POZNAN UNIVERSITY OF TECHNOLOGY ACADEMIC JOURNALS
No 90 Electrical Engineering 2017

DOI 10.21008/j.1897-0737.2017.90.0014

Daria CIURKO*
Jacek HANDKE*

ALGORYTMY PRZESUNIĘCIA FAZOWEGO W ELEKTROENERGETYCZNEJ AUTOMATYCE ZABEZPIECZENIOWEJ – CZĘŚĆ II

W artykule omówiono i porównano algorytmy wykorzystywane do pomiaru przesunięcia fazowego pomiędzy dwoma sygnałami, pod kątem wykorzystania w elektroenergetycznych układach zabezpieczeniowych. Dokonano teoretycznej analizy algorytmów pomiarowych wykorzystujących przekształcenia: Fouriera, Hilberta oraz bazujących na metodach: detekcji przejścia sygnałów przez zero oraz demodulacji kąta fazowego. Przedstawiono szczegółowo sposób implementacji wybranych algorytmów w naukowym środowisku Scilab oraz przeanalizowano ich zachowanie przy różnych parametrach pracy dla zdefiniowanych testowych sygnałów wejściowych. Przeanalizowano otrzymane na drodze symulacji wyniki i porównano otrzymane dane. Część druga artykułu opisuje implementację wymienionych algorytmów.

SŁOWA KLUCZOWE: pomiar przesunięcia fazowego, elektroenergetyczna automatyka zabezpieczeniowa, SciLab

1. WPROWADZENIE

W części pierwszej artykułu przedstawiono opis matematyczny czterech wybranych algorytmów estymacji przesunięcia fazowego dwóch sygnałów, wykorzystujących: transformatę Fouriera, transformatę Hilberta, detekcję przejść wartości chwilowych sygnałów przez zero oraz demodulację kąta fazowego.

W drugiej części dokonano szczegółowego opisu implementacji wymienionych algorytmów. Porównano wpływ czynników takich jak: częstotliwość próbkowania, długość okna pomiarowego, występowanie wyższych harmonicznych, zakłóceń losowych oraz składowej nieokresowej w badanym sygnale. Na wstępie scharakteryzowano cztery klasy sygnałów wykorzystywanych podczas testów zaimplementowanych algorytmów.

* Politechnika Poznańska.

2. CHARAKTERYSTYKA SYGNAŁÓW TESTOWYCH

Na potrzeby badań w. w. algorytmów zdefiniowano 4 klasy dyskretnych sygnałów testowych o dobrze znanych właściwościach. Klasy te oznaczono literami A – D.

Sygnały klasy A posiadają takie same częstotliwości podstawowe f_c i nie zawierają wyższych harmonicznych. Ich przesunięcie fazowe można zdefiniować jako: $\varphi = \varphi_2 - \varphi_1$. Opisują je zależności:

$$x_1(n) = \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_1) \tag{1}$$

$$x_2(n) = \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_2) \tag{2}$$

Klasa B to sygnały testowe o takich samych częstotliwościach podstawowych, zawierające dodatkowo nieparzyste harmoniczne (trzecią i piątą). Wyrażone są wzorami:

$$x_1(n) = \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_1) + \boldsymbol{0{,}3} \; \boldsymbol{sin}(2\pi \; 3f_c \; n) + \boldsymbol{0{,}1} \; \boldsymbol{sin}(2\pi \; 5f_c \; n) \tag{3}$$

$$x_2(n) = \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_2) + \boldsymbol{0{,}3} \; \boldsymbol{sin}(2\pi \; 3f_c \; n) + \boldsymbol{0{,}1} \; \boldsymbol{sin}(2\pi \; 5f_c \; n) \tag{4}$$

Sygnały klasy C są opisane zależnościami zbliżonymi do (3) i (4), dodatkowo jednak zakłócone sygnałem $r(n)$ generowanym losowo .

$$\begin{aligned} x_1(n) = {} & \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_1) + \boldsymbol{0{,}3} \; \boldsymbol{sin}(2\pi \; 3f_c \; n) + \\ & + \boldsymbol{0{,}1} \; \boldsymbol{sin}(2\pi \; 5f_c \; n) + r_1(n) \end{aligned} \tag{5}$$

$$\begin{aligned} x_2(n) = {} & \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_2) + \boldsymbol{0{,}3} \; \boldsymbol{sin}(2\pi \; 3f_c \; n) + \\ & \boldsymbol{0{,}1} \; \boldsymbol{sin}(2\pi \; 5f_c \; n) + r_2(n) \end{aligned} \tag{6}$$

Sygnały oznaczone jako D, mają właściwości zbliżone do B, dodatkowo zakłócone są składową nieokresową $e^{-T_i n}$. Opisują je zależności:

$$x_1(n) = \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_1) + \boldsymbol{0{,}3} \; \boldsymbol{sin}(2\pi \; 3f_c \; n) + \boldsymbol{0{,}1} \; \boldsymbol{sin}(2\pi \; 5f_c \; n) + e^{-T_i n} \tag{7}$$

$$x_2(n) = \boldsymbol{sin}(2\pi \; f_c \; n + \varphi_2) + \boldsymbol{0{,}3} \; \boldsymbol{sin}(2\pi \; 3f_c \; n) + \boldsymbol{0{,}1} \; \boldsymbol{sin}(2\pi \; 5f_c \; n) + e^{-T_i n} \tag{8}$$

Za stałą zanikania składowej nieokresowej przyjęto $T_i = \boldsymbol{10}$.

3. IMPLEMENTACJA WYBRANYCH ALGORYTMÓW ESTYMACJI PRZESUNIĘCIA FAZOWEGO WYKORZYSTUJĄCEGO W ŚRODOWISKU SCILAB

Implementacji wszystkich algorytmów estymacji przesunięcia fazowego dokonano w środowisku Scilab w wersji 5.5.2. Przeprowadzono analizę błędów estymacji i porównano zauważone cechy algorytmu z opisem teoretycznym zawartym w pierwszym rozdziale artykułu.

Założono, że sygnały A, B, C oraz D są sygnałami o częstotliwości równej $f_c = 50$ Hz . W symulacjach tych przyjęto ponadto, że $\varphi_1 = 60°$, $\varphi_2 = 10°$.

3.1. Algorytm wykorzystujący szybką transformatę Fouriera

Poniżej przedstawiono część kodu źródłowego algorytmu pomiarowego, estymującego kąt przesunięcia fazowego przy użyciu szybkiej transformaty Fouriera.

```
fs=16000; //def. częstotliwości próbkowania
ts=1/fs; //obliczenie czasu próbkowania [Hz]
sTime=4; //def. okresu próbkowania [s]
f=50; //częstotliwość sygnału badanego

phase_deg_1 = 60; //def. przesunięcia fazowego syg. 1
phase_deg_2 = 0; //def. przesunięcia fazowego syg. 2

//definicje sygnałów testowych x1 oraz x2 (A)
phase_rad_1 = %pi*phase_deg_1/180;
phase_rad_2 = %pi*phase_deg_2/180;
n=0:(1/fs):sTime;
x1=sin(2*%pi*f*n+phase_rad_1);
x2=sin(2*%pi*f*n+phase_rad_2);

//transformata Fouriera badanych sygnałów
x1_fft=fft(x1);
x2_fft=fft(x2);

//wyznaczenie przesunięcia fazowego
x1_phase=atan(imag(x1_fft),real(x1_fft));
x2_phase=atan(imag(x2_fft),real(x2_fft));
x_phase_diff = x2_phase - x1_phase;

//uśrednienie wyników, konwersja na stopnie
D=((length(x1_fft))/fs)*f;
phase_diff = x_phase_diff(D);
WYNIK=(phase_diff*180)/%pi;
```

Wyniki działania algorytmu dla sygnałów z grup A, B, C oraz D przedstawiono w poniższej tabeli.

Tabela 1. Estymacja kąta przesunięcia fazowego przy wykorzystaniu transformaty Fouriera – zestawienie wyników symulacji

Lp.	f_s	sTime	f_{c1}	f_{c2}	Bezwzględny błąd estymacji			
					Sygnał A	Sygnał B	Sygnał C	Sygnał D
	Hz	s	Hz	Hz	°	°	°	°
1.	16000	1	50	50	0,30	0,10	6,60	34,5
2.	1200	10	50	50	0,04	0,01	0,13	0,64
3.	1200	1	50	50	0,31	0,12	2,75	5,96
4.	1200	0,8	50	50	0,39	0,15	5,19	7,35
5.	1200	0,4	50	50	0,78	0,31	7,81	13,22
6.	1200	0,08	50	50	3,46	1,95	14,21	27,86
7.	1200	0,04	50	50	4,39	3,32	8,36	33,26
8.	120	1	50	50	1,87	2,96	7,17	1,37
9.	70	1	50	50	2,65	0,48	1,82	5,51

Cechą charakterystyczną algorytmów estymacji kąta przesunięcia fazowego, wykorzystujących przekształcenie Fouriera jest transformowanie tych sygnałów podczas analizy z dziedziny czasu do dziedziny częstotliwości. Dzięki tej operacji algorytm wykazuje odporność na wyższe harmoniczne występujące w sygnale. Algorytm wykazuje wrażliwość na zakłócenia losowe w sygnale oraz składową nieokresową. Widać tendencję wzrostu wpływu tych zakłóceń przy malejącym czasie analizy sygnałów. W najgorszym przypadku błąd estymacji przy zakłóceniach losowych wyniósł ponad 14°, a dla składowej nieokresowej ponad 33°. Z punktu widzenia automatyki zabezpieczeniowej rozpatrywane są czasy rzędu kilkudziesięciu milisekund. Można zauważyć także, że na wyniki wpływ miała częstotliwość próbkowania sygnału. W przypadku analizy sygnałów A – B im wyższa była częstotliwość próbkowania, tym bardziej malała wartość błędu bezwzględnego estymacji kąta fazowego.

3.2. Algorytm wykorzystujący transformatę Hilberta

Dokonano analizy kodu źródłowego algorytmu estymującego kąt przesunięcia fazowego przy użyciu transformaty Hilberta. Przy analizie kodu programu pominięto część związaną z deklarowaniem sygnałów badanych x1 i x2 oraz prezentacją przebiegów i błędu pomiaru.

```
Hx1=hilbert(x1); //transformata Hilberta sygnału x1
Hx2=hilbert(x2); //transformata Hilberta sygnału x2
```

```
y1=x1+%i*Hx1;
y2=x2+%i*Hx2;

//obliczenie liczby sprzężonej do y1
ReY1=real(y1);
ImY1=imag(y1);
y1s=[real(y1)-%i*imag(y1)];

//iloczyn liczb zespolonych y1s oraz y2
M=y1s.*y2;

//wydzielenie części rzeczywistej i urojonej z M
ReM=real(M);
ImM=imag(M);

//wyznaczenie przesunięcia fazowego
WYNIK=atan(ImM,ReM);

//uśrednienie wyników, konwersja na stopnie
D=sTime/(1/f);
phase_diff=[((sum(WYNIK))/((fs/f)*D))*(180/%pi)];
```

Wyniki symulacji dla sygnałów A, B, C oraz D przedstawiono w tabeli 2.

Tabela 2. Estymacja kąta przesunięcia fazowego przy wykorzystaniu transformaty Hilberta – zestawienie wyników symulacji

Lp.	f_s	sTime	f_{c1}	f_{c2}	Bezwzględny błąd estymacji Δ			
					Sygnał A	Sygnał B	Sygnał C	Sygnał D
	Hz	s	Hz	Hz	°	°	°	°
1.	16000	1	50	50	$2{,}63\cdot10^{-3}$	14,12	13,44	12,11
2.	1200	10	50	50	$3{,}87\cdot10^{-3}$	14,11	13,51	13,84
3.	1200	1	50	50	$3{,}41\cdot10^{-3}$	14,18	13,62	12,23
4.	1200	0,8	50	50	0,04	14,21	13,36	11,71
5.	1200	0,4	50	50	0,09	14,22	13,46	9,56
6.	1200	0,08	50	50	0,42	14,51	12,82	9,81
7.	1200	0,04	50	50	0,84	14,92	16,25	21,72
8.	120	1	50	50	0,37	14,72	14,17	12,27
9.	70	1	50	50	101,13	93,53	93,41	87,93

Algorytm Hilberta cechuje, w przeciwieństwie do algorytmu korzystającego z transformacji Fouriera, wrażliwość na wszelkie zakłócenia występujące w sygnale. Podczas symulacji uwidocznił się znaczny wpływ wyższych harmonicznych na wartości otrzymywanych estymat kąta przesunięcia fazowego. Dla sygnałów zawierających jedynie podstawową harmoniczną, których częstotliwość próbkowania spełniała twierdzenie o próbkowaniu otrzymywano każdorazowo bardzo dokładne wyniki. Błąd bezwzględny nie przekraczał 1°. Dodanie do sygnału wyższych harmonicznych zwiększyło błąd każdorazowo o około 14°. Dodatkowe wprowadzenie do sygnałów badanych szumu oraz składowej nieokresowej przyczyniło się do zwiększenia błędu pomiaru tylko w przypadku krótkich czasów analizy.

3.3. Algorytm wykorzystujący metodę demodulacji kąta fazowego

Poniżej dokonano analizy kodu źródłowego algorytmu estymującego kąt przesunięcia fazowego przy wykorzystaniu algorytmicznego demodulatora kąta fazowego. Podobnie jak w punkcie 3.2, pominięto część kodu związaną z deklarowaniem sygnałów badanych x1 i x2 oraz prezentacją przebiegów i błędu pomiaru.

```
//wygenerowanie sygnałów pomocniczych
y11=x1.*sin(2*%pi*(f)*n);
y12=-x1.*cos(2*%pi*(f)*n);

y21=x2.*sin(2*%pi*(f)*n);
y22=-x2.*cos(2*%pi*(f)*n);

//Filtracja przebiegów w celu eliminacji składników o
częstotliwości 100 Hz oraz 50 Hz
fon=0.9*(1/fs)*f; //znormalizowana częstotliwość odcięcia
filtru
r=5; //r - rząd filtru.
myfilter=iir(r,'lp','butt',[fon 0], [0 0]); //def. filtru IRR

//sygnały wyjściowe z filtra dolnoprzepustowego
output_y11 = flts(y11, myfilter);
output_y12 = flts(y12, myfilter);
output_y21 = flts(y21, myfilter);
output_y22 = flts(y22, myfilter);

//sygnały zespolone
z1=output_y11+%i*output_y12;
z2=output_y21+%i*output_y22;

//estymacja kąta fazowego
phase_diff1=atan(imag(z1),real(z1));
phase_diff2=atan(imag(z2),real(z2));

//wyznaczenie przesunięcia fazowego
WYNIK=phase_diff2-phase_diff1;

//uśrednienie wyników, konwersja na stopnie
D=sTime/(1/f);
phase_diff=[((sum(WYNIK))/((fs/f)*D))*(180/%pi)];
```

Wybrano te same sygnały oraz te same warunki symulacji tak, aby możliwe było porównanie otrzymanych estymat kąta przesunięcia fazowego. Wyniki symulacji przedstawiono w tabelach 3 oraz 4. W pierwszej części symulacji zbadano wpływ wysokości rzędu *r* wybranego filtra na poprawność wykonywanych symulacji. W drugiej części zwrócono uwagę na wpływ parametrów: częstotliwości próbkowania, czasu trwania symulacji, obecności wyższych harmonicznych, zakłóceń losowych oraz składowej nieokresowej.

Tabela 3. Estymacja kąta przesunięcia fazowego przy wykorzystaniu demodulacji kąta fazowego – zestawienie wyników symulacji – wpływ rzędu filtra na bezwzględny błąd estymacji

Lp.	f_s	sTime	f_{c1}	f_{c2}	Bezwzględny błąd estymacji Δ		
					Sygnał A r = 5	Sygnał A r = 10	Sygnał A r = 15
	Hz	s	Hz	Hz	°	°	°
1.	16000	1	50	50	0,59	–[1]	–[1]
2.	1200	10	50	50	0,05	0,13	–[1]
3.	1200	1	50	50	0,53	1,28	44,74
4.	1200	0.8	50	50	0,66	1,59	48,67
5.	1200	0.4	50	50	1,32	3,20	46,65
6.	1200	0.08	50	50	6,61	15,92	45,26
7.	1200	0.04	50	50	13,21	31,65	40,79
8.	120	1	50	50	20,11	9,09	11,47
9.	70	1	50	50	–[2]	–[2]	–[2]

Wyniki symulacji ukazują, że wysokość rzędu filtra dolnoprzepustowego silnie oddziałuje na poprawność uzyskiwanych wartości estymat kąta przesunięcia fazowego. Im wyższy rząd filtra, tym dłuższy czas potrzebny jest na ustalanie się jego odpowiedzi. Dla krótkich czasów symulacji, obecność filtra prowadziła do znacznych zafałszowań wyników. Przy częstotliwości próbkowania wynoszącej 16 kHz oraz wysokości rzędu filtra równej 15 oraz 30, liczba operacji, jaką musiał wykonać program symulacyjny była zbyt duża.

Na podstawie otrzymanych wyników można stwierdzić, że istnieje pewna graniczna wartość częstotliwości próbkowania, powyżej której bezwzględny błąd estymacji nie różni się znacząco. Natomiast dla częstotliwości próbkowania rzędu częstotliwości sygnału analizowanego uzyskuje się błędy wynoszące od ponad 20° do prawie 44°. Algorytm okazał się nieczuły na zakłócenia w postaci wyższych harmonicznych. Wartość błędu bezwzględnego dla sygnałów zawierających wyższe harmoniczne była większa od sygnałów zawierających jedynie podstawową harmoniczną ale różnica ta była nieznaczna, najczęściej rzędu 0,03°. Znaczną różnicę zauważono jedynie w przypadku, gdy częstotliwość próbkowania wynosiła 120 Hz. Występowanie zakłóceń losowych oraz składowej nieokresowej w badanych przebiegach, także nie doprowadziła do znaczących zmian w otrzymywanych wynikach symulacji.

Tabela 4. Estymacja kąta przesunięcia fazowego przy wykorzystaniu demodulacji kąta fazowego – zestawienie wyników symulacji – wpływ pozostałych czynników

Lp.	f_s	sTime	f_{c1}	f_{c2}	Bezwzględny błąd estymacji Δ			
					Sygnał A r = 5	Sygnał B r = 5	Sygnał C r = 5	Sygnał D r = 5
	Hz	s	Hz	Hz	°	°	°	°
1.	16000	1	50	50	0,59	0,62	0,59	0,61
2.	1200	10	50	50	0,05	0,05	0,02	0,05
3.	1200	1	50	50	0,53	0,56	0,67	0,52
4.	1200	0.8	50	50	0,66	0,7	0,48	0,65
5.	1200	0.4	50	50	1,32	1,41	1,67	1,30
6.	1200	0.08	50	50	6,61	7,03	6,84	7,1
7.	1200	0.04	50	50	13,21	14,56	14,34	14,99
8.	120	1	50	50	20,11	42,42	34,7	43,78
9.	70	1	50	50	–[2]	–[2]	–[2]	–[2]

–[1] – błąd wynikający z ograniczonej pamięci programu, która była niewystarczająca do wykonania symulacji o narzuconych parametrach
–[2] – błąd programu wynikający z wykonywania procedury filtracji, przy częstotliwości próbkowania niezgodnej z twierdzeniem Nyquista

3.4. Algorytm wykorzystujący metodę detekcji przejścia wartości sygnału przez zero

Poniżej dokonano analizy kodu źródłowego algorytmu estymującego kąt przesunięcia fazowego przy wykorzystaniu określenia chwil przejścia badanych sygnałów przez zero. Pominięto część związaną z deklarowaniem sygnałów badanych x1 i x2 oraz prezentacją przebiegów i błędu pomiaru.

```
//Wyznaczenie pochodnej funkcji signum badanego sygnału x1
find_zero_x1 = diff(sign(x1));

//Wyznaczenie punktów przejścia sygnału x1 przez zero
indx_up_x1 = find(find_zero_x1>0);
indx_down_x1 = find(find_zero_x1<0);

//Wyznaczenie pochodnej funkcji signum badanego sygnału x2.
find_zero_x2 = diff(sign(x2));

//Wyznaczenie punktów przejścia sygnału x2 przez zero
```

```
indx_up_x2 = find(find_zero_x2>0);
indx_down_x2 = find(find_zero_x2<0);
//Określenie obszaru poszukiwań miejsc zer. syg. x1
fs_zakres_x1=indx_up_x1(1,1)+(0.1*(fs/sTime));

//odszukanie miejsc zerowych syg. x1 w danym zakresie
ii=0;
while (find_zero_x1(1,fs_zakres_x1+ii) ~= 2);
 ii=ii+1
 x1_pt_loc=fs_zakres_x1+ii
end

//Określenie obszaru poszukiwań miejsc zerowych sygnału x2
fs_zakres_x2=indx_up_x2(1,1)+(0.1*(fs/sTime));

//odszukanie miejsc zerowych syg. x2 w danym zakresie
ii=0;
while (find_zero_x2(1,fs_zakres_x2+ii) ~= 2);
 ii=ii+1
 x2_pt_loc=fs_zakres_x2+ii
end

//Wyznaczenie kąta przesunięcia fazowego
dn=abs(fs_zakres_x1-fs_zakres_x2);
diff_phase=(360*dn/fs)/(1/f);
```

Wyniki symulacji przedstawiono w tabeli 5. Podczas modelowania zwrócono uwagę na wpływ parametrów analizowanych w poprzednich symulacjach: częstotliwości próbkowania, czasu trwania symulacji, obecności wyższych harmonicznych, zakłóceń losowych oraz składowej nieokresowej.

Analiza wyników potwierdza, że metoda estymacji kąta przesunięcia fazowego przy wykorzystaniu określenia momentu przejścia sygnału przez zero jest stosunkowo mało dokładna. Dla najwyższej częstotliwości próbkowania, wynoszącej 16 kHz, przy czasie trwania symulacji 1 s błąd bezwzględny metody wyniósł 0,63°. Zmniejszenie częstotliwości do poziomu 1,2 kHz powiększyło błąd względny odczytu kąta przesunięcia fazowego do 10%. Wpływ długości okna pomiarowego, w przypadku sygnału niezakłóconego nie odgrywał żadnej roli. W tym wypadku uwidocznił się jedynie wpływ częstotliwości próbkowania. Punkty określone metodą graficzną zgadzały się z punktami wyznaczonymi przez program. Następnie wyznaczono różnice pomiędzy chwilami przejścia przez zero sygnałów x1 i x2. Otrzymane różnice czasowe przeskalowano na odpowiednie wartości kątowe, znając częstotliwości podstawowe analizowanych sygnałów.

W przypadku zakłócenia sygnału wyższymi harmonicznymi uzyskano dużo gorsze rezultaty estymacji. Zauważono też silny wpływ zakłóceń losowych na estymowaną wartość kąta przesunięcia fazowego oraz rozbieżność w otrzymywanych, w tym przypadku wynikach. Po wykonaniu symulacji stwierdzono, że algorytm nie jest w stanie poprawnie dokonywać estymacji kąta fazowego w przypadku występowania w sygnale składowej nieokresowej. Bez względu na zmiany parametrów sygnału, czy długości okna pomiarowego wynikiem kąta przesunięcia fazowego każdorazowo był kąt równy 0°.

Tabela 5. Estymacja kąta przesunięcia fazowego przy wykorzystaniu metody określenia momentu przejścia sygnału przez zero – zestawienie wyników symulacji

Lp.	f_s	sTime	f_{c1}	f_{c2}	Bezwzględny błąd estymacji Δ			
					Sygnał A	Sygnał B	Sygnał C	Sygnał D
	Hz	s	Hz	Hz	°	°	°	°
1.	16000	1	50	50	0,63	15,25	28,75	50,0
2.	1200	10	50	50	5,0	10,0	20,0	50,0
3.	1200	1	50	50	5,0	10,0	175,0	50,0
4.	1200	0.8	50	50	5,0	10,0	10,0 [2]	50,0
5.	1200	0.4	50	50	5,0	10,0	175,0 [2]	50,0
6.	1200	0.08	50	50	–[1]	–[1]	–[1]	–[1]
7.	1200	0.04	50	50	–[1]	–[1]	–[1]	–[1]
8.	120	1	50	50	50	10,0	100,0	50,0
9.	70	1	50	50	207,14	207,14	207,14	50,0

–[1] – błąd programu podczas wykonywania iteracyjnej procedury obliczeniowej, analizowany przebieg był zbyt krótki

[2] – błędy w tym przypadku wynosiły od 5° do 175° w zależności od losowo wygenerowanych zakłóceń

4. WNIOSKI

Algorytm wykorzystujący dyskretną transformatę Fouriera cechowała odporność na występujące w sygnale wyższe harmoniczne. Występowanie harmonicznych rzędu trzeciego i piątego w żaden sposób negatywnie nie oddziaływało na wyniki pomiarów. Wpływ zakłóceń losowych, miał wpływ w przypadku, gdy czas obserwacji był krótszy niż 1 sekunda. Gdy częstotliwość próbkowania wynosiła 1,2 kHz, a czas symulacji 0,08 sekund, występowanie szumu w sygnale doprowadziło do wzrostu błędu aż o 12,26°. Niepożądany wpływ na wyniki miała także składowa nieokresowa. Dla najkrótszego czasu obserwacji wynoszącego 0,04 s błąd spowodowany obecnością składowej nieokresowej wyniósł ponad 30°.

Kolejny zaimplementowany algorytm wykorzystywał transformatę Hilberta. W przypadku analizy sygnału niezakłóconego, zawierającego jedynie podstawową harmoniczną otrzymywano znacznie lepszą dokładność niż w przypadku użycia transformaty Fouriera. Kąt estymowany był z dokładnością sięgającą nawet do 0,003°. Niestety, algorytm okazał się bardzo wrażliwy na występujące w sygnale wyższe harmoniczne – dodanie trzeciej i piątej harmonicznej zwiększyło błąd estymacji do około 14°. Na podkreślenie zasługuje fakt, że błąd ten, w przypadkach, gdy spełnione było twierdzenie Nyquista, był stały i nie zależał od częstotliwości próbkowania sygnału oraz od czasu trwania obserwacji. Dodanie do badanych sygnałów szumu pomiarowego oraz składowej nieokresowej nie wpłynęło znacząco na otrzymywane wyniki.

Trzecim rozpatrywanym sposobem estymacji kąta przesunięcia fazowego był algorytmiczny demodulator kąta fazowego. W tym przypadku trudniejsze jest dokonanie analizy błędów estymacji w odniesieniu do wcześniej omówionych algorytmów. Błąd estymacji silnie zależał od parametrów filtra dolnoprzepustowego użytego w procesie filtracji sygnału. Zauważono, że wzrost wysokości rzędu filtra nie zawsze prowadzi do poprawy otrzymywanych wyników. Do dalszych symulacji wybrano rząd filtra $r = 5$. Błąd estymacji wzrastał wraz ze skracaniem się czasu obserwacji. Dla 10 s błąd ten wynosił 0,05° natomiast dla 0,04 s ponad 13°. Algorytm wykazywał dużą odporność na zakłócenia pochodzące od wyższych harmonicznych oraz od sygnałów losowych lub składowej nieokresowej. Przyczyniła się do tego filtracja sygnałów, która miała miejsce we wstępnym etapie przetwarzania sygnałów.

Kolejny badany algorytm wykorzystywał metodę określenia chwili przejścia sygnału przez zero. Algorytm ten wykazywał najgorszą dokładność z dotychczas badanych. Przy wysokiej częstotliwości próbkowania wynoszącej 16 kHz błąd wyniósł 0,63°. Obniżenie częstotliwości próbkowania do poziomu 1,2 kHz doprowadziło do wzrostu błędu aż do 5°. Dodanie do sygnału harmonicznych rzędu trzeciego i piątego podwoiło błąd do 10°. Algorytm nie działał poprawnie w sytuacjach, gdy czas obserwacji był krótszy od 0,1s. W przypadku analizy przebiegów zawierających szum losowy, błąd przekraczał w niektórych przypadkach 175°. Przy obecności składowej nieokresowej wynikiem estymacji był każdorazowo kąt 0°, można więc wysunąć wniosek, że algorytm ten nie nadaje się do analizy stanów nieustalonych i przebiegów silnie odkształconych.

Po przeanalizowaniu wyników estymacji kąta przesunięcia fazowego dla czterech wyżej omówionych metod można stwierdzić, że są one zgodne z zebranymi na ich temat danymi literaturowymi i odpowiadają cechom charakterystycznym badanych algorytmów.

LITERATURA

[1] Proakis J. G., Manolakisi D. G., Scilab code for digital signal processing. Principles, algorithms and applications, Prentice Hall India, 2010.
[2] Matthieu P., Roux P., Scilab – from theory to practice, I. Fundamentals, Editions D–BookeR 2014.
[3] Strona pomocy progamu SciLab, https://help.scilab.org/, dostęp: 20.01.2017.

PHASE SHIFT MEASUREMENT ALGORITHMS IN ELECTRIC POWER PROTECION SYSTEMS, PART II

This article discusses and compares algorithms for the phase shift measurement between the two signals and used in the electric power protection systems. The selected algorithms were theoretically analysed: two using Fourier and Hilbert transformation, based on zero crossing detection and demodulation of the phase angle. The details of implementation in Scilab scientific environment were shown and their behavior under various operating parameters compared. The test signals were specified and later used in algorithms evaluation. Part two of the article presents the implementation of selected measurement algorithms.

(Received: 30. 01. 2017, revised: 15. 02. 2017)