

SŁOTA Adam¹
WIĘK Tomasz²

KONCEPCJA UKŁADU STEROWANIA POZYCYJNO-SIŁOWEGO Z WYKORZYSTANIEM STEROWNIKA NEXTMOVE

W pracy przedstawiono przegląd sposobów programowania wybranych z dostępnych na rynku układów sterowania położeniem. Wskazano układ umożliwiający budowę aplikacji wbudowanych, udostępniający możliwość modyfikacji działania sterownika oraz przedstawiono możliwe miejsca modyfikacji. Przedstawiono zasadę sterowania admitancyjnego oraz zaproponowano sposób implementacji tego rodzaju sterowania w sterowniku NextMove ESB2.

AN IDEA OF POSITION-FORCE CONTROL SYSTEM WITH THE USE OF NEXTMOVE CONTROLLER

A short review of programming methods of selected motion controllers available in the market is presented. One controller, which provides capabilities of embedded programming and thus modification of default controller operation has been chosen and described with more details. Possible modifications are pointed out. Bases of admittance control are presented and an idea of implementation of this control method in NextMove ESB2 controller is presented.

1. WSTĘP

Dostępne aktualnie na rynku układy sterowania położeniem, znajdujące głównie zastosowanie w maszynach technologicznych i robotach przemysłowych, udostępniają zaawansowane funkcje sterowania położeniem zespołu roboczego. Jeden kierunek rozwoju sterowania położeniowego dotyczy możliwości programowania złożonych profili ruchu (interpolacja spline, krzywe Beziera), coraz większej liczby obsługiwanych (interpolowanych) osi oraz wzrostu dokładności realizacji zaprogramowanej trajektorii. Zakłada się tu, że zaprogramowany ruch ma być wiernie odtworzony przez układ wykonawczy bez względu na działające na układ zakłócenia (dynamika układu, zmienne obciążenia). Mają to zapewnić odpowiednio dobrane i dostrojone pętle sprzężenia zwrotnego oraz wyprzedzającego (feedforward). Drugi kierunek rozwoju dotyczy układów,

¹Politechnika Krakowska, Wydział Mechaniczny, 31-864 Kraków, Al. Jana Pawła II 37. Tel: +48 12 374 32 41,
Fax: +48 12 374 32 02, E-mail: slota@mech.pk.edu.pl

²Politechnika Krakowska, Wydział Mechaniczny, 31-864 Kraków, Al. Jana Pawła II 37. Tel: +48 12 374 32 63,
Fax: +48 12 374 32 02, E-mail: wiek@m6.mech.pk.edu.pl

w których istotny wpływ na działanie układu, poza narzuconym programem, ma środowisko, w którym program jest wykonywany. W układach takich wyniki pomiarów wykonanych przed rozpoczęciem programu lub wykonywanych na bieżąco w trakcie realizacji programu mają wpływ na jego wykonanie. Przykładem może tu być lokalizacja łączonych elementów przed rozpoczęciem spawania, czy zadanie uchwycenia przedmiotu poruszającego się na przenośniku taśmowym. W wielu przypadkach oddziaływanie środowiska ma charakter siłowy, wynikający z kontaktu elementu wykonawczego z otoczeniem. Ze względu na brak precyzyjnego opisu otoczenia (wynikający z niedokładności pomiaru, jak również z występujących zmian środowiska), w procesie sterowania konieczne jest uwzględnienie informacji ze sprzężenia zwrotnego siłowego. Sytuacja taka występuje na przykład przy realizacji montażu [1], działania wśród ludzi robotów usługowych [3,5], czy w układach dwuramiennych [7].

Wprowadzenie układów sprzężenia zwrotnego siłowego wymaga stosowania drogich elementów pomiarowych, szczególnie w układach takich jak roboty przemysłowe, które wymagają pomiaru sił i momentów w układzie przestrzennym. Dlatego proponowane są rozwiązania umożliwiające wyznaczenie wielkości oddziaływania układu z otoczeniem (sił/momentów) na podstawie pomiaru prądu silnika napędowego [4]. Rozwiązanie takie upraszcza konstrukcję układu i obniża jego koszty, wymaga jednak, ze względu na pośredni charakter pomiaru, uwzględnienia modelu dynamicznego układu w celu wyznaczenia siły oddziaływania układu z otoczeniem.

Dostępne na rynku układy sterowania umożliwiają wybór trybu sterowania położeniem lub siłą/momentem. Uwzględnienie w sterowaniu ruchem zarówno sprzężenia położeniowego, jak i siłowego, wymaga zwykle modyfikacji oprogramowania układu sterowania. W większości komercyjnych układów sterowania możliwość taka nie jest jednak dostępna.

Celem pracy jest przedstawienie propozycji budowy hybrydowego układu sterowania pozycyjno-siłowego w oparciu o komercyjny układ sterowania *NextMove ESB2* firmy Baldor.

W rozdziale 2 przedstawiono krótką charakterystykę wybranych i dostępnych na rynku wieloosiowych układów sterowania położeniem oraz omówiono możliwości programowej modyfikacji działania układu *NextMove ESB2*. Rozdział 3 zawiera propozycję budowy układu sterowania admityncyjnego z wykorzystaniem sterownika *NextMove ESB2*.

2. UKŁADY STEROWANIA POŁOŻENIEM

2.1 Sposoby programowania

Wieloosiowy sterownik ruchu *Gold Maestro* firmy Elmo [13,15,19] z systemem operacyjnym Linux z rozszerzeniem czasu rzeczywistego umożliwia sterowanie do 16 osi równocześnie. Udostępnia tryb sterowania położeniem, prędkością, momentem, interpolacji, bazowania. Programowany jest w języku MAXL przypominającym składnią język C, lub zgodnie z normą IEC 61131-3, z wykorzystaniem języka tekstu strukturalnego (ST), diagramów bloków funkcyjnych (FBD), języka drabinkowego (LD) oraz języka sekwencji działań (SFC). Trajektoria ruchu generowana jest na podstawie wektora PVT, określającego kolejne punkty, prędkości, czasy przez funkcje z biblioteki Motion Library. Dostępny jest również interfejs API (Multi-Axes Supervisor API Library), umożliwiający z

poziomu programu użytkownika komunikację ze sterownikiem oraz wywoływanie dostępnych przez interfejs funkcji.

Wielosiowe sterowniki ruchu z rodziny *ACR 9000* firmy Parker [8,9], umożliwiają sterowanie do 16 osi, w tym interpolację liniową 8 osi. Mogą być programowane w języku AcroBASIC, który umożliwia obsługę wejść oraz zaawansowane funkcje sterowania ruchem lub w innych językach zgodnych ze standardem IEC 61131-3. Aplikacja ComACRServer umożliwia komunikację pomiędzy sterownikiem a komputerem PC i jest kompatybilna z technologią OLEAutomation, np.: Microsoft.Net, Visual Basic, Visual C++ (C#), Delphi czy pakietami oprogramowania zgodnymi z COM, takimi jak Wonderware's Factory Suite 2000, LabView.

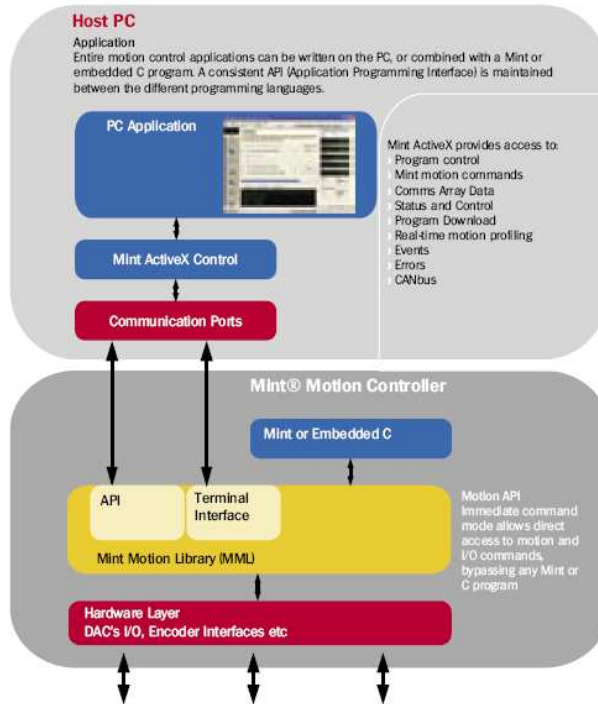
Wielosiowy sterownik ruchu *EPOS P 24/5* firmy Maxon [10,11] posiada wbudowany moduł mocy dla silników DC oraz BLDC o mocy do 120W. Umożliwia sterowanie do 32 osi w trybie sterowania położeniem, prędkością lub prądem. Środowisko EPOS Studio umożliwia programowanie zgodnie ze standardem IEC 61131-3 (ST, IL, FBD, LD, SFC) z wykorzystaniem dostępnych bloków funkcyjnych sterowania ruchem.

Rodzina sterowników ruchu *NextMove* firmy Baldor [18] jest zbudowana w oparciu o procesory sygnałowe firmy Texas Instruments. Sterowniki NextMove umożliwiają sterowanie zarówno w pętli otwartej - silnikami krokowymi, jak również w pętli zamkniętej z wykorzystaniem standardu $\pm 10V$ – układy serwo. Sterownik wersji NextMove e100 umożliwia sterowanie i interpolację do 16 osi. Programowanie układu realizowane jest w języku Mint, z wykorzystaniem aplikacji MintWorkbench. Język Mint, zbudowany w oparciu o składnię języka BASIC, umożliwia programowanie zadań ruchu z wykorzystaniem dostępnej biblioteki, obsługę sygnałów wejścia/wyjścia oraz udostępnia programowanie wielowątkowe (multitasking). Dostępna biblioteka profili ruchu zawiera: ruchy interpolowane z możliwością łączenia segmentów liniowych i kołowych, interpolację śrubową i typu spline, funkcję ruchu stycznego do krzywej 2D (tangential knife), funkcję elektronicznej przekładni, elektronicznej krzywki.

Udostępniony interfejs w postaci kontrolki ActiveX pozwala na zarządzanie sterownikiem (uaktualnianie Firmware, kompilację i przesyłanie programów, przesyłanie danych) z programu użytkownika na PC za pośrednictwem aplikacji MILServer. W porównaniu z przedstawionymi wcześniej sterownikami, istotną różnicą jest jego otwarta architektura. Producent udostępnia biblioteki funkcji w języku C, umożliwiające budowę wbudowanych aplikacji czasu rzeczywistego, zastępujących interpreter Mint. Ze względu na znaczenie tej funkcjonalności dla przedstawianego zadania zostanie ona omówiona bardziej szczegółowo.

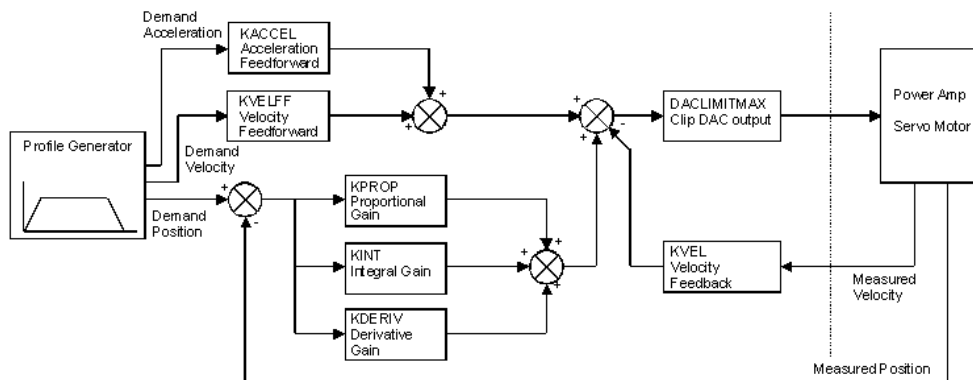
2.2 Niskopoziomowe programowanie sterownika NextMove ESB2

Udostępniona biblioteka funkcji (Mint Motion Library - MML) umożliwia obsługę sprzętu oraz kształtowanie profili ruchu i stanowi ten sam zestaw funkcji, z którego korzysta interpreter języka Mint oraz kontrolka ActiveX. Lokalizację biblioteki MML w strukturze logicznej sterownika przedstawia rysunek 1. W sterowniku może działać firmowy interpreter języka Mint lub utworzona przez użytkownika, zbudowana z wykorzystaniem biblioteki MML, aplikacja wbudowana.



Rys.1. Schemat blokowy struktury oprogramowania sterownika NextMove ESB2 [14]

Biblioteki MML pozwalają na zbudowanie aplikacji modyfikującej domyślną strukturę pętli sterowania czy generowania wartości zadanej. Na rysunku 2 przedstawiono schemat blokowy predefiniowanej w sterowniku pętli sprzężenia zwrotnego.



Rys.2. Schemat blokowy struktury pętli sterowania sterownika NextMove ESB2 [20]

Użytkownik ma możliwość modyfikacji działania układu przez zbudowanie i zainstalowanie w aplikacji wbudowanej funkcji [21]:

- Pre-Servo Loop – wywoływanej po odczytaniu rzeczywistej wartości położenia, a przed obliczeniem wartości uchybu

```
typedef void TPreServoLoop ( __uint8 ucAxis );  
void InstallPreServoLoop ( __uint8 ucAxis,  
TPreServoLoop *pHandler );
```
- Main Servo Loop – wywoływanej w miejscu układu PIDVFA w celu wyznaczenia według algorytmu użytkownika wartości wyjściowej do napędu

```
typedef void TMainServoLoop ( __uint8 ucAxis );  
void InstallMainServoLoop ( __uint8 ucAxis,  
TMainServoLoop *pHandler );
```
- Post Servo Loop – wywoływanej bezpośrednio przed wysłaniem wyznaczonej wartości wyjściowej do napędu

```
typedef void TPostServoLoop ( __uint8 ucAxis );  
void InstallPostServoLoop ( __uint8 ucAxis,  
TPostServoLoop *pHandler );
```
- Motion Profiler – wyznaczającej wartość zadaną położenia do zrealizowania przez napęd

```
typedef void TProfileLoop ( __uint8 ucAxis )  
void InstallProfileLoop ( __uint8 ucAxis, TProfileLoop  
*pHandler )
```
- Obsługi zdarzeń, np. zmiany wartości w tablicy komunikacji Comms

```
typedef void TCommsEventHandler ( __int32  
lCommsEventPending );  
void InstallCommsEventHandler ( TCommsEventHandler  
*pHandler );
```

Zawarte w bibliotece struktury danych udostępniają:

- dane pętli sterowania – rzeczywistą pozycję/prędkość, zadaną pozycję/ prędkość/ przyspieszenie, uchyb położenia, wartości parametrów pętli sterowania (struktura `_TServoLoopData`),
- dane osi napędowej – poprzednią pozycję zadaną, przyrost pozycji, współczynnik skalowania (struktura `_TAxisData`),
- dane osi pomocniczych – bieżącą pozycję, przyrost pozycji, współczynnik skalowania (struktura `_TAuxAxisData`),
- dane wejść sterownika – wartości sygnałów wejściowych cyfrowych i analogowych (struktura `_TIOData`).

Sterownik NextMove ESB2 zbudowany jest w oparciu o procesor sygnałowy firmy Texas Instruments TMS320VC33PGE. Budowa aplikacji wbudowanej wymaga kompilatora TI obsługującego wspomniany procesor (w pracy wykorzystano kompilator C v.5.11 ze środowiska Code Composer Studio 4.10) [17].

3. STEROWANIE ADMITANCYJNE – PROPOZYCJA IMPLEMENTACJI W STEROWNIKU NEXTMOVE ESB2

Do realizacji połączenia sterowania położeniowego ze sterowaniem siłowym wybrano regulator admitancyjny [6]. Impedancja mechaniczna określa relację pomiędzy siłą działającą na obiekt inercyjny, a jego prędkością. Admitancja mechaniczna jest określona jako odwrotność impedancji mechanicznej. Parametry regulatora admitancyjnego to inercja, tłumienie oraz sztywność [2]. Związek pomiędzy siłą F a prędkością V przedstawia zależność (1):

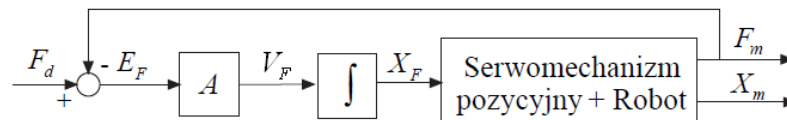
$$M \frac{d}{dt} V(t) + BV(t) + K \int V(t) dt = F(t) \quad (1)$$

gdzie: M – współczynnik inercji
 B – współczynnik tłumienia
 K – współczynnik sztywności

Stosując do powyższego równania transformatę Laplace'a przy zerowych warunkach początkowych otrzymujemy wyrażenie na admitancję (2):

$$A(s) = \frac{V(s)}{F(s)} = \frac{s}{Ms^2 + Bs + K} \quad (2)$$

Na rysunku 3 przedstawiono schemat układu z regulatorem admitancyjnym.



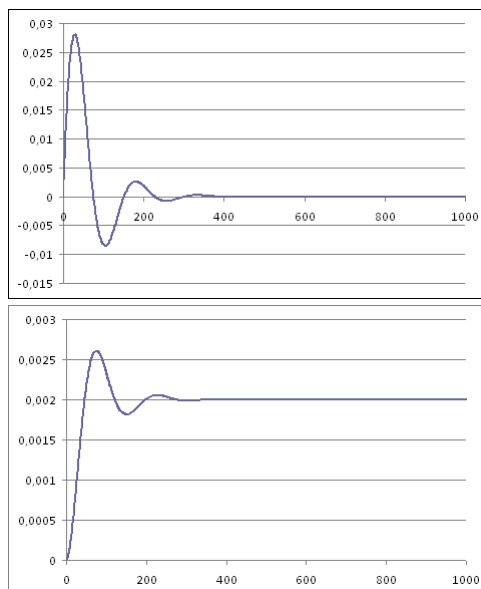
Rys.3. Schemat blokowy układu sterowania admitancyjnego [6]

Wielkością wejściową regulatora jest sygnał uchybu siły E_F , a wielkością wyjściową sygnał prędkości zadanej do napędu V_F . Zapisując równanie (1) w postaci dyskretnej oraz przekształcając go otrzymujemy wyrażenie na wartość prędkości V_F dla kolejnych przebiegów pętli sterowania.

$$V_F(n) = \frac{E_F(n) - KT \sum_{i=0}^{n-1} V_F(i) + M \frac{V_F(n-1)}{T}}{\frac{M}{T} + KT + B} \quad (3)$$

gdzie: T – okres próbkowania

Przykładową odpowiedź jednostkową regulatora admitancyjnego ($V_F(n)$ oraz $X_F(n)$) dla parametrów: $M=1$, $B=15$, $K=500$, $T=0,002$ przedstawiono na rysunku 4.



Rys.4. Odpowiedzi jednostkowe regulatora admitancyjnego (prędkość i położenie)

Do wyznaczenia wielkości oddziaływania układu (wirnika silnika) z otoczeniem zostanie wykorzystany pomiar prądu. Wartość momentu siły, z jaką wałek silnika działa na otoczenie, przy założeniu zerowej podatności i tłumienia układu przedstawia zależność (4):

$$F_m(t) = hi(t) - J\varepsilon(t) \quad (4)$$

gdzie: h – stała momentowa silnika

$i(t)$ – prąd twornika

J – moment bezwładności wirnika wraz z dołączonymi elementami

$\varepsilon(t)$ - przyspieszenie kątowe wirnika silnika

Po przekształceniu do postaci dyskretniej otrzymujemy (5):

$$F_m(n) = hi(n) - J \frac{\alpha(n) - 2\alpha(n-1) + \alpha(n-2)}{T^2} \quad (5)$$

gdzie: α – położenie kątowe wirnika silnika

T – okres próbkowania

Przedstawione zależności posłużą do modyfikacji działania sterownika NextMove. Zależność (5) zostanie wprowadzona do funkcji TProfileLoop i posłuży do wyznaczenia uchybu momentu $E_F(n)$. Zależność (3) zostanie wykorzystana do wyznaczenia

wartości sygnału *Demand Velocity* (rys. 2). Całka i pochodna tego sygnału to – odpowiednio - *Demand Position* oraz *Demand Acceleration* (rys. 2). Wyznaczone w tej funkcji wartości stanowią będą sygnały wejściowe do układu sterowania położeniem.

4. PODSUMOWANIE

W pracy przedstawiono przegląd sposobów programowania wybranych, dostępnych na rynku, układów sterowania położeniem. Większość układów udostępnia własny firmowy język programowania zadań ruchu lub możliwość programowania zgodnie z normą IEC 61131-3. Producenci układów sterowania udostępniają również biblioteki, zwykle zgodne ze standardem OLE, umożliwiające zarządzanie sterownikiem z programu użytkownika na komputerze PC. Tylko jeden z przedstawianych układów, sterownik NextMove, pozwalał na ingerencję użytkownika w sposób działania sterownika, poprzez możliwość opracowania własnej aplikacji wbudowanej. Dla tego sterownika zaproponowano sposób implementacji sterowania admitancyjnego.

Kolejnym etapem prac będzie budowa stanowiska badawczego i doświadczalna weryfikacja zaproponowanej koncepcji. Głównymi elementami stanowiska będą: wieloosiowy sterownik położenia NextMove ESB2 firmy Baldor, serwowzmacniacz ADS 50/5 firmy Maxon umożliwiający pomiar prądu, silnik DC 403A403 z enkoderem firmy Globe Motors oraz zasilacz DR 120/24 firmy Mean Well.

5. BIBLIOGRAFIA

- [1] Huang S., Schimmels J.M.: *Admittance selection for force-guided assembly of polygonal parts despite friction*. IEEE Transactions on Robotics, pp. 817–829, October 2004.
- [2] Jezierski E.: *Dynamika robotów*, Warszawa, WNT 2006.
- [3] Kumar R., Berkelman P., Gupta P., Barnes A., Jensen P.S., Whitcomb L.L., Taylor R.H.: *Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation*. Proc. Int. Conf. Robotics and Automation, pp.610-617, 2000.
- [4] Li X.: *Development of current sensor for cutting force measurement in turning*. IEEE Transactions on Instrumentation and Measurement, pp. 289–296, 2005.
- [5] Winiarski T., Zieliński C.: *Stanowisko do badania algorytmów sterowania pozycyjno-siłowego robotów*. Postępy robotyki. Sterowanie robotów z percepcją otoczenia, str. 85–94, WKŁ 2005.
- [6] Winiarski T., Zieliński C.: *Podstawy sterowania siłowego w robotach*. Pomiar Automatyka Robotyka 6/2008, str. 5-10
- [7] Winiarski T., Zieliński C.: *Sterowanie siłowe w systemach dwuramiennych*. Postępy Robotyki. Systemy i współdziałanie robotów, 2006, str. 267-276.
- [8] *ACR Command Language Reference*,
http://www.naic.edu/~phil/hardware/byuPhasedAr/floor/ACR_UG1_Command_Ref.pdf
- [9] *ComACRServer6 User's Guide*,
http://divapps.parker.com/divapps/emn/pdf/ACR/ComACRServer6_User_Guide.pdf

- [10] *EPOS P / MCD EPOS P, Programmable Positioning Controller, Programming Reference*,
http://test.maxonmotor.com/docsx/Download/Product/Pdf/EPOS_P_Programming_Reference_E.pdf
- [11] *EPOS P programmable position controller*,
http://www.electromate.com/db_support/downloads/296297.pdf
- [12] *Maxon motor control Operating Instructions*,
http://www.electromate.com/db_support/downloads/ADSUserManual.pdf.
- [13] *Maestro Software Manual*,
<http://www.elmomc.com/support/manuals/MAN-MASSW.pdf>
- [14] *Mint, Automation Software and Applications*,
<http://www.smartdrive.co.uk/BR1202-BMint.pdf>
- [15] *Motion Control Library, Tutorial*,
<http://www.elmomc.com/support/Samples-Tutorial/MAN-MLT.pdf>
- [16] *NextMove ESB2, Motion Controller Installation manual*,
http://www.baldor.com/downloads/manuals/_downloads/MN1957_01-10.pdf
- [17] *NextMove MINT™ Motion Library Programmer's Manual*,
http://www.baldor.com/downloads/manuals/_downloads/mn00112-598.pdf
- [18] *NextMove, Multi-axis Motion Controllers*,
http://www.baldor.com/pdf/literature/BR1202-C_NextMove.pdf
- [19] <http://www.elmomc.com/products/multi-axis-machine-motion-controller-g.htm>
- [20] Mint WorkBench – pomoc do programu.
- [21] *MintMT Embedded Programming Guide*,
<http://www.q-tech.hu/pdf/Baldor/MintMT%20Embedded%20Programming%20Guide.pdf>