie jako kryterium podtrzymania pracy aplikacji, wybrano osiągnięcie zadanego przez użytkownika minimalnego czasu  $T_r$ , dla którego jeszcze powinna być przeprowadzana minimalizacja fazy.

## 5. APLIKACJA

Ze względu na szerokie możliwości tworzenia zaawansowanego oprogramowania, kontroli pracy aplikacji i prezentacji wyników, do realizacji programu wybrano środowisko programistyczne LabVIEW. Biorąc pod uwagę sposób przetwarzania danych za właściwe uznano zastosowanie szablonu maszyny stanów. Jej zadanie to realizacja kolejnych działań, odczyt danych pomiarowych, iteracyjna redukcja fazy i prezentacja wyników.

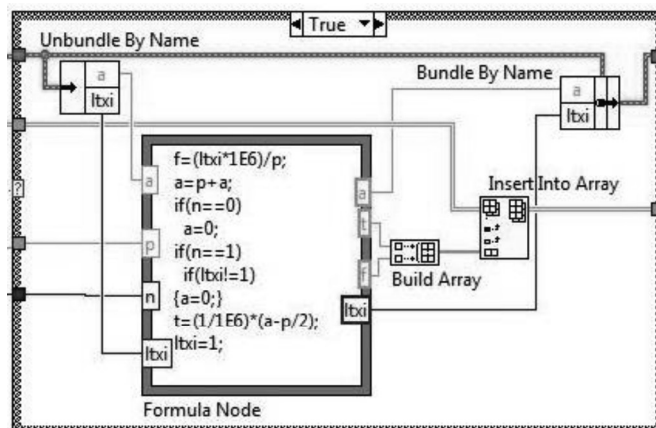
Rozpoczęcie pracy programu uruchamia procedurę odczytu parametrów sygnału testowego. Ze względów praktycznych przyjęto, że użyteczne będzie tworzenie nazwy pliku z danymi z użyciem parametrów zadawanego sygnału testowego. Zawarto w niej kolejno częstotliwość wielkości fizycznej  $x(t)$  o wartości 0,5 Hz, składową stałą zmiennej częstotliwości sygnału wejściowego przetwornika "częstotliwość-kod" o wartości 5010 Hz, oraz amplitudę zmiennej częstotliwości o wartości 4990 Hz.

Na rysunku 6 pokazano fragment diagramu realizujący odczyt parametrów sygnału testowego z nazwy pliku z danymi. Po odczycie ścieżki dostępu do pliku obiektem „Data\_in” następuje konwersja w ciąg tekstowy obiektem „Path To String”. Ciąg jest przeszukiwany obiektami „Match Pattern” w celu odnalezienia symboli - znaczników poprzedzających parametry sygnału testowego: częstotliwość wielkości  $x(t)$ , składową stałą i amplitudę przebiegu testowego, są to odpowiednio " F ", " $f_0$ " i "  $f_m$  ". Tekst po wymienionych znacznikach jest przetwarzany obiektami „Fract/Exp String To Number” w wartości liczbowe i zapisywany w odpowiadających znacznikom wskaźnikach.



Rys. 6. Fragment diagramu aplikacji odczytujący parametry sygnału testowego z nazwy pliku  
 Fig. 6. A fragment of the application diagram that reads the test signal parameters from the file name

Na rysunku 7 pokazano fragment diagramu realizujący wyliczenie kolejnych wartości częstotliwości odpowiadających wartościom liczbowym odczytanym z przetwornika  $f/N$ .

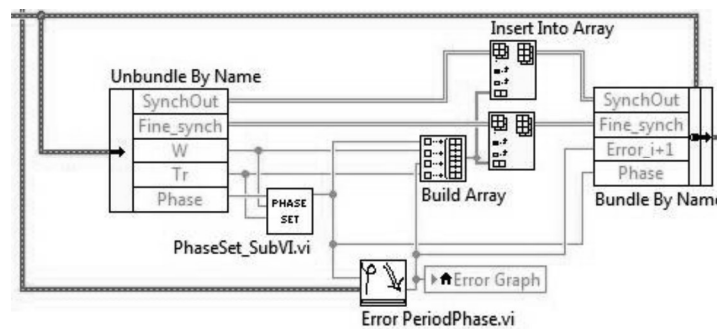


Rys.7. Fragment diagramu wyliczający częstotliwość sygnału wyjściowego z przetwornika  $X/f$   
 Fig. 7. Fragment of the block diagram calculating the frequency of the output signal from the  $X/f$  converter

W każdej iteracji odczytywane są z klastra danych obiektem „Unbundle By Name” dane do obliczeń w oknie „Formula Node”. Wyniki zestawiane są obiektem „Build Array” w dwuelementowy wektor, który następnie obiektem „Insert Into Array” jest umieszczony w tablicy zawierającej obliczone częstotliwości sygnału wejściowego przetwornika  $f/N$  wraz z chwilami czasowymi wystąpienia. Dodatkowo dane, do obliczeń w kolejnej iteracji, są przekazywane do klastra danych obiektem „Bundle By Name”. Zestawienie

wartości częstotliwości sygnału częstotliwościowego jest czynnością wstępną przed porównaniem ich z wartościami wyliczonymi z parametrów zadanych generatorowi sygnału testowego.

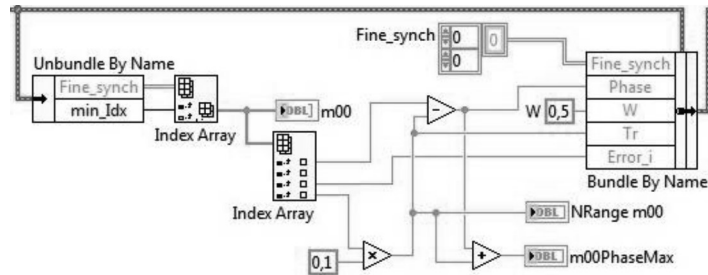
Kolejny fragment diagramu (rys. 8) pokazuje wyliczenie fazy podprogramem "PhaseSet\_SubVI.vi" z danych odczytanych z klastra danych „Unbundle By Name”. Po obliczeniach następuje przekazanie wyniku do podprogramu „Error PeriodPhase.vi” obliczającego minimum  $\Delta_{\Sigma}$  dla danego współczynnika wagowego. Dane wejściowe i wyniki powyższych obliczeń są zestawiane obiektem „Build Array” i zapisywane obiektami „Insert Into Array” i „Insert Into Array” do dwu tablic "SynchOut" i "Fine\_synch", celem odpowiednio archiwizacji wyników i dalszego przetwarzania fazy w kolejnych etapach pracy aplikacji. Finalnie zmodyfikowane tablice, faza i aktualne minimum błędu zapisywane są obiektem „Bundle By Name” w klastrze danych.



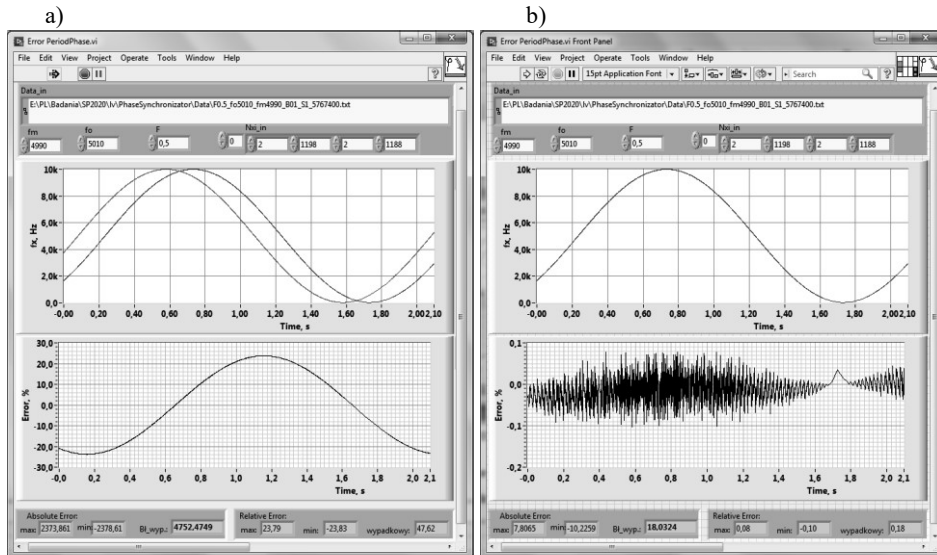
Rys. 8. Obliczenie błędu dla wybranej wartości przesunięcia fazowego  
Fig. 8. Calculating of the error for the selected phase shift value

Następny fragment diagramu (rys.9) pokazuje dobór parametrów dla kolejnego cyklu obliczeń. Wstępnie z klastra danych odczytywana jest obiektem „Unbundle By Name” tablica zawierająca zestawienie wyników oraz indeks wiersza tablicy, na którym znajduje się minimalna wartość błędów  $\Delta_{\Sigma}$ . Pierwszy obiekt „Index Array” pozwala na wydzielenie z tablicy wiersza z minimalną wartością  $\Delta_{\Sigma}$ . Drugi obiekt „Index Array” umożliwia odczyt zawartości poszczególnych komórek wiersza. Są to: wartość współczynnika korekcyjnego  $W$  powodująca najmniejszą różnicę  $\Delta_{\Sigma}$ , odpowiadająca mu faza, która staje się inicjującą dla kolejnego przedziału czasu, oraz bieżący czas  $T_r$ . Dla kolejnego cyklu obliczeń czas  $T_r$  jest obliczany jako jedna dziesiąta poprzedniego. Obiekt „Bundle By Name” zestawia w klastrze danych wartości inicjujące dla kolejnego cyklu doboru współczynnika wagowego. Są to kolejno:

zainicjowana tablica „Fine\_synch”, faza, współczynnik wagowy  $W$ , czas  $T_r$  i wyszukana wartość minimum  $\Delta\Sigma$ .



Rys. 9. Tworzenie ustawień dla kolejnego cyklu minimalizacji przesunięcia fazowego  
Fig. 9. Creating settings for the next cycle of phase shift minimization

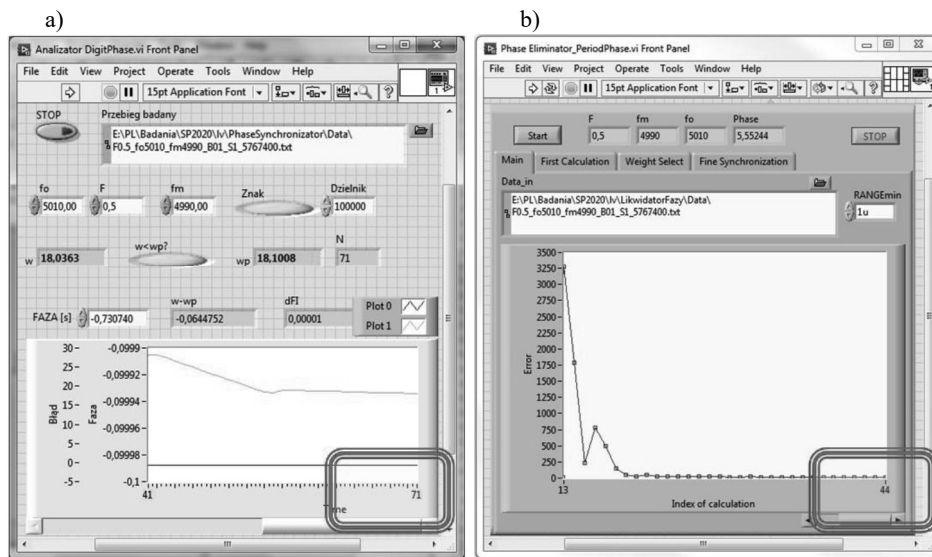


Rys. 10. Przykładowe wyniki pracy aplikacji, a - widok przebiegów w trakcie redukcji fazy,  
b - przebiegi po redukcji

Fig. 10. Sample results of application work, a - view of waveforms during phase reduction,  
b - waveforms after reduction

Rysunek 10 pokazuje fragment panelu aplikacji prezentujący wskaźniki graficzne przebiegów: zadanego i odtworzonego w części górnej i różnicę wartości chwilowych przebiegów w części dolnej. Kolorem czerwonym oznaczono przebieg wejściowy wytworzony w generatorze. Parametry przebiegu: amplituda 4990 Hz, składowa stała 5010 Hz i częstotliwość przebiegu  $x(t)$  0,5 Hz. Kolorem niebieskim oznaczono przebieg odtworzony.

Przebiegi na pozycji „a” są zauważalnie przesunięte względem siebie. Widać dużą wartość błędu przetwarzania. Na pozycji „b” pokazano efekt minimalizacji fazy oraz finalnie uzyskaną krzywą błędów. Jest na nim widoczny wpływ błędu kwantowania w pomiarze dużych wartości częstotliwości reprezentujących wielkość fizyczną  $x(t)$  i wpływ błędu uśredniania mierzonych częstotliwości dla małych wartości tej wielkości. Różnica (9) zmalała z wartości około 4752 Hz do około 18 Hz.



Rys. 11. Panele aplikacji realizujących synchronizację przebiegów różnymi metodami  
Fig. 11. Panels of applications that synchronize signals with different methods

Istotną informację niosą również panele główne aplikacji realizujących zadawanie korekty fazy z wykorzystaniem wymienionych metod doboru fazy (rys. 11). Zaprezentowano panele po przeprowadzeniu procesu minimalizacji przesunięcia fazowego dla tego samego zestawu danych. Panel z lewej strony (pozycja „a”) prezentuje pracę aplikacji zadającej fazę krokowo. Panel z prawej strony (pozycja „b”) prezentuje pracę aplikacji zadającej fazę z wykorzystaniem algorytmu wagowego. Można zauważyć, że do osiągnięcia porównywalnego efektu aplikacja z pozycji „a” potrzebowała 71 iteracji, natomiast aplikacja z pozycji „b” potrzebowała zrealizować tylko 44 iteracje. W związku z powyższym można uznać, że w zakresie przeprowadzonych badań, algorytm wagowy pozwolił na przeprowadzenie procesu synchronizacji danych pomiarowych szybciej niż przez zmianę liczby reprezentującej fazę metodą krokową.



## 6. PODSUMOWANIE

Opracowane oprogramowanie potwierdziło swoją przydatność. Prezentacja graficzna błędu przetwarzania w badanym układzie przy każdym, poddanym w analizie obliczeniowej, przesunięciu fazowym pozwala na bieżąco oceniać zaawansowanie procesu synchronizacji. Rozwiązanie jest niewątpliwie przydatne w pracach metrologicznych nad doskonaleniem systemów pomiarowych.

Program do synchronizacji danych może okazać się przydatny w przypadkach, gdy wymagana jest minimalizacja przesunięcia fazowego pomiędzy przebiegami uzyskanymi doświadczalnie i drogą symulacyjną.

Dokonano porównania wymaganej liczby iteracji, niezbędnej do eliminacji przesunięcia fazowego pomiędzy sinusoidalnym przebiegiem testowym o parametrach: częstotliwość  $F = 0,5$  Hz, amplituda  $f_m = 4990$  Hz i składowa stała  $f_0 = 5010$  Hz a danymi pomiarowymi uzyskanymi podczas przetwarzania tego przebiegu w systemie pomiarowym dla programów realizujących eliminację przesunięcia fazowego obydwoma omawianymi metodami.

Wagowa zmiana przesunięcia fazowego wymagała 44 iteracji. Praca z wykorzystaniem algorytmu krokowej zmiany poszczególnych cyfr fazy wymagała 71 iteracji doboru liczby reprezentującej fazę minimalizującą przesunięcie fazowe pomiędzy przebiegiem testowym i odtworzonym. Wobec powyższego uzasadnione jest stwierdzenie, że przedstawiony algorytm wagowego doboru fazy do synchronizacji danych jest efektywniejszy od synchronizacji poprzez dobór kolejnych miejsc dziesiętnych w liczbie reprezentującej fazę. Przyspieszenie pracy aplikacji po zastosowaniu algorytmu wagowej zmiany przesunięcia fazowego pozwala na efektywniejsze prowadzenie badań.

Omówiony algorytm można zastosować również do przebiegów testowych o innych kształtach, niezbędny jest wówczas wzór opisujący zadawany przebieg lub zestaw próbek reprezentujący przetwarzane w torze pomiarowym wartości chwilowe wielkości fizycznej  $x(t)$ .

## LITERATURA

1. Świsulski D., Pawłowski E., Dorozhovets M.: Digital Processing of Frequency-Pulse Signal in Measurement System, Analysis and Simulation of Electrical and Computer Systems, Springer, Cham, 2018.
2. Jakubiec J.: Błędy i niepewności danych w systemie pomiarowo-sterującym, Wydawnictwo Politechniki Śląskiej, Gliwice 2010.

3. Warda P.: Error of conversion of “physical quantity-to-frequency” converter output signal, Proceedings of SPIE, doi: 10.1117/12.2244496, 2016.
4. Shpak N.O., Kirianaki N.V., Yurish S.Y., and Deynega V.P.: Data acquisition and signal processing for smart sensors, Baffins Lane: John Wiley & Sons, Ltd, 2001.
5. Warda P.: Generator z modulacją częstotliwości jako źródło sygnału testującego dla toru pomiarowego z częstotliwościowym nośnikiem informacji, Przegląd Elektrotechniczny, 86 (2010), nr 8, s.235-238.
6. Pawłowski E.: Reconstruction of input signal of sensor with frequency output, Methods and Models in Automation and Robotics (MMAR), 2015, pp. 909-914.
7. Pawłowski E.: Spectrum analysis of measuring signals in sensors circuits with frequency output. Optoelectronic and Electronic Sensors IV, 06/13-06/16/00 Gliwice, Poland, Proceedings of SPIE, (2001) Vol. 4516, pp. 181-186.
8. Pawłowski E.: Design and evaluation of a flow-to-frequency converter circuit with thermal feedback, Measurement Science & Technology - nr 5, vol. 28, pp. 054004-054013, 2017.
9. Cheng-Ta Ch., Chi-Shen W. and Yu-Chung H.: A CMOS Integrated Capacitance-to-Frequency Converter with Digital Compensation Circuit Designed for Sensor Interface Applications, Proc. of IEEE Sensors 2007, pp. 954-957.
10. Warda P.: Adaptive correction of the quantisation error in the frequency-to-code converter, Methods and Models in Automation and Robotics, 2017, pp. 226-231, doi: 10.1109/MMAR.2017.8046829.
11. Warda P.: Wybrane problemy z oszacowaniem błędu przetwarzania w torze z częstotliwościowym nośnikiem informacji, Przegląd Elektrotechniczny, 89 (2013), nr 5, s. 272-274.