

# SYNCHRONIZACJA PRACY SERWONAPĘDÓW AC W REALIZACJI ZŁOŻONYCH RUCHÓW ELEMENTÓW WYKONAWCZYCH ROBOTÓW I MASZYN

Mateusz MUSIAŁ, Krzysztof CHRAPEK, Mateusz GŁÓWKA

**Streszczenie:** W artykule opisano przeprowadzone badania efektów synchronizacji pracy dwóch serwonapędów AC w sposób programowy (sterowanie niezależne dwóch serwonapędów). W tym celu zbudowano stanowisko pomiarowe oparte o serwonapęd AC z prowadnicą z paskiem zębatym, na którym wykonano pomiary dokładności pozycjonowania i odwzorowania trajektorii dla interpolacji liniowej i kołowej elementu wykonawczego maszyny. Wyniki zestawiono na wykresach i w tabelach.

**Słowa kluczowe:** synchronizacja pracy, sterowanie, serwonapęd AC, interpolacja

## 1. Wstęp

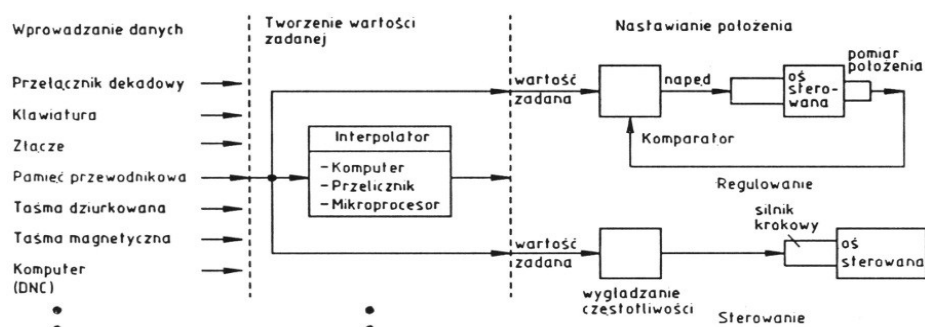
Elementy wykonawcze robotów i maszyn (np. chwytaki, narzędzia skrawające) wykonują złożone ruchy w celu realizacji określonej operacji technologicznej. Wymagania technologiczne sprawiają, że cechą charakterystyczną robotów, jak i współczesnych obrabiarek, są indywidualne napędy w każdej osi, dlatego wraz z rozwojem nowoczesnych urządzeń, następuje postęp w dziedzinie sterowania, które to polegają na synchronizacji pracy napędów wszystkich osi wykorzystywanych do wygenerowania danej trajektorii. Jakość synchronizacji przemieszczeń napędów ma kluczowe znaczenie dla dokładności wykonywanych operacji technologicznych [1,2].

Wysokiej klasy układy napędowe wykorzystywane są zarówno jako napędy elektryczne poszczególnych osi sterowanych w robotach, jak również jako napędy posuwu obrabiarek sterowanych numerycznie (NC, CNC). Serwonapędy (serwomechanizmy) są układami mechaniczno – elektryczno – elektroniczno – mikroprocesorowymi spełniającymi zadanie automatycznej regulacji położenia, prędkości i momentu. Wraz z odpowiednim układem mechanicznym umożliwiają wykonywanie przemieszczeń z dokładnością do tysięcznych części milimetra [3].

Za sterowanie układami wykonawczymi odpowiada zazwyczaj sterownik PLC sprzężony z interpolatorem i interpretatorem kodu programu. Sposób sterowania zależy od rodzaju napędów zastosowanych w maszynie. Pritschow [3] wyróżnia dwie konfiguracje układu sterowania w zależności od sposobu nastawiania wartości zadanej (np. położenia): jedną dla osi sterowanej opartej na serwonapędzie ze sprzężeniem zwrotnym (regulacja) oraz drugą z wykorzystaniem silnika krokowego i otwartej pętli sterowania, gdzie położenie zadawane jest impulsowo (rys. 1).

Dokładna kontrola trajektorii jest podstawowym wymogiem dla nowoczesnych obrabiarek CNC. Możemy starać się polepszać dokładność pozycjonowania każdej pojedynczej osi sterowanej poprzez stosowanie różnych strategii kontroli, jak regulator P o dużym wzmocnieniu, kontrolery typu feed-forward i ZPETC (zero phase error tracking controller). Jednak dobre wyniki śledzenia dla poszczególnych osi nie gwarantują

zmniejszenia błędu konturu wieloosiowego ruchu [4]. Termin "błąd konturu" określa tu wartość odchylenia położenia dla zadanej trajektorii i jest zdefiniowany jako odległość do niej w kierunku prostopadłym [5]. Inną metodą na zmniejszenie błędu konturu jest sterowanie typu cross-coupling (ze sprzężeniem krzyżowym). Koren [6,7] wprowadza symetryczną strukturę tego typu sterownika w celu poprawy dokładności odwzorowania trajektorii. Wśród metod sterowania synchronicznego wykorzystujących sprzężenie od wartości błędu pozycji wynikowej (synchronicznej) wyróżnić można [5]: dwuosiowe sterowanie ze sprzężeniem krzyżowym (cross-coupled), optymalne synchroniczne śledzenie ścieżki oraz metodę ciągłej kontroli konturu.



Rys. 1. Sterownice numeryczne – przetwarzanie informacji w układzie sterowania [3]

Xiao i Zhu [8] opisują konieczność zastosowania algorytmów synchronicznego sterowania ze sprzężeniem od wszystkich napędów, argumentując, że w konwencjonalnych systemach sterowania, jakkolwiek błąd spowodowany zaburzeniami w jednej pętli korygowany jest tylko wewnątrz tej pętli, podczas gdy druga pętla realizuje swoje dotychczasowe działanie. Ta, ze względu na brak komunikacji, może powodować niedopuszczalne błędy synchronizacji w ścieżce wynikowej. Sterowanie w układzie ze sprzężeniem krzyżowym umożliwia rozwiązanie tego problemu poprzez dzielenie informacji zwrotnej obu pętli regulacyjnych. Feng [9] zaproponował sterowanie w takim układzie dla synchronizacji napędów poszczególnych kół w robocie mobilnym w celu dokładnego odwzorowania ścieżki poprzez kompensację błędów orientacji. Sun i wsp. [10] zastosowali koncepcję cross-coupling do koordynacji ruchów przy współpracy wielu robotów w zadaniach montażu.

Możliwe są także inne podejścia do synchronizacji, m.in. wykorzystanie regulatora logiki rozmytej ze sprzężeniem krzyżowym czy neuro-regulatora. Jednakże, główną wadą tych technik jest to, że nie uwzględniają bezpośrednio niepewności modelu przy formułowaniu struktury układu sterowania. Metody te nie są odpowiednie w szczególności, kiedy podstawowym wymogiem projektu systemu sterowania jest wysoka dokładność [8].

W niniejszej pracy przedstawiono efekty wstępnych prac nad tematem sterowania synchronicznego serwonapędów przy maksymalnie uproszczonej strukturze układu sterowania. Stanowią one podstawę do rozwijanego aktualnie tematu sterowania i synchronizacji pod kątem budowy robota dydaktycznego.

## 2. Cel i zakres badań

Głównym celem prowadzonych prac jest badanie zachowania serwonapędu przy sterowaniu z wyzwaniem krokowym pod kątem możliwości realizacji tym sposobem

założonych przebiegów. Badania te prowadzone są w kierunku opracowania algorytmów interpolacji i sterowania dla synchronizacji pracy kilku napędów przy realizacji złożonych ruchów układów wykonawczych. W zadaniu tym niezwykle ważne są dwa aspekty: dokładność odwzorowania planowanych ruchów każdego z napędów oraz ich wzajemna synchronizacja czasowa dla uzyskania ruchu w danej interpolacji.

W pracy zaprezentowano wyniki realizacji synchronizacji programowej dwóch serwonapędów przy założeniu portalowej budowy układu wykonawczego w układzie kartezjańskim. Sprawdzenie możliwości dokładności odwzorowania trajektorii przeprowadzono przy:

- założonym sposobie sterowania,
- opracowanym algorytmie sterowania,
- opracowanym programie realizującym sterowanie w zadanej interpolacji.

### **3. Sposób sterowania**

Do przeprowadzenia wstępnych badań, których opis i wyniki zawarte są w tym opracowaniu, przyjęty został sposób sterowania wynikający z połączenia dwóch koncepcji rozwiązania układu napędowego przedstawionych przez Pritschowa [3]. Oś sterowana oparta jest na serwonapędzie z wbudowanym układem regulacji położenia, ale położenie zadawane jest impulsowo (krokowo), podobnie jak dla silników krokowych.

Opis programowania ruchów maszyn i robotów oparto na zasadzie programowania obrabiarki CNC. Najważniejszą częścią programu sterującego pracą robotów i maszyn jest część związana z programowaniem ruchu elementu wykonawczego. Do prawidłowego zaprogramowania tego ruchu konieczne są dane o: punkcie początkowym i końcowym ruchu, prędkości oraz torze [11].

#### **3.1. Założenia dla sterowania**

W przeprowadzonych badaniach synchronizacja polega na sterowaniu niezależnie podłączonymi napędami przy pomocy jednego programu sterującego, aby wykonywały one ściśle określony ruchy. Dla opracowywania sposobu synchronizacji programowej przyjęto następujące założenia:

- jednostką sterującą jest komputer PC wyposażony w port LPT,
- algorytm obliczeniowy utworzony w środowisku MatLab,
- napędy sterowane krokowo w trybie CW/CCW sterownika,
- programowane trajektorie to linia prosta i łuk.

Zdecydowano, że sterownik w trakcie synchronizacji będzie pracował w trybie CW/CCW, w którym sterowanie jest stosunkowo proste i intuicyjne – wysłanie zbocza narastającego na jedno wyjście portu powoduje obrót wału silnika o zadany kąt zgodnie z ruchem wskazówek zegara lub przeciwnie.

Programowane trajektorie to linia prosta oraz łuk, ponieważ są to podstawowe trajektorie wykonywane przez maszyny i roboty.

#### **3.2. Programowanie ruchu maszyn i robotów**

Tor ruchu jest to trajektoria, po której porusza się element wykonawczy, w trakcie przejścia od punktu początkowego do punktu końcowego, określane w procesie interpolacji. Interpolację należy interpretować jako procedurę sprzężenia ze sobą

niezależnie wykonywanych ruchów przez napędy w poszczególnych osiach, w celu otrzymania założonej, wypadkowej trajektorii narzędzia lub końcówki manipulatora [11].

Jednym z podstawowych ruchów jest ruch liniowy od punktu początkowego do punktu końcowego. W popularnym języku programowania obrabiarek CNC opartym na G-kodach, ruch ten oznaczany jest jako interpolacja G1. W trakcie programowania pracy maszyny konieczne jest jedynie podanie punktu końcowego wykonywanej trajektorii. Bardziej złożonym problemem jest ruch elementu wykonawczego po łuku, oznaczany jako G2 lub G3. W trakcie programowania takiego ruchu konieczne jest podanie dodatkowej informacji na temat toru, a mianowicie promienia okręgu, po jakim porusza się człon roboczy. Określając jedynie te trzy parametry, istnieją cztery możliwe trajektorie, jakie może wykonać narzędzie. Często przyjmuje się założenie, że ruch jest wykonywany po kącie z przedziału (0–180°), co ogranicza ilość możliwych trajektorii do dwóch. Rozróżnienie tych dwóch rozwiązań odbywa się poprzez podanie kierunku ruchu (G2 – zgodnie z ruchem wskazówek zegara, G3 – przeciwnie).

#### 4. Algorytm sterowania

W celu zaprogramowania założonych trajektorii, utworzono dwa oddzielne programy – jeden dla interpolacji liniowej, drugi dla interpolacji kołowej. Pomimo, że programy te różnią się od siebie, to posiadają podobną strukturę ogólną. W związku z tym opracowano ogólny algorytm programu dla obu przypadków:

- wczytanie danych wejściowych,
- wykonanie podstawowych obliczeń potrzebnych do wygenerowania trajektorii,
- generowanie trajektorii ruchu,
- generowanie sygnałów sterujących przemieszczeniem w dwóch osiach,
- wysłanie sygnałów sterujących.

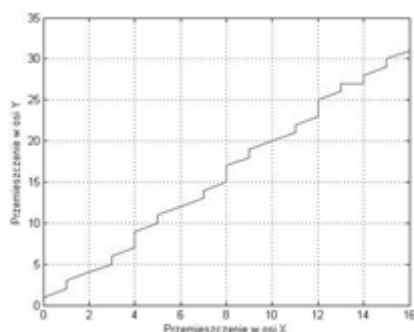
W pierwszej części programów wczytywane są zarówno dane konieczne do generowania trajektorii oraz dane dodatkowe, takie jak:

- rozdzielczość kątowa pracy silnika: 512imp/obr,
- odległość, jaką pokonuje element wykonawczy na jeden obrót silnika: 78mm,

#### 5. Program sterujący

W obu przypadkach generowana trajektoria ma postać funkcji schodkowej (rys. 2). Wynika to z impulsowego przebiegu sterowania napędu. Na osiach wykresu nie są przedstawione przemieszczenia podane w jednostce odległości, lecz ilości punktów, na jakie została podzielona cała trajektoria. Liczba tych punktów zależy od zastosowanej rozdzielczości. W wygenerowanej trajektorii przedstawione są minimalne skoki przemieszczeń w osiach. Takie przedstawienie trajektorii ułatwia generowanie sygnałów sterujących pracą serwonapędu.

Generowanie sygnałów sterujących, w obu przypadkach wygląda analogicznie. Sygnały sterujące, są ciągiem zer i jedynek. Program w pętli sprawdza, czy w kolejnych punktach trajektorii nastąpiło przemieszczenie w danej osi i w daną stronę o minimalną odległość. Jeżeli takie przemieszczenie nastąpiło, do wektora sterującego zapisywana jest wartość 1, natomiast, gdy nie nastąpiło – wartość 0. W ten sposób powstają cztery ciągi sygnałów sterujących (w obie strony dla osi X i Y). Następnie ciągi sterujące otrzymane w poprzednim kroku zostają wysłane na odpowiednią linię portu LPT.



Rys. 2. Fragment trajektorii generowanej w programie sterującym

Program sterujący przemieszczeniem liniowym, został napisany z wykorzystaniem interpolacji liniowej G1. Poniżej zamieszczono opis części programu specyficznych dla trajektorii liniowej.

W pierwszej części programu zostają wczytane współrzędne punktu końcowego trajektorii  $K=(x_k; y_k)$  konieczne do zdefiniowania w sposób jednoznaczny przemieszczenia elementu wykonawczego po linii prostej. Następnie obliczana jest długość toru ruchu  $s$  oraz liczba punktów na torze  $l_p$  przy zadanej rozdzielczości  $rozd_{z_{mm}}$ . Generowana jest właściwa trajektoria ruchu, która stanowi zbiór współrzędnych  $(x; y)$ , przedstawiający ilość przebytych odległości pomiędzy punktami dzielącymi trajektorię w każdej osi. Dla przemieszczenia liniowego, kolejne współrzędne generowane są z dwóch analogicznych zależności, które są wykonywane w kolejnych przejściach pętli programu:

$$p_x(i) = \sim \left( \frac{i \cdot x_k \cdot rozd_{z_{mm}}}{l_p} \right), \quad (1)$$

$$p_y(i) = \sim \left( \frac{i \cdot y_k \cdot rozd_{z_{mm}}}{l_p} \right), \quad (2)$$

gdzie:  $p_x(i)$ ,  $p_y(i)$  – obliczone kolejne współrzędne punktów trajektorii,  $i$  – liczba określająca numer kolejnej współrzędnej,  $x_k$ ,  $y_k$  – współrzędne punktu końcowego trajektorii,  $rozd_{z_{mm}}$  – rozdzielczość liniowej pracy sterownika,  $l_p$  – liczba punktów podziału toru ruchu.

Program sterujący przemieszczeniem elementu wykonawczego po łuku opracowano z wykorzystaniem interpolacji kołowej G2/G3. W pierwszej części drugiego programu zostają wczytane dane początkowe, opisujące programowany łuk: współrzędne punktu końcowego łuku  $(x_k, y_k)$ , promień łuku  $R$  oraz wskaźnik kierunku ruchu CW/CCW. Jednak, jak zaznaczono wcześniej, przy takim opisie istnieją cztery możliwe trajektorie w kształcie łuku. Zakładając, że element wykonawczy pokonuje krótszą z możliwych dróg, pozostają dwie możliwości, do odróżnienia których wykorzystywany jest trzeci z wymienionych parametrów. W kolejnej części programu dokonano obliczenia dwóch możliwych środków okręgów, dla których spełnione są powyższe dane początkowe i otrzymano współrzędne  $S_1=(x_{S1}; y_{S1})$  oraz  $S_2=(x_{S2}; y_{S2})$ . Następnie wykonywane są obliczenia, w wyniku których zostaje określone, który ze środków odpowiada za przemieszczenie w kierunku zgodnym, a który przeciwnym do ruchu wskazówek zegara. W celu odróżnienia, który ze środków

opisuje szukany łuk obliczono współrzędne wektorów łączących środki okręgów z punktami początku i końca łuku oraz skorzystano z podstawowych właściwości iloczynu wektorowego. Iloczyn wektorowy dwóch wektorów jest wektorem prostopadłym do płaszczyzny rozpiętej na tych dwóch wektorach. Dodatkowo wektor ten jest zorientowany w taki sposób, aby trójka wektorów była zgodna z orientacją prawoskrętnego układu współrzędnych OXYZ.

Korzystając z tych informacji zauważono, że iloczyny wektorowe wektorów W1 (S<sub>1</sub>A) i W2 (S<sub>1</sub>B) oraz W3 (S<sub>2</sub>A) i W4 (S<sub>2</sub>B), skierowane są wzdłuż osi Z, ale mają przeciwne zwroty. Istnieje zależność pomiędzy znakiem z-owej współrzędnej iloczynu wektorowego, a kierunkiem trajektorii łuku – gdy współrzędna ta jest ujemna, otrzymuje się trajektorię w kierunku przeciwnym do ruchu wskazówek zegara, natomiast gdy współrzędna jest dodatnia – w kierunku zgodnym.

Następnie obliczono długość toru ruchu  $s$  oraz kąt  $\beta$ , na którym rozpięty jest łuk, a także liczba punktów na łuku zgodnie z określoną rozdzielczością.

W kolejnej części programu wygenerowano trajektorie ruchu elementu wykonawczego w postaci zbioru współrzędnych  $(x_t; y_t)$ , korzystając z równania parametrycznego okręgu:

$$\begin{cases} x_t = x_s + R \cos \gamma \\ y_t = y_s + R \sin \gamma \end{cases} \quad (3)$$

gdzie:  $(x_t; y_t)$  – współrzędne punktów okręgu;

$(x_s; y_s)$  – współrzędne środka okręgu;

R – promień okręgu;

$\gamma$  – kąt, na jakim rozpięty jest łuk

W programie kolejne współrzędne trajektorii generowane są na podstawie dwóch analogicznych zależności:

$$p_x(i) = \sim((x_s + R \cos(\alpha_s)) \cdot rozd_z_{mm}) \quad (4)$$

$$p_y(i) = \sim((y_s + R \sin(\alpha_s)) \cdot rozd_z_{mm}), \quad (5)$$

gdzie:  $p_x(i)$ ,  $p_y(i)$  – obliczone kolejne współrzędne punktów trajektorii;

$(x_s; y_s)$  – współrzędne środka okręgu;

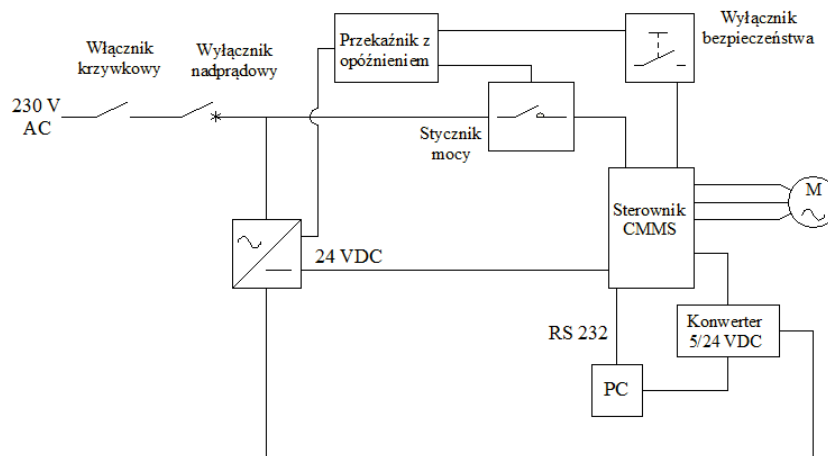
R – promień łuku;  $\alpha_s$  – przedziały kątów, na których rozpięte są szukane łuki;

$rozd_z_{mm}$  – rozdzielczość liniowej pracy napędu.

## 6. Stanowisko badawcze

Stanowisko badawcze opracowano w celu sprawdzenia rzeczywistej dokładności odwzorowania trajektorii przy założonym sposobie sterowania oraz opracowanym algorytmie i programie, które opisano w poprzednich rozdziałach. Oparte jest ono na napędzie firmy Festo złożonym z serwośilnika AC serii EMMS-AS, sterownika CMMS-AS oraz prowadnicy liniowej z paskiem zębatym serii EGC-70.

Stanowisko pomiarowe złożone jest serwonapędu wraz ze sterownikiem i prowadnicą liniową firmy Festo, układu zasilania i bezpieczeństwa oraz komputera PC i sterownika PLC do rejestracji sygnałów z enkodera (rys. 3).



Rys. 3. Schemat ideowy połączenia elementów na stanowisku

Sterownik CMMS firmy Festo w trakcie synchronizacji pracy napędów pracuje w trybie Synchronisation i ustawiony jest na sterowanie CW/CCW. Na wejścia CW i CCW podawany jest binarny sygnał sterujący generowany przez program.

## 7. Diagnostyka działania

Diagnostykę pracy serwonapędu przeprowadzono poprzez analizę sygnałów z enkodera podłączonego do osi z paskiem zębatym. W celu przymocowania enkodera do napędu wykorzystano jedną z piast przygotowanych na podłączenie silnika. Jako enkoder wykorzystano optoelektroniczny inkrementalny przetwornik obrotowo-impulsowy o rozdzielczości 2048imp/obr, który podłączono do osi napędu.

Dane z enkodera odbierane są poprzez sterownik PLC CX5020, który wyposażono w moduł interfejsu enkodera inkrementalnego EL5152.

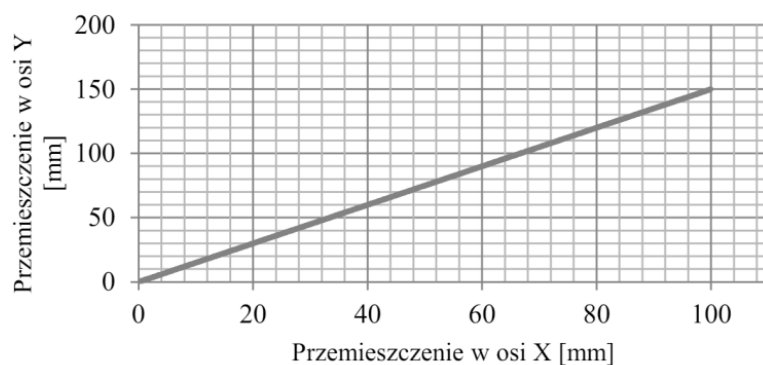
Diagnostykę przeprowadzono dla trajektorii:

- liniowej o punkcie końcowym (100;150)mm,
- kołowej o punkcie końcowym (100;120)mm i promieniu  $R=150$ mm.

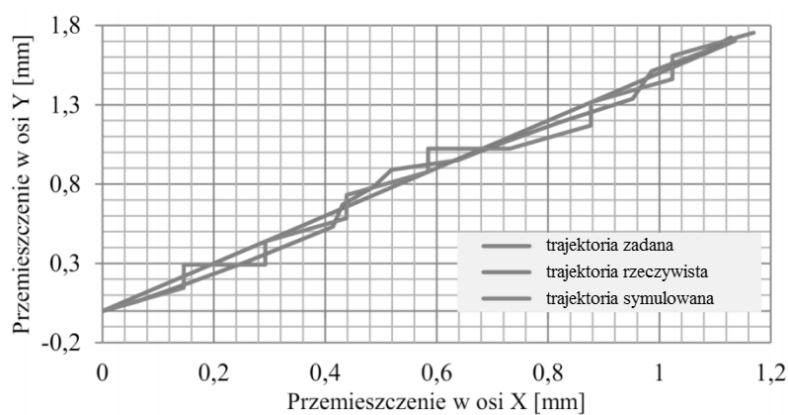
W obu przypadkach dane ze sterownika wysyłano dla rozdzielczości napędu 512imp/obr., dla której dokładność pozycjonowania jest na poziomie 0,1523mm.

Zarejestrowany przebieg dla interpolacji liniowej przedstawiono na rys. 4. Otrzymano założoną trajektorię, czyli linię prostą o punkcie końcowym (100; 150)mm.

Celem dokładniejszej analizy otrzymanego toru ruchu powiększono fragment uzyskanego przebiegu (rys. 5) i dodano również przebieg trajektorii zadanej (linii prostej) oraz trajektorii symulowanej. Przebiegi te nie pokrywają się, co wynika z idei sterowania impulsowego – silnik wykonuje przemieszczenia w sposób skokowy, które rozpoczynają się w momencie wysłania sygnału do sterownika. Dodatkowo współrzędne poszczególnych punktów są obliczane w sposób przybliżony ze skończoną dokładnością. Rzeczywisty punkt końcowy trajektorii ma współrzędne (99,978; 149,97)mm, więc element wykonawczy osiągnął finalne położenie z założoną dokładnością (rzędu  $\pm 0,15$ mm).

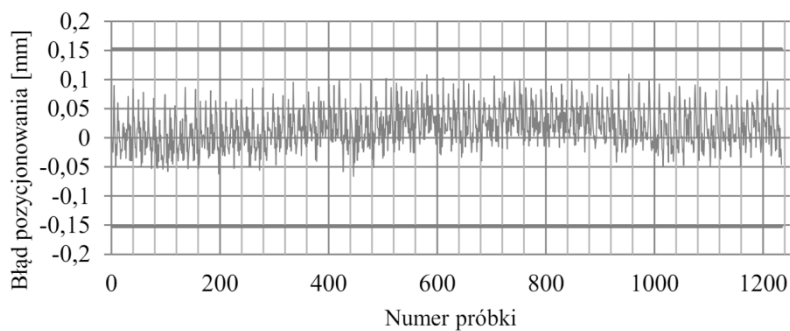


Rys. 4. Przebieg trajektorii odczytanej z enkodera, dla linii prostej



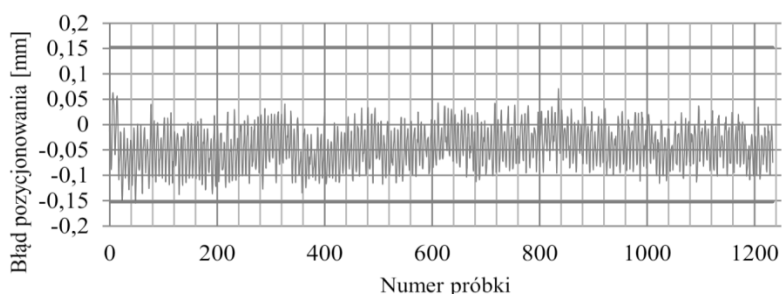
Rys. 5. Przebiegi dla przemieszczenia po linii prostej

W celu określenia maksymalnego błędu pozycjonowania obliczono różnice współrzędny trajektorii idealnej i rzeczywistej. Otrzymano w ten sposób dwie charakterystyki, przedstawiające odpowiednio błąd pozycjonowania w osi X (rys. 6) oraz Y (rys. 7).



Rys. 6. Błąd pozycjonowania w osi X dla przemieszczenia po linii prostej





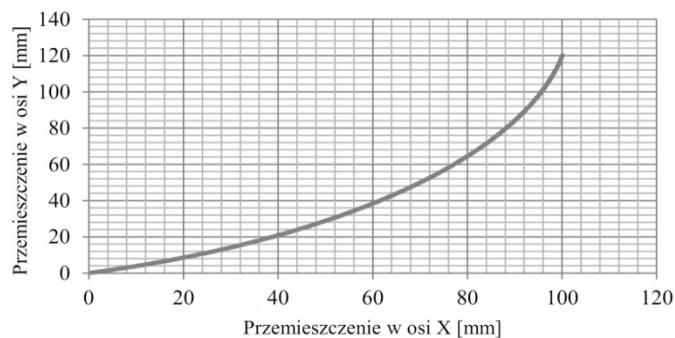
Rys. 7. Błąd pozycjonowania w osi Y dla przemieszczenia po linii prostej

Na wykresach przedstawiono dodatkowo dopuszczalne błędy pozycjonowania, wynikające z ustawionej rozdzielczości napędu. Na całej długości trajektorii wartości graniczne dokładności pozycjonowania zostały przekroczone w nielicznych przypadkach, co świadczy o dobrej jakości sposobu sterowania. Krytyczne wartości błędu pozycjonowania dla obu osi zestawiono w Tabeli 1.

Tab. 1. Krytyczne wartości błędu pozycjonowania w osi X i Y

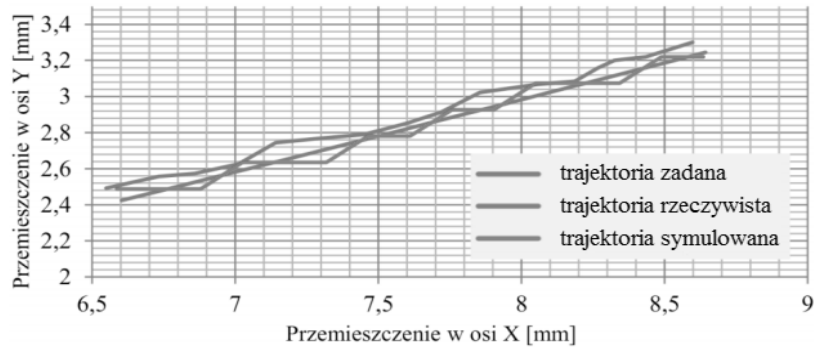
Maksymalny błąd pozycjonowania	Wartość błędu [mm]
w stronę dodatnią dla osi X	0,1092
w stronę ujemną dla osi X	-0,0663
w stronę dodatnią dla osi Y	0,0701
w stronę ujemną dla osi Y	-0,1548

Dla interpolacji kołowej, analogicznie jak dla diagnostyki przemieszczenia po linii prostej, pierwszą informacją jaką otrzymano, był kształt otrzymanej, rzeczywistej trajektorii dla założonego łuku (rys. 8). Otrzymano trajektorię w kształcie łuku, o współrzędnych punktu końcowego (100;120).



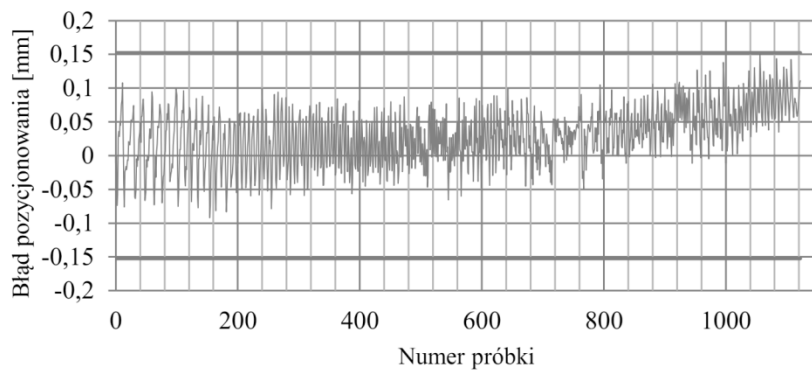
Rys. 8. Przebieg trajektorii odczytanej z enkodera, dla łuku

Na rys. 9 zestawiono przebiegi 3 trajektorii: zadanej, symulowanej, oraz rzeczywistej. Dla tego przypadku wszystkie otrzymane trajektorie są do siebie zbliżone, ewentualne niedokładności występują z tych samych powodów, co określono dla przebiegów trajektorii liniowej.

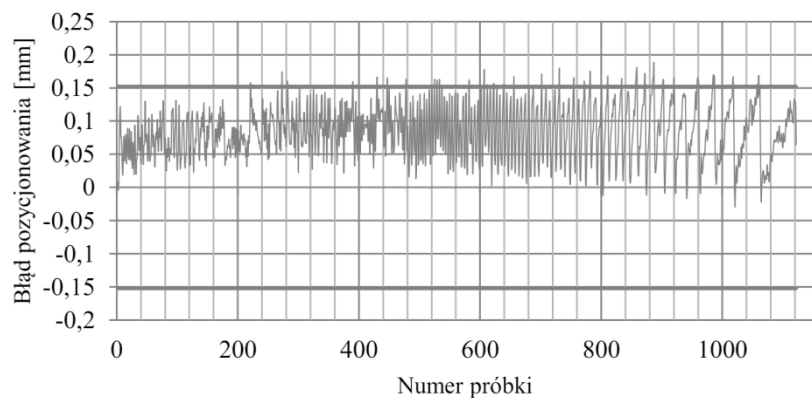


Rys. 9. Przebiegi trajektorii idealnej, symulowanej oraz zarejestrowanej

W kolejnej części dokonano analizy błędów pozycjonowania w osi X oraz w osi Y, które przedstawiono na rys. 10 i 11. O ile błąd pozycjonowania w osi X mieści się w założonych granicach dokładności pozycjonowania, o tyle dla przemieszczenia w osi Y można zaobserwować przesunięcie wartości w stronę dodatnią. W Tabeli 2 przedstawiono krytyczne wartości błędu pozycjonowania dla każdej osi.



Rys. 10. Błąd pozycjonowania w osi X dla przemieszczenia po łuku



Rys. 11. Błąd pozycjonowania w osi Y dla przemieszczenia po łuku

Tab. 2. Krytyczne wartości błędu pozycjonowania w osi X i Y

Maksymalny błąd pozycjonowania	Wartość błędu [mm]
w stronę dodatnią dla osi X	0,1506
w stronę ujemną dla osi X	-0,0919
w stronę dodatnią dla osi Y	0,1866
w stronę ujemną dla osi Y	-0,0298

## 8. Podsumowanie i wnioski

W pracy scharakteryzowano sposoby programowania przemieszczeń dla obrabiarek CNC, skupiając się na przemieszczeniu po linii prostej oraz po łuku, ponieważ właśnie takie ruchy są najczęściej wykorzystywane.

Utworzenie programów sterujących na komputerze PC, z wykorzystaniem środowiska MatLab, jest wygodne z punktu widzenia programowania oraz wykonywania obliczeń matematycznych, jednak w praktycznych zastosowaniach, nie jest to optymalny sposób sterowania serwonapędami ze względu na małą szybkość wykonywania operacji, co w konsekwencji powoduje zmniejszenie prędkości pracy napędu. Należy rozważyć wykorzystanie PLC bądź mikrokontrolera jako jednostki sterującej. Zoptymalizowałoby to cały proces pod względem osiągniętych prędkości pracy napędu.

W pracy badano realizację sterowania pracą napędów poprzez wysyłanie binarnych sygnałów sterujących z rozdzielczością 512imp/obr. Rozdzielczość ta wynika głównie z małej szybkości pracy jednostki sterującej. Przy zwiększaniu częstotliwości wysyłania danych, w sposób liniowy zmniejszają się możliwe do osiągnięcia prędkości.

Badania przeprowadzono dla sterowania krokowego pracą serwonapędu co wpłynęło na znaczne uproszczenie układu sterowania, w którym komputer PC wraz z opracowanym programem podłączony jest bezpośrednio ze sterownikami serwonapędów. Na komputerze wykonywane są zarówno obliczenia interpolacji jak i generowane sygnały sterujące.

Rzeczywiste przemieszczenia elementu wykonawczego, są zadowalające. Szczególnie dla trajektorii liniowej – różnice pomiędzy trajektorią rzeczywistą, a zadaną mieszczą się w dopuszczalnych granicach błędu, opisanego przez dokładność pozycjonowania dla danej rozdzielczości pracy napędu. Dla trajektorii kołowej różnice te są nieco większe oraz częściej przekraczana jest wartość graniczna (dla osi Y). Należy skorygować przyczynę powstawania takich niedokładności, która wiązać się może z błędami zaokrągleń w obliczeniach. Zwiększenie rozdzielczości pracy sterownika polepszy znacznie jakość odwzorowania trajektorii, gdyż obecna (na poziomie 0,15mm) jest niewystarczająca.

## Literatura

1. Morecki A., Knapczyk J.: Podstawy robotyki: Teoria i elementy manipulatorów i robotów, WNT, Warszawa, 1999.
2. Kosmol J.: Serwonapędy obrabiarek sterowanych numerycznie, WNT, Warszawa, 1998.
3. Pritschov G.: Technika sterowania obrabiarkami i robotami przemysłowymi, Oficyna Wyd. PWr, Wrocław 1995.
4. Yi-Ti Shin, Chin-Sheng Chen, An-Chen Lee: A novel cross-coupling control design

- for Bi-axis motion, *International Journal of Machine Tools & Manufacturing* 42 (2002), s. 1539-1548.
5. Miyoshi T., Terashima K., Maki Y.: Optimum Synchronous Control for Multiple-Axis Servo Systems, *Industrial Electronics Society, IECON 2000*, s. 1080-1086.
  6. Koren Y.: Cross-coupled biaxial computer control for manufacture systems, *ASME Journal Of Dynamic System, Measurement and Control* 102, s. 265-272.
  7. Koren Y., Lo C. C.: Variable-gain cross-coupling controller for contouring, *Annals of the CIRP* 40 (1), 371-374.
  8. Xiao Y., Zhu K. Y.: Optimal Synchronization Control of High-Precision Systems, *IEEE Transactions on Industrial Electronics* (53), (2006), s. 1160-1169.
  9. Feng L., Koren Y., Borenstein J.: Cross-coupling motion controller for mobile robots, *IEEE Control Syst. Mag.* (1993), vol. 13, no. 6, s. 35–43.
  10. Sun D., Mills J. K., Adaptive synchronized control for coordination of multirobot assembly tasks, *IEEE Trans. Robot. Autom.* (2002), vol. 18, no. 4, s. 498–510.
  11. Nikiel G.: Programowanie obrabiarek CNC na przykładzie układu sterowania Sinumerik 810D/840D, ATH, Bielsko-Biała, 2004.

Mgr inż. Mateusz MUSIAŁ,  
Dr inż. Krzysztof CHRAPEK,  
Inż. Mateusz GŁÓWKA  
Instytut Technologii Maszyn i Automatyzacji  
Politechnika Wrocławska  
50-371 Wrocław, ul Łukasiewicza 5  
tel. (0-71) 320 41 84  
e-mail: mateusz.musial@pwr.wroc.pl  
krzysztof.chrapek@pwr.wroc.pl  
mateusz.glowka90@gmail.com