Mateusz ZAPART[1], Cezary WOREK[1]

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, AGH, Instytut Elektroniki (1)
ORCID: 1. 0000-0002-6342-0737; 2. 0000-0003-3741-5501

doi:10.15199/48.2022.10.59

Analiza symulacyjna dyskretnego modelu układu SOGI-FLL wraz z uwzględnieniem behawioralnych modeli układów peryferyjnych mikrokontrolera

Streszczenie. *Artykuł przedstawia analizę modelu SOGI-FLL (Second Order General Integrator – Frequency Locked Loop) pod kątem odtwarzania częstotliwości napięcia sieci dla zastosowań w falownikach wyspowych sterowanych cyfrowo. W celu analizy układu SOGI-FLL zrealizowano jego model, odtwarzający funkcjonalność odpowiadającą programowej realizacji z użyciem mikrokontrolera. Wykonano porównanie realizacji obliczeń stałoprzecinkowych i zmiennoprzecinkowych pod kątem dokładności ustalania częstotliwości oraz odpowiedzi impulsowej.*

Abstract*. The presented article shows an analysis of the SOGI-FLL (Second Order General Integrator – Frequency Locked Loop) model for frequency recovery in digitally controlled off-grid inverters. The model of SOGI-FLL is created which reproduces the software functionality of the microcontroller realization. A comparison was made for fixed and floating-point arithmetic to determine the precision of frequency tracking and its dynamics, by observing step response. (**Simulation analysis of the SOGI-FLL discrete model including behavioral modeling of the microcontroller peripheral.**).*

Słowa kluczowe: SOGI-FLL, Ngspice, ADC, symulacja, cyfrowy, SOGI, SOGI-PLL
Keywords: SOGI-FLL, Ngspice, ADC, simulations, digital, SOGI, SOGI-PLL.

Wstęp

W artykule przedstawiono wyniki symulacji układu SOGI-FLL (ang. *Second Order General Integrator Frequency Locked Loop*) w realizacji dyskretnej, która miała na celu przetestowanie i dobór parametrów filtru związanych z implementacją algorytmu w mikrokontrolerze klasy STM32G474.

W ogólnym przypadku układ SOGI-FLL spełnia funkcję układu całkującego drugiego rzędu i zachowuje się podobnie do filtru opartego na obwodzie rezonansowym z wyjściem kwadraturowym. Poprzez dodanie sprzężenia stabilizującego amplitudę oraz przestrajającego częstotliwość otrzymuje się układ do śledzenia jednej ze składowych częstotliwości nawet odkształconego sygnału podanego na wejście. Zaletą tego typu obwodu jest łatwa realizacja w dziedzinie dyskretnej, która wymaga kilku prostych obliczeń, co pozwala na implementację w ogólnodostępnych mikrokontrolerach o niewielkich zasobach sprzętowych [1].

Układy odtwarzania częstotliwości są powszechnie używane w falownikach sieciowych, gdyż w urządzeniach tego typu konieczna jest synchronizacja fazy i częstotliwości między falownikiem, a siecią publiczną lub pomiędzy urządzeniami pracującymi w układzie wyspowym (ang. *Off-Grid*). Istnieje kilka rozwiązań realizujących taką funkcjonalność do których należą między innymi układy DPLL (ang. *Digital Phase Locked Loop*), oparte o transformatę Park'a, która wymaga dużo bardziej złożonych obliczeń z wykorzystaniem funkcji trygonometrycznych [2]. Układ SOGI wybrano ze względu na realizację prac badawczo-rozwojowych nad falownikiem jedno-fazowym, ze względu na brak przekształceń trygonometrycznych i prostotę realizacji. Kolejną zaletą wybranego układu jest posiadanie wyjścia kwadraturowego sygnałów, co z kolei pozwala, wraz z użyciem obwodu pomiaru prądu i napięcia, w łatwy sposób szacować na bieżąco moc czynną i bierną [3].

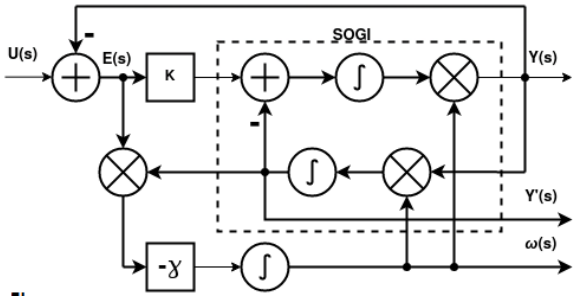
Istnieją również rozwiązania układu SOGI-PLL, gdzie wykorzystuje się filtr SOGI jako detektor fazy w obwodzie pętli fazowej [4]. Porównanie między SOGI-FLL, a SOGI-PLL wykazuje, że wersja FLL jest stabilniejsza w stanie ustalonym, a PLL szybsza w stanach nieustalonych [5]. Z perspektywy opisywanej aplikacji różnica jest na tyle mało znacząca, że wybrano prostszą implementację. Istnieją, oczywiście rozwiązania ulepszające układ SOGI-FLL względem zaprezentowanej podstawowej realizacji, do których należy na przykład maszyna stanów poprawiająca stabilność. Jest to związane z odtwarzaniem częstotliwości w przypadku zapadów napięcia jak i jego wzrostów [6]. Alternatywnym rozwiązaniem odtwarzania częstotliwości prezentowanym w literaturze jest użycie metody TKEO (ang. *Teager-Kayser Energy Operator*), pozwalająca poprawić wrażliwość na zmiany fazy przy detekcji częstotliwości w stosunku do pętli fazowej [7]. Następnym rozwiązaniem jest metoda DESA-2 (ang. Discrete Energy Separation Algorithm), która jest również prosta obliczeniowo i prawidłowo odtwarza częstotliwość nawet przy pracy z mocno zniekształconymi sygnałami [8].

Prezentowany artykuł w stosunku do dostępnych prac, gdzie analizy są przedstawiane w sposób analityczny lub operatorowy z wykorzystaniem narzędzi takich jak MATLAB, skupia się na realizacji zadania używając narzędzia symulacyjnego Ngspice (link: ngspice.sourceforge.net) o otwartym kodzie źródłowym. Układ SOGI-FLL zaimplementowano w wersji dyskretnej w oparciu o metodę Eulera w przód [1] w języku C w środowisku Ngspice, używając rozszerzenia XSPICE. Pozwoliło to na symulowanie systemu przerwań w oparciu o który zbierane są próbki z przetwornika analogowo-cyfrowego, w sposób bardzo zbliżony do rzeczywistej implementacji z wykorzystaniem mikrokontrolera.

Układ SOGI-FLL budowa i zasada działania

Na rysunku 1 przedstawiono schemat blokowy układu całkującego drugiego rzędu wraz z pętlą odtwarzania częstotliwości. Układ składa się z dwóch układów całkujących, połączonych w pętlę. Wejściem sygnału do obwodu całkującego, jest wyjście z bloku wzmocnienia oznaczonego literą κ. Po odjęciu sygnału z wyjścia drugiego układu całkującego, po przejściu przez blok całkujący, sygnał wynikowy podawany jest na wyjście Y(s) poprzez układ mnożący. Można przyjąć, że jest to iloczyn przez stałą liczbę, która odpowiada częstości kątowej. W wyniku tego iloczynu otrzymywany jest sygnał wyjściowy Y(s), który podawany jest na drugi blok całkujący, którego wyjście, jest drugim wyjściem układu SOGI. Sygnał Y'(s) jest przesunięty o 90 stopni względem Y(s). Układ SOGI jest obwodem

drugiego rzędu, zachowującym się jak obwód rezonansowy. Jest to obwód nietłumiony, dlatego w docelowej realizacji amplituda wyjściowa jest kontrolowana przy pomocy członu proporcjonalnego. Różnica między wejściem U(s), a wyjściem Y(s) tworzy sygnał błędu, który mnożony przez stałą κ i jest podawany na wejście układu SOGI. Ostatnim elementem jest pętla przestrajania częstotliwości. Sygnał błędu E(s) z wyjściem kwadraturowym Y'(s) tworzy iloczyn, skalowany przez stałą -γ, a następnie całkowany. Wynikiem jest częstość kątowa ω(s) odpowiadająca sygnałowi wejściowemu. Kiedy częstotliwość wejściowa, równa jest częstotliwości środkowej układu SOGI, to sygnał błędu Ef(s) jest równy zeru. W sytuacji, gdy częstotliwość wejściowa sygnału U(s) jest mniejsza od aktualnej częstotliwości podawanej na wejście SOGI ω(s), to sygnał błędu E(s) oraz Y'(s) mają tą samą fazę, więc Ef(s) jest również dodatnie, co w efekcie zmniejsza wartość całki. W przypadku sygnału wejściowego o częstotliwości większej od częstotliwości środkowej układu SOGI, sygnał Ef(s) staje się ujemny, co powoduje zwiększanie wartości całki.



Rys.1. Schemat blokowy układu SOGI-FLL

Przedstawiony model musi zostać zdyskretyzowany przed implementacją w mikrokontrolerze. W wyniku dyskretyzacji metodą Eulera w przód, otrzymuje się następujące wzory:

(1) $e[n]=u[n]-y[n-1]$

(2) $y[n]=y[n-1]+(e[n]\cdot k-y^{'}[n])\cdot\omega[n-1]\ T_s$

(3) $y'[n]=y'[n-1]+y[n]\cdot\omega[n-1]\cdot T_s$

(4) $\omega[n]=\omega[n-1]-\gamma\cdot y'[n]\cdot e[n]\cdot T_s$

gdzie $e[n]$ – sygnał błędu, $y[n]$ – wyjście, $y'[n]$ – wyjście kwadraturowe

Przedstawione równania są bardzo proste do realizacji z wykorzystaniem mikrokontrolera i mogą być szybko wykonywane.

Realizacja na liczbach stałoprzecinkowych

Realizacja na liczbach stałoprzecinkowych pozwala na zastosowanie prostych mikrokontrolerów lub układów FPGA, które nie zawierają jednostek zmiennoprzecinkowych. W tym celu przygotowano realizację wykorzystując zmienne 32- bitowe. Przeprowadzone symulacje pozwoliły określić wykorzystanie zakresu zmiennych, występowanie przepełnień oraz ocenić dokładność obliczeń. W takim wypadku realizacja w oparciu o wzory 1-4 przedstawia się następująco (reprezentacja w języku C):

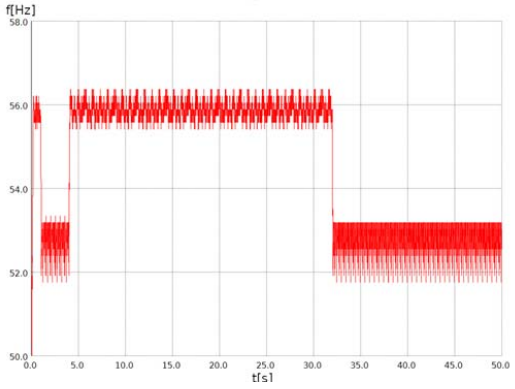
```
int32_t e = input – prevOutput;
int32_t amplified_error = (e * k) / 100;
int32_t stab_fb_out = amplified_error –prevOutq;
int32_t outputInc= (stab_fb_out * prevOutω) / fs;
int32_t Output = sogi->prevOutput + outputInc;
int32_t Outputq = prevOutq + (Output * prevOutω ) / fs;
int32_t ωInc = (gamma * Outputq * e) / 100;
int32_t Outputω = prevOutputω – ωInc /fs;
```

gdzie: input – wartość próbki sygnału wejściowego, prevOutput – wartość próbki sygnału wyjściowego z poprzedniej iteracji, k – współczynnik dobroci, prevOutq – wartość próbki z poprzedniej iteracji dla wyjścia kwadraturowego, fs – stała reprezentująca częstotliwość przetwarzania, z którą działa układ SOGI, Output - sygnał wyjściowy, Outputω – wartość odtworzonej pulsacji

Przedstawiona realizacja zawiera zmienne pomocnicze, użyte dla zapewnienia czytelności zapisu. Pierwszym istotnym elementem jest wzmocniony sygnał błędu e[n], który jest mnożony przez stałą i dzielony przez 100. Testy wykazały, że takie przesunięcie przecinka jest wystarczające dla odpowiedniej rozdzielczości zapewniającej strojenie obwodu poprzez zmiany współczynnika k. Drugi współczynnik skalujący występuje w linii, w której jest obliczana zmiana częstotliwości *ωInc.* Liczba ta jest dzielona najpierw przez 100, a potem przez częstotliwość przetwarzania. W ten sposób tracona jest informacja o dokładnej częstotliwości co jest wadą rozwiązania. Zapewnienie dynamiki zmiennych stałoprzecinkowych oraz utrzymanie możliwie dużej rozdzielczości odtwarzania częstotliwości wydaje się złożonym zagadnieniem. Dlatego dużo lepszym rozwiązaniem jest zastosowanie liczb zmiennoprzecinkowych.

Symulacja z użyciem liczb stałoprzecinkowych

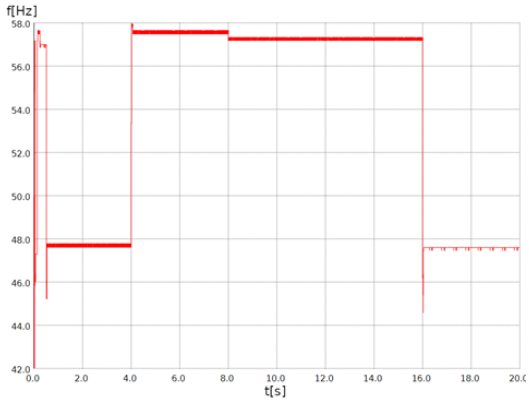
Symulację wykonano podając na wejście modelowanego przetwornika sygnał sinusoidalny o amplitudzie 100mV ze składową stałą 1V. Składowa ta jest następnie odejmowana po próbkowaniu, tak aby odczyty z przetwornika oscylowały wokół wartości zero, ponieważ układ SOGI-FLL w prezentowanej wersji nie odtwarza poprawnie częstotliwości z obecną składową stałą na wejściu. Wynik symulacji pokazuje rezultat przy pracy z sygnałem sinusoidalnym podawanym na wejście i z wykorzystaniem przetwornika 8-bitowego.



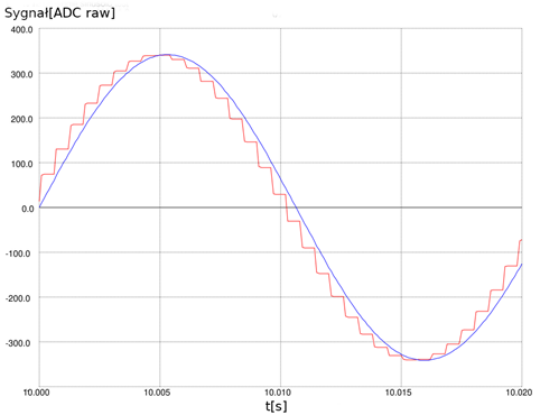
Rys.2. Wyjście odtwarzanej częstotliwości układu SOGI, realizacja stało-przecinkowa, fin=47 Hz, ADC 8-bit, gamma=6000 k=200 fs=1000 Hz, Amplituda sygnału podawanego na wejście 100 mV, ADCRef=1,2 V

Przedstawiony na rysunku 2 wykres pokazuje, że w wersji 8- bitowej i z próbkowaniem 1000 Hz układ błędnie określa częstotliwość. Zwiększanie częstotliwości przetwarzania nie pomaga rozwiązać problemu, nadal występują przeskoki na wyjściu, które nie zbliżają się do wartości rzeczywistej 47 Hz. Testy dla różnych współczynników i częstotliwości sygnału wejściowego w zakresie 40 Hz – 60 Hz pokazują analogiczne problemy. Odczytywana częstotliwość jest zawsze zawyżona i zmienia się skokowo.

Kolejne testy wykonywano z przetwornikiem 12-bitowym, zwiększono również częstotliwość wykonywania obliczeń i próbkowania do 2 kHz, ponieważ dla 1 kHz, układ nadal nie odtwarzał prawidłowo częstotliwości. Współczynnik gamma jest tutaj znacząco mniejszy w stosunku do wersji 8-bitowej. Wynika to z faktu, że zakres liczb wejściowych przy wersji 12-bitowej został powiększony, gdzie zakres przyjmowanych wartości jest od 0 do 4095 zamiast od zera do 255.



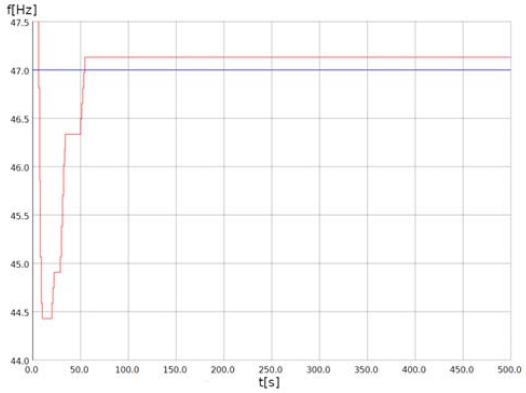
Rys.3. Realizacja stało-przecinkowa, Wyjście odtwarzanej częstotliwości układu SOGI-FLL, fin=47 Hz, ADC 12-bit, gamma=50 k=100, fs=2000 Hz, vinamp=100 mV, ADCRef=1,2V



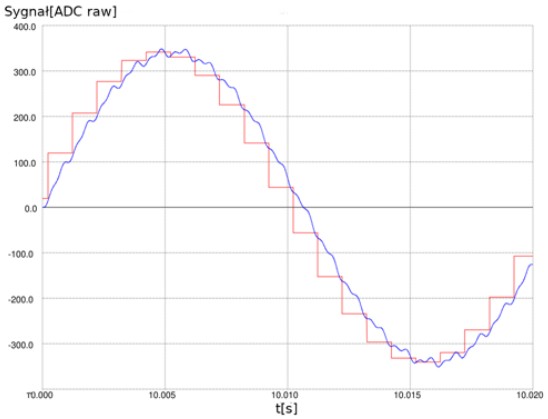
Rys.4. Realizacja stało-przecinkowa, Przebiegi wejścia/wyjścia. Niebieski - sygnał wejściowy, Czerwony - wyjście układu SOGI-FLL, fin=47 Hz, ADC 12-bit, gamma=50, k=100, fs=2000 Hz, vinamp=100 mV, ADCRef=1,2V

Na rysunku 3 i 4 pokazano, że układ dostroił się do wartości bliskiej 47 Hz. Jest widoczne niewielkie przesunięcie fazowe na przebiegu czasowym, przebieg wyjściowy jest lekko opóźniony co wskazuje, że częstotliwość środkowa układu SOGI nie jest idealnie dostrojona względem sygnału wejściowego. Na rysunku 3 pokazującym odtwarzaną częstotliwość widoczne jest przesterowanie do 58 Hz podczas startu układu. Jako częstotliwość początkową dla układu SOGI-FLL wybrano 50 Hz. Obwód w początkowej fazie ustalił się na około 48 Hz, a następnie utracił synchronizację i do niej powrócił. Testy z różnymi wzmocnieniami pokazały, że nie jest osiągalny wynik z wartościami pomiędzy tymi liczbami, co sugeruje, że jest to efekt numeryczny realizacji stałoprzecinkowej.

Wykonano również symulacje pod kątem analizy zachowania obwodu w sytuacji, gdy na wejściu nie jest podawany idealny sygnał sinusoidalny. Nałożono składową o niewielkiej amplitudzie i częstotliwościach 1 kHz (Vpp=4 mV) i 2 kHz (Vpp=3 mV). Wyniki przedstawiono na kolejnych rysunkach 5 i 6, pokazując wyjście częstotliwości i odfiltrowany sygnał.



Rys.5. Realizacja stało-przecinkowa, Czerwony - Wyjście odtwarzanej częstotliwości układu SOGI-FLL, Niebieski- linia odniesienia 47 Hz, fin=47 Hz, ADC 12bit. gamma=20, k=100, fs=1000 Hz, vinamp=100 mV wraz z sygnałami 1 kHz i 2 kHz



Rys.6. Realizacja stało-przecinkowa, Przebiegi wejścia/wyjścia. Niebieski - sygnał wejściowy, Czerwony-wyjście układu SOGI-FLL, fin=47 Hz, ADC 12 bit, gamma=20, k=100, fs=1000 Hz, Vinamp=100 mV wraz z sygnałami 1 kHz i 2 kHz

Wyniki pokazują, że układ zachowuje się znacząco lepiej w stosunku do testu z idealnym przebiegiem sinusoidalnym. Stan ustalony na wyjściu częstotliwościowym, osiągany jest po czasie mniejszym niż 100ms, gdzie w wersji z mniejszą ilością bitów było to około 17 sekund. Na rysunku 6 można zauważyć, że nie występuje przesunięcie fazy między wejściem, a wyjściem jak to było na poprzednim rysunku.

Zauważono również, że amplituda sygnału wejściowego znacząco wpływa na stabilność części odpowiedzialnej za detekcję częstotliwości sygnału wejściowego. Konieczna jest zatem korekcja współczynnika gamma, przy zmianie amplitudy z 100mV na 200mV współczynnik wymaga zmniejszenia. Dla dużych amplitud gamma powinno być małe, a dla małych duże. Nie powinno to stanowić problemu w rzeczywistym układzie, ponieważ amplituda sygnału sinusoidalnego nie zmienia się znacząco, podczas prawidłowego działania. Kolejnym problemem jest zależność od amplitudy wejściowej, która implikuje konieczność doboru współczynników. Istotne znaczenie ma gamma, ponieważ jego zmiany powodują, że obwód albo zaniża, albo zawyża śledzoną częstotliwość. Wersja z wykorzystaniem liczb stało przecinkowych sprawia problemy, choć realizacja jest możliwa, ale konieczne są duże nakłady pracy do weryfikacji poprawności działania. Odrzucono to rozwiązanie ze względu na możliwość łatwego użycia liczb zmiennoprzecinkowych.

Testy symulacyjne wykazały, że jedną z istotnych przyczyn nieprawidłowego działania jest problem zakresu wartości liczb. Jako przykład można podać skrajną sytuację, gdy częstotliwości próbkowania są duże, a w kolejnych krokach zmiany sygnałów są tak małe, że pojawia

się dzielenie przez zero uniemożliwiając działanie. Zmiana przetwornika z 8 bitów na 12 bitów znacząco poprawia działania układu SOGI. Należy dodać, że amplitudy wejściowe stają się większe, ze względu na większy zakres liczb wejściowych z przetwornika od 0 do 4095 zamiast 0 do 255. Może zdarzyć się przepełnianie zmiennych 32 bitowych, w wyniku czego krytyczny staje się dobór wzmocnienia k i współczynnika gamma. Kolejnym problemem jest utrata dokładności przy niskich częstotliwościach próbkowania, która widoczna jest na wyjściu odtwarzania częstotliwości. Układ zapewnia wystarczającą filtrację przebiegu wejściowego, ale przy sygnale wejściowym o częstotliwości 47 Hz wyjście układu pokazuje, że jest to 50 Hz i więcej. Układ nie radzi sobie również z małymi amplitudami sygnału wejściowego, a na wyjściu pojawia się wtedy błędna wartość zero.

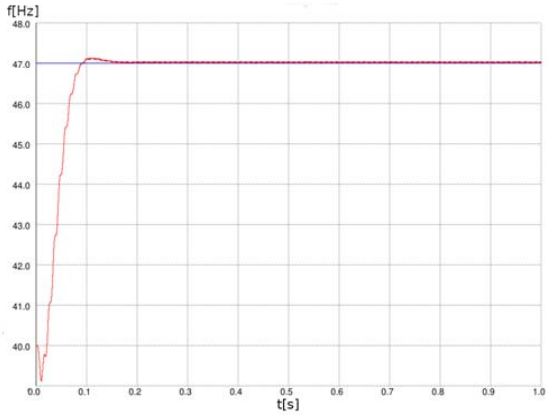
Realizacja na liczbach zmiennoprzecinkowych

Aby rozwiązać problemy z dynamiką zmiennych występujące w realizacji stałoprzecinkowej zrealizowano model używający zmiennych typu float32. Koprocesory zmiennoprzecinkowe 32-bitowe dostępne są obecnie w dużej klasie mikrokontrolerów z rdzeniem Cortex-M4, więc taka realizacja może być niemal tak szybka jak stałoprzecinkowa. Kod zrealizowano z wykorzystaniem języka C w analogiczny sposób jak przedstawiony wcześniej.

```
float e = (float)input - *prevOutput;
float amplified_error = (e * k);
float stab_fb_out = amplified_error – prevOutputq;
float outputInc= (stab_fb_out * prevOutω) / fs;
float Output = sogi->prevOutput + outputInc;
float Outputq = prevOutq + (Output * prevOutω ) / fs;
float ωInc = (gamma * Outputq * e);
float Outputω = prevOutputω - ωInc/fs;
```

gdzie: input - wartość próbki sygnału zmierzona przez ADC, prevOutput – wartość próbki sygnału wyjściowego z poprzedniej iteracji, k – współczynnik, prevOutq – wartość próbki z poprzedniej iteracji dla wyjścia kwadraturowego, fs– stała reprezentująca częstotliwość przetwarzania, z którą działa SOGI, *Output - sygnał wyjściowy, Outputω – wartość odtworzonej pulsacji.*

Zaprezentowana implementacja różni się jedynie brakiem współczynników skalujących względem wcześniej przedstawionej wersji.



Rys.7. Realizacja zmiennoprzecinkowa, Czerwony - Wyjście odtwarzanej częstotliwości układu SOGI-FLL, Niebieski - linia odniesienia 47 Hz, fin=47 Hz, ADC 12-bit, gamma=0,06, k=0,6, fs=50 kHz, vinamp=100 mV wraz z sygnałem 1kHz (Vpp=20 mV)

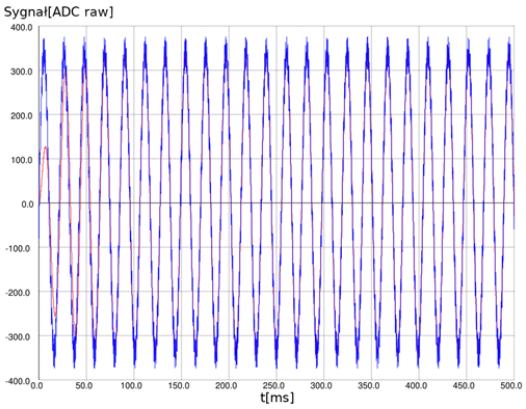
Symulacja z użyciem liczb zmiennoprzecinkowych

Zaprezentowano symulację dla przetwornika 12-bitowego, ponieważ 8-bitowe są obecnie spotykane wyłącznie w najprostszych wersjach mikrokontrolerów, a ponadto nie dają dobrych wyników, co pokazano przy-realizacji stałoprzecinkowej. Na wejście SOGI-FLL podano sygnał sinusoidalny o częstotliwości 47Hz z niewielką składową o wartości 1kHz i 2kHz. Początkową wartość prędkości wykonywania obliczeń jak i bezpośrednio powiązaną częstotliwość próbkowania przyjęto na 50kHz otrzymując znaczące nadpróbkowanie. Takie działanie upraszcza uruchamianie. Sygnał odtwarzanej częstotliwości przedstawiono na rysunku 7.

Wyjście bloku dostrajania częstotliwości ma niewielkie tendencje do oscylacji/przesterowania. Można zmniejszyć ten efekt zmniejszając współczynnik gamma, co nieznacznie wydłuży czas ustalania częstotliwości.

Sygnał wyjściowy podczas startu szybko osiągnął docelową amplitudę, co przedstawiono na rysunku 8 w formie czerwonego przebiegu.

Dodatkowo na rysunku 8 widoczny jest start układu, osiąganie stanu ustalonego. Amplituda stabilizuje się szybko w ciągu jednego okresu. Należy zaznaczyć, że układ startuje od napięcia zerowego, faza między generowanym przebiegiem przez układ SOGI, a sinusoidalnym sygnałem wejściowym jest zachowana, a jedynie występuje różnica w częstotliwości.



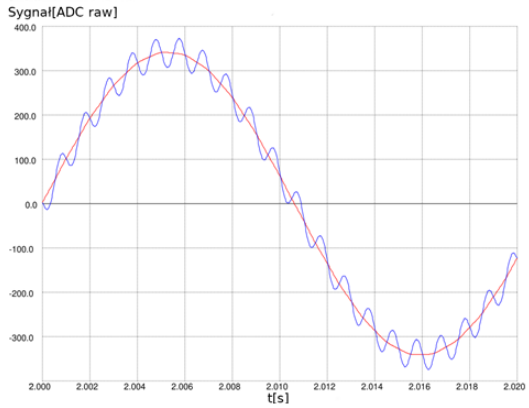
Rys.8. Realizacja zmiennoprzecinkowa, przebiegi czasowe podczas startu, Niebieski - sygnał wejściowy, Czerwony-wyjście układu SOGI-FLL, fin=47 Hz, ADC 12-bit, gamma=0,06, k=0,6, fs=50 kHz, vinamp=100 mV, Vinamp=100 mV wraz z sygnałem 1 kHz(Vpp=20 mV).

Na rysunku 9 przedstawiono przebiegi sygnału wejściowego oraz wyjściowego dla jednego okresu. Ze względu na dużą częstotliwość próbkowania dyskretyzacja jest praktycznie niewidoczna. Przebiegi są ze sobą w fazie, obserwowalna jest skuteczność filtrowania, zakłócenie o częstotliwości 1kHz jest na niezauważalnym poziomie.

W realizacji z wykorzystaniem mikrokontrolera, zaprezentowana szybkość przetwarzania w naszym zastosowaniu jest nieosiągalna, ze względu na ilość dodatkowych obliczeń i strukturę oprogramowania. Kolejne symulacje wykonano dla mniejszych częstotliwości, które pozawalają zapewnić zapas czasu na obliczenia.

Jako pierwszą częstotliwość próbkowania przetestowano 500 Hz. W wyniku otrzymywane jest 10 próbek na okres przebiegu 50 Hz. Testy tak niskiej szybkości przetwarzania pokazują, że nie udaje się uzyskać dostrojenia do 47 Hz, obwód zaniża częstotliwość o niecały 1 Hz.

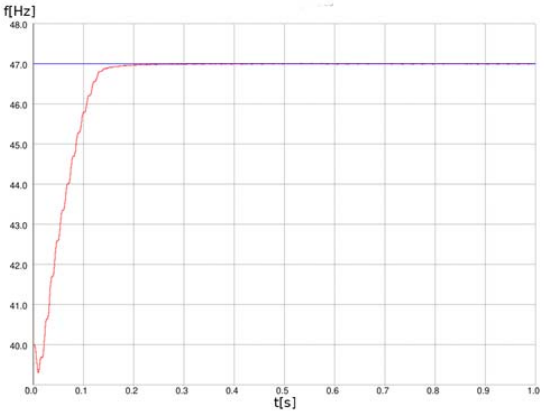
Kolejne testy wykonano z częstotliwością 1 kHz, gdzie wynikowa częstotliwość jest również zaniżona tym razem do 46,86 Hz. Gdy zostaną wybrane współczynniki jak dla realizacji stałoprzecinkowej odpowiednio przeskalowane z rysunku 5 i 6, to wyjście częstotliwościowe ulega oscylacjom o amplitudzie 1,8 Hz. Podobnie jest dla współczynników jak dla symulacji z rysunku 7.



Rys.9. Realizacja zmiennoprzecinkowa, stan ustalony, Niebieski - sygnał wejściowy, Czerwony-wyjście układu SOGI-FLL, fin=47 Hz, ADC 1-2bit, gamma=0,06. k=0,6. fs=50 kHz, wraz z sygnałem 1 kHz (Vpp=20 mV)

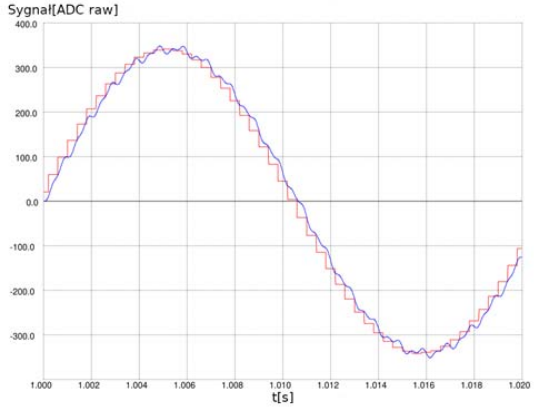
Dopiero przy częstotliwości fs=2,5 kHz otrzymano wynik na poziomie 46,999 Hz, co przedstawiono na rysunku 10. Współczynniki zostały zmodyfikowane, aby uniknąć przesterowania jak na rysunku 7, co tylko w niewielkim stopniu spowalnia dochodzenie do stanu ustalonego.

Na rysunku 11 przedstawia przebieg czasowy wejścia i wyjścia dla częstotliwości próbkowania 2,5 kHz. Filtracja przebiegu i śledzenie częstotliwości działa poprawnie. W celu sprawdzenia czy stabilność się nie zmienia przy zmianie amplitudy wejściowej jak to ma miejsce w przypadku realizacji stałoprzecinkowej, zmniejszono dwukrotnie amplitudę sinusoidy wejściowej pozostawiając dodatkowe sygnały 1 kHz i 2kHz bez zmiany amplitudy. Otrzymano odpowiedź taką samą jak dla symulacji z rysunku 10, co potwierdza poprawne działanie względem realizacji stałoprzecinkowej.

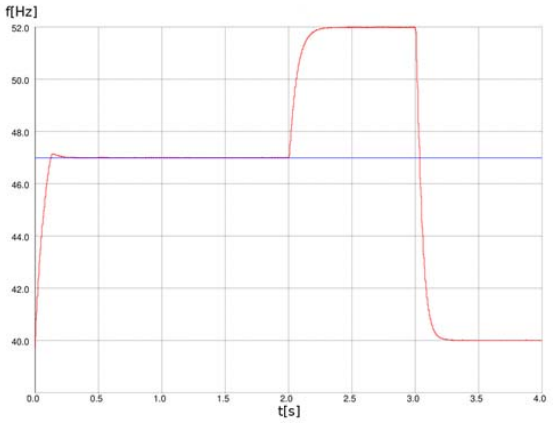


Rys.10. Realizacja zmiennoprzecinkowa, Czerwony - Wyjście odtwarzanej częstotliwości układu SOGI-FLL, Niebieski - linia odniesienia 47 Hz, fin=47 Hz, ADC 12-bit, gamma=0,04, k=0,9, fs=2,5 kHz, vinamp=100 mV wraz z sygnałem 1 kHz (Vpp=4 mV) i 2 kHz(Vpp=3 mV)

W następnej kolejności przedstawiono test stabilności dla sygnałów wejściowych o częstotliwościach różnych od 47 Hz oraz dla stanów przejściowych. Wykonano testy skoku częstotliwości sygnału podawanego na wejście układu SOGI-FLL, a wynik pokazano na rysunku 12. W chwili czasowej t=2 s, zamiast sygnału sinusoidalnego o częstotliwości 47 Hz, podawany jest przebieg 52 Hz, a następnie w t=3 s przebieg 40 Hz. W wyniku skoku częstotliwości stan stabilny jest otrzymywany po około 200ms. Odpowiedź jest stabilna i dość szybka jak na wymagania falowników sieciowych w naszym zastosowaniu.



Rys.11. Realizacja zmiennoprzecinkowa, stan ustalony, Niebieski - sygnał wejściowy, fin=47 Hz, ADC 12-bit, gamma=0,04, k=0,9, fs=2,5 kHz, vinamp=100mV wraz z sygnałem 1 kHz (Vpp=4 mV) i 2 kHz (Vpp=3 mV)



Rys.12. Realizacja zmiennoprzecinkowa, skoki częstotliwości, Niebieski – odniesienie 47 Hz, Czerwony-wyjście częstotliwości układu SOGI-FLL, fin=47 Hz, ADC 12-bit, gamma=0,04, k=0,9, fs=2,5 kHz, wraz z sygnałem 1 kHz (Vpp=4 mV) i 2 kHz(Vpp=3 mV)

Podsumowanie

Przedstawiona analiza pokazuje, że układ z obliczeniami wykonywanymi na liczbach stałoprzecinkowych może działać poprawnie, ale uzyskanie takiego efektu wymaga podjęcia dodatkowych działań ze względu na wrażliwość na zmiany amplitudy sygnału wejściowego, które znacząco pogarszają pracę obwodu. Dodatkowo układ, wykazuje problemy z śledzeniem częstotliwości sygnału wejściowego. Symulacje pokazały, że wersja z przetwornikiem analogowo cyfrowym 12-bitowym względem 8-bitowego dla realizacji obliczeń ze stałym przecinkiem poprawia sytuację, ale czasami uniemożliwia prawidłowe działanie obwodu ze względu na przepełnianie zmiennych 32-bitowych. Rozwiązanie z wykorzystaniem obliczeń na bazie systemu float32 działa już prawidłowo. Układ zachowuje się stabilnie przy częstotliwości próbkowania 2,5 kHz, przy czym testowany układ radzi sobie bez problemu ze skokami częstotliwości w zakresie 40-52 Hz dochodząc do stanu ustalonego w około 200 ms. Testy wykazały użyteczność układu SOGI-FLL jako prostego i szybkiego rozwiązania do śledzenia częstotliwości napięcia sieci. Prezentowane rozważania pokazują, że można wykorzystać układ SOGI-FLL w implementacji na mikrokontroler opartym o rdzeń Cortex-M4F zostawiając duży margines czasowy na inne operacje matematyczne. Dodatkowo ze względu na to, że znacząca część obecnie dostępnych mikrokontrolerów wyposażona jest w 12-bitowe przetworniki analogowo-cyfrowe, można

bez większych przeszkód wykorzystać współczesne mikrokontrolery w falownikach jedno-fazowych do synchronizacji fazy i częstotliwości między falownikiem, a siecią publiczną lub pomiędzy urządzeniami pracującymi w układzie wyspowym (ang. *Off-Grid*).

Praca finansowana z subwencji naukowo-badawczej Instytutu Elektroniki Akademii Górniczo-Hutniczej w Krakowie.

***Autorzy**: mgr inż. Mateusz Zapart, AGH Akademia Górniczo-Hutnicza, Instytut Elektroniki, al. Adama Mickiewicza 30, 30-059 Kraków, E-mail: mzapart@agh.edu.pl; dr hab. inż. Cezary Worek, AGH Akademia Górniczo-Hutnicza, Instytut Elektroniki,, al. Adama Mickiewicza 30, 30-059 Kraków, E-mail: worek@agh.edu.pl.*

LITERATURA

[1] Możdżyński K., Rafał K., Bobrowska-Rafał M., Application of the second order generalized integrator in digital control systems, *ARCHIVES OF ELECTRICAL ENGINEERING*, 63 (2014), 423-437

[2] Sakamoto S., Izumi T., Yokoyama T., Haneyoshi T., A New Method for Digital PLL control Using Estimated Quadrature Two Phase Frequency Detection, *Proceedings of the Power Conversion Conference-Osaka,* (2002), 671-676

[3] Kherbachi A, Bendib A, Kara K., Chouder A, ARM based implementation of sogi-fll method for power calculation in single-phase power system, *5th International Conference on Electrical Engineering – Boumerdes,* (2017)

[4] Ikken N, Bouknadel A., Haddou A, Tariba N., El omari H., El omari H, PLL Synchronization Method Based on Second-Order Generalized Integrator for Single Phase Grid Connected Inverters Systems during Grid Abnormalities, *International Conference on Wireless Technologies, Embedded and Intelligent Systems,* (2019)

[5] Royan, Andromeda T, Facta M., Hermawan,Setiawan I., Trias Andromeda, Comparison of SOGI-FLL with SOGI-PLL for Single-Phase Grid-Connected Inverters, *The 4th International Conference on Energy, Environment, Epidemiology and Information System*, 125(2019)

[6] Nejad S., Matas J., Martin H., Hoz J., Al-Turki Y., New SOGI-FLL Grid Frequency Monitoring with a Finite State Machine Approach for Better Response in the Face of Voltage Sag and Swell Faults, *Electronics,* 9(2020), nr. 4, 612

[7] Sahoo A, Thattai K., Ravishankar J., Ciobotaru M., Power-Sharing Based on Open-Loop Synchronization of Inverters in an Islanded AC Microgrid, *44th Annual Conference of the IEEE Industrial Electronics Society,* (2018), 237-242

[8] Sugiura Y., Usukura K., Aikawa N., Instantaneous frequency estimation for a sinusoidal signal combining DESA-2 and notch filter, *European Signal Processing Conference (EUSIPCO)*, (2015), 2676-2680