

Grzegorz Barański*, Rafał Sochaczewski*, Jacek Czarnigowski*

WYKORZYSTANIE PROGRAMU LABVIEW DO BADAŃ UKŁADÓW NAPĘDOWYCH*

WPROWADZENIE

Ciągle rosnący poziom zawansowania technologii współczesnego świata wymaga niemniej zawansowanych badań tych obiektów. Jednocześnie rosną też możliwości pomiarowe. Dostępne są układy pomiarowe o coraz większej precyzji pomiaru, mierzące różne wielkości fizyczne od ciśnienia przez temperaturę do przyspieszeń.

Korzystając z czujników możliwa jest obserwacja zjawisk zachodzących w obiekcie, określenie charakterystyk statycznych i dynamicznych czy też diagnostyka obiektu. Pomiaru takie pozwalają zatem nie tylko na poznanie obiektu ale także na jego sterowanie oraz diagnostykę. Sygnały z czujników pomiarowych mogą mieć charakter: prądowy, napięciowy lub rezystancyjny. Przykładem takiego obiektu jest silnik spalinowy.

Dostępność szerokiej gamy czujników pomiarowych oraz duża różnorodność zadań związanych z pomiarem wymaga odpowiedniego i uniwersalnego systemu pomiarowego. Systemy takie opierają się na elektronicznych układach pomiarowych obsługiwanych przez odpowiednie oprogramowanie. Obecnie najbardziej rozwiniętym i nowoczesnym oprogramowaniem systemów pomiarowych jest LabVIEW™ (z ang. **L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench), będący produktem firmy National Instruments Corporation.

LabVIEW jest środowiskiem programistycznym służącym do tworzenia własnych programów użytkowych podobnie jak środowiska Borland C, Borland Pascal, i inne. Jedyną różnicą jest sposób tworzenia kodu źródłowego programu. W innych środowiskach kod źródłowy jest sekwencyjnym zapisem tekstowym w określonym języku programowania (Basic, C, Pascal), natomiast w LabVIEW kod źródłowy sprowadza się do graficznego przedstawienia operacji funkcji *wejścia* (z klawiatury i przyrządów), *wyjścia* (na ekran, drukarkę i przyrządy), przenoszenia i przetwarzania informacji oraz połączenia między blokami. Język graficzny G, przedstawia zapis programu w postaci ikon, zacisków i połączeń za pomocą, których można zbudować dowolny wirtualny przyrząd pomiarowy. W zależności od potrzeb i oprogramowania może

* Grzegorz BARAŃSKI, Rafał SOCHACZEWSKI, Jacek CZARNIGOWSKI – Katedra Silników Spalinowych i Transportu, Wydział Mechaniczny, Politechnika Lubelska.

to być wirtualny oscyloskop, analizator widma, rejestrator, multimetr czy inny nietypowy przyrząd.

LabVIEW (w skrócie LV) posiada rozbudowane biblioteki funkcji i podprogramów do większości zadań programistycznych. Zawiera ponadto bogate biblioteki programowych sterowników różnych przyrządów pomiarowych, sterowniki interfejsów GPEB, VXI i szeregowych oraz biblioteki funkcji do zbierania, analizy i prezentacji danych. LabVIEW, tak jak i inne środowiska programistyczne, daje szerokie możliwości uruchamiania programu i poszukiwania błędów. Można ustawiać pułapki, animować diagram przepływu informacji, a także uruchamiać program krokowo śledząc wyniki w poszczególnych jego częściach.

STRUKTURA PROGRAMU

LabVIEW zawiera biblioteki funkcji i narzędzia zaprojektowane specjalnie do zbierania danych i sterowania przyrządami pomiarowymi. Programy, napisane za pomocą LV, noszą nazwę przyrządów wirtualnych, ang. Virtual Instruments, (VI), ponieważ ich wygląd i działanie imituje przyrządy rzeczywiste. Przyrządy VI posiadają płytę czołową (interfejs użytkownika), kod źródłowy (diagram przepływu informacji) oraz przyjmują parametry od przyrządów VI wyższego poziomu.

Do programowania wykorzystuje się narzędzia udostępnione w *Show Tools Palette* (rys. 1). Za pomocą nich możliwe jest, w pierwszej kolejności: wybór obiektów oraz ich umieszczanie na panelu lub wstawianie do programu graficznego, następnie łączenie, przemieszczanie, uszeregowywanie i rozmieszczanie poszczególnych obiektów



Rys. 1. Narzędzia do projektowania panelu i programowania graficznego

tów lub całych grup, usuwanie zbędnych obiektów, edycja tekstów, zadawanie parametrów, uruchamianie, testowanie i zatrzymywanie programu.

Tworzenie programu opiera się o dwa podstawowe panele:

- interfejs użytkownika;
- diagram przepływu informacji.

Interfejs użytkownika

Panel stanowi interaktywny interfejs z użytkownikiem, stanowiąc symulację płyty czołowej rzeczywistego przyrządu. Można na nim umieszczać zadajniki (*Controls*) oraz wskaźniki (*Indicators*) pełniących funkcje terminali. Zadajniki służą do wprowadzania danych do programu, np.: przełączniki, pokręta, natomiast wskaźniki służą do wyprowadzania danych z programu do użytkownika np.: wyświetlacze, wykresy. Do obsługi elementów na panelu wykorzystywana jest mysz lub klawiatura, natomiast wyniki przedstawiane są na monitorze komputera.

W celu umieszczenia obiektu na panelu użytkownika, należy skorzystać z palety *Controls* przedstawionej na rys. 2. Obiekty w przedstawionym oknie podzielone są na poszczególne grupy, które zawierają następujące obiekty:

- *Numeric* – elementy do wprowadzania do programu i odczytywania z programu danych liczbowych (np. pokręta, wskaźniki cyfrowe);
- *Boolean* – elementy do wprowadzania do programu i odczytywania z programu danych typu logicznego (np. przełączniki, wskaźniki w postaci diody LED);
- *String & Patch* – do wprowadzania do programu i odczytywania z programu danych w postaci tekstu lub ścieżki dostępu do pliku;
- *Array & Luster* – do wprowadzania do programu i odczytywania z programu danych w postaci macierzy lub klastrów;
- *List & Table* – do wprowadzania do programu informacji w postaci listy lub tabeli;
- *Graph* – elementy do odczytywania z programu informacji w postaci wykresów;
- *Ring & Enum* – elementy do wprowadzania do programu informacji wybranej z listy;
- *Containers* – elementy *Tab Control SubPanel* i *ActiveX Container*;
- *I/O* – elementy do wprowadzania do programu informacji dotyczących operacji wejścia/wyjścia;



Rys. 2. Okno *Controls*

- *Dialog Controls* – elementy do tworzenia formularzy;
- *Classic Controls* – obiekty ze starszych wersji LabVIEW (styl klasyczny);
- *Refnum* – elementy do identyfikacji operacji wejścia/wyjścia;
- *Decorations* – elementy dekoracyjne;
- *Express* – elementy typu *Numeric*, *Buttons*, *Switches* i *Text* z podziałem na wejściowe i wyjściowe;
- *Select a Control...* – wyszukiwanie obiektów zapisanych w pliku umieszczonym w dowolnym miejscu na dysku;
- *User Controls* – własne elementy użytkownika umieszczone w katalogu *User.lib*.

Diagram przepływu informacji

Przyrządy VI pobierają instrukcje z diagramu przepływu informacji, który jest napisany w języku G. Diagram jest piktograficznym rozwiązaniem zagadnienia programistycznego, jest kodem źródłowym. W jego skład wchodzi terminal, funkcje, struktury i przewody. Aby wprowadzić obiekty funkcyjne i procedury do diagramu należy skorzystać z okna *Functions Palette* (rys. 2). Udostępnia ona grupy bibliotek oraz kaskadowo rozwijane pola, które zawierają obiekty funkcyjne, umożliwiające przetwarzanie danych oraz kontrolowanie ich przepływem w programie graficznym. Obiekty funkcyjne i procedury można podzielić na dwie grupy. Pierwsza z nich zawiera obiekty niskiego poziomu – podstawowe, do których można zaliczyć np.: operatory operacji arytmetycznych lub funkcje trygonometrycznych. Natomiast druga grupa



Rys. 3. Okno *Controls*

zawiera obiekty wysokiego poziomu – zaawansowane, do których zaliczają się między innymi: sterowniki urządzeń zewnętrznych i obiekty umożliwiające analizę danych.

W poszczególnych grupach znajdują się następujące elementy:

- *Structures* – sekwencje, instrukcje wyboru, pętle, struktura do wpisywania wzorów, zmienne globalne i lokalne;
- *Numeric* – operacje na liczbach;
- *Boolean* – operacje na zmiennych typu logicznego;
- *String* – operacje na łańcuchach znaków;
- *Array* – operacje na macierzach;
- *Cluster* – operacje na elementach typu klaster;
- *Comparison* – operacje porównania;
- *Time & Dialog* – funkcje związane z czasem i tworzenie okien dialogowych;
- *File I/O* – funkcje do obsługi plików;
- *NI Measurements* – funkcje do obsługi kart akwizycji sygnałów pomiarowych, obsługi napędu oraz akwizycji i przetwarzania obrazu;
- *Waveform* – operacje na przebiegach sygnałów;
- *Analyze* – funkcje do analizy sygnałów;
- *Instrument I/O* – funkcje do obsługi urządzeń dołączonych do komputera;
- *Application Control* – funkcje do sterowania działaniem aplikacji;
- *Graphics & Sound* – funkcje związane z tworzeniem grafiki i dźwięków.

W celu uzyskania powtarzalności zdarzeń, elementy programu muszą być umieszczone w pętlach. Wyróżniamy dwa rodzaje pętli: *For Loop* i *While Loop*. Umieszczona w ramce struktury *For Loop* część programu jest wykonywana tyle razy, ile nosi liczba dołączona do zacisku z literą „N” w lewym górnym rogu ramki. Sygnał dołączony do zacisku z literą „i” informuje o numerze aktualnie wykonywanej iteracji. W trakcie pierwszego wykonywania pętli do zacisku wejściowego zostaje pobrana wartość z zewnątrz, dlatego możemy podłączyć do niego wartość inicjującą. W trakcie wykonywania pętli po raz ostatni, zacisk wyjściowy może przekazać wartości z pętli dalej do programu.

Umieszczona w ramce struktury *While Loop* część programu jest przetwarzana tak długo, aż zmienna dołączona do zacisku w postaci czerwonego kółka przybierze wartość *True* lub *False* w zależności od ustawień konfiguracji. Znaczenie zacisku iteracji „i” jest takie same w pętli *For Loop*.

PRZYKŁADY ZASTOSOWANIA

Dla zobrazowania możliwości programu oraz sposobu jego zastosowania poniżej przedstawiono dwa przykłady.

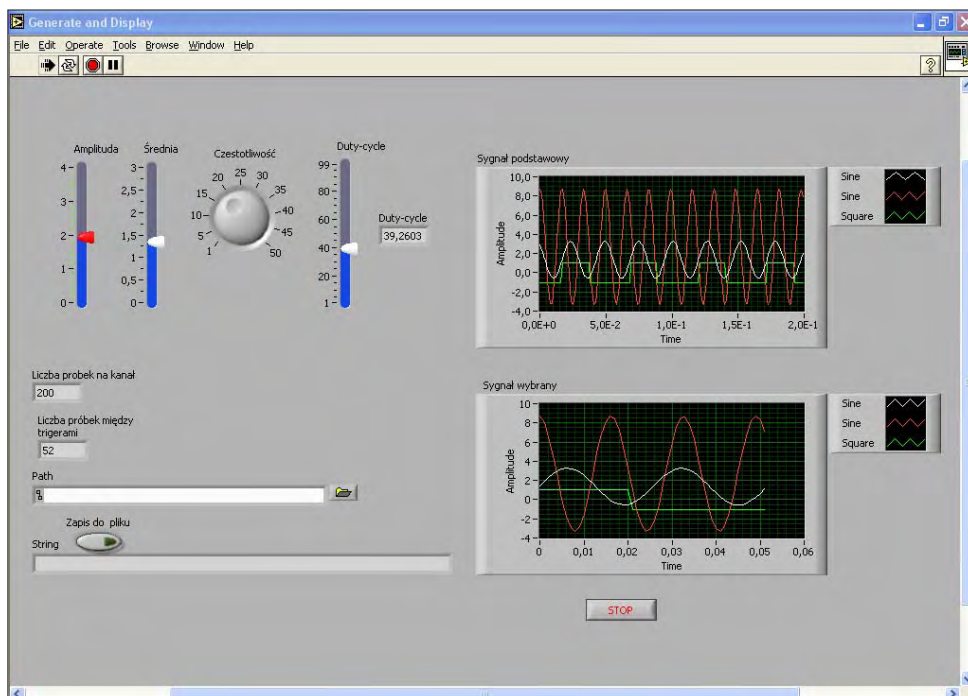
Generowanie i analiza trzech sygnałów

Program LabView umożliwia wykonanie analizy sygnału zarówno otrzymanego z urządzeń pomiarowych jak też z generatorów. Zastosowanie generatorów pozwala na przygotowanie i przetestowanie algorytmów analizy danych przed ich zastosowaniem do analizy danych pochodzących z pomiarów.

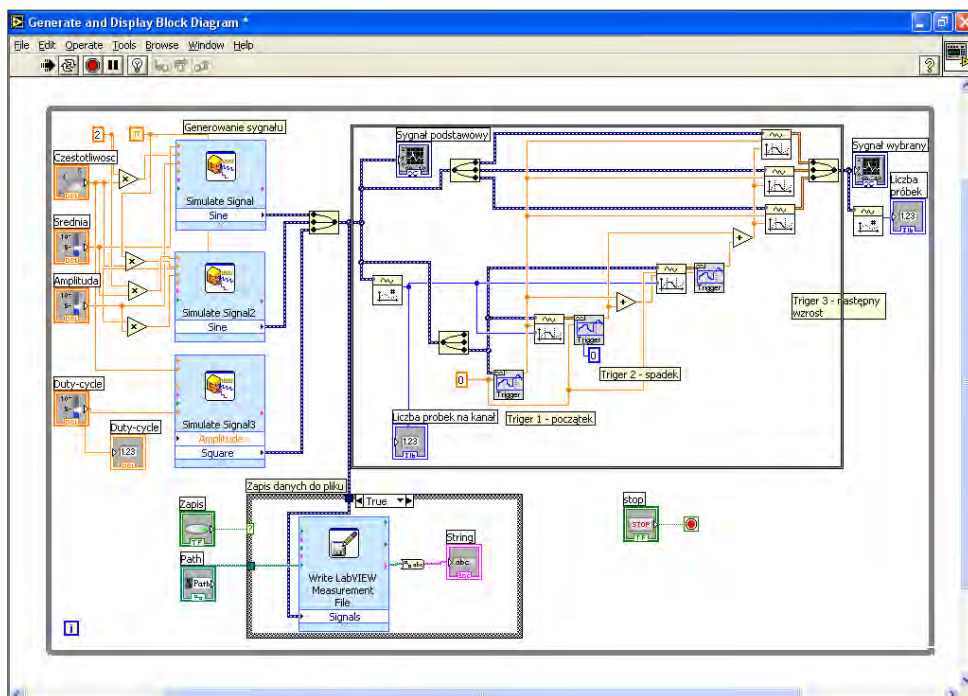
Przykład takiego programu przedstawiono na rysunku 4 i 5.

W programie tym wykorzystano 3 generatory sygnałów (Simulate Signal na rys. 4) do uzyskania strumienia danych składającego się z dwóch fal sinusoidalnych oraz jednej prostokątnej. Jednocześnie umożliwiono ustawienie zarówno częstotliwości, amplitudy jak i wartości średniej sygnałów, w przypadku sygnału prostokątnego możliwe jest także ustawienie wypełnienia sygnału (Duty Cycle).

Tak otrzymany strumień danych jest poddany analizie. W tym przypadku polega ona na wyodrębnieniu z sygnału podstawowego zakresu odpowiadającego jednemu okresowi zmian sygnału prostokątnego. Do tego celu zastosowano Trigger, odnajdujący zbocze rosnące i opadające sygnału. Na podstawie pracy tego elementu wycinany jest pożądany fragmenty strumienia danych (zastosowano funkcję *WDT Get Waveform Subset*). Tak uzyskane dane są wyświetlane na wykresie dolnym – rysunek 4.



Rys. 4. Panel programu do generowania i analizy trzech sygnałów



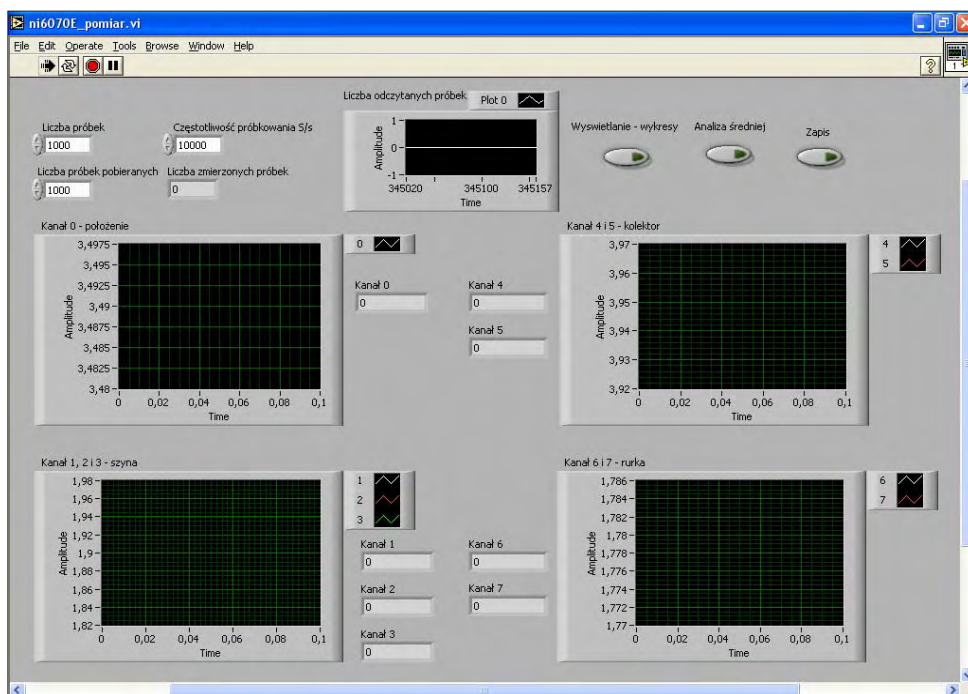
Rys. 5. Diagram programu do generowania i analizy trzech sygnałów

Jednocześnie wprowadzono także funkcję zapisu danych do pliku tekstowego. Funkcja *Write LabView Measurement File* pozwala na zapisanie danych w postaci kodu ASCII zawierającego zarówno same dane jak i opis kanałów, czas wykonania zapisu i parametry pomiaru (lub generowania). Taki plik może być następnie analizowany w dowolnym programie numerycznym (Excel, Matlab, Grapher) lub wprowadzony ponownie do LabView w celu ich dalszej analizy.

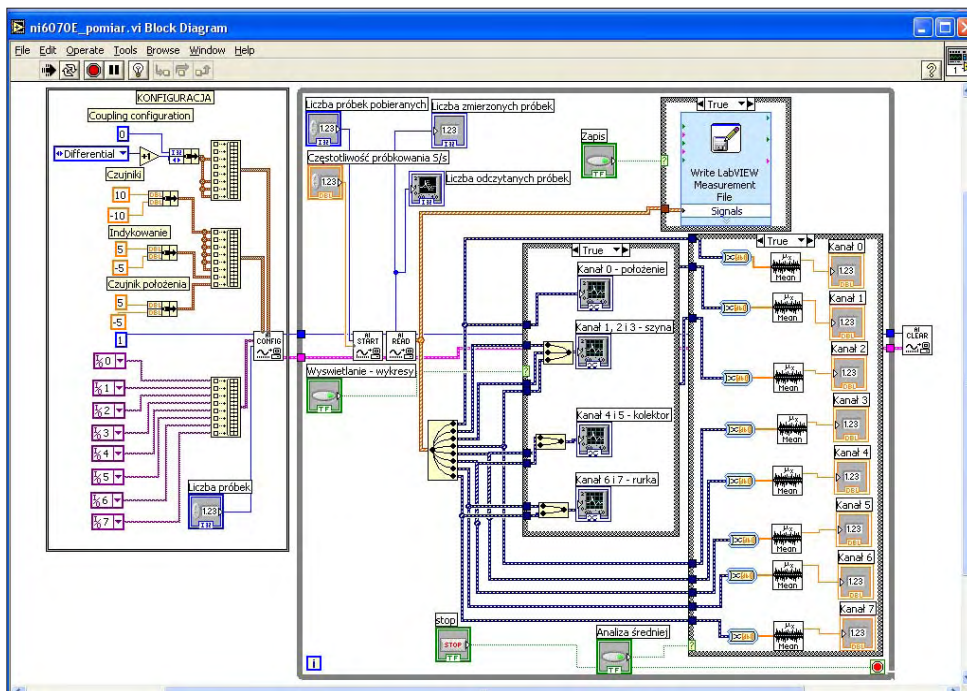
Wirtualny ośmiokanałowy przyrząd pomiarowy

Zbudowano wirtualny ośmiokanałowy przyrząd pomiarowy (rys. 6) służący do obsługi urządzenia National Instrument NI 6070E. Obsługiwany za pomocą funkcji DAQ. Pierwszym krokiem było zbudowanie bloku konfiguracji wykorzystując funkcję *AI Config.vi* za pomocą której możliwy jest wybór sposobu pomiaru: *Differential* lub *Refferendet*. Przy pomiarze z ustawieniem *Differentia* pomiar zmiennych z poszczególnych kanałów przebiega niezależnie, natomiast wybór funkcji *Refferendet* powoduje wykonywanie pomiarów zmiennych ze wszystkich kanałów względem wspólnej masy. Dodatkowo istnieje możliwość wyboru mierzonego kanału oraz jego zakresu napięć. W ostatniej fazie tworzenia bloku konfiguracji założono liczbę próbek pomiaru. Kolejnym elementem układu pomiarowego jest blok *AI Start* służący

do rozpoczęcia pomiaru. W bloku tym zadajemy wartości częstotliwości próbkowania oraz liczbę próbek pobieranych. W celu odczytywania wartości próbek oraz ich ilości wstawiono blok *AI Read*. Kończącą fazą w tworzeniu układu pomiarowego było zastosowanie bloku: wizualizacji, uśredniania oraz zapisu do pliku. Za pomocą bloku *Waveform Graph* wartości mierzone są wyświetlane na wykresach przedstawionych na rysunku 5. Wykonywanie pomiaru powoduje dość znaczne obciążenie komputera pomiarowego, dlatego wykonano wizualizację w strukturze *Case*, dzięki której możliwe jest wyłączenie procesu wizualizacji na wykresach. Struktura ta daje możliwość odciążenia układu i uniknięcia występowania „gubienia” próbek podczas zapisu do pliku. Dodatkowo wstawiono również funkcję statystyczną *Mean* za pomocą której otrzymujemy wartości średnie. Elementy te umieszczono w strukturze *Case* uruchamiającej wyświetlanie wartości średnich na panelu użytkownika. Ponadto umieszczono funkcję *Write Labview* służącą do zapisu danych do pliku tekstowego zawierającą takie same funkcje jak w poprzednim przykładzie. Zastosowano również funkcję *AI Clear* służącą do czyszczenia bufora pamięci.



Rys. 6. Panel programu do pomiaru ośmiu wielkości charakteryzujących pracę silnika spalinowego



Rys. 7. Diagram programu do pomiaru ośmiu wielkości charakteryzujących pracę silnika spalinowego

PODSUMOWANIE

Graficzne programowanie w środowisku LabVIEW cechuje duża łatwość i intuicyjność obsługi. Użytkownik nie musi znać skomplikowanego języka programowania, gdyż program jest zbudowany z gotowych elementów, którym nadajemy odpowiednie parametry. LabVIEW umożliwia przygotowanie zarówno prostych aplikacji jak i oprogramowania złożonych systemów – możliwości programu wydają się prawie nieograniczone.

Program umożliwia zarówno obsługę danych z urządzeń pomiarowych jak i generowanie sygnałów. Tak uzyskane dane mogą być następnie analizowane lub/i zapisywane do plików w celu ich dalszej analizy. Jednocześnie możliwa jest wizualizacja danych w czasie pracy programu.

Cyfrowe układy przetwarzania sygnałów pozwalają na przetwarzanie i rejestrację dużej liczby danych pomiarowych w pamięci komputera i zachowanie ich do dalszej analizy. Ponadto daje nam możliwość rejestracji wielkości szybkozmiennych, których pomiar przyrządami analogowymi do tej pory był utrudniony. Możliwa też jest realizacja pomiarów wielokanałowych oraz filtrowanie i wygładzanie sygnałów pomiarowych.

Przedstawione powyżej przykłady są dowodem uniwersalności i funkcjonalności wirtualnego oprogramowania, jakim jest LabVIEW, ułatwiającego przeprowadzanie badań naukowych.

LITERATURA

1. Świsulski D.: Komputerowa technika pomiarowa. Warszawa 2005.
2. Świsulski D.: Przyrządy wirtualne jako nowa generacja przyrządów pomiarowych. Pierwsze Sympozjum Naukowe Aktualne problemy w metrologii. APM 2000, Gdańsk 2000.
3. Tłaczała W.: Środowisko LabVIEW w eksperymencie wspomaganym komputerowo,. Warszawa 2002.

Streszczenie

W artykule przedstawiono krótką charakterystykę programowania w środowisku LabVIEW. Opisano również ogólną budowę środowiska programu wraz z podstawowymi jego funkcjami. Ponadto, na przykładzie dwóch aplikacji, zaprezentowano technikę programowania w tym środowisku.

USE OF LABVIEW SOFTWARE FOR PROPULSION SYSTEMS RESEARCH

Summary

The article describes shortly programming procedures in LabView software. General structure of the software and principal functions were also presented. Moreover, programming technique was shown on two examples.