

Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora (22)

Analiza widmowa z zastosowaniem transformacji Fouriera

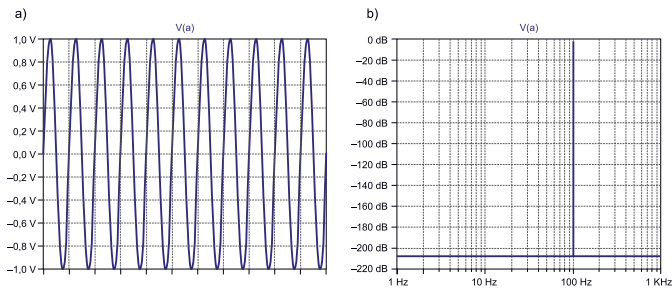


Jeszcze w czasach użytkowania oscyloskopów analogowych, dokładnie w roku 1965, dwaj matematycy amerykańscy: James William Cooley i John Tukey opublikowali dokument „An Algorithm for the Machine Calculation of Complex Fourier Series”, w którym podali algorytm szybkiego obliczania dyskretnej transformacji Fouriera. Algorytm, znany dzisiaj jako Fast Fourier Transform (FFT) jest obecnie szeroko stosowany do analizy widmowej sygnałów.

Niemal 150 lat musiało upłynąć zanim opracowana przez Jeana Baptiste Josepha Fouriera teoria doczekała się praktycznego zastosowania. Dopiero gdy technika cyfrowa pozwoliła na próbkowanie sygnałów analogowych, a tym samym na gromadzenie dyskretnych próbek tego sygnału, możliwe stało się opracowanie metod dyskretnej przekształcenia Fouriera – DFT (*Discrete Fourier Transform*). Problem polega na tym, że operacja ta wymaga bardzo dużej liczby obliczeń, co początkowo praktycznie eliminowało tę metodę

do zastosowań w aplikacjach czasu rzeczywistego. Nawet analiza wykonywana już po zakończeniu gromadzenia próbek była przy dostępnych mocach obliczeniowych pierwszych komputerów dość złożona i długotrwała. Zaproponowany przez Cooley'a i Tukey'a algorytm FFT na tyle zmniejszył liczbę operacji matematycznych, że obliczenie transformacji Fouriera nawet w układach czasu rzeczywistego stało się realne.

Problem był właściwie jeden, za to bardzo poważny – nie istniały jeszcze praktycznie oscyloskopy cyfrowe, w których FFT mogłaby być implementowana. Przełom nastąpił na przełomie lat 80. i 90. XX. wieku, właściwie bliżej początku ostatniej dekady minionego stulecia. Od tego momentu oscyloskopy cyfrowe zaczęły wypierać oscyloskopy analogowe, ale to też jeszcze nie był czas, w którym FFT była masowo implementowana. W pierwszych oscyloskopach cyfrowych funkcja ta nie była jeszcze dostępna. Na szczęście postępowanie w tym zakresie było bardzo szybkie, i wkrótce oscyloskopy bez FFT znajdowały coraz mniej nabywców. Dziś funkcję FFT znajdujemy we wszystkich rodzajach oscyloskopów, czy to typu desktop czy przystawek USB. Analiza FFT jest także dostępna w programach matematycznych (Mathcad, Excel), a także symulatorach, m.in. w LTspice.



Rysunek 1. Czysty przebieg sinusoidalny a) w dziedzinie czasu, b) w dziedzinie częstotliwości

Implementacja FFT w symulatorze LTspice

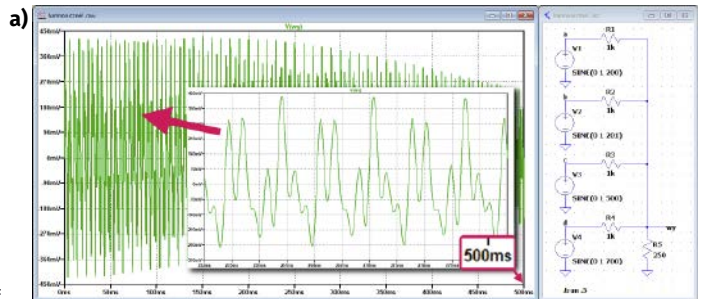
Przekształcenie Fouriera i algorytmu DFT i FFT jest dość złożonym zagadnieniem matematycznym, na tyle że nie będziemy go omawiać w artykule. Zajmiemy się praktycznym zastosowaniem analizy FFT w symulatorze LTspice. Rozpatrzmy, jaki wpływ na wyniki analizy mają parametry akwizycji danych i funkcji FFT.

Do przeprowadzenia analizy widmowej z użyciem FFT konieczne jest zgromadzenie określonej liczby próbek sygnału analogowego. Zasadniczo algorytm wymaga, aby liczba ta była równa 2^k , gdzie k jest liczbą naturalną. Inaczej mówiąc, liczba próbek musi być potęgą liczby 2. Intuicyjnie spodziewamy się, że wraz ze wzrostem liczby k będzie zwiększała się liczba próbek, którą będziemy dysponować, a zatem obliczenia będą dokładniejsze. Czy jest to jedyny parametr decydujący o dokładności obliczeń? Postaramy się to sprawdzić.

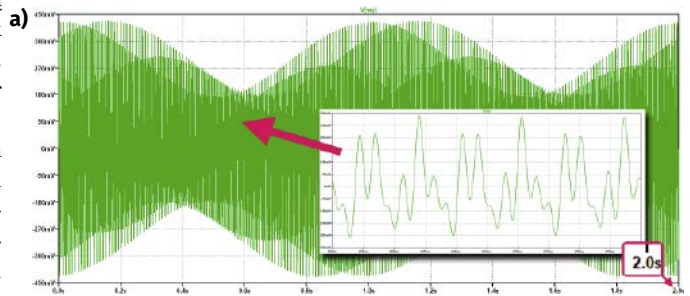
Zrozumienie związku między interpretacją jakiegoś zjawiska przedstawionego w dziedzinie czasu i w dziedzinie częstotliwości prawdopodobnie stwarza przeciętnemu człowiekowi podobny problem jak zrozumienie związku między szybkością a przyspieszeniem. Dla przykładu proponuję mały test polegający na narysowaniu wykresu przyspieszenia dla obiektu ruszającego z miejsca, poruszającego się przez chwilę ze stałą prędkością, a następnie hamującego do zera. Rozwiązanie na końcu artykułu.

Wracamy jednak do zagadnień związanych z analizą widmową. Rozpatrzmy czysty sygnał sinusoidalny. Jego interpretacja w dziedzinie czasu jest funkcją ciągłą (rysunek 1a), zaś w dziedzinie częstotliwości będzie to pojedynczy prążek odpowiadający częstotliwości sinusoidy (rysunek 1b). Teraz przeanalizujemy pewien okształcony przebieg okresowy, który zostanie poddany próbkowaniu. Przebieg okształcony składa się w naszym przykładzie z określonej liczby składowych, które na wykresie widmowym wystąpią jako odrębne prążki. Założmy, że w rozpatrywanym przypadku mamy składowe: 200 Hz, 201 Hz, 500 Hz i 700 Hz, wszystkie o jednakowych amplitudach. Analiza przebiegu w dwóch etapach. Pierwszy obejmuje gromadzenie N próbek w rekordzie akwizycji. Wiemy już, że potrzebny jest do tego czas T . Z transformacji Fouriera wynikają ważne wnioski: po pierwsze – obliczone widmo nie będzie ciągłe, po drugie – jeśli w rekordzie akwizycji znajdzie się N próbek, to całe widmo będzie się składało z N dyskretnych częstotliwości. Kolejnym ważnym wnioskiem jest to, że odstęp między dyskretnymi wartościami częstotliwości widma, czyli rozdzielczość, jest równa odwrotności czasu akwizycji (czasu pełnego zapelnienia bufora). Na wykresach widma, które są domyślnie rysowane w skali logarytmicznej, jest to najmniejsza widoczna na wykresie częstotliwość. Oznaczmy ją jako f_1 lub r . W analizatorach widma parametr ten jest oznaczany zwykle jako RBW (Resolution Bandwidth).

Przeprowadźmy teraz eksperyment, w którym wyświetlimy półsekundowy fragment naszego sygnału okształconego. Zgodnie z powyższymi wnioskami uzyskamy rozdzielczość widmową $R=1/0,5=2$ Hz. Prawdopodobnie więc przy takich parametrach akwizycji danych nie będziemy w stanie rozróżnić składowych 200 Hz i 201 Hz badanego sygnału. Na rysunku 2a pokazano półsekundowy wykres czasowy przebiegu okształconego, a na rysunku 2b jego widmo wyznaczone



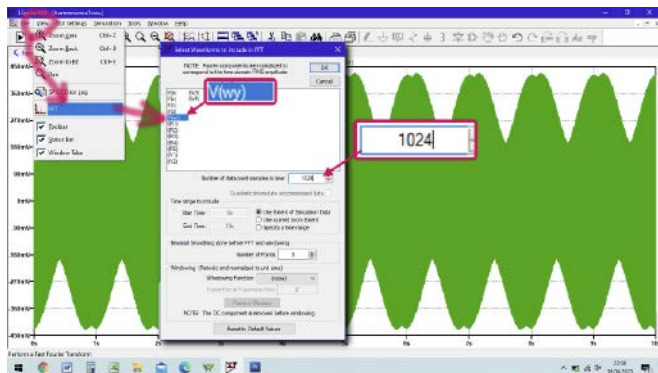
Rysunek 2. Półsekundowy przebieg okształcony a) w dziedzinie czasu, b) w dziedzinie częstotliwości



Rysunek 3. Dwusekundowy przebieg okształcony a) w dziedzinie czasu, b) w dziedzinie częstotliwości

przez symulator LTspice. Dodatkowo na rysunku tym możemy przyrzeć się powiększonemu fragmentowi widma w okolicach 200 Hz. Rzeczywiście prążka 201 Hz nie widzimy na tym wykresie. Wydłużmy więc czas akwizycji do 2 sekund. Pozwoli to uzyskać rozdzielczość widma równą 0,5 Hz, a więc prążki 200 Hz i 201 Hz powinny być już widoczne. Wykresy ilustrujące ten przypadek zostały pokazane na rysunku 3. Jak widać, rzeczywiście można na nich rozróżnić składowe widma o częstotliwościach 200 Hz i 201 Hz, chociaż ze względu na niewielką rozdzielczość są one wyraźnie rozmyte.

Spróbujmy teraz zbadać jak na wynik analizy FFT wpływa liczba próbek. W przypadku symulacji parametr ten ma znaczenie nieco abstrakcyjne, ponieważ tak naprawdę nie wiemy jak wygląda proces próbkowania realizowany przez symulator. W pomiarach oscyloskopowych zwykle lepiej wiemy jak duży jest rekord akwizycji, znana jest też szybkość pracy układu akwizycji. Są to parametry podawane w specyfikacji technicznej urządzeń. W programie LTspice argument określający liczbę próbek nosi nazwę „Number of data point samples in time”. Domyślnie jest on równy 262144, i taka właśnie wartość jest przyjmowana przy każdym uruchomieniu analizy FFT.



Rysunek 4. Okno uruchamiania obliczeń FFT z polem określającym liczbę elementów analizy

Sprawdźmy na przykładzie przebiegu rozpatrywanego wcześniej, jak liczba próbek użyta do analizy FFT w symulatorze LTspice wpływa na wynik. Załóżmy, że chcemy uzyskać rozdzielczość widma równą 0,1 Hz, konieczny do tego czas pomiaru jest więc równy $1/0,1=10$ s. Wykonamy pomiary dla liczby próbek równych 2048, 4096, 8192, 16384, 32768 i 65536.

Do rozpoczęcia symulacji FFT aktywne musi być okno wykresu czasowego. Wybieramy polecenie „View → FFT”, wskazujemy interesujący nas węzeł i wprowadzamy ewentualnie odpowiednią wartość w polu „Number of data point samples in time” (rysunek 4). Ciekawostką może wydawać się fakt, że wprowadzana liczba próbek nie musi być w symulatorze LTspice potęgą liczby 2. Wyjaśnienia, jak ten problem rozwiązano w programie LTspice raczej nigdzie nie znajdziemy, gdyż jest to tajemnica handlowa Linear Technologies. Prawdopodobnie brakujące próbki do najbliższej liczby spełniającej warunek $N=2^k$, są dodawane sztucznie na podstawie interpolowanych wartości przebiegu czasowego, albo uzupełniane wartościami stałymi. Wyniki symulacji zostały pokazane na rysunkach 5a...5f.

Od liczby próbek zależy maksymalna częstotliwość widoczna na wykresie obliczonego widma. Można więc tym parametrem regulować pośrednio zakres analizy. Wiemy już, że minimalna częstotliwość jest równa odwrotności czasu symulacji, górna zaś jest równa:

$$f_2 = \frac{N}{2} \cdot \frac{1}{T} - f_1 = \frac{N}{2} \cdot \frac{1}{T} - \frac{1}{T} = \frac{N-2}{2T}$$

gdzie:

N – liczba próbek (Number of data point samples in time),

T – czas symulacji,

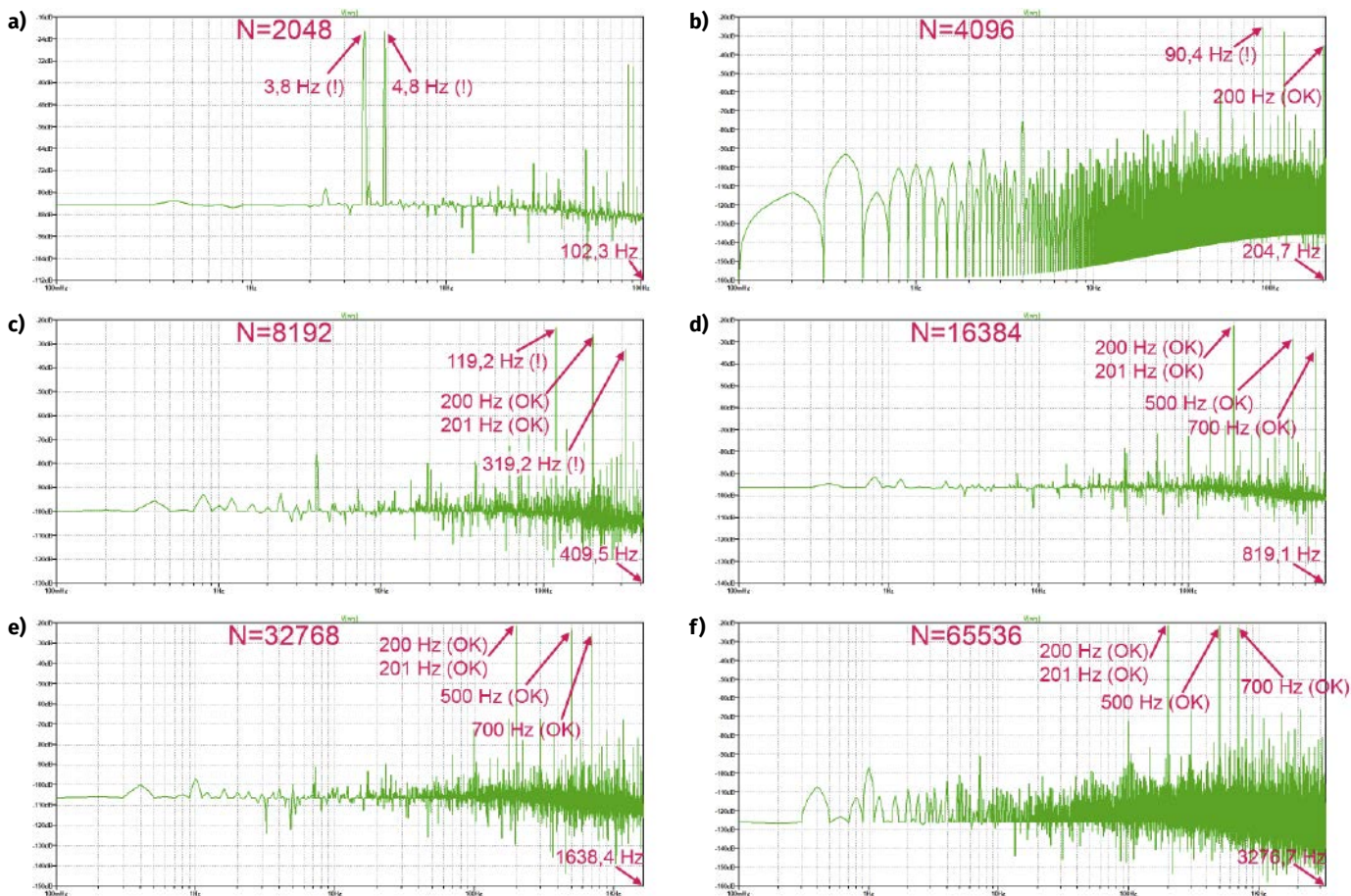
f_1 – częstotliwość początkowa widma,

f_2 – częstotliwość końcowa.

Jak wiadomo w procesie próbkowania sygnału analogowego może wystąpić zjawisko aliasingu. Będzie tak, jeśli nie zostanie spełniony warunek określony w twierdzeniu o próbkowaniu Nyquista-Shanona. Mówi on, że częstotliwość próbkowania musi być co najmniej 2 razy większa od maksymalnej częstotliwości występującej w sygnale, Warto więc tę częstotliwość znać. W symulatorze jest to parametr wirtualny, ale jego znaczenie jest dokładnie takie samo jak w przypadku próbkowania rzeczywistych sygnałów na przykład w oscyloskopie. Wirtualną częstotliwość próbkowania w programie LTspice można wyznaczyć ze wzoru:

$$f_p = \frac{N}{T}$$

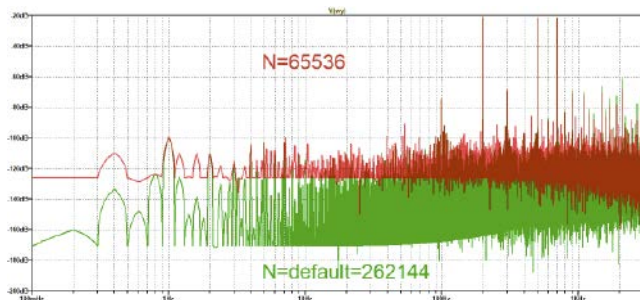
Wróćmy jeszcze raz do przykładu omawianego wcześniej. Początkowo ustalaliśmy liczbę próbek równą 2048. Przy czasie symulacji równym 10 s dawało to częstotliwość próbkowania równą 204,8 Hz. Takie parametry nie mogą dać prawidłowego wyniku analizy FFT, gdyż nie jest spełniony warunek określony w twierdzeniu o próbkowaniu. Maksymalna częstotliwość musiałaby być 2-krotnie mniejsza od częstotliwości próbkowania, tymczasem składowe rozpatrywanego przebiegu zaczynają się od 200 Hz. Choć na wykresie występują blisko siebie dwa prążki (rysunek 5a), to są one wynikiem aliasingu i mają częstotliwości 3,8 Hz ($204,8-201=3,8$) i 4,8 Hz ($204,8-200=4,8$)



Rysunek 5. Wyniki symulacji przebiegu odkształconego a) N=2048, b) N=4096, c) N=8192, d) N=16384, e) N=32768, f) N=65536

Tabela 1. Zestawienie obejmujące częstotliwość początkową f_1 i końcową f_2 wykresu FFT oraz wirtualną częstotliwość próbkowania f_p w zależności od liczby próbek N użytych do obliczeń przy stałym czasie akwizycji $T=10$ s, f_{max} – maksymalna częstotliwość, która może wystąpić w sygnale

N	$f_1=R$	f_2	f_p	f_{max}
16	0,1	0,7	1,6	0,8
32	0,1	1,5	3,2	1,6
64	0,1	3,1	6,4	3,2
128	0,1	6,3	12,8	6,4
256	0,1	12,7	25,6	12,8
512	0,1	25,5	51,2	25,6
1024	0,1	51,1	102,4	51,2
2048	0,1	102,3	204,8	102,4
4096	0,1	204,7	409,6	204,8
8192	0,1	409,5	819,2	409,6
16384	0,1	819,1	1638,4	819,2
32768	0,1	1638,3	3276,8	1638,4
65536	0,1	3276,7	6553,6	3276,8
131072	0,1	6553,5	13107,2	6553,6
262144	0,1	13107,1	26214,4	13107,2
524288	0,1	26214,3	52428,8	26214,4



Rysunek 6. Porównanie szumu w obliczeniach FFT dla N=65536 i N=262144

zamiast 200 Hz i 201 Hz. W analizie naszego przebiegu pozbywamy się aliasingu dopiero przy liczbie próbek równej 16384 (trzymając się jednak zasady, że liczba próbek jest potęgą liczby 2). Wobec tego „odkrycia” przestaje więc dziwić domyślna, wydawać by się mogło, bardzo duża liczba próbek programu LTspice. W tabeli 1 zestawiono częstotliwość początkową f_1 i końcową f_2 wykresu FFT oraz wirtualną częstotliwość próbkowania f_p w zależności od liczby próbek N użytych do obliczeń. Przyjęto stały czas akwizycji $T=10$ s. W zestawieniu uwzględniono ponadto teoretycznie maksymalną częstotliwość f_{max} , która może wystąpić w sygnale. Parametr „Number of data point samples in time” wpływa ponadto na poziom szumów wyświetlanych na wykresie widma. Zwiększenie liczby próbek zmniejsza poziom szumów, co zilustrowano na rysunku 6. W pokazanym na nim eksperymencie pomiar widma był uruchomiony z domyślną liczbą próbek ($N=262144$ – wykres zielony), a następnie z ręcznie wprowadzoną wartością $N=65536$ (wykres czerwony). W celu łatwiejszego porównywania wykresów wprowadzono następnie takie same zakresy na obu osiach.

Wpływ kompresji danych i precyzji obliczeń na symulację FFT

Program LTspice w trakcie wykonywania obliczeń, m.in. FFT, tworzy pliki dyskowe takie jak *.log, *.net, *.raw. W plikach *.raw są zapisywane dane związane z wykresem czasowym. Dane te są następnie

używane do obliczeń FFT. W zależności od stopnia złożoności układu symulowanego i parametrów symulacji, w szczególności czasu symulacji i liczby przebiegów, pliki te mogą osiągać bardzo duże wielkości. Zwykle po zakończeniu pracy z programem nie są one potrzebne i w zależności od opcji „Save .raw, .plt and .log data file in a specific directory” mogą być kasowane lub nie. Warto tę opcję wyłączyć, aby nie zaśmiecać dysku zbędnymi plikami. Nie jest jednak możliwa praca symulatora bez tworzenia tych plików. Łatwo domyślić się, że obsługa dużych plików będzie wpływała niekorzystnie na szybkość działania programu. Autorzy zastosowali więc domyślną ich kompresję, niestety stratną. Może to skutkować pewnymi niedokładnościami podczas obliczeń FFT, ponieważ funkcja ta korzysta z danych zapisanych w plikach *.raw. Jeśli więc planujemy obliczanie widma, warto kompresję wyłączyć umieszczając na pulpicie symulatora opcję:

```
.options plotwinsize=0
```

Spowolni ona działanie symulatora, ale może poprawić efekt obliczeń. Zwykle zastosowanie tej opcji skutkuje obniżeniem poziomu szumów. Efekt ten może być wzmocniony zwiększeniem precyzji obliczeń np. po zastosowaniu opcji:

```
.options numdgt=7
```

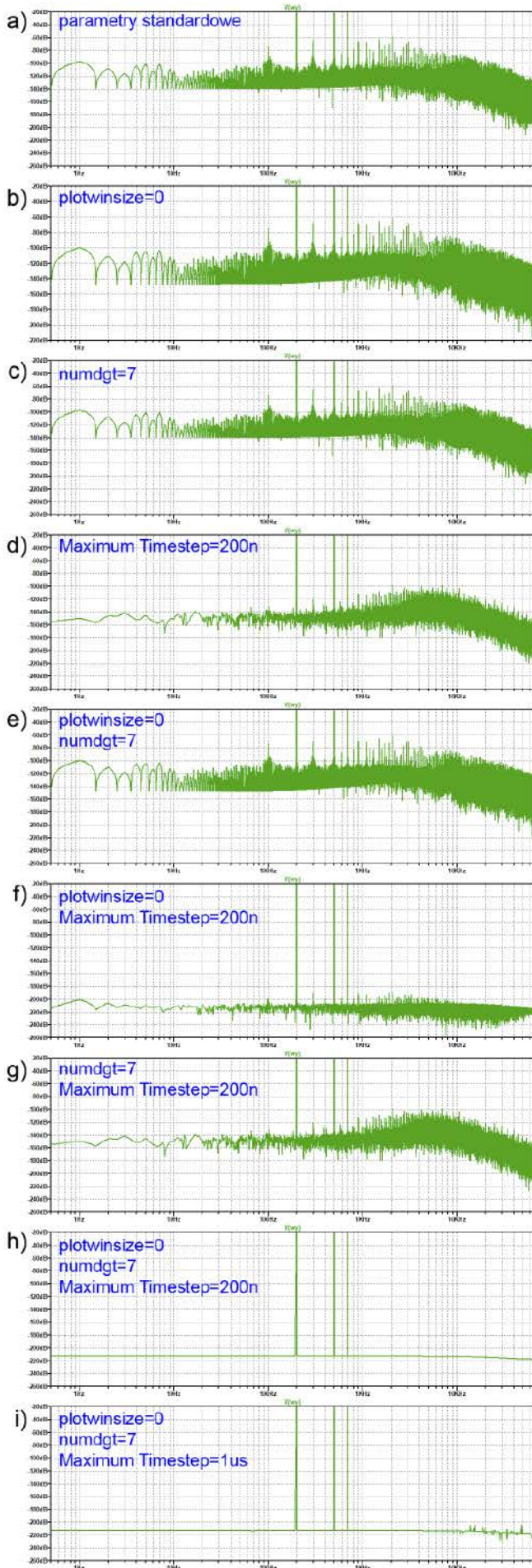
Parametr `numdgt>6` powoduje włączenie podwójnej precyzji obliczeń, co oczywiście skutkuje dalszym zwiększeniem wielkości plików *.raw.

Należy pamiętać, że ustawianie w oknie „Control panel” (ikonka z młotkiem) parametrów związanych z kompresją nie daje stałego efektu. Wprowadzone wartości są ważne tylko w danej sesji pracy z LTspice. Wyjście z programu i jego ponowne uruchomienie przywraca parametry domyślne. Jeśli zależy nam na trwałym wyłączeniu kompresji, to jest to możliwe tylko dla konkretnej symulacji zapisanej w pliku *.asc po wcześniejszym umieszczeniu na schemacie wymienionych wcześniej opcji.

Maximum timestep

Wymienione środki optymalizacji analizy FFT nie są jedynymi. Co więcej nie są najbardziej skuteczne. Jak już było wcześniej powiedziane nie wiemy dokładnie jak działa algorytm obliczania FFT zastosowany w symulatorze LTspice. Wiadomo, że wynik analizy widmowej zależy od liczby próbek, ale musimy pamiętać, że program nie dysponuje fizycznym przebiegiem, który może próbować w zależności od parametrów jakie się temu procesowi zada. Wejściowy przebieg poddawany analizie w symulatorze jest tworzony cyfrowo, a dodatkowo standardowo jest jeszcze kompresowany. Istnieje podejrzenie, że przy zadanej określonej liczbie próbek analizy FFT w przypadku, gdy ich braknie podstawowy bufor danych jest rozszerzany poprzez interpolację. Istnieje jednak możliwość fizycznego zwiększenia zasobu danych przez ustalenie maksymalnego kroku czasowego na etapie generowania wszystkich przebiegów w symulacji TRANSCIENT. Jest to opcja „Maximum Timestep” komendy .TRAN. Musimy jednak zgodzić się na drastyczne wydłużenie czasu obliczeń po zastosowaniu wszystkich opcji. Może jednak w określonych sytuacjach warto je stosować. Z pewnością ciekawe będzie porównanie możliwych kombinacji opcji stosowanych do obliczeń FFT. Wykonajmy takie porównanie. Zbadamy przebieg zastosowany w pierwszej symulacji. Wiemy, że występują w nim dwie składowe oddalone od siebie i 1 Hz (200 Hz i 201 Hz), przyjmijmy więc minimalną rozdzielczość pozwalającą rozróżnić obie składowe. Niech to będzie 0,5 Hz. Wynikający z tego założenia czas symulacji jest równy $1/0,5$ s=2 s.

Gdybyśmy niezależnie badali znaczenie takich parametrów jak: liczba próbek (standardowa lub ustalona), precyzja (standardowa lub podwójna), kompresja (włączona lub wyłączona) i maksymalny krok czasowy (standardowy lub ustalony), otrzymalibyśmy 16 wyników. Zachęcam Czytelników do przeprowadzenia własnych analiz, w artykule zostaną zaprezentowane tylko wybrane. Zrezygnujemy z badania wpływu liczby próbek na wynik analizy FFT, co ograniczy liczbę kombinacji do 8. We wszystkich pokazanych wykresach



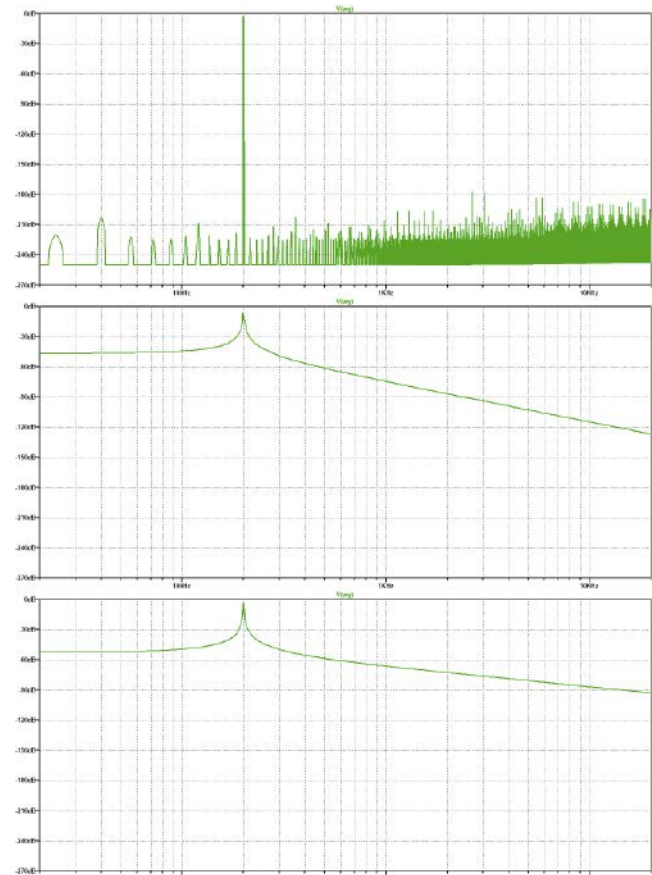
Rysunek 7. Porównanie wpływu parametrów plotwinsize, numdgt i Maximum Timestep na wyniki obliczeń FFT

zapewniono jednolity zakres na osi pionowej. Punktem odniesienia jest analiza ze standardowymi parametrami. Jak widać, wyłączenie kompresji i zwiększenie precyzji obliczeń wpływa zwykle korzystnie w mniejszym lub większym stopniu na jakość wyniku analizy FFT, ale największe znaczenie ma jednak parametr Max Timestep polecenia .TRANSIENT. Decyduje on o gęstości kroku czasowego podczas obliczania zależności czasowych, a zatem również o ich dokładności. Parametry domyślnie zoptymalizowano tak, aby uzyskać kompromis między czasem wykonywania obliczeń a ich dokładnością. Jeśli bardzo nam zależy na czystości analizy FFT warto poeksperymentować z tym parametrem. Trudno jednak udzielić uniwersalnej porady w tym względzie. Przykładowo, w naszym eksperymencie drobne artefakty na wykresie FFT zaczynają się pojawiać dla maksymalnego kroku czasowego ok. 1 μ s (rysunek 7i).

Wyciek widma

Transformacja Fouriera daje w wyniku ciągle widmo od minus nieskończoności do plus nieskończoności, zakłada też, że przebieg jest okresowy. Obliczenia transformacji dyskretnej ograniczają się do ściśle określonego czasu – okna. Jeśli w tym czasie zmieści się pełna krotność okresu badanego sygnału, obliczenia można uznać za poprawne (z pominięciem efektów wynikających z dyskretyzacji procesu). Jeśli jednak czas obliczeń obejmuje pewną liczbę pełnych okresów plus „wiszący” fragment kolejnego okresu, dochodzi do tzw. wycieku widma. Na wykresie widma zobaczymy oczywiście prążek odpowiadający częstotliwości sygnału, ale będzie on rozmyty i jakby rozciągnięty na sąsiednie częstotliwości. Sprawdźmy ten efekt praktycznie.

Tym razem rozpatrujemy czysty sygnał sinusoidalny o amplitudzie 1 V i częstotliwości np. 200 Hz. Okres tego sygnału jest równy 5 ms. Załóżmy, że analiza FFT powinna objąć dokładnie 100 okresów,

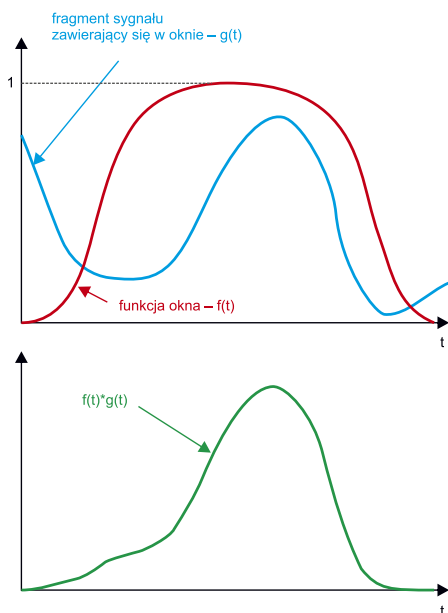


Rysunek 8. Wykresy FFT dla przypadków bez i z wyciekem widma – częstotliwość sygnału 200 Hz a) czas symulacji T=500 ms (obejmuje pełną liczbę okresów przebiegu) – brak wycieku widma, b) czas symulacji T=502,5 ms dodana połowa okresu – wyciek widma, c) czas symulacji T=501,25 ms dodane ćwierć okresu – wyciek widma

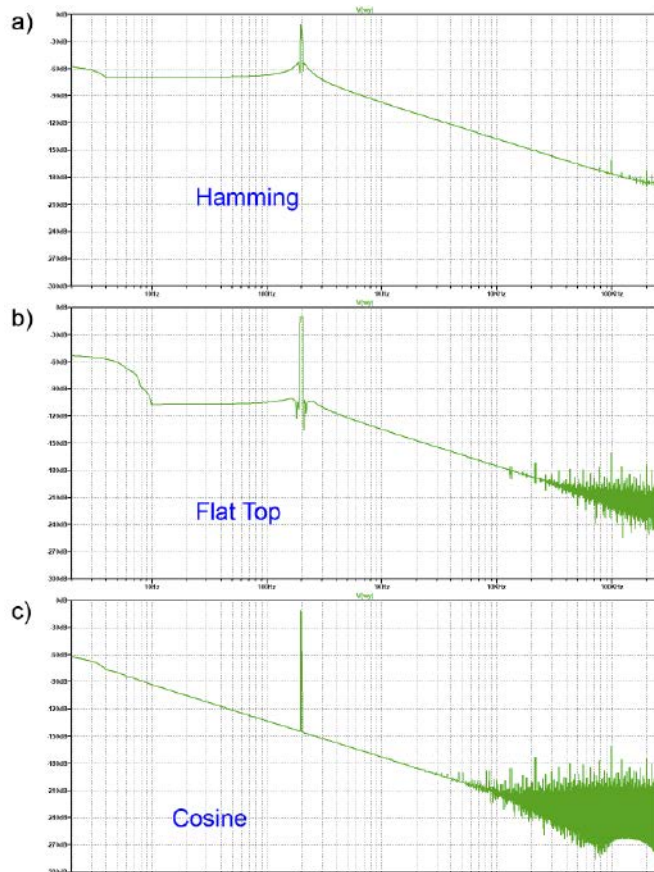
powinniśmy zatem uruchomić symulację TRANSIENT z czasem 500 ms. Dla zwiększenia dokładności obliczeń przyjmujemy ponadto maksymalny krok czasowy 10 μ s, włączymy podwójną precyzję obliczeń i wyłączymy kompresję danych. Analizę FFT prowadzimy ze standardową liczbą próbek (262144). Wynik obliczeń został pokazany na **rysunku 8a**. Ograniczamy zakres wyświetlanych częstotliwości do 20 Hz...20 kHz, Teraz do okna czasowego dodajemy pół okresu przebiegu, czas symulacji wydłużamy zatem do 502,5 ms i tak jak w poprzednim przypadku ograniczamy zakres wyświetlania widma do takiego samego zakresu (**rysunek 8b**). Warto jeszcze sprawdzić, jak będzie wyglądał wykres widma dla przypadku, w którym czasowe okno analizy zostanie rozszerzone o ćwierć okresu. Czas symulacji jest więc równy 501,25 ms. Skutki wycieku widma są nadal widoczne, chociaż rozmycie jest o kilka decybeli niższe.

Nietrudno wyobrazić sobie, że w świecie rzeczywistym spełnienie warunku, aby w oknie czasowym mieścił się pełny okres przebiegu jest praktycznie niemożliwe. Najczęściej w ogóle nie mierzymy przebiegu okresowego, tylko złożenie wielu przebiegów zupełnie nieskorelowanych ze sobą. Praktycznie więc zawsze stosując analizę FFT będziemy mieć do czynienia z wyciekami widma. Jak więc poradzić sobie z minimalizacją skutków tego efektu?

Rozwiązanie oczywiście istnieje, chociaż siłą rzeczy musi wprowadzać pewną deformację cyfrowej postaci przebiegu oryginalnego. Jeśli transformacja Fouriera zakłada okresowość sygnału, metoda eliminacji wycieku widma powinna w jakiś sposób tak przekształcić sygnał oryginalny, aby algorytm obliczający FFT „myślał”, że ma do czynienia z sygnałem okresowym. Cechą funkcji okresowej jest to, że wartości tej funkcji na początku okresu i na końcu są takie same (przy założeniu, że punkt końcowy okresu jest jednocześnie punktem początkowym kolejnego). Jeśli sygnał przed poddaniem go analizie FFT wymnożymy przez jakąś funkcję, która na krańcach okna będzie równa zero, to uzyskamy efekt, o którym była mowa. Na **rysunku 9** zilustrowaną ten pomysł. Niestety, ta operacja zniekształca przebieg oryginalny, ale zwykle poprawia skutki wycieku danych z okna czasowego analizy FFT. Funkcje okna mogą być różne. W praktyce istnieje kilka najczęściej używanych, które powinny być dobierane w zależności od potrzeb. Często oknom tym nadawane są nazwy pochodzące od nazwisk ludzi, którzy je zdefiniowali. Poszczególne typy okien różnie wpływają na pomiar amplitudy i częstotliwości. Przykładowo, okno Hamminga zapewnia dobrą rozdzielczość częstotliwościową, ale gorszy pomiar amplitudy. W przeciwieństwie do niego okno Flat Top zapewnia dobry pomiar amplitudy przy słabszym określeniu



Rysunek 9. Ilustracja zasady działania okna czasowego w obliczeniach FFT



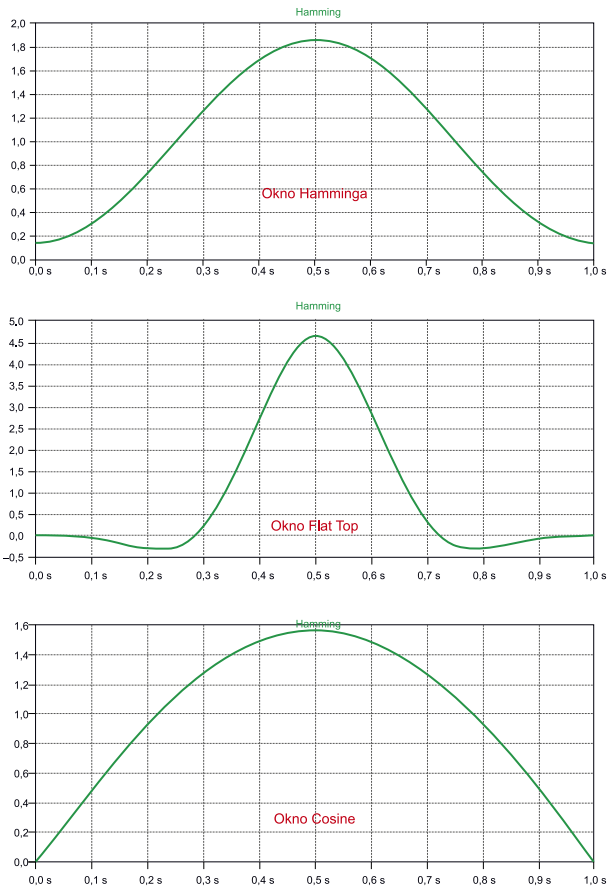
Rysunek 10. Wynik analizy FFT z zastosowaniem różnych okien czasowych a) okno Hamminga, b) okno Flat Top, c) okno Cosine

częstotliwości. Zobaczmy, czy i jak poprawią się wykresy widma z rysunku 8 po zastosowaniu kilku najczęściej stosowanych okien. Zachowamy parametry symulacji z wyciekami połowy okresu w oknie czasowym. Uruchamiamy więc symulację `.tran 0 502.5m 0 10u`, a po wykreśleniu przebiegu czasowego uruchamiamy 3-krotnie analizę FFT, za każdym razem zmieniając typ okna (*Windowing Function*). Na **rysunku 10** pokazano działanie okna Hamminga, Flat Top i Cosine, a na **rysunku 11** pokazano każde z tych okien.

Pomiar sygnału rzeczywistego

Do tej pory rozpatrywaliśmy symulację całego toru pomiarowego włącznie ze źródłem sygnału. Spróbujmy teraz włączyć do symulacji element świata rzeczywistego. W tym celu korzystając z programu Audacity nagrany sygnał eksportujemy w formacie WAV do pliku `warkot.wav`, a następnie opisywanymi już we wcześniejszych odcinkach cyklu metodami importujemy w symulatorze. Nagranie trwa 4 sekundy. Schemat pomiarowy, wykres czasowy i analizę FFT pokazano na **rysunku 12a**. Widmo obliczono bez żadnego okna czasowego. Na **rysunku 12b** widzimy natomiast widmo tego samego sygnału obliczone w programie Audacity. Można odnotować dużą zgodność obu wykresów.

Pozostaje jeszcze wyciągnięcie jakichś wniosków z tej obserwacji. Można stwierdzić, że dość ewidentne maksimum sygnału występuje dla zakresu częstotliwości ok. 44...46 Hz, co odpowiadałoby obrotom 2640...2760 rpm. To bardzo dużo. Należałoby się spodziewać maksymalnych obrotów rzędu 1400...1600 rpm. Wentylator był jednak zamknięty w urządzeniu, być może rezonowała obudowa, możliwe również, że wyższa wysokość dźwięku wynika z tego, że dźwięk jest generowany nie tylko na skutek drgań wirnika, ale również przez obracające się skrzydełka, których jest kilka. Mamy więc pewien problem interpretacyjny. Aby jakoś go rozwiązać wykonałem dodatkowy pomiar innym narzędziem z zastosowaniem akcelerometru.

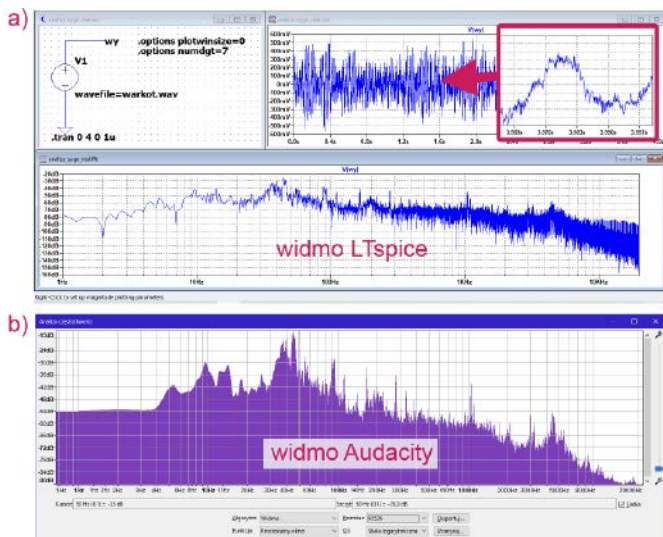


Rysunek 11. Przykładowe okna czasowe analizy FFT

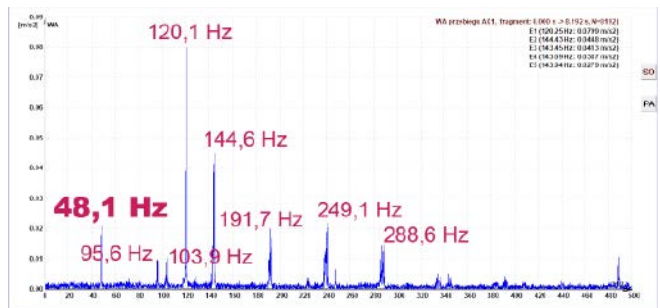
Hmmm... zamiast rozstrzygnąć wątpliwości pojawiły się kolejne. Owszem na wykresie widoczny jest wyraźny prążek w okolicach 48 Hz (rysunek 13), ale maksimum widma wypada na ok. 120 Hz. Odpowiedź może być tylko jedna. Nie mierzymy tylko prędkości obrotowej wentylatora, lecz również bicie jego łożyska, co do kiepskiej jakości którego faktycznie nie mam wątpliwości.

A co by było gdyby...

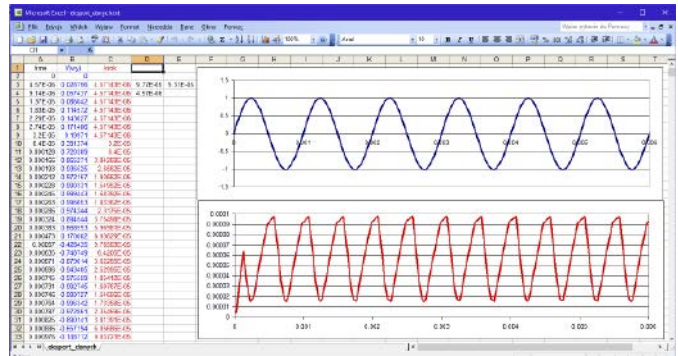
Zapewne uważni Czytelnicy zadali sobie pytanie: a co by się stało, gdyby wynik analizy FFT poddać analizie FFT? Nawet nie wnikając w zawiłości matematyczne można sądzić, że przeprowadzimy w ten



Rysunek 12. Pomiar i analiza sygnału akustycznego zarejestrowanego mikrofonem, a) układ pomiarowy zastosowany w symulatorze wraz z wykresem czasowym przebiegu badanego, b) analiza FFT w programie LTspice, c) analiza FFT w programie Audacity



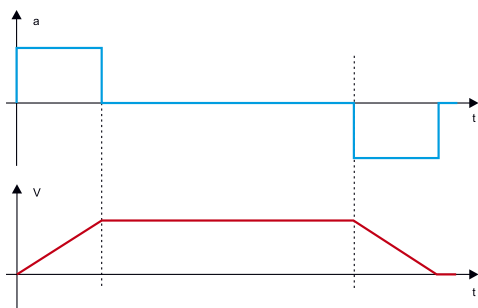
Rysunek 13. Analiza widmowa zjawiska pokazanego na rysunku 12 z zastosowaniem akcelerometru



Rysunek 14. Wykres zmienności kroku w analizie czasowej LTspice

sposób odwrotną transformację Fouriera, czyli przejście z dziedziny częstotliwości na dziedzinę czasu. Nie jest to operacja wykonywana często, gdyż zwykle nie dysponujemy narzędziem, które dostarczałoby widmo wprost, bez wcześniejszego obliczenia go z przebiegu czasowego. Nie ma więc potrzeby przeprowadzania operacji odwrotnej, gdyż i tak odpowiednimi danymi dysponujemy. Jest jednak pewna dość nieoczywista sytuacja, w której odwrotna transformacja może być bardzo przydatna.

LTspice ma jedną, być może dość zaskakującą wielu użytkowników cechę ujawniającą się podczas tworzenia przebiegów czasowych. Załóżmy, że chcielibyśmy zastosować wygenerowany w symulatorze przebieg czasowy do obliczeń w jakimś zewnętrznym programie, np. w Excelu. Rysujemy schemat, uruchamiamy symulację TRANS i uzyskujemy wykres czasowy. Teraz eksportujemy dane tego wykresu do pliku tekstowego. Uaktywniamy okno wykresu i wybieramy polecenie: „File → Export data as text”. W oknie tego polecenia wybieramy lokalizację i nazwę pliku, np. *sinus.txt*. Otwieramy Excel i z menedżera plików przeciągamy plik *sinus.txt* do pustego obszaru roboczego Excela. Można dla wygody zamienić wykładniczą postać danych na normalną. Dane wprowadzone do arkusza nadają się do wszelkiego rodzaju obliczeń. Można na przykład wykonać wykres czasowy przebiegu. Gdybyśmy jednak chcieli analizować dane próbka po próbce, moglibyśmy napotkać na pewien niewygodny problem. Przeanalizujmy krok próbkowania zastosowany przez LTspice. W tym celu w polu C3 zapisujemy formułę: A3-A2 i kopiujemy tę komórkę w całej kolumnie C aż do końca danych. Patrzymy na dane... i okazuje się, że przy dłuższych czasach symulacji symulator nie używa przy tworzeniu wykresu czasowego stałego kroku próbkowania. Krok ten zmienia się cyklicznie, co zilustrowano na rysunku 14. Gdy przebieg wypłaszcza się (pochodna zbliża się do zera), symulator zwiększa krok czasowy. Niestety, taki niestały krok obliczeń może utrudniać pewne operacje matematyczne. Metodą poprawienia tej niewygodnej własności symulatora jest dwukrotne przeprowadzenie transformacji Fouriera, czyli FFT na przebiegu czasowym i FFT na widmie. W rezultacie otrzymujemy ponownie przebieg czasowy, ale jeśli wykonamy eksport danych tak otrzymanego wykresu czasowego do Excela, tak jak to robiliśmy wcześniej, to okaże się, że tym razem krok czasowy jest już stały.



Rysunek 15. Rozwiązanie zagadki

Inne opcje analizy FFT

Dla porządku należy jeszcze wspomnieć o innych opcjach obliczeń FFT w symulatorze LTspice. Do tej pory była mowa o tym, że do obliczania FFT jest wymagana liczba próbek będąca potęgą liczby 2. Algorytm zastosowany w programie LTspice omija jednak ten problem. Nie wiemy jednak jak, gdyż Linear Technologies tego nie ujawnia. Sądzymy, że brakujące dane są dodawane metodą interpolacji. Cechy te umożliwiają dość dowolne definiowanie zakresu obliczania transformaty FFT. Można więc liczyć ją z całego zakresu czasowego symulacji, można wskazać konkretny przedział czasowy wprowadzając numeryczne dane „Start time” i „Stop time”, można też obliczać

FFT z fragmentu widocznego na ekranie po zastosowaniu lupki powiększającej (zoom).

Ostatni trik, jakiego można używać podczas obliczeń FFT to uśrednianie dwumianowe („*Binomial Smoothing done before FFT and windowing*”). Opcja ta jest domyślnie włączona i operuje na trzech próbkach, ale zakres uśredniania można zmieniać. Zabieg teoretycznie powinien minimalizować szum, ale jego działanie jest bardzo subtelne.

Podsumowanie

Analiza FFT jest doskonałym narzędziem pozwalającym na diagnozowanie wielu nieprawidłowości działania różnych układów i urządzeń. Nieumiejętne jej stosowanie może jednak prowadzić do wyciągania mylnych wniosków. Największym wrogiem tego narzędzia jest kompresja danych i dodawanie brakujących danych przez interpolację. Prowadzi to do rozmywania się widma i dodawania składników, które nie wystąpiłyby w równoważnym sygnale rzeczywistym. Doświadczony elektronik powinien wyłapywać większość niepożądanych artefaktów, ale nie zawsze jest to oczywiste. Jednym z lepszych środków zaradczych jest ograniczenie kroku czasowego, co jednak bardzo niekorzystnie wpływa na szybkość obliczeń.

Jarosław Doliński, EP

REKLAMA



KOMPUTERY RASPBERRY PI I MODUŁY ARDUINO



AVT SPV Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 11
Sklep stacjonarny czynny jest od poniedziałku do piątku w godzinach: 8:00–16:00, tel. 22 257 84 66
e-mail: handlowy@avt.pl, tel. 22 257 84 51 (w godzinach: 9:00–15:00)

<http://sklep.avt.pl>