

Stanowisko badawcze układów sterowania wieloosiowego napędami liniowymi poprzez magistralę EtherCAT

Streszczenie. W artykule przedstawiono stanowisko badawcze przeznaczone do badania wieloosiowych układów sterowania z serwonapędami liniowymi. Stanowisko składa się z komputera pełniącego rolę sterownika nadrzędnego, serwo sterowników oraz silników liniowych. Sterownik pracuje pod kontrolą systemu Linux z jądrem czasu rzeczywistego. Zaimplementowano również programowy stos EtherCAT do komunikacji z napędami. Przedstawiono budowę układu sterowania oraz strukturę oprogramowania sterownika. Przedstawiono również planowane badania, które będą realizowane na stanowisku dotyczące metod generacji optymalnych trajektorii ruchu, implementacji algorytmów regulacji położenia i prędkości w sterowniku oraz wpływu magistrali EtherCAT na jakość sterowania.

Abstract. The article presents a laboratory stand designed for researching multi-axis control systems with linear servodrives. The stand consists of a PC-based controller, servocontrollers and linear motors. The controller runs under Linux operating system with a real-time kernel. A software EtherCAT stack for communicating with drives was also implemented. The control system structure and the structure of the controller's software is presented. Research planned to be carried out on the laboratory stand is also presented. The research will deal with development of optimal trajectory generation methods, implementation of position and velocity loops in the controller and the influence of EtherCAT fieldbus with different communication parameters on control quality. (**Laboratory stand for researching control systems with linear servo drives controlled over EtherCAT fieldbus.**)

Słowa kluczowe: magistrala EtherCAT, silniki liniowe, Linux RTAI.

Keywords: EtherCAT fieldbus, linear motors, Linux RTAI.

Wstęp

Wieloosiowe maszyny sterowane numerycznie stanowią jedno z najważniejszych narzędzi stosowanych w wielu gałęziach przemysłu. Zaliczyć do nich można m.in. frezarki, tokarki, wycinarki ale również roboty i inne specjalistyczne urządzenia technologiczne [1]. Rosnące wymagania co do wydajności i dokładności realizowanych zadań zmuszają producentów maszyn i układów sterowania do ciągłego udoskonalania swoich produktów. Jedną z metod poprawy wydajności i/lub dokładności maszyn wieloosiowych jest implementacja w sterownikach wieloosiowych zaawansowanych metod sterowania wykorzystujących m.in. algorytmy optymalizacyjne czy też metody sztucznej inteligencji [2,3,4]. Z tego względu jest to obszar badawczy interesujący zarówno naukowców jak i przedsiębiorstwa produkujące i wykorzystujące maszyny CNC.

Serwonapędy liniowe z silnikami PMLSM (Permanent Magnet Linear Synchronous Motor) zdobywają coraz większą popularność wśród producentów maszyn ze względu na eliminację układu przeniesienia napędu w formie łożysk kulowych, przekładni pasowych lub zębatach. Brak dodatkowych układów mechanicznych pomiędzy silnikiem a suportem maszyny pozwala na znaczącą redukcję luzów i osiąganie dużych prędkości posuwu i przyspieszeń. Dzięki temu napędy te doskonale sprawdzają się w wysokowydajnych maszynach wieloosiowych takich jak wycinarki laserowe czy manipulatory.

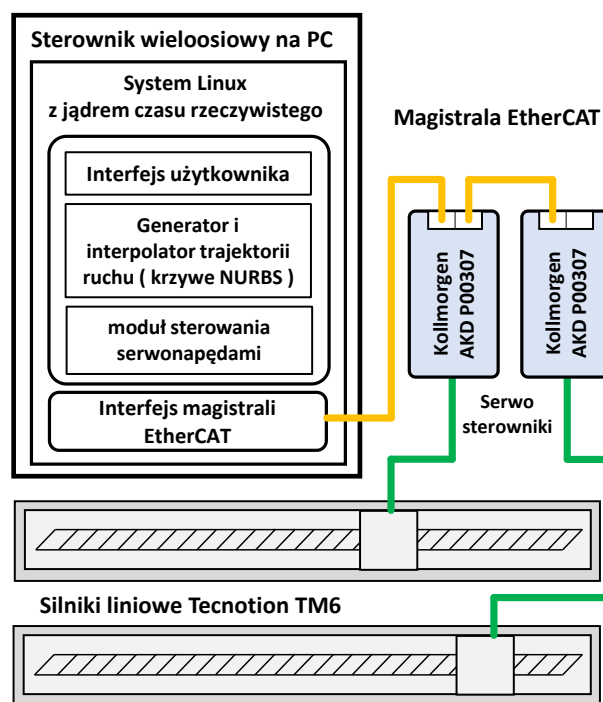
Aby możliwa była implementacja i badanie różnych metod sterowania maszynami wieloosiowymi opracowano prototypowy układ sterowania oraz zbudowano stanowisko badawcze. Układ sterowania jest otwarty co oznacza, że możliwa jest implementacja dowolnych metod generacji trajektorii ruchu, profilowania prędkości, interpolacji. Jednym z założeń było wykorzystanie standardowych, komercyjnych serwonapędów. Ponadto dzięki zastosowaniu bardzo szybkiej magistrali EtherCAT do komunikacji sterownika z serwonapędami, możliwa jest implementacja układu regulacji częściowo w sterowniku. Otwiera to możliwość połączenia zalet komercyjnych serwonapędów oraz zaawansowanych metod sterowania.

W artykule przedstawiono stanowisko badawcze przeznaczone do badania wieloosiowych układów

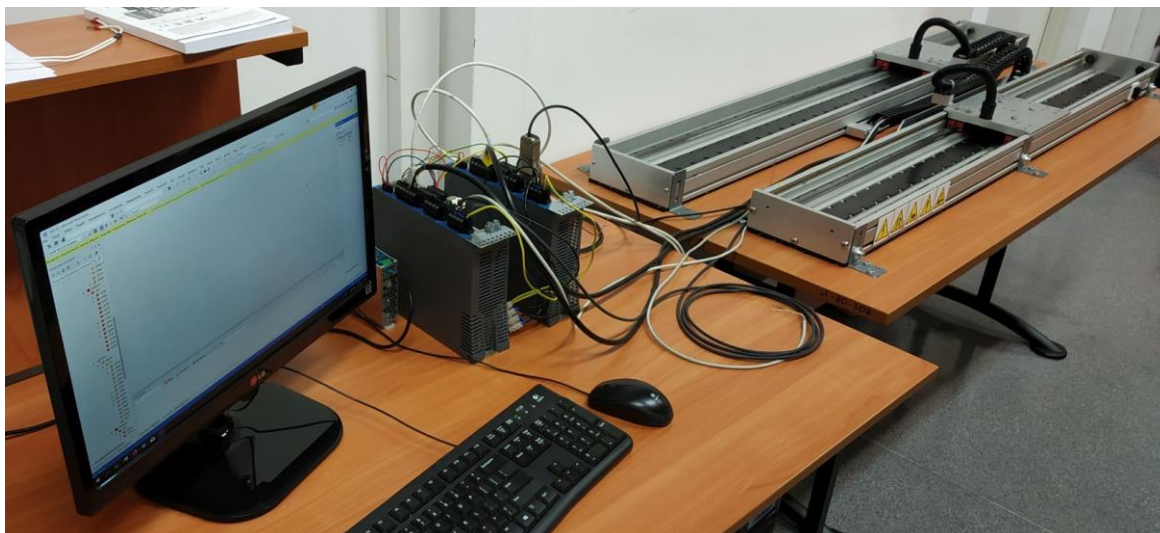
sterowania z napędami liniowymi. Omówiono budowę stanowiska ze szczególnym uwzględnieniem otwartego sterownika na bazie komputera PC z magistralą EtherCAT. Przedstawiono również przykładowe badania, które będą realizowane na stanowisku. Przedstawione stanowisko stanowi rozwinięcie prac autorów w zakresie opracowywania otwartych układów sterowania dla maszyn wieloosiowych z wykorzystaniem protokołów komunikacyjnych i systemów operacyjnych czasu rzeczywistego [5,6].

Budowa stanowiska badawczego

Schemat stanowiska badawczego przedstawiono na rysunku 1. Zdjęcie stanowiska przedstawiono na rysunku 2.



Rys.1. Schemat blokowy stanowiska badawczego



Rys.2. Widok stanowiska badawczego

Główne elementy składowe stanowiska to sterownik na bazie PC, serwo sterowniki oraz jednostki ruchu liniowego z silnikami liniowymi. Do budowy jednostek zastosowano silniki liniowe rdzeniowe Tecnotion TM6. Magnesy trwałe umieszczono na ramie aluminiowej. Siłownik (ang. forcer) połączono z blatem aluminiowym do którego dołączono dwie pary wózków osadzonych na prowadnicach tocnych. Silniki umożliwiają uzyskanie siły 120 N przy prądzie nominalnym 3 A i 240 N przy dopuszczalnym dwukrotnym przeciążeniu. Pozwala to na osiągnięcie maksymalnego przyspieszenia ok. 40 m/s² bez dodatkowej masy zamontowanej na wózku. Maksymalna osiągalna prędkość silników to 7 m/s lecz ze względów praktycznych (ograniczenia mechaniki, długość jednostek, maksymalne przyspieszenie) osiągalna jest prędkość 4m/s.

Sprzężenie zwrotne od położenia zapewniają linały optyczne firmy Renishaw. W jednej jednostce zastosowano linał inkrementalny TONIC generujący sygnał sinus - cosinus o rozdzielczości 20 μm na jeden okres. Druga jednostka wyposażona jest w linał absolutny RESOLUTE o rozdzielczości 5nm i interfejsie szeregowym BiSS-C.

Do sterowania silnikami wykorzystano serwo sterowniki Kollmorgen AKD P00307 o prądzie nominalnym 3 A i możliwości przeciążenia do 9 A. Serwo sterowniki komunikują się ze sterownikiem nadrzędnym za pośrednictwem magistrali EtherCAT. Każdy serwo sterownik wyposażony jest w dwuportowy przełącznik EtherCAT co umożliwia zastosowanie topologii liniowej bez dodatkowych urządzeń sieciowych. Możliwe jest również bezpośrednie sterowanie napędów z programu serwisowego za pośrednictwem dodatkowego portu Ethernet.

Sterownik wieloosiowy na bazie komputera PC

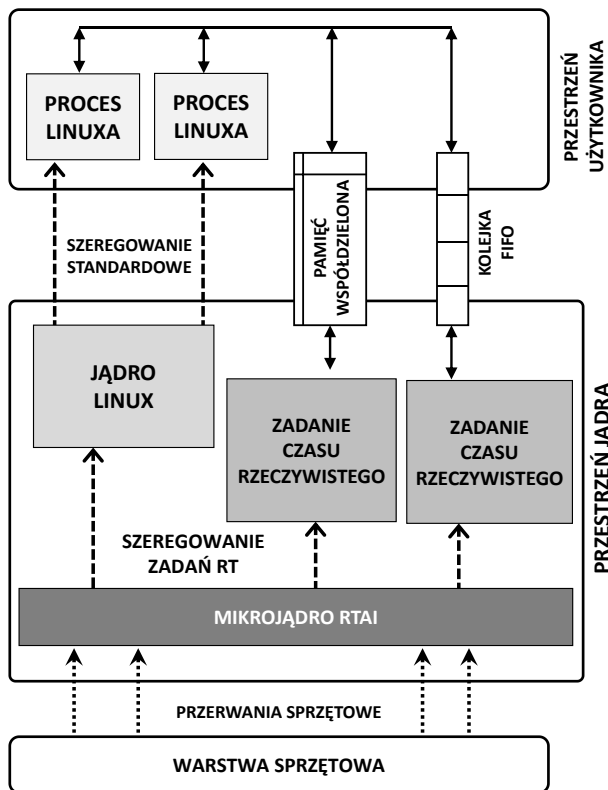
Sterownik wieloosiowy zaimplementowano na standardowym komputerze PC. W sterowaniu wieloosiowym wartości zadane (położenie, prędkość, prąd) przesyłane są do serwo sterowników w stałych odstępach czasu (najczęściej poniżej 1 ms). Wymaga to uzyskania determinizmu czasowego pracy sterownika. Stosując standardowe systemy operacyjne ogólnego przeznaczenia takie jak Windows czy Linux nie jest możliwe uzyskanie determinizmu czasowego. Dzieje się tak ze względu na konieczność zapewnienia równomiernego dostępu do zasobów sprzętowych dla wielu jednocześnie uruchomionych procesów (np. przeglądarki internetowej, odtwarzacza filmów, edytora tekstu). W tradycyjnych

rozwiązaniach determinizm czasowy, niezbędny do implementacji algorytmów sterowania napędami, uzyskuje się stosując zewnętrzne platformy sprzętowe na bazie mikroprocesorów lub układów FPGA [7]. W tego typu układach komputer pełni przede wszystkim rolę platformy dla środowiska programistycznego, interfejsu użytkownika oraz akwizycji danych.

Założeniem przy tworzeniu sterownika wieloosiowego było wyeliminowanie dedykowanych rozwiązań sprzętowych. Wykorzystanie komputera do sterowania wieloosiowymi układami napędowymi możliwe jest dzięki zastosowaniu systemów operacyjnych czasu rzeczywistego (ang. real-time operating system, RTOS). W systemach tego typu procesy krytyczne czasowo (np. algorytm sterowania, komunikacja z napędami) realizowane są zawsze w pierwszej kolejności zgodnie z ustalonymi priorytetami. Pozostałe procesy niekrytyczne czasowo (np. interfejs graficzny, obsługa plików) wykonywane są dopiero, gdy zostaną zrealizowane procesy krytyczne. Zasadniczym celem algorytmu szeregowania zadań systemu jest realizacja wszystkich zadań krytycznych przed upływem limitu czasowego (ang. deadline). Przekroczenie limitu traktowane jest jako awaria. Obecnie, ze względu na powszechne stosowanie procesorów wielordzeniowych, systemy RTOS umożliwiają podział zadań na różne rdzenie procesora tak by zadania krytyczne i niekrytyczne czasowo realizowane były tylko przez dedykowane zasoby sprzętowe.

Dostępnych jest szereg systemów operacyjnych czasu rzeczywistego. Można wyróżnić systemy dedykowane np. QNX, VxWorks oraz modyfikacje lub rozszerzenia systemów ogólnego przeznaczenia np. Linux RTAI, Windows RTX, TwinCAT, Linux Preempt-RT [8]. W artykule, ze względu na założenie utworzenia otwartego układu sterowania, zdecydowano się na zastosowanie otwartej, bezpłatnej modyfikacji systemu Linux – Linux RTAI. W systemie Linux RTAI modyfikacji uległo jądro systemu operacyjnego. Oprócz standardowego jądra systemu Linux funkcjonuje drugie jądro tzw. mikrojądro (ang. microkernel), którego zadaniem jest zarządzanie procesami czasu rzeczywistego. Mikrojądro jest nadrzędne w stosunku do jądra standardowego, które traktowane jest wraz z całym systemem Linux jako proces tła (niekrytyczny czasowo). Dzięki temu możliwe jest uruchamianie na jednym komputerze typowych aplikacji użytkowych systemu Linux oraz modułów czasu rzeczywistego. Moduły czasu rzeczywistego są kompilowane jako moduły jądra systemu.

Aby powiązać je z mikrojądrem konieczne jest wykorzystanie dedykowanego interfejsu programistycznego systemu czasu rzeczywistego. Wymiana danych pomiędzy poszczególnymi modułami czasu rzeczywistego oraz modułami przestrzeni użytkownika pracującymi pod kontrolą standardowego jądra Linux realizowana jest z wykorzystaniem mechanizmu kolejek FIFO oraz pamięci współdzielonej. Schemat blokowy architektury systemu czasu rzeczywistego Linux RTAI przedstawiono na rysunku 3.



Rys.3. Schemat blokowy struktury systemu operacyjnego czasu rzeczywistego Linux RTAI

Jako środowisko uruchomieniowe wykorzystano otwarte oprogramowanie do sterowania maszynami wieloosiowymi LinuxCNC [9]. LinuxCNC udostępnia pełen interfejs użytkownika oraz generator trajektorii ruchu umożliwiając zbudowanie w pełni funkcjonalnego sterownika maszyn CNC. W niniejszym stanowisku wykorzystano jedynie środowisko uruchomieniowe tego oprogramowania (Hardware Abstraction Layer, HAL) oraz interfejs programistyczny (Real Time Application Interface, RTAPI). Środowisko to umożliwia tworzenie własnych modułów czasu rzeczywistego pisanych w języku C i uruchamianie ich jako procesów czasu rzeczywistego. Środowisko HAL posiada tekstowy interfejs użytkownika pozwalający na ładowanie skompilowanych wcześniej modułów, powiązanie modułów z wątkami czasu rzeczywistego oraz przekazywanie danych między modułami. Możliwe jest również ładowanie modułów przestrzeni użytkownika, nie wymagających pracy w trybie czasu rzeczywistego. Moduły te są zwykłymi aplikacjami systemu Linux realizującymi np. interfejs użytkownika, odczyt, wizualizację i archiwizację danych pomiarowych czy też wstępne przetwarzanie programu sterującego. Uruchomienie modułów przestrzeni użytkownika w środowisku HAL umożliwia wymianę danych z modułami czasu rzeczywistego.

Podstawowy sposób wymiany danych polega na łączeniu ze sobą za pośrednictwem pamięci współdzielonej zmiennych udostępnionych przez każdy z modułów.

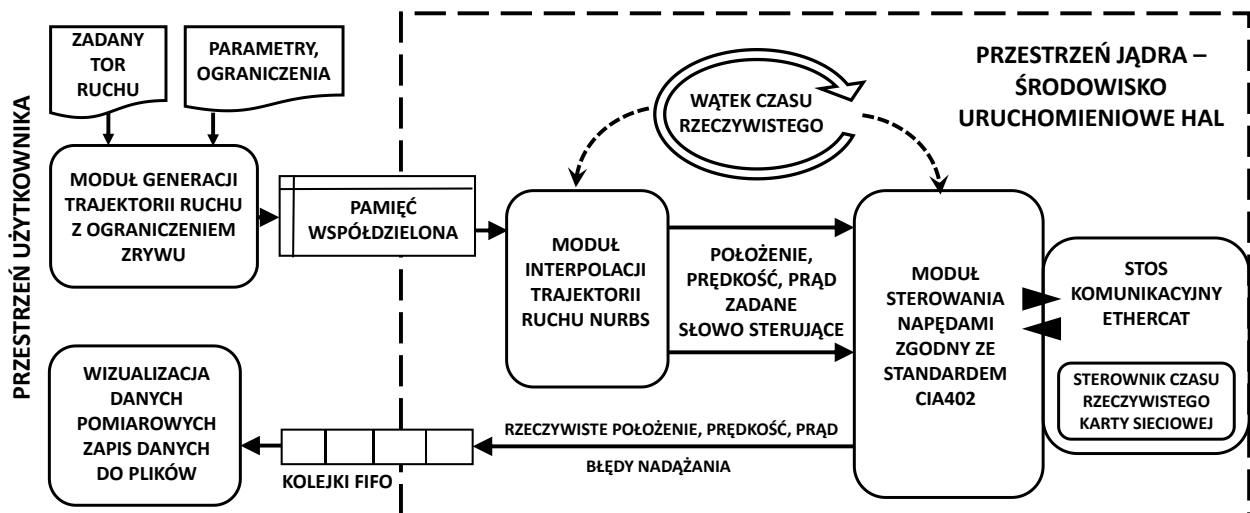
Zmienne określane są jako „piny” natomiast połączenia jako sygnały przez analogię do łączenia ze sobą elementów elektronicznych. Dzięki temu możliwa jest szybka konfiguracja i połączenie różnych modułów bez konieczności ich rekompilacji. Każdy z modułów może udostępnić funkcje, które można połączyć z wątkami czasu rzeczywistego. Funkcje połączone z wątkami są wywoływane cyklicznie w zależności od czasu cyklu danego wątku. Pozwala to na implementację różnych algorytmów sterowania np. regulatorów lub innych bloków funkcyjnych np. interfejsów komunikacyjnych. Każdy wątek i funkcja ma swój priorytet, o którym decyduje kolejność tworzenia wątków i przypisywania im funkcji. Nie wszystkie funkcje udostępniane przez moduły muszą być przypisane do wątków. Moduł może udostępnić funkcje innym modułom pełniąc rolę biblioteki ładowanej i usuwanej z pamięci w zależności od potrzeb.

Każdy sterownik wieloosiowy składa się z kilku zasadniczych bloków funkcyjnych. Interfejs użytkownika umożliwia wprowadzanie danych (np. zadanej trajektorii ruchu, parametrów, ograniczeń) oraz wyświetlanie danych odczytanych z urządzeń. Generator trajektorii ruchu wyznacza na podstawie zadanej toru ruchu chwilową prędkość styczną do toru ruchu. Wyznaczony w ten sposób profil prędkości uwzględnia ograniczenia maksymalnej prędkości, przyspieszenia i zrywu w poszczególnych osiach. Jest to o tyle istotne, że profilowanie prędkości pozwala na uniknięcie stanów nasycenia napędu oraz ogranicza błędy nadążania poszczególnych osi. Interpolator wyznacza położenia zadane przekazywane w stałych odstępach czasu do każdej z osi. Położenia zadane wyznaczone są na podstawie wygenerowanego profilu prędkości oraz zadanej toru ruchu.

W standardowej wersji oprogramowania LinuxCNC zaimplementowano relatywnie prosty algorytm trapezoidalnego profilowania prędkości. W tym wariancie profilowania ograniczone jest jedynie przyspieszenie natomiast zryw (pochodna przyspieszenia) nie jest uwzględniany. Tor ruchu może być zadany w postaci linii prostych lub łuków. W przypadku definiowania złożonych kształtów pociąga to za sobą konieczność ich aproksymacji dużą liczbą krótkich odcinków liniowych. Nieciągłość zadanej toru ruchu skutkuje skokowymi zmianami prędkości posuwu co negatywnie wpływa na dokładność odtwarzania zadanej toru ruchu. W programie LinuxCNC zaimplementowano algorytm wygładzający tor ruchu. Powoduje on dodatkową utratę dokładności odwzorowania pierwotnego toru ruchu i spadek prędkości posuwu. Ponadto całość algorytmu generacji trajektorii ruchu, interpolacji i funkcji dodatkowych zawarto w jednym module czasu rzeczywistego. Powoduje to ograniczoną elastyczność rozwiązania i trudność we wprowadzaniu zmian w poszczególnych funkcjach systemu.

W opisywanym stanowisku autorzy opracowali i zaimplementowali zaawansowany algorytm profilowania prędkości uwzględniający ograniczenie zrywu (tzw. profilowanie po krzywej S) oraz automatycznie redukujący prędkość posuwu w zależności od krzywizny zadanej toru ruchu. We fragmentach toru ruchu o małej krzywiznie działanie algorytmu powodować będzie utrzymywanie możliwie wysokiej prędkości posuwu. Prowadzić to będzie do skrócenia czasu realizacji trajektorii ruchu.

Równocześnie zaimplementowano interpolator toru ruchu opisanego krzywymi wielomianowymi NURBS (Non-Uniform Rational B-Spline). Tor ruchu opisany krzywymi wielomianowym cechuje ciągłość geometryczna i gładkość dzięki czemu unika się nagłych zmian prędkości i przyspieszenia.



Rys.4. Schemat blokowy modułów programowych zaimplementowanych w układzie sterowania

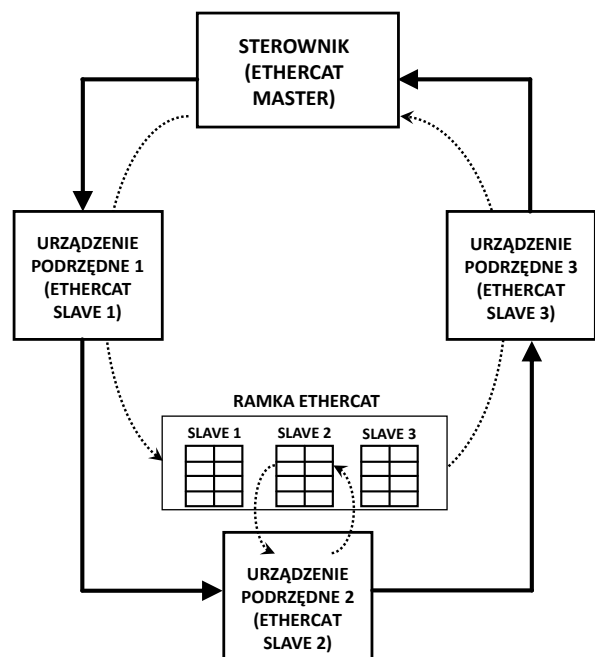
Krzywe NURBS są powszechnie stosowane w oprogramowaniu graficznym i CAD do opisu złożonych kształtów. Bezpośrednia interpolacja krzywych NURBS pozwala uniknąć dodatkowych etapów przetwarzania i aproksymacji toru ruchu dzięki czemu wzrasta dokładność jego odtwarzania. Algorytm generacji profilu prędkości i interpolacji NURBS, pomimo większej złożoności obliczeniowej działa on-line tzn. w trakcie realizacji zadanej trajektorii ruchu. Wprowadzenie nowego algorytmu profilowania ma szczególne znaczenie w przypadku badań dotyczących sterowania napędami liniowymi, mogącymi uzyskać wysokie prędkości i przyspieszenia. Schemat blokowy modułów programowych zaimplementowanych w sterowniku wykorzystanym w budowie stanowiska przedstawiono na rysunku 4.

Komunikacja z serwonapędami poprzez magistralę EtherCAT

W opisywanym stanowisku badawczym komunikacja pomiędzy sterownikiem na bazie PC oraz serwonapędami z silnikami liniowymi realizowana jest z wykorzystaniem magistrali EtherCAT. EtherCAT jest magistralą przemysłową wykorzystującą warstwę fizyczną Ethernet. Wyższe warstwy zostały zmodyfikowane tak aby zapewnić determinizm czasowy komunikacji. W większości przemysłowych magistral komunikacyjnych determinizm uzyskuje się poprzez cykliczne odpytywanie (ang. polling) poszczególnych urządzeń wykonawczych slave w sieci przez urządzenie nadrzędne (sterownik) tzw. master. W magistrali EtherCAT master wysyła jedną ramkę Ethernetową, która przekazywana jest po kolei do każdego urządzenia podrzędnego. Dedykowany układ komunikacyjny w węzłach sieci odczytuje dane z przychodzącej ramki oraz zapisuje dane zwrotne. Ponieważ operacja ta realizowana jest sprzętowo, wprowadzane opóźnienia są minimalne. Odczytane dane są następnie przekazywane do wyższych warstw stosu, gdzie są interpretowane i przekazywane do algorytmu sterowania. Ostatnie urządzenie w sieci retransmituje ramkę do mastera, który otrzymuje w ten sposób dane zwrotne ze wszystkich urządzeń w sieci. Protokół ten pozwala na uzyskanie krótszych czasów cyklu komunikacyjnego w porównaniu z odpytywaniem po kolei poszczególnych urządzeń w sieci [10,11]. Schemat komunikacji w sieci EtherCAT przedstawiono na rysunku 5.

Na komputerze pełniącym rolę sterownika wieloosiowego zaimplementowano procedury obsługi

magistrali EtherCAT (tzw. stos magistrali). Ponieważ standard EtherCAT jest otwarty, każdy może zaimplementować i wykorzystać własną wersję stosu magistrali. W opisywanym stanowisku wykorzystano otwarty (licencja GPLv2) kod źródłowy IGH EtherCAT Master. Dostępny otwarty kod źródłowy umożliwia realizację komunikacji z wykorzystaniem warstwy aplikacyjnej CANOpen. CANOpen został opracowany jako warstwa aplikacyjna magistrali CAN. Później zastosowano go również w magistralach EtherCAT oraz Ethernet Powerlink. W standardzie CANOpen definiuje się zbiór obiektów komunikacyjnych (Object Dictionary) stanowiący zbiór wszystkich zmiennych (procesowych, diagnostycznych, konfiguracyjnych) wykorzystywanych w komunikacji. Każda zmienna określana jest jednoznacznym identyfikatorem tzw. indeksem oraz w niektórych przypadkach także sub-indeksiem. Ze zmiennych zawartych w Object Dictionary część stanowią zmienne procesowe przesyłane cyklicznie w każdym cyklu komunikacyjnym.



Rys.5. Schemat komunikacji w sieci EtherCAT

Zmienne te przyporządkowane są do obiektu danych procesowych tzw. Process Data Object (PDO). Są to przede wszystkim wartości zadane, wartości rzeczywiste i słowa kontrolne i statusowe danego urządzenia. Możliwa jest również transmisja acykliczna realizowana w razie potrzeby poprzez obiekt danych serwisowych tzw. Service Data Object (SDO). Są to najczęściej dane konfiguracyjne np. nastawy regulatorów, tryb działania, kody błędów itp.

W opisywanym stanowisku wykorzystano serwonapędy serii Kollmorgen AKD. Serwonapędy te umożliwiają komunikację z wykorzystaniem magistrali EtherCAT z minimalnym cyklem komunikacyjnym 250 us. Do komunikacji wykorzystywana jest warstwa aplikacyjna CANOpen. Standard CANOpen określa szereg profili sprzętowych dla różnych rodzajów urządzeń. Dla napędów elektrycznych jest to profil CiA402. Profil ten definiuje standardowe adresy Object Dictionary dla najczęściej wykorzystywanych zmiennych procesowych. Są to m.in. położenie zadane, położenie rzeczywiste, prędkość zadana i rzeczywista, prąd zadany i rzeczywisty, błąd nadążania. Indeksy te są takie same w napędach różnych producentów zgodnych ze standardem CiA402. Oprócz tego zdefiniowana jest standardowa maszyna stanów napędu określająca sposób działania podczas uruchamiania, wystąpienia błędu, zatrzymania awaryjnego. Stan maszyny stanów wymuszany jest przez słowo kontrolne (Control Word), które musi być przesyłane w ramach transmisji cyklicznej. Informację o aktualnym stanie można odczytać ze słowa statusowego (Status Word), które również jest przesyłane w każdym cyklu.

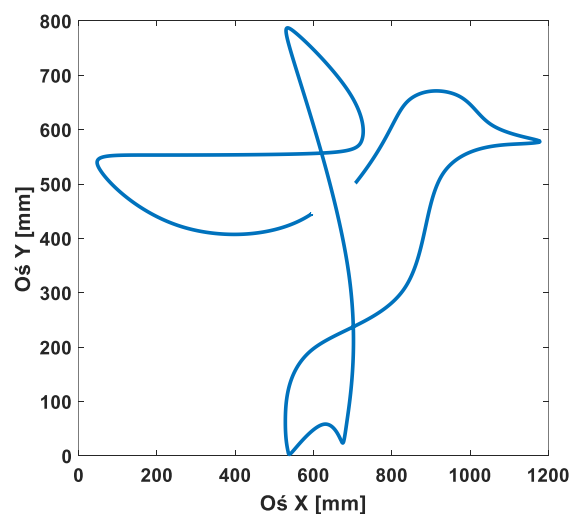
W wykorzystywanym kodzie źródłowym stosu magistrali EtherCAT nie było zaimplementowanych żadnych profili sprzętowych. Autorzy musieli dokonać implementacji sterowania napędem zgodnego ze standardem CiA402. Implementację wykonano w formie modułu czasu rzeczywistego bezpośrednio sprzężonego z modułem stosu komunikacyjnego. Dzięki rozdzieleniu obydwu warstw stosu komunikacyjnego możliwa jest niezależna modyfikacja i aktualizacja obydwu bloków programowych. Komunikacja pomiędzy modułem sterowania napędami a modułem stosu komunikacyjnego realizowana jest poprzez wywołania funkcji udostępnionych przez moduł stosu. Wymiana danych z modułami nadrzędnymi np. modułem interpolatora lub innymi modułami realizowana jest przez „piny” środowiska HAL. Moduł sterowania napędami udostępnia piny będące odpowiednikami zmiennych Object Dictionary przesyłanych cyklicznie przez magistralę EtherCAT. Dzięki modularnej budowie sterowanie poprzez magistralę EtherCAT można rozszerzać o dodatkowe moduły sterujące do urządzeń różnych typów. Autorzy dodali możliwość komunikacji z rozproszonymi modułami wejść i wyjść zgodnymi z profilem sprzętowym CiA401. W przyszłości możliwa jest implementacja profili sprzętowych obsługujących inne rodzaje urządzeń.

Komunikacja realizowana jest z cyklem komunikacyjnym zależnym od częstotliwości realizacji głównego wątku czasu rzeczywistego. Wątek ten wywołuje funkcję modułu sterowania napędami, która z kolei wywołuje funkcję realizującą przygotowanie i wysłanie ramki EtherCAT oraz przetworzenie danych zwrotnych. Minimalny osiągalny czas cyklu zależy przede wszystkim od stabilności cyklu wątku (rozrzut czasowy cyklu), czasu realizacji wszystkich funkcji powiązanych z wątkiem oraz minimalnego czasu cyklu obsługiwanego przez napędy i inne urządzenia w sieci. W przypadku opisywanego stanowiska najkrótszy czas cyklu to 250 μs wynikający z ograniczeń serwonapędów Kollmorgen AKD.

Badania realizowane na stanowisku

Głównym tematem badawczym realizowanym na stanowisku jest opracowanie metod wyznaczania optymalnych trajektorii ruchu dla napędów liniowych. Na obecnym etapie prac opracowano algorytm generacji trajektorii ruchu działający on-line uwzględniający ograniczenia prędkości, przyspieszenia i zrywu oraz krzywiznę trajektorii ruchu opisaną jako krzywe NURBS.

W celu zademonstrowania działania generatora trajektorii ruchu zaimplementowanego na stanowisku przeprowadzono testy realizując przykładowy tor ruchu zdefiniowany w formie krzywej NURBS (rysunek 6). Na potrzeby eksperymentu przyjęto, że napędy liniowe działają w kinematyce kartezjańskiej. Generator trajektorii ruchu skonfigurowano tak aby ograniczyć prędkość, przyspieszenie i zryw do odpowiednio 2,5 m/s, 25 m/s² i 500 m/s³. Interpolacja toru ruchu dla wygenerowanej trajektorii ruchu realizowana była z cyklem 250 μs. Z takim samym cyklem przesyłano do serwonapędów położenia zadane wygenerowane w wyniku interpolacji.



Rys. 6. Testowa trajektoria ruchu NURBS

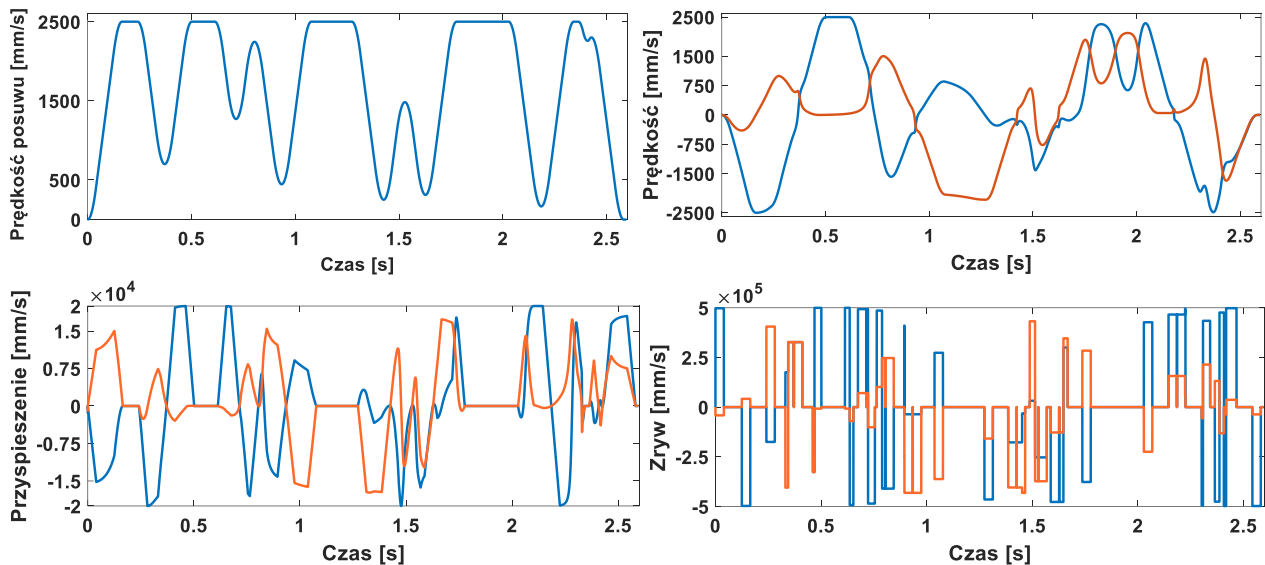
Na rysunku 7 przedstawiono przebiegi prędkości posuwu (stycznej do toru ruchu) oraz przebiegi prędkości, przyspieszenia i zrywu w obu jednostkach liniowych. Prędkość styczna jest automatycznie redukowana w fragmentach toru ruchu o dużej krzywiznie. Pozwala to na ograniczenie błędów nadążania w osiach maszyny i przyczynia się do dokładniejszego odwzorowania zadanego toru ruchu. Jednocześnie należy zauważyć, że prędkości, przyspieszenia i zrywy w osiach nie przekraczają założonych wartości maksymalnych. W ramach dalszych prac algorytm zostanie rozbudowany o blok optymalizacyjny dodatkowo korygujący wygenerowany profil prędkości. Pozwoli to w większym stopniu wykorzystać możliwości serwonapędów liniowych. Może to mieć miejsce np. w fragmentach trajektorii gdzie, żadna z wielkości nie osiąga wartości maksymalnych. Algorytm optymalizacyjny może uwzględniać dodatkowe kryteria takie jak maksymalne dopuszczalne błędy odtwarzania trajektorii ruchu lub maksymalny dopuszczalny prąd w serwonapędach.

Kolejnym tematem badawczym, realizowanym na stanowisku, jest implementacja regulatorów położenia i prędkości w sterowniku wieloosiowym. Pozwoli to na implementację zaawansowanych metod sterowania regulacji np. regulatorów adaptacyjnych, predykcyjnych, skrośnych, powtarzalnych i innych [12,13,14]. do sterowania komercyjnymi napędami.

Stanowisko posłuży również do badania komunikacji z wykorzystaniem protokołu EtherCAT pod kątem doboru

odpowiedniej konfiguracji sprzętowej komputera i systemu czasu rzeczywistego. Ponadto możliwe będzie testowanie

innych urządzeń EtherCAT np. modułów komunikacyjnych.



Rys. 7. Przebiegi prędkości stycznej do toru ruchu oraz prędkości, przyspieszenia i zrywy w osiach x (niebieski) i y (pomarańczowy)

Podsumowanie

W artykule przedstawiono stanowisko badawcze przeznaczone do realizacji badań dotyczących sterowania wieloosiowego serwonapędami liniowymi. Serwonapędy liniowe cechują się wysoką dynamiką i dokładnością pozycjonowania. Stosowane są w wysokowydajnych maszynach takich jak wycinarki laserowe. Omówiono budowę stanowiska bazującego na sterowniku zaimplementowanym na komputerze PC. Przedstawiono architekturę opracowanego sterownika, w którym zaimplementowano algorytm generacji trajektorii ruchu i interpolacji toru ruchu w postaci krzywych wielomianowych NURBS. Omówiono komunikację poprzez magistralę EtherCAT oraz jej implementację w opracowanym stanowisku. Przedstawiono planowane prace badawcze wraz z przykładowymi wynikami realizacji trajektorii ruchu przez algorytm generacji i interpolacji trajektorii ruchu. Opracowane stanowisko posłuży do realizacji szeregu prac badawczych dotyczących zwiększenia wydajności i dokładności układów sterowania wieloosiowego z napędami liniowymi.

Badania zostały sfinansowane ze środków Ministerstwa Nauki i Szkolnictwa Wyższego przeznaczonych na działalność statutową wspierającą rozwój młodych naukowców i doktorantów (grant wewnętrzny 1035-F)

LITERATURA

[1] Fitzpatrick M., Machining and CNC technology, *McGraw Hill Higher Education*, (2013)
 [2] Paprocki M., Erwiński K., Grzesiak L. M., Minimalizacja błędów nadążania w wieloosiowych maszynach CNC poprzez zastosowanie sterowania predykcyjnego z neuronowym modelem obiektu regulacji., *Przegląd Elektrotechniczny*, 88 (2012), n.4b, 169-174
 [3] Erwiński K., Paprocki M., Wawrzak A., Grzesiak L. M., PSO based feedrate optimization with contour error constraints for NURBS toolpaths, *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, (2016), 1200-1205
 [4] Szczepański R., Erwiński K., Paprocki M., Accelerating PSO based feedrate optimization for NURBS toolpaths using parallel computation with OpenMP, *22nd International Conference on*

Methods and Models in Automation and Robotics (MMAR), (2017), 431-436
 [5] Paprocki M., Wawrzak A., Erwiński K., PC-based CNC machine control system with LinuxCNC software, *Measurement Automation Monitoring*, 63 (2017), 15-19
 [6] Erwiński K., Paprocki M., Grzesiak L. M., Karwowski K., Wawrzak A., Application of Ethernet Powerlink for communication in a Linux RTAI open CNC control system, *IEEE Transactions on Industrial Electronics*, 60 (2013), n.2, 628-636
 [7] Przybył, A. Smola, J. Kimla, P., Rozproszony system sterowania obrabiarką numeryczną bazujący na sieci Ethernet Czasu Rzeczywistego, *Przegląd Elektrotechniczny*, 86 (2010), n.2, 342-346
 [8] Hambarde P., Rachit V., Jha S., he survey of real time operating system: RTOS, *International Conference on Electronic Systems, Signal Processing and Computing Technologies*, (2014)
 [9] Paprocki M., Wawrzak A., Erwiński K., Kłosowiak M., Elastyczny układ sterowania CNC maszyn bazujący na PC, *Mechanik*, 91 (2018), n.4, 299-303
 [10] Zurawski R., *Industrial Communication Technology Handbook*, CRC Press, (2014)
 [11] Wu X., Xie L., Performance evaluation of industrial Ethernet protocols for networked control application, *Control Engineering Practice*, 84 (2019), 208-217
 [12] Szczepański R., Tarczewski T., Grzesiak L.M., Zieliński M., Adaptive state feedback speed controller for PMSM based on Artificial Bee Colony algorithm, *Applied Soft Computing*, 83 (2019)
 [13] Wu J., Liu C., Xiong Z., Ding H., Precise contour following for biaxial systems via an A-type iterative learning cross-coupled control algorithm, *International Journal of Machine Tools and Manufacture.*, 93 (2015), 10-18
 [14] Tarczewski, T., Skiwski, M., Niewiara, L. J., Grzesiak, L. M., High-performance PMSM servo-drive with constrained state feedback position controller, *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 66, n. 1., 49-58, (2018)

Autorzy: dr inż. Krystian Erwiński, E-mail: erwin@fizyka.umk.pl; mgr inż. Andrzej Wawrzak, E-mail: awawrzak@fizyka.umk.pl; dr inż. Marcin Paprocki, E-mail: zwirek@fizyka.umk.pl; Uniwersytet Mikołaja Kopernika w Toruniu, Katedra Automatyki i Systemów Pomiarowych, ul. Grudziądzka 5, 87-100 Toruń; prof. dr hab. inż. Lech M. Grzesiak, Politechnika Warszawska, Instytut Sterowania i Elektroniki Przemysłowej; ul. Koszykowa 75, 00-662 Warszawa, E-mail: lmg@isep.pw.edu.pl;