

Stanisław MIKULSKI*

STANOWISKO LABORATORYJNE DO CYFROWEGO PRZETWARZANIA SYGNAŁÓW Z WYKORZYSTANIEM ŚRODOWISKA MATLAB ORAZ PLATFORMY PROGRAMISTYCZNEJ .NET

W artykule opisano stanowisko laboratoryjne służące do cyfrowej analizy i przetwarzania sygnałów elektrycznych. Stanowisko to zostało stworzone w oparciu o oprogramowanie MATLAB, dowolną kartę pomiarową oraz interfejs graficzny programu zaimplementowany w języku C# platformy .NET. Przedstawione zostały także pomiary wykonane za pomocą prezentowanego w artykule układu oraz analiza wpływu zastosowania dwóch środowisk programistycznych na czas uzyskania wyników obliczeń oraz ich wizualizację.

SŁOWA KLUCZOWE: cyfrowe przetwarzanie sygnałów, filtracja cyfrowa, Matlab, C#.NET

1. KONCEPCJA STANOWISKA LABORATORYJNEGO

Głównym założeniem opisanego w artykule stanowiska było skonstruowane narzędzia pozwalającego na rejestrację, przetwarzanie i analizę sygnałów elektrycznych bez wykorzystania do tego zaawansowanych układów opartych o procesory sygnałowe DSP. Zamiast procesorów sygnałowych do obliczeń wykorzystany został komputer klasy PC z zainstalowanym systemem MATLAB/Simulink. Aby stanowisko mogło spełniać zarówno cele naukowe jak i dydaktyczne wyposażone zostało zarówno w symulator generatora funkcyjnego oraz arbitralnego, jak i możliwość komunikacji z kartą pomiarową. W przypadku komunikacji z kartą pomiarową stanowisko miało mieć możliwość zarówno akwizycji sygnału jak i generowania sygnałów na wyjściu karty. Program komputerowy został także wyposażony w możliwość wyświetlania przebiegów czasowych sygnałów, widm częstotliwościowych oraz odczyt podstawowych parametrów sygnału.

Ze względu na funkcjonalność stanowisko można podzielić na cztery moduły:

- moduł akwizycji danych – w skład, którego wchodzi źródło sygnału oraz blok programu sterujący akwizycją danych. W zależności od trybu pracy, jako źródło sygnału można wyróżnić: symulację w środowisku MATLAB, odczyt

* Politechnika Poznańska.

- zarejestrowanego wcześniej przebiegu sygnału zapisanego w formacie wav, kartę dźwiękową komputera lub kartę pomiarową. W trakcie pracy programu komputerowego można zmienić takie parametry jak częstotliwość próbkowania, czas lub liczbę pobranych próbek. W przypadku trybu symulacyjnego można ustawić dodatkowo kształt sygnału wejściowego, amplitudę, częstotliwość, fazę początkową oraz czas symulacji,
- moduł analizy częstotliwościowej – w programie analiza widma częstotliwościowego sygnału odbywa się za pomocą szybkiej transformaty Fouriera FFT, zgodnie z algorytmem Colley-Tuckey'a opisanego w [2, 4],
 - moduł filtracji cyfrowej sygnału – moduł ten pozwala na dobranie współczynników filtru cyfrowego rekursywnego IIR lub nierekursywnego FIR. Do projektowania filtrów FIR wykorzystano metodę okien opisaną w [3]. Wyboru okien można dokonać pomiędzy oknami: prostokątnym, trójkątnym, Bartletta oraz Hamminga,
 - moduł rejestrujący sygnał – pozwala na zapis otrzymanych przebiegów do pliku wave oraz ich późniejsze odtworzenie.

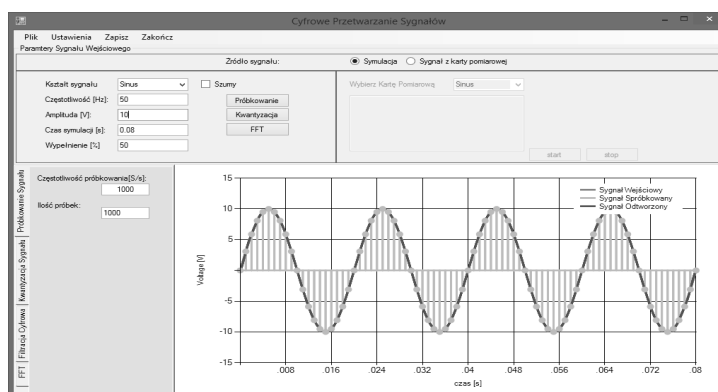
2. REALIZACJA STANOWISKA

2.1. Interfejs programu

Interfejs programu został zaimplementowany w środowisku programistycznym Visual Studio 2012, przy użyciu języka C#. Takie rozwiązanie pozwoliło na stworzenie jednocześnie zaawansowanego i przyjaznego dla użytkownika interfejsu aplikacji. Graficznie interfejs składa się z trzech części. Pierwsza część interfejsu, przedstawionego na rysunku 1, pozwala na wybór jednego z dwóch trybów pracy stanowiska: trybu symulacyjnego lub trybu pomiaru sygnałów za pomocą karty pomiarowej.

Druga część interfejsu pozwala na ustawienie parametrów toru przetwarzania sygnałów, takich jak: częstotliwość próbkowania, parametry filtracji cyfrowej oraz analizy częstotliwościowej sygnałów. W dolnej części interfejsu prezentowane są przebiegi sygnałów (takich jak sygnał wejściowy, wyjściowy) oraz wyniki pomiarów.

Komunikacja pomiędzy aplikacją a środowiskiem Matlab odbywa się za pomoci technologii MATLAB COM Automation Server, zgodnej ze standardem COM. Program w momencie rozruchu tworzy niewidoczną dla użytkownika instancję środowiska MATLAB (tzw. serwer automatyzujący). Wszystkie parametry obliczeń wprowadzone jako dane wejściowe zostają przetłumaczone na komendy zgodne z konstrukcją języka programowania MATLAB-a. Następnie są wysyłane do przestrzeni roboczej (z ang. *Workspace*) serwera automatyzującego w postaci komend lub zmiennych. Po wykonaniu obliczeń wyniki pobierane są z przestrzeni roboczej do pamięci aplikacji za pomocą odnośników do nazw zmiennych.

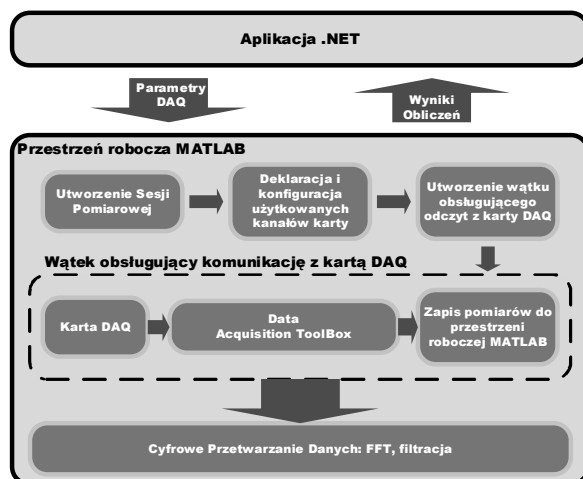


Rys. 1. Interfejs graficzny stanowiska laboratoryjnego

2.2. Komunikacja z kartą pomiarową

Przesyłanie informacji pomiędzy kartą pomiarową, a aplikacją odbywa się pośrednio przez środowisko MATLAB z wykorzystaniem modułu Data Acquisition Toolbox. Cały proces komunikacji przedstawiono na schemacie z rysunku 2. Pomiędzy przestrzenią roboczą a kartą pomiarową tworzona jest sesja pomiarowa. Do jej utworzenia potrzebna jest indywidualna nazwa danej karty pomiarowej w systemie operacyjnym. Po utworzeniu sesji niezbędne jest przypisanie do niej karty pomiarowej oraz wykorzystywanych kanałów karty. Użytkownik może zdefiniować używane kanały oraz ustawienia karty pomiarowej takie jak: częstotliwość próbkowania czy zakres. Sesja pomiarów z karty może się odbywać w trybie skończonym i ciągłym. Po rozruchu sesji pomiarowej karta zapisuje do bufora próbki sygnału. W momencie zapełnienia bufora próbki zostają zapisane do przestrzeni roboczej Matlaba.

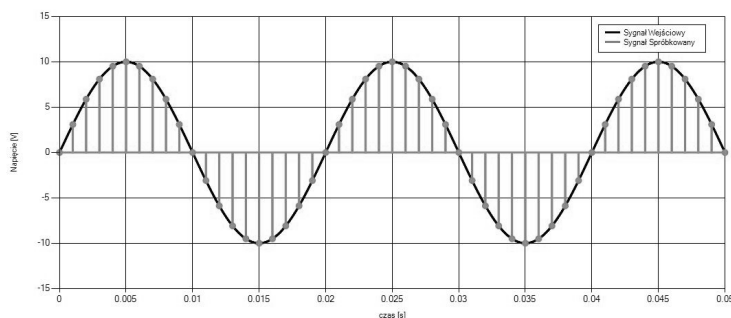
W programie zastosowano zaimplementowaną w Data Acquisition Toolbox wielowątkowość, dzięki czemu zachowano ciągłość komunikacji pomiędzy interfejsem a użytkownikiem. Aby komunikacja pomiędzy kartą pomiarową a przestrzenią roboczą Matlaba odbywała się z wykorzystaniem wątku pobocznego, musi zostać zdefiniowane odwołanie do funkcji wykonywanej w tym wątku w sytuacji, gdy bufor danych zostanie zapełniony. Wielkość bufora danych (liczba pobranych z karty próbek sygnału) jest konfigurowalna. Mechanizm Data Acquisition Toolbox wprowadza jednak ograniczenia dotyczące liczby odczytów i zapisów danych z bufora karty pomiarowej. Skutkiem tego ograniczenia jest brak możliwości przesyłu danych z karty „próbka po próbce”, a więc aplikacja nie jest w stanie działać online. Obliczenia odbywają się cyklicznie, na podstawie pakietów danych.



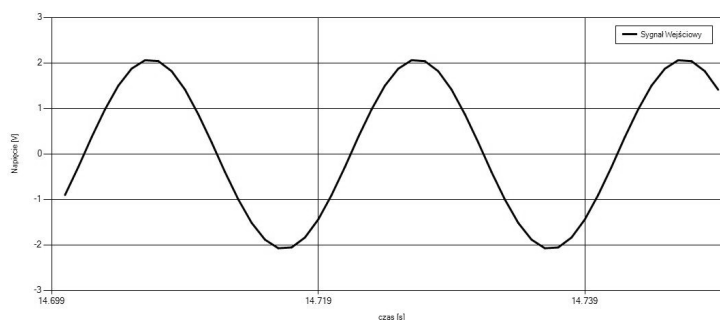
Rys. 2. Schemat komunikacji pomiędzy aplikacją .NET a przestrzenią roboczą Matlab w przypadku pomiarów z wykorzystaniem karty pomiarowej

3. ANALIZA SYGNAŁÓW Z WYKORZYSTANIEM STANOWISKA LABORATORYJNEGO

Badania zostały przeprowadzane na komputerze klasy notebook z procesorem głównym Intel Core-i5 3317U. Układ ten skonstruowany w technologii 22 nm, posiada 3 MB pamięci cache. Częstotliwość taktowania procesora wynosi 1,7 GHz. W jednostce tej zaimplementowana jest także technologia Turbo Boost pozwalająca na zwiększenie częstotliwości taktowania maksymalnie do 2,6 GHz. Notebook wyposażono także w 8 GB pamięci podręcznej oraz kartę graficzną ATI Mobility Radeon HD 8720. Stanowisko wyposażono także w kartę pomiarową firmy National Instruments NI USB-6251. Karta ta posiada 16 wejść analogowych oraz 16 bitowy przetwornik analogowo-cyfrowy. Maksymalna częstotliwość próbkowania każdego wejścia to 1,25 MS/s.

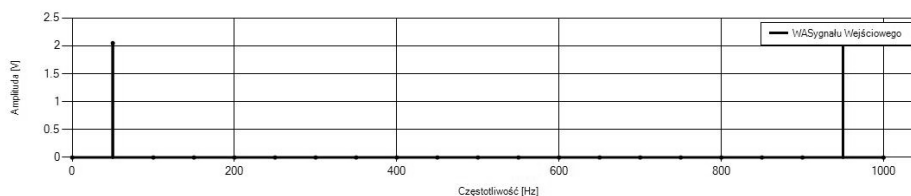


Rys. 3. Wynik symulacji (linia ciągła) oraz dyskretyzacji sygnału wejściowego (prążki)



Rys. 4. Przebieg sygnału zarejestrowanego z karty pomiarowej

Na rysunkach 3 i 4 pokazano dwa zarejestrowane za pomocą aplikacji przebiegi sygnałów. Na rysunku 3 widać przebieg sygnału sinusoidalnego 50 Hz o amplitudzie 2 V otrzymanego w wyniku symulacji. Na rysunek 4 przedstawiono zarejestrowany za pomocą karty pomiarowej sygnał elektryczny o takich samych parametrach (kształt, częstotliwość i amplituda), jak w przypadku sygnału symulowanego na rysunku 3. Dla tego sygnału, przy zachowaniu takiej samej konfiguracji całego stanowiska przeprowadzono analizę widma amplitudowego przy pomocy algorytmów FFT. Wyniki przedstawiono na rysunku 5.



Rys. 5. Widmo amplitudowe sygnału z karty pomiarowej

W trakcie badań częstotliwość próbkowania karty pomiarowej była ustawiona na 1000 S/s.

4. WNIOSKI I PODSUMOWANIE

W artykule przedstawiono przykład praktycznego zastosowania połączenia platformy programistycznego .NET oraz środowiska obliczeniowego Matlab, przy wykorzystaniu interfejsu MATLAB Automation Server.

Oprogramowania Matlab jest jednym z wiodących narzędzi służących do zaawansowanych obliczeń inżynierskich. Zaimplementowana jest w nim szeroka gama narzędzi obliczeniowych, pozwalających na wykonywanie symulacji, analizy oraz przetwarzania cyfrowego m. in. sygnałów, obrazów. Niestety, pomimo bardzo szerokich możliwości obliczeń, interfejs graficzny MATLAB GUI tego środowiska

jest narzędziem o bardzo ograniczonych możliwościach. Dlatego w opinii autora zastosowanie techniki wymiany danych pomiędzy opisanymi w artykule platformami ma swoje uzasadnienie przede wszystkim w zwiększeniu funkcjonalności Matlab'a. Z drugiej strony, w trakcie tworzenia aplikacji na platformie .NET przeniesienie obliczeń do Matlab'a pozwala na ominięcie żmudnego procesu tworzenia algorytmów i kodu obliczeń, co skraca czas tworzenia aplikacji oraz redukuje prawdopodobieństwo popełnienia błędów obliczeniowych.

Po analizie współpracy pomiędzy aplikacją a kartą pomiarową autor stwierdza, że komunikacja między interfejsem graficznym i DAQ odbywała się w sposób płynny i poprawny. Karta zwiększa możliwości stanowiska laboratoryjnego o fizyczny pomiar sygnałów elektrycznych. Dzięki temu otrzymano proste stanowisko do przetwarzania sygnałów cyfrowych wraz z implementacją filtrów cyfrowych. Stanowisko takie może być alternatywą dla specjalistycznych układów z procesorami sygnałowymi. Wadą połączenia karty pomiarowej z Matlabem w stosunku do układów z procesorami sygnałowymi jest ograniczona liczba odczytów próbek z bufora w jednostce czasu.

W ramach artykułu nie został przeanalizowany wpływ komunikacji pomiędzy środowiskiem MATLAB a platformą .NET na czas wykonania obliczeń. Autor planuje dalsze badania w tej tematyce.

LITERATURA

- [1] Oppenheim, A. V. and R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, 1989.
- [2] Cooley, J. W. and J. W. Tukey, "An Algorithm for the Machine Computation of the Complex Fourier Series," *Mathematics of Computation*, Vol. 19, April 1965, pp. 297-301.
- [3] Zieliński T., *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, Warszawa 2007, ISBN 978-83-206-1640-8 2007.
- [4] www.mathworks.com, Fourier Analysis and Filtering, 2014.02.12 15:00
- [5] www.msdn.microsoft.com, Component Object Model, 2014.02.09 13:30
- [6] www.ni.com/white-paper/3215/en/, NI-DAQmx Hardware-Timed Single Point Lateness Checking, 2014.01.20 15:30

IMPLEMENTATION OF DSP LABORATORY STAND IN .NET PLATFORM CONNECTED TO MATLAB

The article describes example of laboratory stand for the analysis and processing of digital signals. This stand has been created with use of the MATLAB Automation Server, newest National Instruments DAQ drivers (NI-DAQmx) and user interface implemented in .NET framework with use of C# language. Stand can work in simulation mode or in data acquisition mode with use of National Instruments Data Acquisition devices connected to computer. The author of the paper describes mechanism of data transmission between two random environments: Matlab Workspace and .NET Framework and show impact of this solution on calculation time.