

Andrzej ZBROWSKI, Jordan MĘŻYK, Piotr CZAJKA

Instytut Technologii Eksploatacji – Państwowy Instytut Badawczy, Radom

WARSTWA PROGRAMOWA W SYSTEMIE STEROWANIA MANIPULATOREM DO POZYCJONOWANIA MIKROFONU POMIAROWEGO W KOMORZE BEZECHOWEJ

Słowa kluczowe

Manipulator, komora bezechowa, LabView, synchronizacja pomiarów.

Streszczenie

W artykule zaprezentowano możliwości wykorzystania środowiska LabView do sterowania manipulatorem do pozycjonowania mikrofonu w komorze bezechowej. Przedstawiono uzasadnienie wyboru tego oprogramowania do realizacji funkcji sterująco-pomiarowej. Zaprezentowano również strukturę układów sterowania oraz model geometryczny manipulatora. Omówiono ważną kwestię programowej synchronizacji modułów odpowiedzialnych za pomiary akustyczne z modułami sterującymi ruchem manipulatora.

Wprowadzenie

Automatyzacja akustycznych procesów pomiarowych w komorze bezechowej, jest zagadnieniem stawiającym szereg nietypowych, trudnych i nierzadko wzajemnie sprzecznych wymagań opisanych w odpowiednich normach [1, 3]. Konstrukcja manipulatora wiąże się zawsze z wprowadzeniem dodatkowych elementów mechanicznych, które zakłócają pole swobodne komory bezechowej [4]. Jednak zysk polegający na znaczącym uproszczeniu procedur pomiarowych

oraz skróceniu czasu wymaganego do przeprowadzenia badań powoduje zainteresowanie tematyką automatyzacji pomiarów akustycznych. Główne zadania opracowanego manipulatora to zwiększenie dokładności, uniwersalności oraz ułatwienie prowadzenia badań o charakterze naukowo-badawczym, dydaktycznym oraz badań przemysłowych [7]. System sterowania manipulatorem powinien mieć strukturę otwartą i posiadać dużą elastyczność funkcjonalną, tak aby możliwe było przeprowadzenie różnorodnych badań z dziedziny akustyki. Przy wyborze oprogramowania należało uwzględnić łatwość tworzenia aplikacji oraz możliwość jej późniejszej modyfikacji w zależności od nowych wymagań.

Efektem zadania badawczego, zrealizowanego przez Akademię Górniczo-Hutniczą i Instytut Technologii Eksploatacji, jest system manipulatora do pozycjonowania mikrofonu pomiarowego w komorze bezdechowej.

1. Środowisko LabView

LabView jest zintegrowanym graficznym środowiskiem programistycznym [6]. Programy tworzone w tym środowisku nazywane są wirtualnymi instrumentami i składają się z kilku składników: interfejsu urządzenia, diagramu blokowego realizującego funkcje programowe oraz sterowników urządzeń zewnętrznych. Dzięki temu tworzenie aplikacji użytkowych jest łatwe i intuicyjne, a sama aplikacja podporządkowana jest funkcji, jaką ma spełniać. Funkcje programu składa się w postaci schematu blokowego bez ustalonej hierarchii. Poszczególne bloki wykonują swoje zadania w momencie otrzymania wszystkich danych wejściowych i kończą wystawieniem danych na swoje wyjścia. Dzięki temu poszczególne zadania mogą wykonywać się jednocześnie, a system sam zarządza wątkami. Program kompilowany jest do postaci aplikacji mogącej funkcjonować w systemie Windows, Linux lub MacOS.

Środowisko LabView jest środowiskiem modułowym i wyposażone jest w szereg specjalizowanych bibliotek, m.in. matematyczna (*Mathematics*), pomiarowa (*Measurement*), do przetwarzania sygnałów (*Signal Processing*), do komunikacji (*Communication*) oraz wizyjna i do sterowania napędami (*Vision and Motion*).

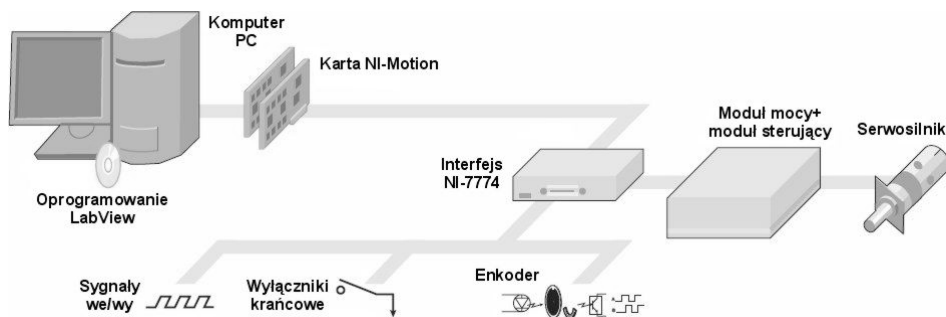
Najważniejszą biblioteką oprogramowania LabView z punktu widzenia potrzeb systemu sterowania manipulatora pomiarowego jest biblioteka NI-Motion. Zawiera ona następujące grupy funkcji sterujących napędami:

- konfiguracja napędów i osi,
- trajektoria,
- sterowanie ruchem (start/stop/miksowanie),
- wejście/wyjście (krańcówki, zezwolenia na ruch, interakcje sprzętowe),
- punkt odniesienia (procedury bazowania),
- elektryczna konfiguracja portów i sygnałów kontrolnych,
- obsługa błędów,
- programowanie na karcie.

Drugą biblioteką stosowaną w systemie manipulatora pomiarowego jest biblioteka Measurement I/O odpowiedzialna za akwizycję danych. Zawiera ona komplet funkcji pomocnych przy pobieraniu danych, począwszy od funkcji komunikacyjnych, poprzez funkcje generujące sygnały testowe do funkcji sterujących wejściami/wyjściami cyfrowymi oraz sygnałami wyzwalania (*trigger*).

2. Układ sterowania

Do sterowania manipulatorem pomiarowym zastosowano kartę NI PXI-7354, zdolną kontrolować do 4 osi. Karta umożliwia współpracę z napędem w trybie silnika krokowego z częstotliwością do 4 MHz lub w trybie serwa. Przy ograniczeniu do dwóch osi wewnętrzny regulator PID może pracować z okresem realizacji algorytmu 62 μ s. Karta ta może również generować sinusoidalne sygnały komutacji dla silników bezszczotkowych. W celu podłączenia karty do układów zewnętrznych wykorzystany został interfejs NI-7774. Schemat układu sterowania jedną osią został przedstawiony na rysunku 1.

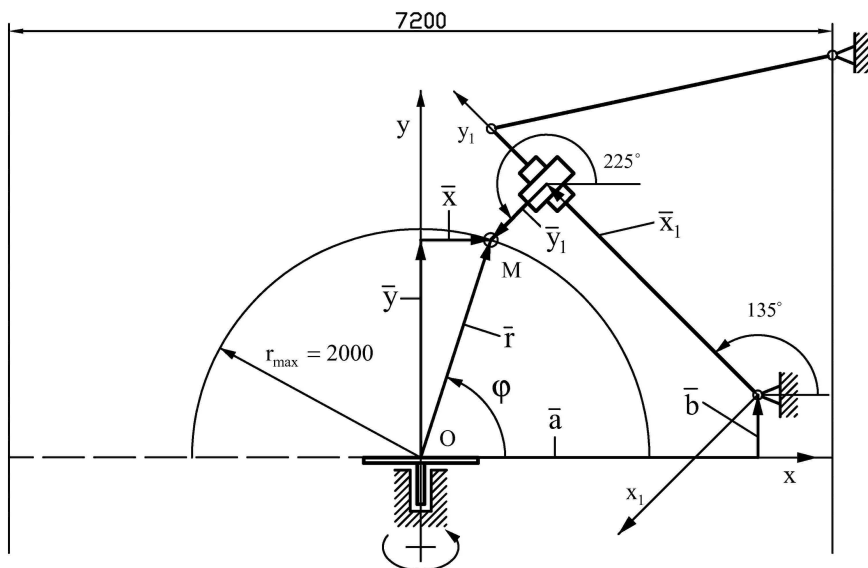


Rys. 1. Schemat układu sterowania pojedynczą osią

Do współpracy z systemem manipulatora karta została ustawiona w tryb sterowania serwonapędami z zadawaniem prędkości [4]. Polega on na tym, że karta wydaje polecenie jazdy z określoną prędkością za pośrednictwem sygnału napięciowego z zakresu ± 10 V, gdzie +10 V oznacza jazdę z pełną prędkością do przodu, a -10 V oznacza jazdę z pełną prędkością do tyłu. Informacja zwrotna o położeniu przekazywana jest w postaci sygnałów z enkoderów inkrementalnych oraz wyłączników krańcowych. Elementami wykonawczymi są napędy Rexroth IndraDrive z serii HCS zasilające silniki serii MSK. Wszystkie sygnały zwrotne z manipulatora przekazywane są do napędów i tam przetwarzane, a następnie przesyłane do karty NI-Motion. Sygnał enkodera odbierany przez kartę jest sygnałem emulowanym przez napęd. Użytkownik ma wpływ zarówno na sposób emulacji, jak i interpretacji tego sygnału.

Model geometryczny

Model matematyczny manipulatora został przedstawiony na rysunku 2. Układ odniesienia związany z elementami wykonawczymi nie jest ortogonalny względem układu przestrzeni pomiarowej, zatem zależność między współrzędnymi docelowymi a nastawianymi na napędach nie jest liniowa [2].



Rys. 2. Model geometryczny manipulatora do pozycjonowania mikrofonu w komorze bezechowej

W rozważanym modelu osiągnięcie dowolnego punktu M w ćwiartce koła zdefiniowanej przez układ OXY wymaga ruchu dwóch modułów liniowych. Zmianę pozycji liniowych modułów opisuje równanie wektorowe:

$$\bar{x} + \bar{y} = \bar{a} + \bar{b} + \bar{x}_1 + \bar{y}_1 \quad (1)$$

Dla modelu geometrycznego prostego obowiązują zależności w postaci:

$$\begin{aligned} x &= a + x_1 \cos 135^\circ + y_1 \cos 225^\circ \\ y &= b + x_1 \sin 135^\circ + y_1 \sin 225^\circ \end{aligned} \quad (2)$$

gdzie:

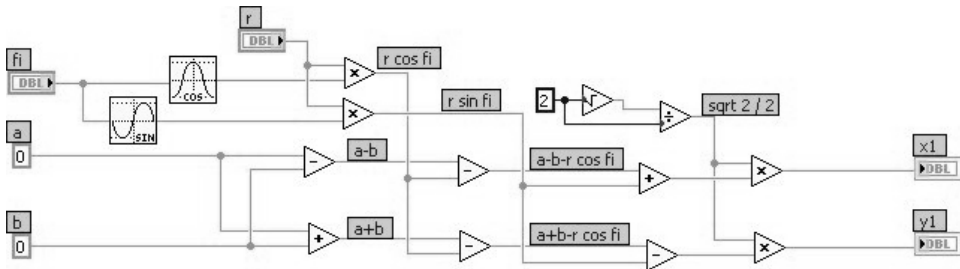
$$x = r \cos \varphi, \quad y = r \sin \varphi \quad (3)$$

Zatem dla modelu geometrycznego odwrotnego:

$$x_1 = \frac{\sqrt{2}}{2} (a - b - r \cos \varphi + r \sin \varphi)$$

$$y_1 = \frac{\sqrt{2}}{2} (a + b - r \cos \varphi - r \sin \varphi)$$
(4)

Zależności dla modelu odwrotnego umożliwiają przeliczenie współrzędnych układu OXY na polecenia sterujące napędami. Do konwersji współrzędnych zbudowano oddzielną aplikację (wirtualny instrument), która następnie została wykorzystana w głównej aplikacji (rys. 3).

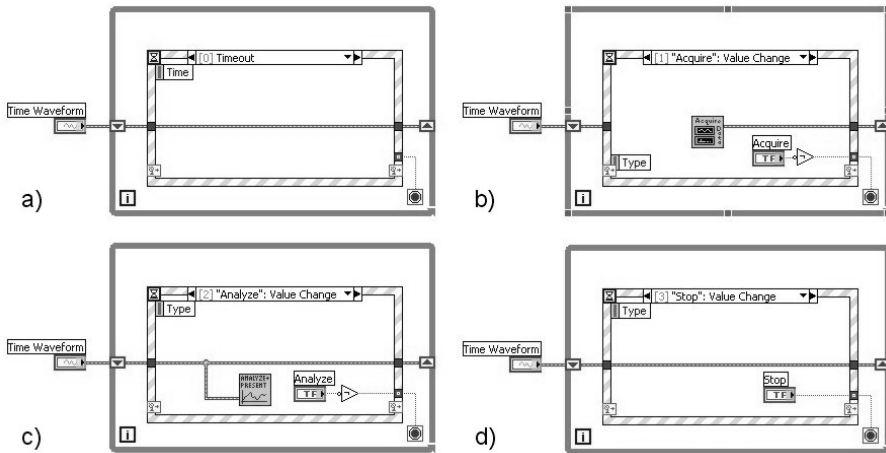


Rys. 3. Fragment aplikacji odpowiedzialny za przeliczenie zadawanych współrzędnych na polecenia sterujące napędami

Synchronizacja pomiarów

Do automatyzacji badań w komorze bezchowej konieczne jest zapewnienie synchronizacji pomiędzy ruchem napędów a akwizycją danych z mikrofonu. W celu wykonywania pomiarów „w locie”, bez zatrzymywania napędów, program cały czas powinien kontrolować pozycję stołu i mikrofonu, tak aby we właściwym momencie wykonywać pomiar. LabVIEW oferuje kilka mechanizmów synchronizacji takich jak: zdarzenie (*event*), wystąpienie (*occurrence*), powiadomienie (*notification*), kolejka (*queue*), semafor (*semaphore*) oraz rendezvous [5]. Tylko niektóre z powyższych mechanizmów są odpowiednie do automatyzacji badań akustycznych.

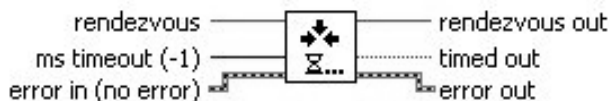
Obsługa zdarzeń (*events*) polega na zastosowaniu specjalnej struktury (rys. 4) pozwalającej utworzyć fragmenty kodu, które są wykonywane, jeśli pojawi się wcześniej zdefiniowane zdarzenie. Standardowo zdefiniowanych jest dużo zdarzeń systemowych związanych z całą aplikacją, z konkretnym przyrządem lub elementami panelu frontowego aplikacji. Elementy schematu blokowego mogą również generować zdarzenia.



Rys. 4. Wykonywanie akcji w programie w zależności od wystąpienia zdarzeń: a) przekroczenie czasu (*Timeout*), b) zmiana wartości przycisku „Acquire”, c) zmiana wartości przycisku „Analyze”, d) zmiana wartości przycisku „Stop”

Mechanizm synchronizacji polegający na obsłudze zdarzeń mógłby być zastosowany do synchronizacji pomiarów, ale najlepiej sprawdza się w interakcji z użytkownikiem aplikacji (np. zmiana wartości przycisków, zadajników itp.).

Kolejny sposób synchronizacji to mechanizm rendezvous (rys. 5), którego sposób działania polega na wstrzymywaniu działania aplikacji lub jej fragmentów tak, aby wszystkie synchronizowane elementy znalazły się w ustalonym punkcie jednocześnie, a dopiero potem wykonywały dalsze swoje działania. Dzięki temu elementy wolniejsze nadążają za szybszymi, ale nie ma kontroli czasu wykonania.

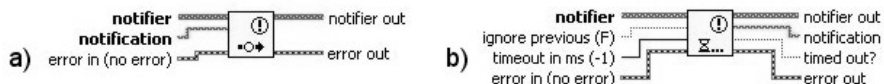


Rys. 5. Obiekt rendezvous – „spotkanie” fragmentów programu w wybranym punkcie

Ten rodzaj synchronizacji bardzo dobrze nadaje się do obsługi akcji wymagających czasu a istotnych dla dalszego działania aplikacji, np. do zapisu lub odczytu danych do/z pliku.

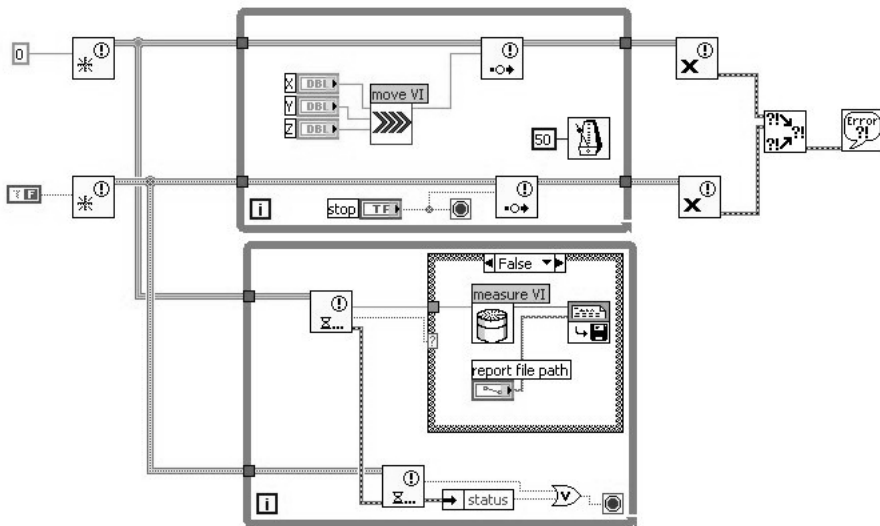
Z punktu widzenia synchronizacji pomiarów interesującym mechanizmem jest powiadomienie (*Notification*). Pozwala on na wstrzymanie wykonywania fragmentu programu do chwili otrzymania powiadomienia od innej części tego

programu. Widok obiektów do obsługi powiadomień został przedstawiony na rysunku 6.



Rys. 6. Obiekty do obsługi powiadomień: a) Send Notification – funkcja wysyłająca powiadomienie, b) Wait for Notification – funkcja oczekująca na powiadomienie

Powyższe rozwiązanie nie gwarantuje wykonania akcji natychmiast, a dopiero w najbliższym cyklu odpowiedzialnej za nie pętli. Rozwiązaniem tego problemu jest umieszczenie odbiornika powiadomienia w drugiej pętli, która nie posiada wbudowanych opóźnień. Dzięki zastosowaniu powiadomień pętla taka nie obciąża nadmiernie procesora i pozwala na wykonanie zadań synchronicznie z główną pętlą. Ponadto powiadomienie może przekazywać wartość (lub zbiór wartości) do drugiej pętli.



Rys. 7. Uproszczony schemat programu sterującego manipulatorem, wykorzystano mechanizm Notification oraz sub-vi

Rysunek 7 przedstawia schematycznie rozwiązanie zastosowane w aplikacji sterującej manipulatorem. W głównej pętli zadawane są kolejne pozycje, dla których wykonywane są pomiary. Za ruch elementów manipulatora odpowiedzialna jest aplikacja *moveVI*, która również generuje powiadomienia o osiągnięciu

nięciu punktów pomiarowych. W drugiej pętli, wykonywanej bez opóźnień, umieszczona została aplikacja *measureVI*, która odpowiedzialna jest za odczytanie wartości z mikrofonu pomiarowego i zapisanie jej do pliku.

Podsumowanie

Strukturę systemu sterowania manipulatorem zbudowano z zastosowaniem nadrzędnego kontrolera w postaci karty NI-Motion, która obsługiwana jest z poziomu oprogramowania LabView. Karta sterująca wraz z oprogramowaniem umożliwia zadawanie i bieżące śledzenie położenia mikrofonu, dzięki czemu możliwa jest synchronizacja pomiarów w „locie” bez zatrzymywania napędów. Takie rozwiązanie charakteryzuje się również dużą elastycznością bowiem położenia punktów pomiarowych nie są zapisane na stałe w napędach. Umożliwia to tworzenie dowolnych trajektorii w ramach dostępnej przestrzeni pomiarowej. Zastosowane oprogramowanie LabView pozwala na tworzenie aplikacji przez osoby nieposiadające doświadczenia w programowaniu. Dzięki obecności bibliotek takich jak Motion i Measurement możliwe jest budowanie w prosty sposób zaawansowanych aplikacji napędowo-pomiarowych. Dostępne mechanizmy synchronizacji zapewniają koordynację pomiędzy ruchem napędów a akwizycją danych z mikrofonu. Środowisko LabView umożliwia tworzenie aplikacji otwartych, które mogą być modyfikowane w zależności od nowych wymagań. Pomimo zapewnienia pełnej otwartości oprogramowania nie ma możliwości uszkodzenia układu mechanicznego manipulatora. Osiągnięto to w wyniku zastosowania zabezpieczeń sprzętowych w postaci wyłączników krańcowych oraz nałożenia ograniczeń na maksymalne wartości prędkości i przyspieszenia, które zostały zaimplementowane w napędach niezależnie od aplikacji sterującej.

Praca naukowa finansowana ze środków Ministra Nauki i Szkolnictwa Wyższego, wykonana w ramach realizacji Programu Wieloletniego pn. „Doskonalenie systemów rozwoju innowacyjności w produkcji i eksploatacji w latach 2004–2008”.

Bibliografia

1. Batko W., Felis J., Flach A., Giesko T., Kamisiński T., Zbrowski A.: A concept of an actuator for positioning of elements of a measurement system in an anechoic chamber. OSA 2007, LIV Otwarte Seminatum z Akustyki: Rzeszów–Przemyśl, Poland, September 10–14, 2007.
2. Felis J., Zbrowski A., Giesko T., Mężyk J.: Structure Synthesis and Mechanical Parameters Choice for a Manipulating Mechanism for Acoustical Measurements in Anechoic Chamber, Solid State Phenomena Vols. 147–149 (2009) pp. 13–18, <http://www.scientific.net/>

3. PN-EN ISO 3745:2007 Akustyka. Wyznaczanie poziomów mocy akustycznej źródeł hałasu na podstawie pomiarów ciśnienia akustycznego. Metody dokładne w komorach bezdechowych i w komorach bezdechowych z odbijającą podłogą.
4. Mężyk J., Kamisiński T., Zbrowski A., Flach A.: Hardware Structure of a Control Section of Manipulation Mechanism for Acoustical Measurement in Anechoic Chamber, *Solid State Phenomena Vols 147–149 (2009)* pp. 7–12, <http://www.scientific.net/>
5. Świsulski D.: Komputerowa technika pomiarowa, Oprogramowanie wirtualnych przyrządów pomiarowych w LabVIEW, Agenda Wydawnicza PAK-u, Warszawa 2005.
6. Winiecki W., Nowak J., Stanik S.: Graficzne zintegrowane środowiska programowe, Wydawnictwo MIKOM, Warszawa 2001.
7. Zbrowski A., Mężyk J., Czajka P.: Modułowy manipulator do pomiarów akustycznych w komorze bezdechowej. *Zeszyty Naukowe Politechniki Rzeszowskiej* Nr 261, seria Mechanika z. 76, Rzeszów 2009, s. 189–195.

Recenzent:
Jan Maciej KOŚCIELNY

Software layer in control system of manipulator for microphone positioning in anechoic chamber

Key-words

Manipulator, anechoic chamber, LabView, measurement synchronization.

Summary

The article presents the possibilities of the use of the LabView environment for the control of a manipulator for microphone positioning in an anechoic chamber. Qualification of LabView for such use is discussed. The structure of control section and geometric model of the manipulator are described. Also, an important matter concerning software synchronisation of modules is discussed for acoustic measurement modules and manipulator motion control modules.

