

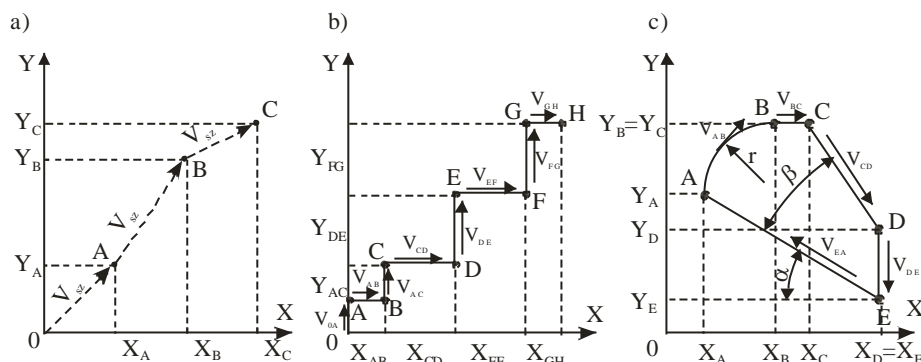
Jan KOSMOL^{1*}

INTERPOLACJA W KOMPUTEROWYCH UKŁADACH STEROWANIA NUMERYCZNEGO

W artykule omówiono zalety i wady interpolatorów spotykanych w układach sterowania numerycznego obrabiarek. Przedstawiono interpolatory stosowane w układach NC I generacji, bazujące na rozwiązaniach sprzętowych, tj. interpolatory liniowe i kołowe oraz interpolatory stosowane w układach II generacji, tj. CNC, bazujące na rozwiązaniach programowych. Porównano interpolatory sprzętowe i programowe z punktu widzenia możliwości eksploatacyjnych obrabiarek sterowanych numerycznie, tj. dokładności interpolacji i maksymalnych prędkości ruchów posuwu, decydujących o wydajności obróbki.

1. WPROWADZENIE

Jedną z najważniejszych właściwości eksploatacyjnych obrabiarek sterowanych numerycznie jest możliwość generowania ruchu narzędzia względem przedmiotu obrabianego po teoretycznie dowolnym torze. Na rys. 1 przedstawiono, w sposób symboliczny, trzy możliwe przypadki tworzenia toru narzędzia względem przedmiotu, zwane sterowaniem punktowym, odcinkowym i kształtowym. Pierwsze dwa przypadki



Rys. 1. Sposoby realizacji toru narzędzia w obrabiarkach sterowanych numerycznie: punktowo (a), odcinkowo (b) i kształtowo (c): V - prędkość posuwu, V_{sz} - przesuw szybki

Fig. 1. Methods of cutting tool path realization in numerically- controlled machine tools: point by point (a), line control (b) and continuous control (c): V - speed of feed, V_{sz} - fast speed of feed

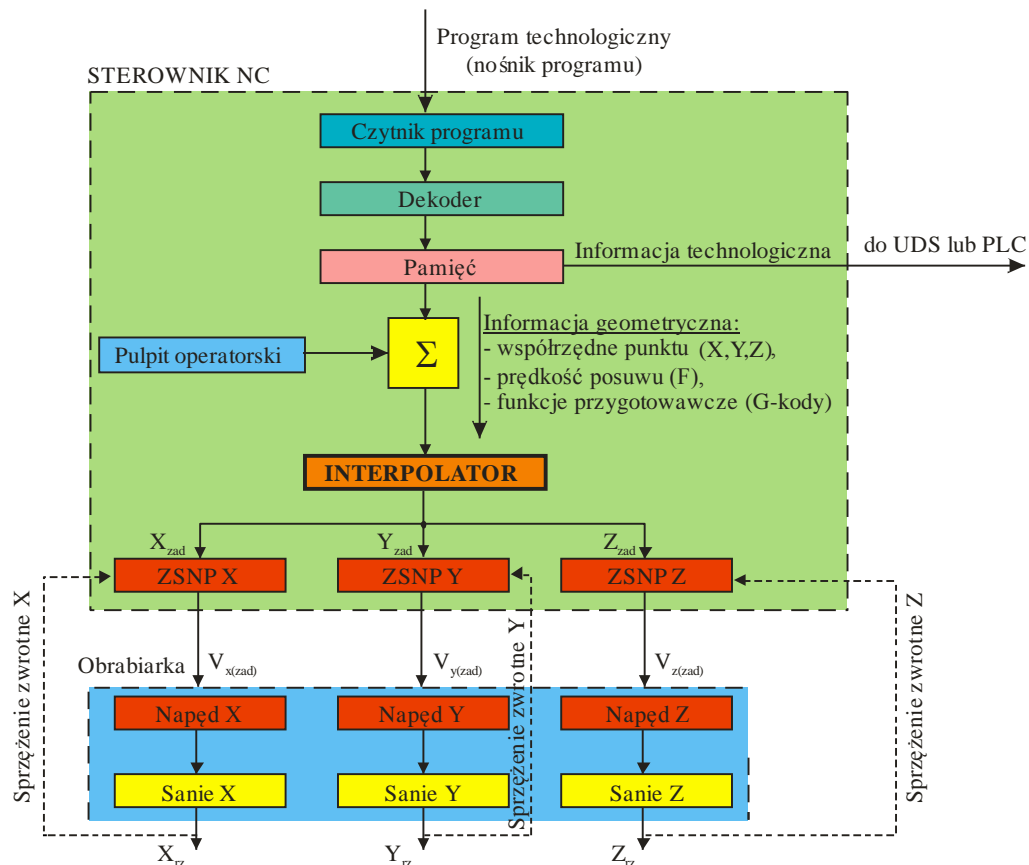
¹ Politechnika Śląska, Katedra Budowy Maszyn, Gliwice

* E-mail: jkosmol@polsl.pl

(punktowy i odcinkowy) można zrealizować również bez sterowania numerycznego, chociaż w praktyce są wykorzystywane sterowniki NC. Natomiast trzeci przypadek, zwany kształtowym, można zrealizować tylko przy użyciu sterowania numerycznego.

Zarys toru, po którym może przemieszczać się narzędzie lub przedmiot zależy przede wszystkim od zespołu (modułu) zwanego interpolatorem oraz od napędów posuwu.

Na rys. 2 przedstawiono, w sposób uproszczony, schemat klasycznego sterowania numerycznego obrabiarek [4].



Rys. 2. Schemat blokowy sterowania numerycznego obrabiarki: ZSNP - Zespoły sterowania napędami posuwu; X_{zad} , Y_{zad} , Z_{zad} - zadane przemieszczenia w kierunkach X, Y, Z; $V_{x(zad)}$, $V_{y(zad)}$, $V_{z(zad)}$ - sygnały wyjściowe ze sterowania NC, jako prędkości zadane [4]

Fig. 2. Block diagram of a numerically- controlled machine tool: ZSPN - control unit of feed drives; X_{zad} , Y_{zad} , Z_{zad} - given value of displacements in X, Y, Z directions, $V_{x(zad)}$, $V_{y(zad)}$, $V_{z(zad)}$ - output signals of CNC controller as given values of speed [4]

Oprócz klasycznych bloków (modułów) jak: czytnik, dekodery, pulpit i inne, w sterowniku znajduje się zespół interpolatora. Wielkościami wejściowymi do interpolatora są współrzędne punktu docelowego, zapisane w kolejnym wierszu programowym, prędkość posuwu i tzw. G-kody, które informują o sposobie interpolacji. Natomiast na wyjściu z interpolatora są sygnały, które można interpretować jako zadane przemieszczenia w poszczególnych osiach, oznakowane jako X_{zad} , Y_{zad} , Z_{zad} . Sygnały te stanowią równocześnie sygnały wejściowe do bloków oznakowanych jako ZSNP, tj. do zespołów

sterowania napędami posuwu. Zespoły te zawierają m.in. regulatory przemieszczenia, jako że sterowanie ruchem w poszczególnych osiach odbywa się na ogół w układzie automatycznej regulacji położenia. Rolą tych zespołów jest wykrywanie uchybu położeniowego w poszczególnych osiach i wygenerowanie sygnałów wyjściowych, które można interpretować jako sygnały prędkości zadanych w poszczególnych osiach (na rys. 2 oznakowane jako $V_{x(zad)}$, $V_{y(zad)}$, $V_{z(zad)}$). Te sygnały stanowią równocześnie wielkości wyjściowe ze sterowania numerycznego i wielkości wejściowe do napędów poszczególnych osi.

Przedstawiona na rys. 2 struktura sterowania numerycznego jest charakterystyczna dla sterowań NC, zwanych także sterowaniami I generacji.

2. INTERPOLATORY I ICH PODZIAŁ

Interpolator jest zespołem sterowania numerycznego, który odpowiada za ruch narzędzia względem przedmiotu po zaprogramowanym torze. W najogólniejszym przypadku, tor może zostać opisany w postaci funkcji:

$$f(x,y,z,\dots)=0 \quad (1)$$

gdzie: $f(x,y,z,\dots)$ - funkcja opisująca krzywą, która ma być torem narzędzia lub przedmiotu.

W obrabiarkach sterowanych numerycznie przyjęto opisywać tory w układzie współrzędnych prostokątnych, ale dla prawidłowej realizacji toru za pomocą niezależnych serwonapędów, funkcja opisująca krzywą musi być przedstawiona w postaci parametrycznej, np.:

$$\begin{aligned} f_x(x(p)) &= 0 \\ f_y(y(p)) &= 0 \\ f_z(z(p)) &= 0 \end{aligned} \quad (2)$$

gdzie: p - parametr,

f_x, f_y, f_z - funkcje parametryczne funkcji (1).

Każda funkcja parametryczna opisuje ruch pojedynczej, niezależnej osi sterowanej numerycznie, a geometryczna suma tych ruchów tworzy tor narzędzia lub przedmiotu.

Typową, często stosowaną interpolacją jest płaska (2D) interpolacja kołowa, której funkcja (1), w przypadku okręgu, którego środek znajduje się w początku układu współrzędnych, przyjmuje postać:

$$X^2 + Y^2 = R^2 \quad (3)$$

gdzie: R – promień okręgu,

a funkcje parametryczne (2) wyglądają następująco:

$$\begin{aligned} X(p) &= R \sin(\omega p) \\ Y(p) &= R \cos(\omega p) \end{aligned} \quad (4)$$

gdzie: p - parametr, którym jest czas t ,

ω - prędkość kątowna interpolacji ($\omega = V/R$, gdzie V - prędkość posuwu).

Zrealizowanie dowolnego toru (dowolnej funkcji (1)) jest trudne, a właściwie to niemożliwe. Dlatego, w praktyce można spotkać się tylko z pewną liczbą interpolatorów, które umożliwiają uzyskiwanie tylko niektórych kształtów torów.

Wszystkie spotykane interpolatory można podzielić, z uwagi na ich konstrukcję, na:

- sprzętowe (ang. hardware solution),
- programowe (ang. software solution).

Niezależnie od konstrukcji interpolatora, przetwarzanie sygnałów w interpolatorze odbywa się zawsze numerycznie. Z tego wynika, że sygnały wyjściowe z interpolatora są dyskretne. Z dyskretnej postaci sygnałów wyjściowych z interpolatora wynikają m.in. dokładność interpolacji (dokładność uzyskiwania funkcji (1), czyli dokładność toru) i prędkość interpolacji, czyli maksymalna prędkość posuwu podczas interpolacji.

Interpolatory sprzętowe są charakterystyczne dla sterowań numerycznych I generacji sterowania NC, gdzie przetwarzanie sygnałów odbywa się sprzętowo. Stosowane są dyskretne elementy elektroniczne, takie jak: liczniki rewersyjne, dzielniki częstotliwości, itp. Ze sprzętowej budowy interpolatorów wynikają ograniczenia co do dokładności interpolacji, jak i prędkości interpolacji.

Interpolatory programowe są charakterystyczne dla sterowań numerycznych CNC II generacji, gdzie przetwarzanie sygnałów odbywa się przy użyciu programów komputerowych, realizujących określone algorytmy obliczeniowe (modele numeryczne). Interpolatory te, w porównaniu do sprzętowych, są dokładniejsze i szybsze.

3. INTERPOLATORY SPRZĘTOWE

Interpolatory sprzętowe nie pozwalają na uzyskiwanie złożonych funkcji (1). Dlatego w sterowaniach I generacji stosowane są prawie wyłącznie interpolatory liniowe i kołowe, bardzo rzadko paraboliczne.

Synteza układów interpolujących opiera się na równaniach różniczkowych, których rozwiązaniem są funkcje parametryczne, opisujące ruch w każdej z osi. Takie postępowanie znane jest w literaturze pod nazwą DDA (Digital Differential Analyzer, czyli cyfrowy analizator równań różniczkowych) [5],[6]. Interpolatory sprzętowe składają się z modułów, tzw. integratorów cyfrowych, które w sposób przybliżony, cyfrowy, wykonują całkowanie tych równań parametrycznych, uzyskując na wyjściu przyrosty przemieszczeń w poszczególnych osiach.

Interpolatory liniowe realizują tory, których funkcja (1) ma postać:

$$Y = k X \quad (5)$$

gdzie: k - współczynnik kierunkowy prostej ($k = \operatorname{tg} \alpha$, gdzie: α - kąt nachylenia prostej), a funkcje parametryczne przyjmują postać:

$$\begin{aligned} X &= V_x t \\ Y &= V_y t \end{aligned} \quad (6)$$

gdzie: V_x , V_y - składowe prędkości posuwu w kierunkach X i Y , przy czym zachodzi związek:

$$V = \sqrt{V_x^2 + V_y^2} = \text{const} \quad (7)$$

gdzie: V - prędkość posuwu.

Z zależności (6 i 7) wynika, że w interpolacji liniowej $V_x = \text{const}$ i $V_y = \text{const}$. Ponadto można wykazać, że zachodzą następujące zależności pomiędzy składowymi prędkościami posuwu i prędkością posuwu:

$$\begin{aligned} V_x &= V \cos(\arctg(k)) = V \cos \alpha \\ V_y &= V \sin(\arctg(k)) = V \sin \alpha \end{aligned} \quad (8)$$

Przyrosty przemieszczenia $dx(t)$ i $dy(t)$ w kierunkach X i Y uzyskuje się różniczkując wyrażenia (6), mianowicie:

$$\begin{aligned} dx(t) &= V_x dt \\ dy(t) &= V_y dt \end{aligned} \quad (9)$$

W [5] i [3] przedstawiono inny, praktycznie stosowany w interpolatorach sprzętowych zapis przyrostów przemieszczeń $dx(t)$ i $dy(t)$, mianowicie:

$$\begin{aligned} dx(t) &= a \omega dt \\ dy(t) &= b \omega dt \end{aligned} \quad (10)$$

gdzie: ω - stała ($\omega = \frac{f}{p^r}$, gdzie: f - częstotliwość generatora, p - podstawa systemu liczenia,

$p=2$ lub $p=10$, r - pojemność rejestru),

a, b - parametry zapisywane do rejestrów, przed rozpoczęciem interpolacji.

Z porównania (9) i (10) wynika, że

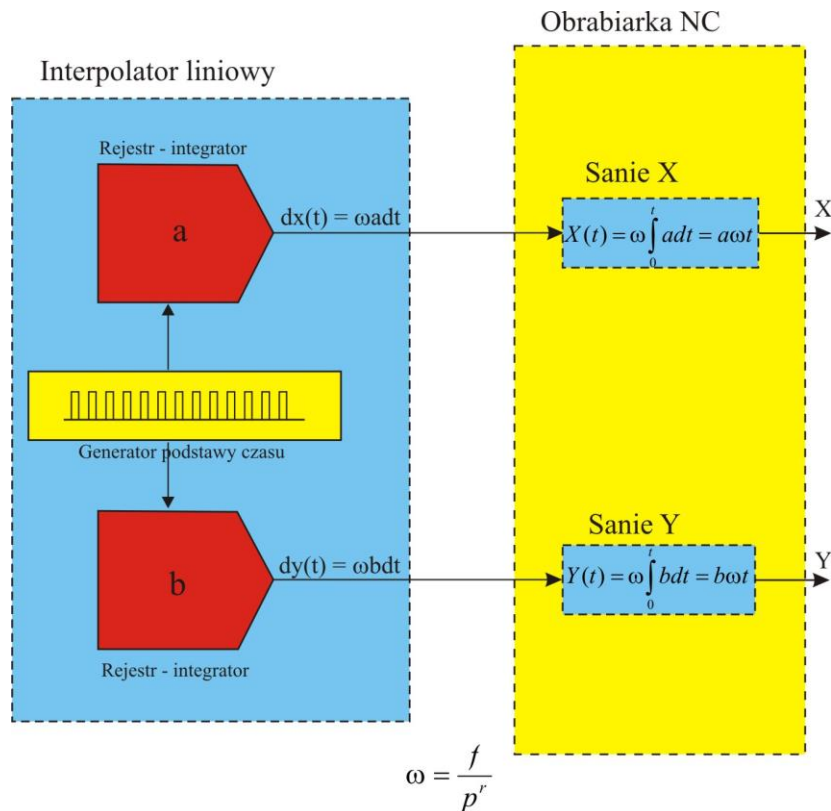
$$\begin{aligned} a &= \frac{V_x}{\omega} = \frac{V \cos \alpha}{\omega} \\ b &= \frac{V_y}{\omega} = \frac{V \sin \alpha}{\omega} \end{aligned} \quad (11)$$

a parametry a i b reprezentują prędkość ruchu w osiach X i Y .

Tak więc, podczas interpolacji liniowej ruchy składowe w osiach X i Y odbywają się ze stałymi prędkościami, których wielkości zależą od zaprogramowanej wartości posuwu V (funkcja F) i kąta nachylenia odcinka prostej ($\alpha = \arctg(k)$). Natomiast informacja o prędkościach ruchów w poszczególnych osiach jest zapisywana w rejestrach w postaci parametrów a i b . Schemat sprzętowego interpolatora liniowego przedstawia rys. 3.

Interpolator liniowy sprzętowy składa się z dwóch rejestrów - integratorów cyfrowych i generatora podstawy czasu o częstotliwości f . Przetwarzanie informacji w tych interpolatorach przedstawia się następująco [2],[10]: w rejestrach, pełniących funkcję integratorów, następuje całkowanie numeryczne, a ich zawartość odpowiada w przybliżeniu chwilowej wartości $X(t)$ i $Y(t)$ (wg (6)), ale zapisanych w postaci sumy impulsów (pojedynczemu impulsowi odpowiada elementarne przemieszczeniu δ , charakterystyczne dla sterowania i obrabiarki NC). Rejestr jest licznikiem impulsów, które albo są dodawane do zawartości rejestru albo odejmowane (zależy to od tego, czy zawartość rejestru jest

funkcją rosnącą czy malejącą). Na wyjściach z rejestrów - integratorów otrzymuje się impulsy - przyrosty $dx(t)$ i $dy(t)$, które w przybliżeniu odpowiadają zależnościom (9). Impulsy te wchodzi następnie do napędów sań (za pośrednictwem zespołów ZSNP (rys. 2)), które także pełnią rolę integratorów przemieszczenia.



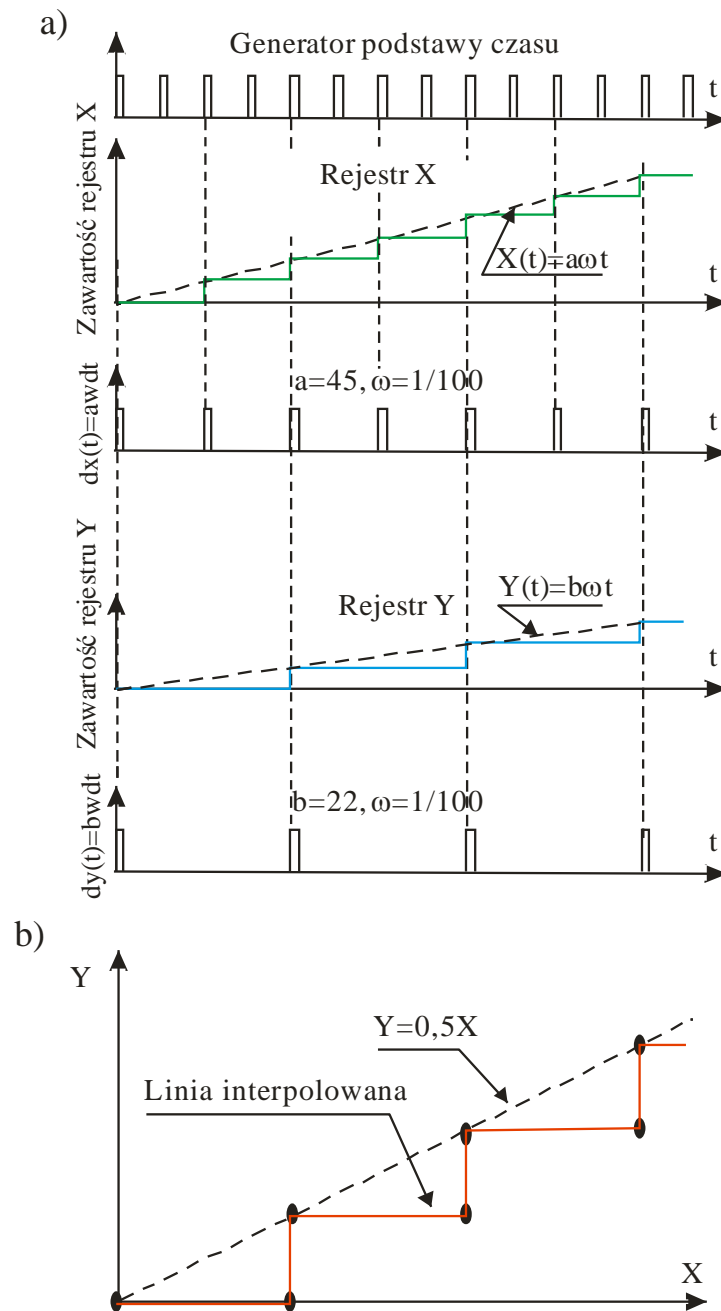
Rys. 3. Przykładowy schemat sprzętowej realizacji interpolatora liniowego: f - częstotliwość generatora, p - podstawa systemu liczenia ($p=2$ lub $p=10$), r - pojemność rejestru, ω - stała [5],[3]

Fig. 3. An example of hardware linear interpolator unit: f - frequency of generator, p - basis of numeral system ($p=2$ or $p=10$), r - buffer capacity, ω - constant [5],[3]

W przypadku interpolatora liniowego, jeszcze przed rozpoczęciem interpolacji, do rejestrów wpisuje się parametry a i b , które zgodnie z zależnościami (11) są miarą prędkości ruchu V_x i V_y w obu osiach. Rejestry - integratory całkują w przypadku interpolatora liniowego funkcje stałe a i b , a przyrosty impulsów $dx(t)$ i $dy(t)$ są wprost proporcjonalne do nich, czyli do prędkości V_x i V_y (na rys. 3 prędkości V_x i V_y są niejawne i ukryte w parametrach a i b).

Na rys. 4 przedstawiono przykładowe przebiegi sygnałów w interpolatorze liniowym sprzętowym. Przyjęto, że współczynnik kierunkowy prostej $k = 0,5$ ($\alpha=26,6^\circ$), co oznacza, że parametry $a = 44,72 \approx 45$ natomiast $b = 22,36 \approx 22$ (dla $p=10$, $r=2$, $\omega=1/100$). Integrator X całkuje dwukrotnie szybciej impulsy z generatora podstawy czasu niż integrator Y. W rezultacie, w rejestrze X przybywa dwukrotnie więcej impulsów niż w rejestrze Y. Ponieważ przemieszczenia w osiach X i Y są wprost proporcjonalne do zawartości rejestrów, to przemieszczenie w osi X jest dwukrotnie większe niż w osi Y.

W interpolatorze sprzętowym jednym z najważniejszych parametrów jest elementarne przemieszczenie δ , które odpowiada przyrostowi stanu licznika o jeden impuls. Wielkość ta jest charakterystyczna dla sterowania numerycznego, zależy od budowy interpolatora (pojemności licznika, maksymalnej częstotliwości generatora), ale także od łańcucha kinematycznego posuwu (skoku śruby pociągowej, przełożenia przekładni mechanicznej i innych). Typowa wielkość δ dla interpolatorów sprzętowych to $2 \mu\text{m}$. Uzyskanie mniejszego elementarnego przemieszczenia jest trudne. Tym samym, jest to jedno z ograniczeń dokładności interpolacji.

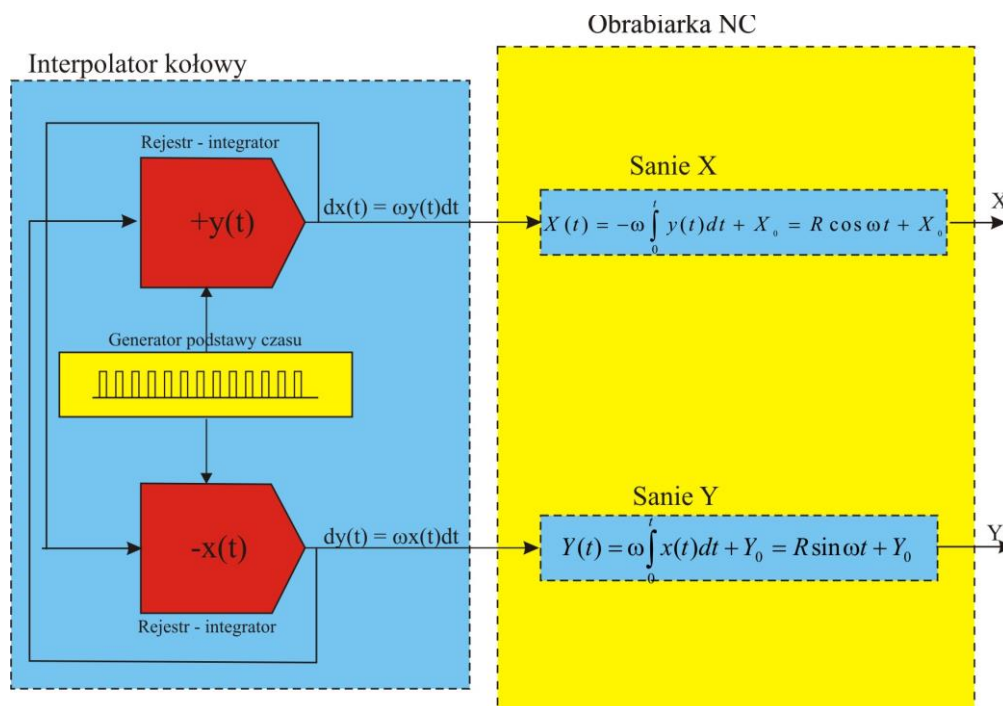


Rys. 4. Przykładowe przebiegi sygnałów w interpolatorze liniowym sprzętowym
 Fig. 4. An example of time performance of signals for linear hardware interpolator

Drugim, istotnym ograniczeniem jest maksymalna prędkość posuwu. Jej wartość zależy od częstotliwości generatora, maksymalnego, wykrywalnego przez sterowanie uchybu położeniowego λ (tzw. podziałka cyklu, $\lambda = \varepsilon_{max}$, gdzie ε to uchyb położeniowy w serwonapędzie), elementarnego przemieszczenia δ oraz współczynnika wzmocnienia prędkościowego k_V ($k_V = V/\varepsilon$). Dla stosowanych w sterowaniach I generacji częstotliwości generatora rzędu 2,5 MHz teoretyczna, maksymalna prędkość ruchu wynosi 5 m/s (dla elementarnego przemieszczenia $\delta = 2 \mu\text{m}$), ale z uwagi na zachowanie odpowiedniego zapasu stabilności serwonapędu, nie może być praktycznie wykorzystana. Zapas stabilności serwonapędu określa się przez największą, dopuszczalną wartość współczynnika wzmocnienia prędkościowego k_V . Dla ówczesnych obrabiarek ze sterowaniami NC (sterowania sprzętowe) maksymalna, dopuszczalna wartość tego współczynnika wynosiła ok. 40 s^{-1} [2]. Wynikająca z tego maksymalna prędkość ruchu V wynosiła ok. 3,6 m/min (wartość tę obliczono, przyjmując, że w sterowaniach sprzętowych, największy, wykrywalny uchyb położeniowy $\lambda = \varepsilon_{max}$, wynosił 2 mm) [2].

Tak więc, sterowania z interpolatorem sprzętowym nie nadają się np. do realizacji technologii HSC (High Speed Cutting). Natomiast, kiedy wymagania dokładności interpolacji nie przekraczają ok. $10 \mu\text{m}$ a prędkości ruchu posuwu ok. 3,6 m/min, to w dalszym ciągu interpolatory sprzętowe są konkurencyjne w stosunku do sterowań z interpolatorami programowymi.

Korzystną cechą interpolatorów sprzętowych jest ich prostota, niezawodność i niska cena. Koszt wytworzenia interpolatora składającego się z kilku elementów elektronicznych typu rejestr cyfrowy, integrator cyfrowy, generator podstawy czasu jest niewielki w stosunku do zaawansowanych interpolatorów programowych.



Rys. 5. Przykładowy schemat sprzętowej realizacji interpolatora kołowego
Fig. 5. An example of hardware realization of circular interpolation unit

Potwierdzeniem takiej oceny interpolatora liniowego może być przedstawiony na rys. 5 interpolator kołowy. Zawiera on dwa rejestry - integratory cyfrowe i generator podstawy czasu, podobnie jak interpolator liniowy. Technicznie nieskomplikowana budowa interpolatora kołowego wynika m.in. z tego, że odpowiednio łączy się wyjścia i wejścia rejestrów, przy czym w jednym z rejestrów impulsy są sumowane a w drugim odejmowane. W rejestrach, w przeciwieństwie do interpolatora liniowego, całkowaniu podlegają zmienne będące funkcją czasu. Przed rozpoczęciem interpolacji do obu rejestrów trzeba wprowadzić informacje o współrzędnych środka okręgu (X_0, Y_0). Dzięki temu, okrąg, który chcemy uzyskać w wyniku interpolacji jest jednoznacznie zdefiniowany.

Uzasadnieniem dla schematu interpolatora kołowego z rys. 5, jest przekształcenie funkcji parametrycznych (4) do postaci (12). Jeżeli zróżniczkować funkcje (4) i jeżeli zamiast parametru p podstawić czas t , to otrzyma się przyrosty funkcji parametrycznych (przyrosty całek $dx(t)$ i $dy(t)$):

$$\begin{aligned} dx(t) &= -R\omega \sin \omega t dt = -\omega Y(t) dt \\ dy(t) &= R\omega \cos \omega t dt = \omega X(t) dt \end{aligned} \quad (12)$$

Jeżeli teraz scałkować przyrosty całek (12) w następnych integratorach (rolę tę pełnią napędy sań) to otrzyma się parametryczne równania okręgu w postaci:

$$\begin{aligned} X(t) &= -\omega \int Y(t) dt + X_0 \\ Y(t) &= \omega \int X(t) dt + Y_0 \end{aligned} \quad (13)$$

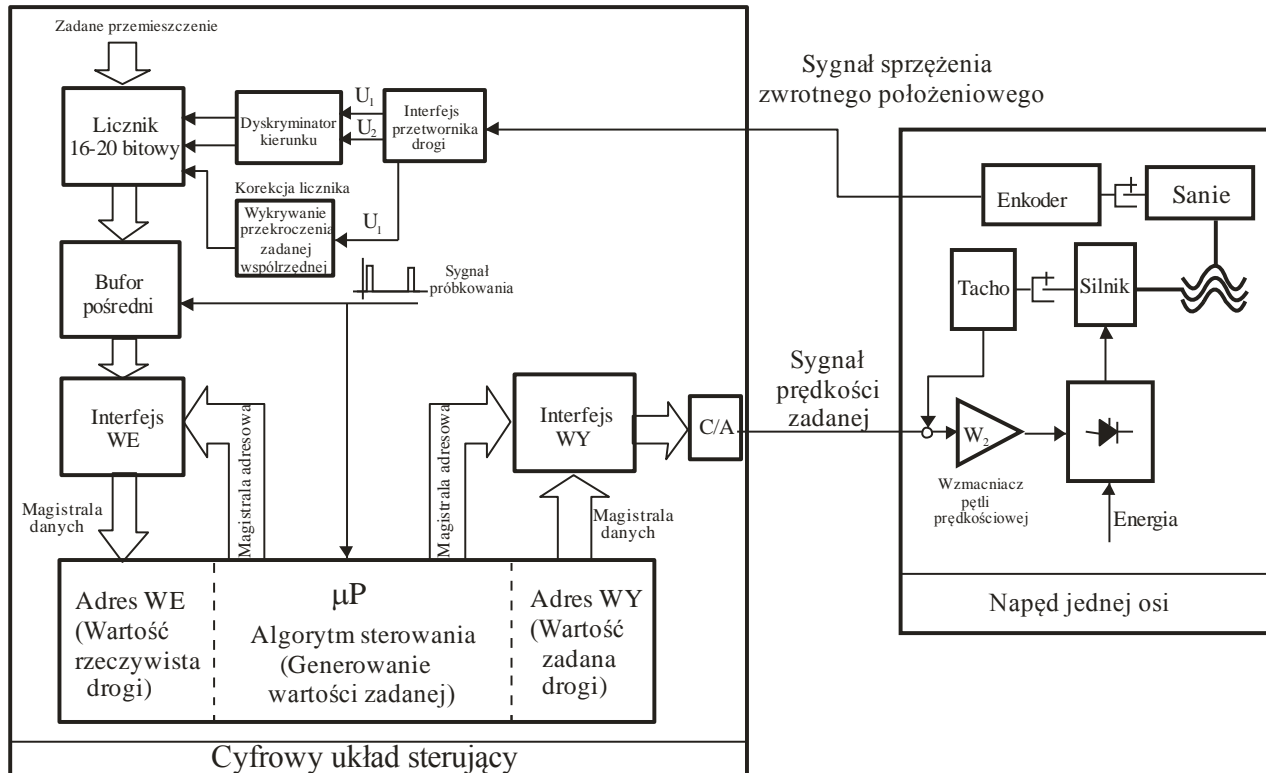
Na rys. 5 zmienne $X(t)$ i $Y(t)$ zapisane są w rejestrach, odpowiednio X i Y, tyle tylko, że w postaci sumy impulsów.

4. INTERPOLATORY PROGRAMOWE

Wprowadzenie komputerowego przetwarzania informacji zmieniło radykalnie możliwości sterowania obrabiarkami. Komputerowa struktura sterowania, bardzo przypominająca klasyczny komputer osobisty PC, zmieniła nie tylko tworzenie programów technologicznych czy wprowadzanie programu technologicznego do sterowania, ale edycję programu, przechowywanie programu, a przede wszystkim komunikowanie się za pośrednictwem magistrali adresowej. Obecność procesora i pamięci operacyjnej zmieniły także filozofię przetwarzania sygnałów. Dotyczy to także, a może przede wszystkim, interpolatora. Na rys. 6 pokazano w sposób schematyczny sterowanie numeryczne komputerowe CNC, a właściwie zespół obliczeniowy i ZSNP.

To, co jest istotne z punktu widzenia interpolatora ukryte jest pod nazwą "Algorytm sterowania (Generowanie wartości zadanej)". Wejściem do tego bloku jest cyfrowy sygnał uchybu położeniowego, a wyjściem sygnał cyfrowy prędkości zadanej w danej osi.

Interpolatory programowe bazują na funkcjach wielomianowych oraz na tzw. krzywych Beziera [1],[2]. Z formalnego punktu widzenia jedne i drugie należą do krzywych parametrycznych, z których do najpopularniejszych należą krzywe sklejane B-spline. Krzywe sklejane B-spline opisywane są za pomocą wielomianów.



Rys. 6. Schemat sterowania numerycznego komputerowego CNC

Fig. 6. The diagram of computer numerical control CNC

Spośród krzywych wielomianowych wyróżnia się ponadto wielomianowe krzywe wymierne (W artykule w dalszym ciągu wielomianowa krzywa wymierna będzie nazywana krzywą wymierną), które zostały zdefiniowane w innym układzie współrzędnych, w tzw. układzie współrzędnych jednorodnych [10]. Pozwalają one modelować krzywe stożkowe (klasyczne krzywe wielomianowe pozwalają modelować tylko krzywe paraboliczne).

Funkcje wielomianowe trzeciego stopnia to funkcje potęgowe w postaci:

$$f(p) = a_0 + a_1 p + a_2 p^2 + a_3 p^3 \quad (14)$$

gdzie:

p - parametr ($p \in [0,1]$),

a_0 - współczynnik odpowiadający współrzędnej punktu poprzedniego,

a_1 - współczynnik odpowiadający współrzędnej punktu następnego, który ma zostać osiągnięty w rozpatrywanym kroku programowym,

a_2, a_3 - współczynniki, które muszą być dobrane w taki sposób, aby uzyskać pożądany kształt toru.

W praktyce nie stosuje się funkcji potęgowych wyższego rzędu niż trzeci, ponieważ komplikują się algorytmy obliczeniowe (wyznaczanie przecięć, ekstremów, obcinanie itp.). Przy pomocy krzywych potęgowych trzeciego stopnia (14) można tworzyć tory pomiędzy dwoma punktami bazując na krzywych parabolicznych, wykładniczych oraz innych krzywych drugiego stopnia, w taki sposób, że tor krzywej przechodzi przez zaprogramowane punkty.

Dla przybliżenia interpolacji wielomianowej drugiego stopnia rozpatrzony zostanie przykład toru parabolicznego o współrzędnych punktu końcowego (0,25;0,5). Wiersz programowy w sterowaniu SINUMERIK 840D wygląda następująco:

N10 POLY PO[Y]=(0,5) PO[X]=(0,25) PL=4,

a jego interpretacja jest następująca:

- PL = 4 - interpolacja z 4 punktami kontrolnymi na krzywej wielomianowej,
- PO[Y] = (0,5) - współrzędna punktu końcowego w kierunku Y,
- PO[X] = (0,25) - współrzędna punktu końcowego w kierunku X.

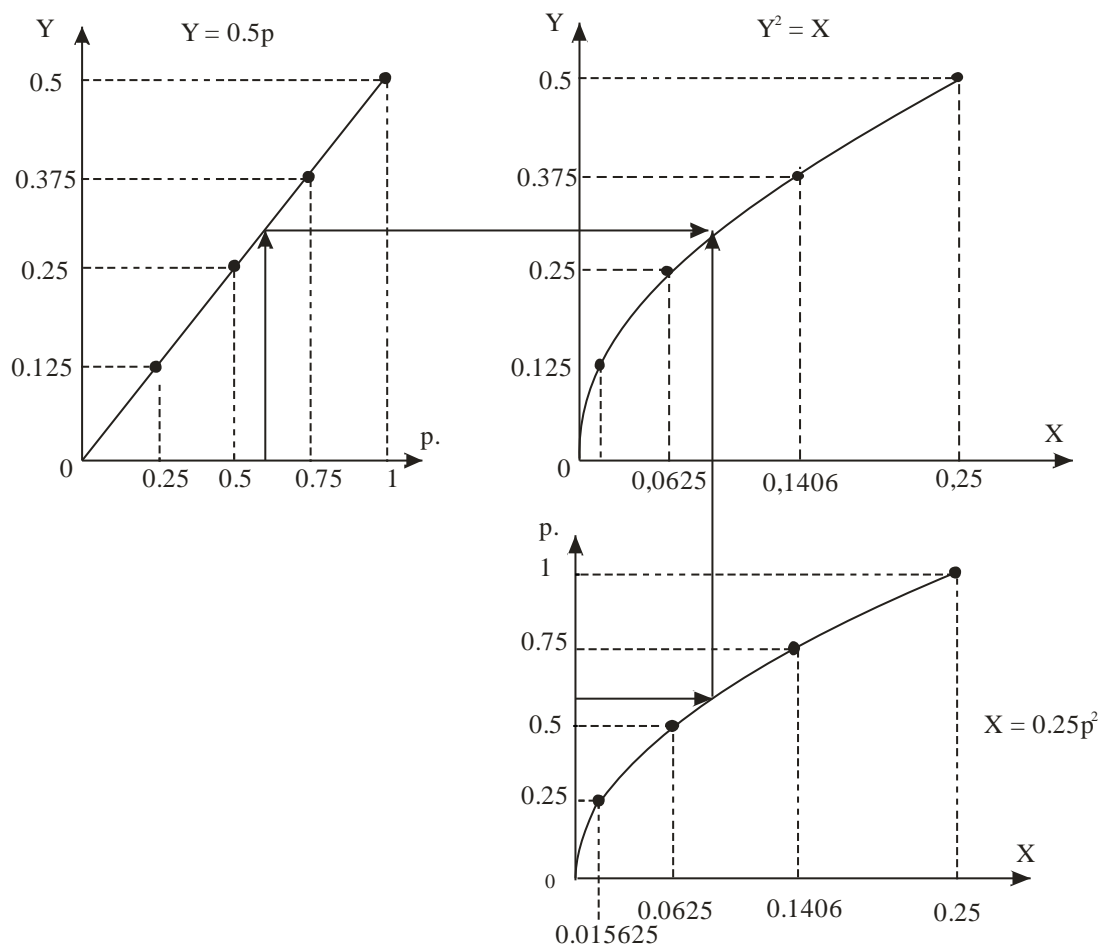
Równania parametryczne krzywej parabolicznej przedstawiają się następująco:

$$\begin{aligned} X &= 0,25p^2 \\ Y &= 0,5p \end{aligned} \quad (15)$$

a krzywa paraboliczna przyjmuje postać:

$$Y^2 = X \quad (16)$$

Graficzną reprezentację funkcji parametrycznych (15) oraz samej funkcji parabolicznej (16) przedstawia rys. 7.



Rys. 7. Graficzna interpretacja funkcji parametrycznych krzywej parabolicznej
Fig. 7. Graphic interpretation of parametric, parabolic function

Szerokie zastosowanie w interpolacji programowej znalazły krzywe Beziery, które są parametrycznymi funkcjami wielomianowymi. Krzywe wielomianowe pozwalają na tworzenie krzywych parabolicznych, a krzywe wielomianowe wymierne, parabolicznych i stożkowych. Na ogół są to funkcje klasy (14), tzn. trzeciego stopnia. Dla przybliżenia tej interpolacji na rys. 8, pokazano krzywą wielomianową Beziery trzeciego stopnia.

Krzywe Beziery są zdefiniowane przez tzw. punkty kontrolne $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$. Znajomość tych punktów kontrolnych jednoznacznie definiuje kształt krzywej wielomianowej. Oznacza to, że w programie CNC musi być zapisana informacja o współrzędnych punktów kontrolnych. Przykładowo, w sterowaniach Heidenhaina iTNC530 wiersze programowe dla krzywej wielomianowej 3 stopnia wyglądają następująco (rys. 8b):

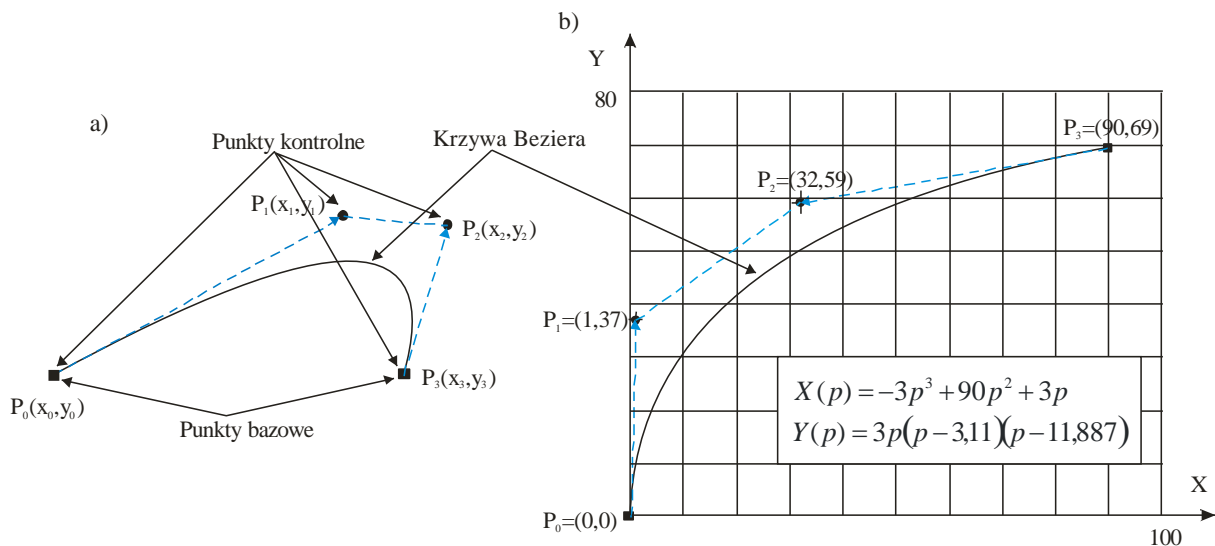
SPL X90 Y69 Z0,0 - wywołanie interpolacji i końcowa pozycja krzywej wielomianowej

K1X1,0 K2X32,0 K3X90,0 - punkty kontrolne

K1Y37,0 K2Y59,0 K3Y69,0 - punkty kontrolne

K1Z0,0 K2Z0,0 K3Z0,0 - punkty kontrolne.

Parametryczna krzywa wielomianowa zawiera także punkty bazowe. Są to najczęściej punkty: początkowy i końcowy odcinka krzywej wielomianowej.



Rys. 8. Przykład krzywej wielomianowej trzeciego stopnia, tj. krzywej Beziery

Fig. 8. An example of 3-rd degree polynomial, i.e. Bezier curve

Postać parametryczna krzywej wielomianowej Beziery wygląda następująco:

$$f(p) = A(1-p)^3 + 3Bp(1-p)^2 + 3Cp^2(1-p) + Dp^3 \quad 0 \leq p \leq 1, \quad (17)$$

a zapis parametryczny w każdej osi:

$$\begin{aligned} X(p) &= A_x(1-p)^3 + 3B_x p(1-p)^2 + 3C_x p^2(1-p) + D_x p^3 \\ Y(p) &= A_y(1-p)^3 + 3B_y p(1-p)^2 + 3C_y p^2(1-p) + D_y p^3 \end{aligned} \quad (18)$$

gdzie: A_x , B_x , C_x , D_x , A_y , B_y , C_y , D_y - współczynniki kształtujące postać krzywej, które są funkcją współrzędnych punktów charakterystycznych.

Punkty bazowe uzyskuje się dla $p=0$ (punkt początkowy) i $p=1$ (punkt końcowy).

Zachodzą następujące relacje:

$$P_0(x_0, y_0) = P_0(A_x, A_y),$$

$$P_3(x_3, y_3) = P_3(D_x, D_y),$$

stąd wynika, że:

$$x(0) = A_x$$

$$y(0) = A_y,$$

$$x(1) = D_x,$$

$$y(1) = D_y.$$

Przyjmując, że punkty kontrolne opisane są przez współrzędne w sposób następujący

$$P_0 = (x_0, y_0),$$

$$P_1 = (x_1, y_1),$$

$$P_2 = (x_2, y_2),$$

$$P_3 = (x_3, y_3),$$

parametryczne krzywe wielomianowe można przedstawić następująco:

$$X(p) = d_x + c_x p + b_x p^2 + a_x p^3 \quad (19)$$

$$Y(p) = d_y + c_y p + b_y p^2 + a_y p^3$$

przy czym zachodzą relacje:

$$\begin{aligned} x_1 &= x_0 + \frac{c_x}{3} & y_1 &= y_0 + \frac{c_y}{3} \\ x_2 &= x_1 + \frac{c_x + b_x}{3} & y_2 &= y_1 + \frac{c_y + b_y}{3} \\ x_3 &= x_0 + c_x + b_x + a_x & y_3 &= y_0 + c_y + b_y + a_y \end{aligned} \quad (20)$$

oraz

$$\begin{aligned} c_x &= 3(x_1 - x_0) & c_y &= 3(y_1 - y_0) \\ b_x &= 3(x_2 - x_1) - c_x & b_y &= 3(y_2 - y_1) - c_y \\ a_x &= x_3 - x_0 - c_x - b_x & a_y &= y_3 - y_0 - c_y - b_y \end{aligned} \quad (21)$$

Zależności (20) i (21) pozwalają na dwie interpretacje konstruowania krzywych wielomianowych:

1. Dla przyjętej postaci krzywej, tzn. dla przyjętych współczynników $a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y$ z zależności (20) można wyznaczyć współrzędne punktów charakterystycznych i wstawić je do wierszy programowych definiujących interpolację wielomianową.
2. Dla przyjętych punktów charakterystycznych, np. przy pomocy programu CAD lub CorelDraw można skonstruować krzywą parametryczną i odczytać współrzędne punktów charakterystycznych, a z zależności (21) można wyznaczyć współczynniki równań parametrycznych (19).

Krzywe wielomianowe (Beziera) postaci (17) lub (18) pozwalają na tworzenie krzywych parabolicznych, np. paraboli. Nie pozwalają natomiast na tworzenie krzywych

stożkowych, np. elipsy, hiperboli czy nawet okręgu. Krzywe stożkowe można tworzyć przy pomocy krzywych Beziera wymiernych.

Wymierna krzywa Beziera to rzut środkowy krzywej wielomianowej Beziera zapisanej we współrzędnych jednorodnych [10] na płaszczyznę $W=1$. Na rys. 9 pokazano graficzną interpretację definicji krzywej wymiernej [13].

Jeżeli dla krzywej wielomianowej Beziera zbiór punktów kontrolnych można przedstawić jako:

$$P_i(x_i, y_i, z_i, \dots)$$

to dla wymiernej krzywej Beziera zbiór punktów kontrolnych wygląda następująco:

$$P_i^H(w_i x_i, w_i y_i, w_i z_i, \dots)$$

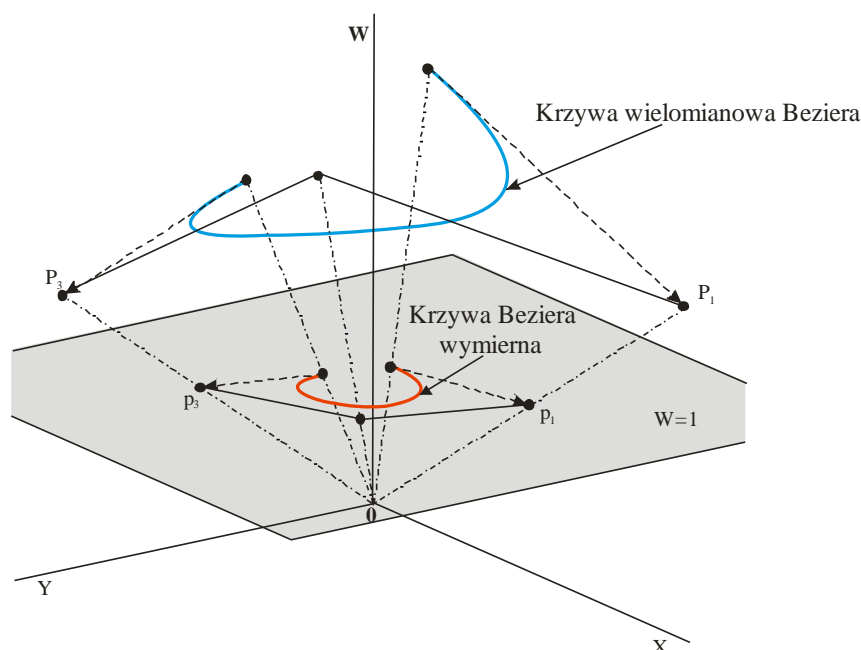
Jeżeli krzywą wielomianową przestrzenną Beziera przedstawić we współrzędnych XYW jednorodnych a następnie dokonać jej rzutu środkowego (rzut względem punktu 0) na płaszczyznę $W=1$, to otrzymana krzywa reprezentuje krzywą Beziera wymierną.

Dowolny punkt (x, y) na krzywej wielomianowej Beziera, opisanej równaniami parametrycznymi (18), można zapisać następująco:

$$(x, y) = \left(\sum_{i=0}^n x_i B_i^n(p), \sum_{i=0}^n y_i B_i^n(p) \right) \quad 0 \leq p \leq 1 \quad (22)$$

gdzie: $B_i^n(p)$ - tzw. wielomian Bernsteina, który jest wielomianem parametrycznym, dla którego punkty kontrolne są naturalnymi współczynnikami takiego wielomianu.

Oznacza to m.in., że w stosunku do zapisu wielomianu w postaci (19), kiedy trzeba dokonywać pewnych przeliczeń (20) lub (21) aby wyznaczyć współczynniki równań parametrycznych (19), w wielomianach Bernsteina nie ma takiej potrzeby.



Rys. 9. Interpretacja graficzna definicji wymiernej krzywej Beziera [13]

Fig. 9. Graphic interpretation of definition of rational Bezier curve [13]

Współrzędne dowolnego punktu (x,y) na krzywej wymiernej można przedstawić następująco:

$$(x, y) = \left(\frac{\sum w_i x_i B_i^n(p)}{\sum x_i B_i^n(p)}, \frac{\sum w_i y_i B_i^n(p)}{\sum y_i B_i^n(p)} \right) \quad 0 \leq p \leq 1 \quad (23)$$

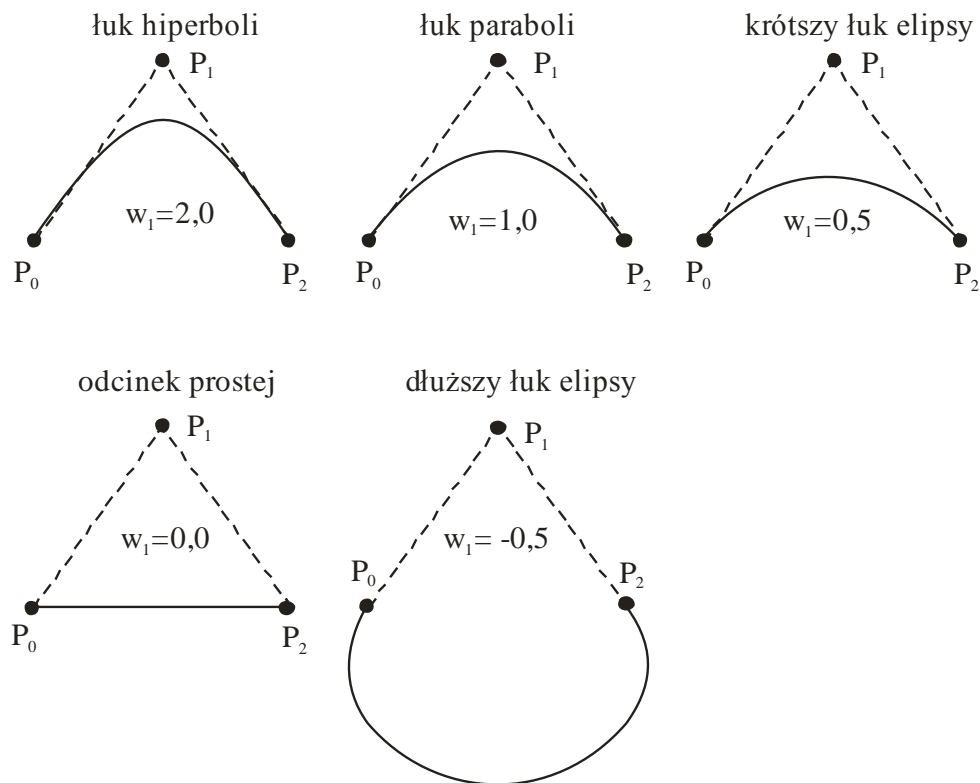
gdzie: w_i – wagi.

W stosunku do krzywej wielomianowej Beziera (17) lub (22) krzywa wielomianowa wymierna zapisana w postaci (23) zawiera dodatkowy współczynnik w postaci wagi w_i , który oprócz punktów kontrolnych (x_i, y_i) , pozwala na jeszcze lepszą kontrolę i znacznie bogatsze kształtowanie zarysu krzywej.

Przykładowo, jeżeli są dane trzy punkty kontrolne P_0, P_1 i P_2 (rys. 10) i jeżeli wagi $w_0 = w_2 = 1$, to waga w_1 definiuje kształt krzywej wymiernej Beziera (tabela 1).

Na rys. 10 pokazano przykłady krzywych stożkowych dla kilku wartości wagi w_1 (wagi $w_0 = w_2 = 1$). Są to łuki hiperboliczne, paraboliczne, eliptyczne czy też odcinek prostej. Przykłady te pokazują łatwość, z jaką można kształtować zarysy linii krzywych, jeżeli sterowanie numeryczne dysponuje interpolatorem z aplikacją wymiernych krzywych Beziera.

W praktycznych zastosowaniach kształtowania przedmiotów obrabianych na obrabiar-kach CNC często występuje konieczność interpolacji kołowej. Wymierne krzywe Beziera także umożliwiają taką interpolację. Obrazuje to rys. 11.



Rys. 10. Przykłady krzywych stożkowych dla kilku wartości wagi w_1 [11]

Fig. 10. Examples of conic curves for several values of weight w_1 [11]

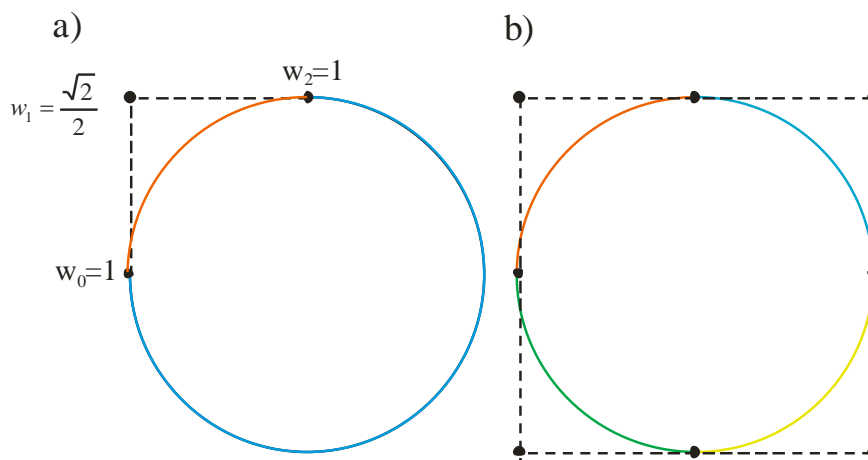
Tabela 1. Wagi w_I definiujące kształt krzywej wymiernej Beziera
 Table 1. Weights w_I defining rational Bezier curves

$w_I > 1$	→	łuk hiperboli
$w_I = 1$	→	łuk paraboli
$0 < w_I < 1$	→	krótszy łuk elipsy lub okręgu
$w_I = 0$	→	odcinek prostej
$-1 < w_I < 0$	→	dłuższy łuk elipsy lub okręgu
$w_I = -1$	→	dwa łuki paraboli
$w_I < -1$	→	dwa łuki hiperboli

Dla uzyskaniu łuku okręgu za pomocą wymiernych krzywych Beziera waga $w_1 = \frac{\sqrt{2}}{2}$.

Przy czym, interpolację kołową można uzyskać za pomocą krótszego lub dłuższego łuku okręgu (rys. 11a) albo tylko krótszych łuków okręgu (rys. 11b).

Jeżeli więc sterowanie CNC dysponuje interpolatorem z aplikacją krzywych wymiernych Beziera to interpolację kołową programuje się, wpisując w wierszu poleceń wagę $w_1 = \frac{\sqrt{2}}{2}$ oraz współrzędne punktu początkowego i końcowego łuku okręgu.

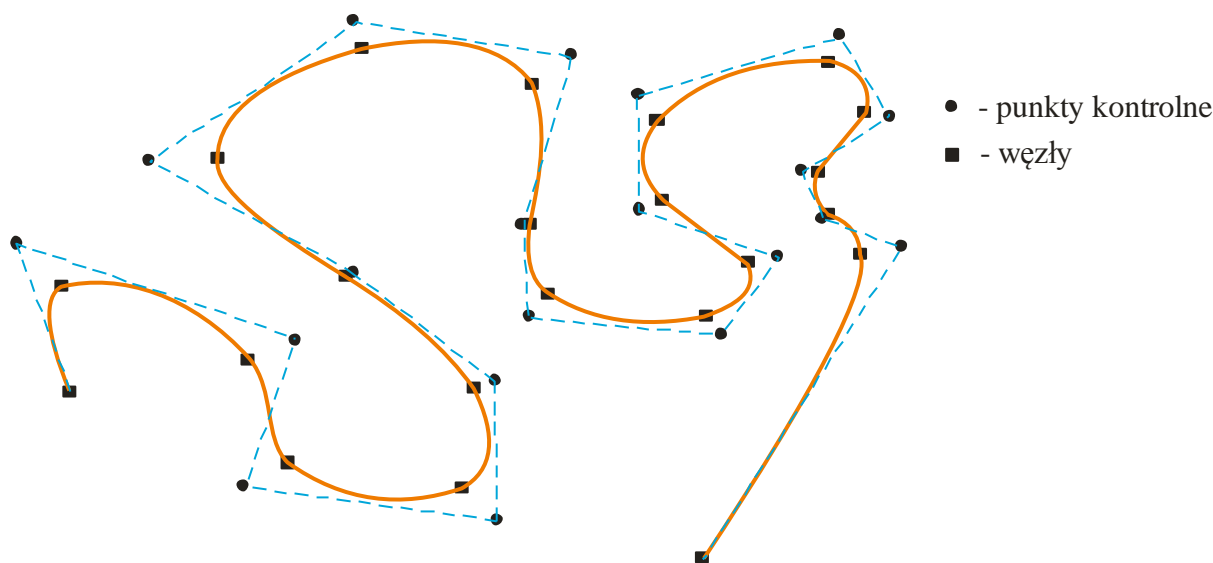


Rys. 11. Kształtowanie krzywych kołowych przy użyciu wymiernych wielomianów Beziera: a) za pomocą dwóch krzywych tworzących krótszy i dłuższy łuk okręgu, b) za pomocą czterech krzywych tworzących krótszy łuk okręgu [11]

Fig. 11. Forming of circle curves using rational Bezier curves: a) using two curves for shorter and longer arc of circle, b) using four curves for shorter arc of circle [11]

Koncepcja interpolacji za pomocą krzywych Beziera (wielomianowych i wielomianowych wymiernych) jest bardzo wygodnym narzędziem tworzenia torów narzędzia na obrabiarce CNC, pod warunkiem, że krzywą da się przybliżyć wielomianem niskiego stopnia (co najwyżej 3 stopnia). Dla bardziej złożonych kształtów, jak np. na rys. 12, interpolacji za pomocą pojedynczej krzywej Beziera nie da się zrealizować.

Wprawdzie, można by podejść do takiego zagadnienia konwencjonalnie, tzn. podzielić całą krzywą na szereg odcinków, definiując początek i koniec każdego za pomocą tzw. węzłów. Dla każdego z takich odcinków można zdefiniować krzywą Beziera, wielomianową lub wielomianową wymierną i w programie CNC wywołać interpolację wielomianową wielokrotnie. Jednak, oprócz dużej pracochłonności takiego postępowania (wielokrotne wyznaczanie punktów kontrolnych), pozostaje pytanie jak dzielić całą krzywą na odcinki oraz co z gładkością krzywej w węzłach.



Rys. 12. Przykład złożonej krzywej, niedającej się opisać pojedynczą krzywą Beziera
 Fig. 12. An example of complex curve, which can't be modeled using a single Bezier curve

Innym, wygodniejszym sposobem opisu złożonych krzywych jest zastosowanie interpolacji splajnowej. Technika ta bazuje na krzywych Beziera sklejanych (ang. Spline). Koncepcja krzywych sklejanych bazuje na dwóch podstawowych założeniach:

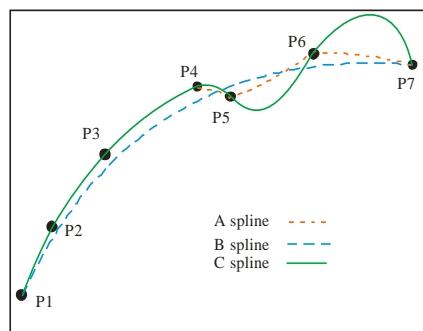
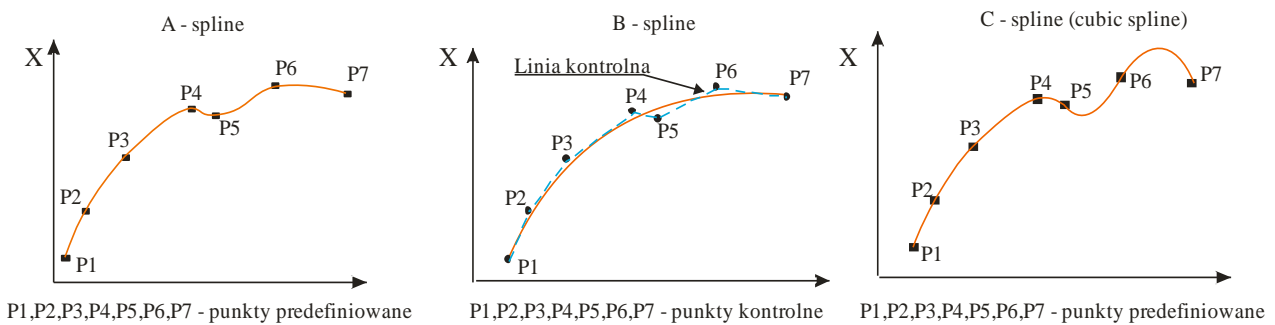
- każdy z odcinków krzywej jest opisywany za pomocą krzywych Beziera (wielomianowych lub wielomianowych wymiernych),
- w węzłach krzywej zachowane zostają warunki gładkości (jest dwukrotnie różniczkowalna).

Ten drugi warunek powoduje, że w praktyce spotyka się trzy klasy krzywych sklejanych (rys. 13):

- klasa A (Akima-spline), krzywa przechodzi przez zaprogramowane punkty interpolacji, a zastosowaną krzywą jest wielomian trzeciego stopnia. W punktach interpolacji nie jest zachowany warunek dwukrotnej różniczkowalności, czyli krzywa nie jest gładka. Natomiast zachowany jest warunek styczności, na podstawie punktów sąsiadujących z punktami interpolacji. Jest to tzw. interpolacja Akimy. Tę metodę stosuje się, kiedy chcemy zachować warunki styczności w punktach interpolacyjnych, a moce obliczeniowe komputera nie są najwyższe.
- klasa B (B-spline), najpopularniejsza w praktyce interpolacja; zaprogramowane punkty to punkty kontrolne krzywej Beziera, a krzywa przechodzi przez węzły zdefiniowane

przy podziale krzywej na odcinki. Używane są wielomiany 1, 2 i 3 stopnia. Interpolacja B-spline jest, z formalnego punktu widzenia aproksymacją, natomiast powszechnie nazywana jest interpolacją. Punkty kontrolne tworzą linię łamaną, tzw. polylinię, która jednoznacznie definiuje krzywą splajnową.

- klasa C (Cubic-spline) jest interpolacją podobną do klasy A-spline, tzn. krzywa obejmuje zaprogramowane punkty, z tą jednak różnicą, że w punktach interpolacji zachowany jest warunek dwukrotnej różniczkowości. Krzywa przechodzi przez węzły z zachowaniem gładkości. Interpolacja jest stosowana, kiedy żądana jest określona krzywizna w określonych punktach



Rys. 13. Klasy krzywych sklejanych

Fig. 13. Types of spline curves

Na rys. 13 pokazano także wszystkie trzy klasy krzywych na jednym wykresie, który uwidacznia najważniejsze różnice pomiędzy nimi.

W krzywych sklejanych parametr p (patrz (17)) mieści się w przedziale $[0,1]$. Ponieważ jednak cała krzywa jest dzielona na szereg podzakresów, to ten parametr w obszarze pojedynczego podzakresu jest odpowiednio mniejszy. Jeżeli węzłów jest $m+1$ (u_0, u_1, \dots, u_m), a stopień wielomianu jest n , to potrzeba $(m-n)$ punktów kontrolnych, a liczba odcinków z których składa się cała krzywa, wynosi $(m-2n)$. Wówczas parametr p w każdym z podzakresów mieści się w przedziale $[u_n, u_{m-n}]$, a nie $[0,1]$.

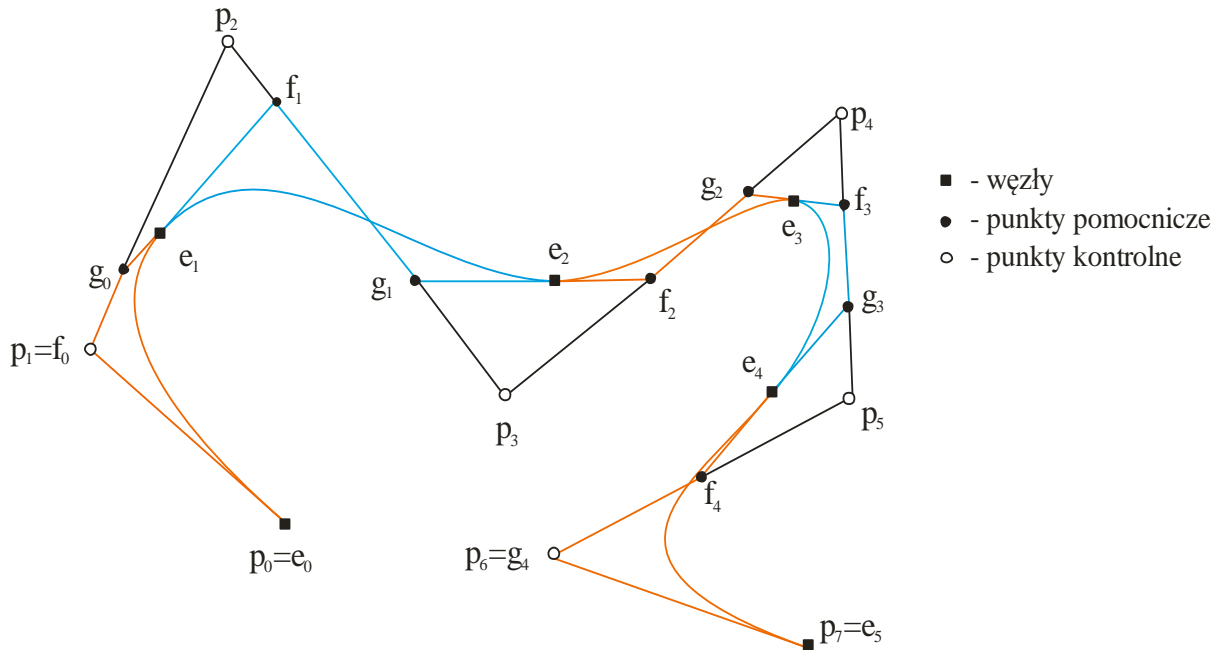
Dowolny punkt na krzywej sklejanej można przedstawić następująco:

$$P(p) = \sum_0^{m-n-1} p_i N_i^n(p) \quad u_n \leq p \leq u_{m-n} \quad (24)$$

gdzie: $N_i^n(p)$ - unormowane funkcje B-sklejane stopnia n .

Dla zobrazowania warunków ciągłości (gładkości) krzywej sklejaney, na rys. 14 przedstawiono ich graficzną interpretację.

Warunki ciągłości (pierwsza i druga pochodna w węzłach) uzyskuje się w taki sposób, że punkty pomocnicze g_0, e_{i+1}, f_{i+1} są współliniowe (pierwsza pochodna) oraz punkty pomocnicze g, f są odpowiednio umiejscowione, znajdują się na prostych przechodzących przez dwa kolejne punkty kontrolne p_i, p_{i+1} (druga pochodna).



Rys. 14. Wielomianowa niejednorodna krzywa B-sklejane trzeciego stopnia (zbudowana z pięciu krzywych Béziera); węzły: $e_0=0,0$, $e_1=0,1$, $e_2=0,4$, $e_3=0,6$, $e_4=0,8$, $e_5=1,0$ [12]

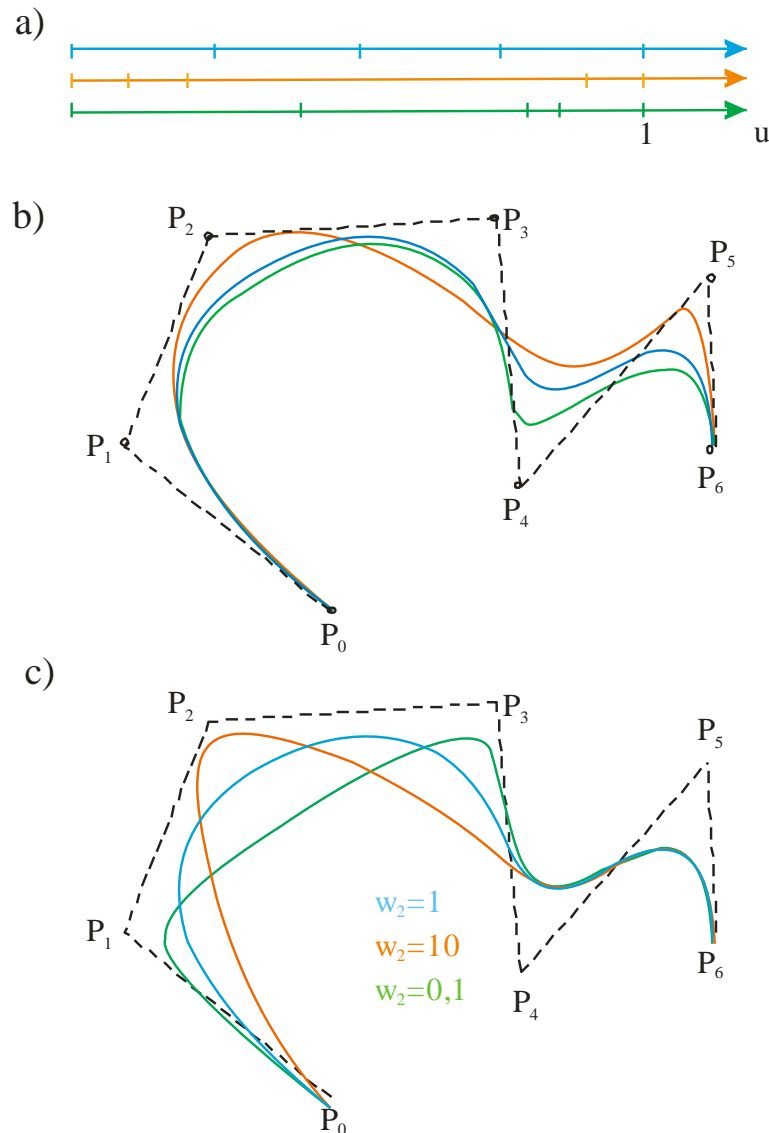
Fig. 14. Polynomial non-uniform B-spline 3-rd degree curve (consist of five Bezier curves); knots: $e_0=0,0$, $e_1=0,1$, $e_2=0,4$, $e_3=0,6$, $e_4=0,8$, $e_5=1,0$ [12]

Szczególne znaczenie i zastosowanie znalazły wymierne funkcje B-sklejane NURBS (Non-Uniform-Rational B-Spline). Łączą one zalety funkcji sklejaneych i wymiernych. Oznacza to, że tak jak w przypadku wymiernych krzywych Beziera, krzywe NURBS posiadają możliwość kształtowania krzywej sklejaney za pomocą punktów kontrolnych, węzłów i wag. Nazwa Non-Uniform oznacza, że krzywa może zostać podzielona na szereg niejednakowych podzakresów.

Dowolny punkt na krzywej sklejaney NURBS można przedstawić, podobnie jak dla krzywej wymiernej (20), następująco (algorytm de Boora) [13]:

$$(x, y) = \left(\frac{\sum_{i=0}^{i=m-n-1} w_i x_i N_i^n(p)}{\sum_{i=0}^{i=m-n-1} x_i N_i^n(p)}, \frac{\sum_{i=0}^{i=m-n-1} w_i y_i N_i^n(p)}{\sum_{i=0}^{i=m-n-1} y_i N_i^n(p)} \right) \quad u_n \leq p \leq u_{m-n} \quad (25)$$

W zależności (25) pojawiła się dodatkowa w stosunku do zależności (22) zmienna, która poszerza możliwość kształtowania krzywej, mianowicie długość podprzedziału u . Na rys. 15 przedstawiono przykłady wpływu tej zmiennej na postać krzywej (rys. 15a i rys. 15b).



Rys. 15. Przykładowy wpływ wielkości podprzedziału (u) dla pięciu ($m=5$) węzłów i wag w_i na kształt krzywej NURBS 3-go stopnia ($n=3$): a) rozkłady pięciu węzłów, b) wpływ rozkładów węzłów na kształty krzywej, c) wpływ wag na kształty krzywej [10]

Fig. 15. An example of influence the range size (u) for five ($m=5$) knots and weight w_i on the NURBS 3-rd degree ($n=3$) curve form: a) distribution of five knots, b) influence of distribution of knots on curve form, c) influence of weights on curve form [10]

Zmieniając rozmieszczenie węzłów w sposób nierównomierny (Non-Uniform) można wpływać na kształt krzywej. Podobnie, zmieniając wagi, możemy wpływać na kształt krzywej. Wszystkie te działania można wykonywać dla niezmiennych punktów kontrolnych.

5. PRZYKŁADY APLIKACJI INTERPOLACJI PROGRAMOWEJ

Zaawansowane układy sterowania numerycznego CNC takie jak Sinumerik 840D, Heidenhain iTNC530 czy GE FANUC, oferują opcje interpolacji wielomianowej oraz z wykorzystaniem krzywych B-spline. W dalszym kolejności przedstawione zostaną praktyczne przykłady zastosowania takich interpolacji w postaci bloków programowych, zawierających wiersze interpolacji wielomianowej.

Bardzo rozbudowaną interpolację programową posiadają sterowania cyfrowe firmy Siemens [9]. Przykładowo, Sinumerik 840D oferuje wszystkie trzy klasy interpolacji splajnowej, tzn. interpolację A-spline, B-spline (NURBS) i C-spline, których wywołanie wygląda następująco:

ASPLINE X Y Z A B C {A-spline}

lub

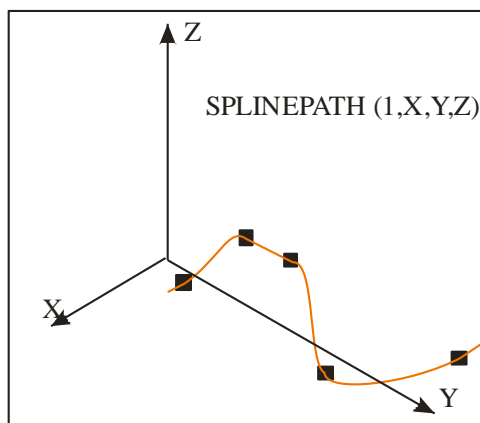
BSPLINE X Y Z A B C {B-spline}

lub

CSPLINE X Y Z A B C {C-spline}.

Wybór osi, w których będzie realizowana interpolacja splajnowa definiuje polecenie:

SPLINEPATH(n,X,Y,Z,...) { $n=1, X,Y,Z$ - nazwy osi} (rys. 16).



Rys. 16. Definicja osi interpolowanych splajnami [9]
Fig. 16. Definition of interpolated axes using splines [9]

Interpolacja B-spline wykorzystuje krzywe NURBS. Dlatego możliwe jest zadeklarowanie w wierszach programowych wartości wag w_i dla każdego punktu kontrolnego w postaci:

PW=n {*PW*-deklaracja wagi: $0 < n < 3$ z krokiem 0,0001}.

Interpolacja B-spline wykorzystuje wielomiany trzeciego lub drugiego stopnia. Wymaga to zadeklarowania w postaci:

SD=2

lub *SD=3* {*SD* - stopień wielomianu}.

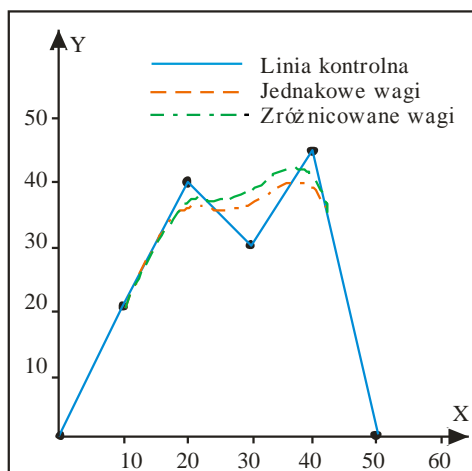
Punkty kontrolne w zasadzie oblicza samo sterowanie. Możliwe jest także "ręczne" zadeklarowanie odległości pomiędzy tymi punktami w postaci polecenia:

PL = zakres odległości pomiędzy punktami kontrolnymi.

W tabeli 2 przedstawiono przykładowy blok programowy interpolacji B-spline dla przypadku, kiedy wagi są takie same ($w_i=1$) i zróżnicowane (rys. 17). Ponadto, pokazano również blok programowy dla wielomianowej linii kontrolnej.

Tabela 2. Przykłady bloków programowych dla interpolacji B-spline
Table 2. Examples of program blocks for B-spline interpolation

Wszystkie $w_i = 1$	Zróżnicowane w_i	Linia łamana kontrolna
<i>N10 G1 X0 Y0 F300 G64</i>	<i>N10 G1 X0 Y0 F300 G64</i>	<i>N10 G1 X0 Y0 F300 G64</i>
<i>N20 BSPLINE</i>	<i>N20 BSPLINE</i>	<i>N20;omitted</i>
<i>N30 X10 Y20</i>	<i>N30 X10 Y20 PW=2</i>	<i>N30 X10 Y20</i>
<i>N40 X20 Y40</i>	<i>N40 X20 Y40</i>	<i>N40 X20 Y40</i>
<i>N50 X30 Y30</i>	<i>N50 X30 Y30 PW=0.5</i>	<i>N50 X30 Y30</i>
<i>N60 X40 Y45</i>	<i>N60 X40 Y45</i>	<i>N60 X40 Y45</i>
<i>N70 X50 Y0</i>	<i>N70 X50 Y0</i>	<i>N70 X50 Y0</i>



Rys. 17. Porównanie krzywych B-spline dla jednakowych i zróżnicowanych wag [9]

Fig. 17. Comparison of B-spline curves for uniform and difference weights [9]

W przypadku interpolacji A-spline i C-spline należy zadeklarować sposób rozpoczynania i kończenia interpolacji (dla B-spline takiego wymogu nie ma). Są to dwie grupy poleceń:

BAUTO, *BNAT* i *BTAN* - sposób rozpoczynania interpolacji,

EAUTO, *ENAT* i *ETAN* - sposób kończenia interpolacji,

BAUTO (*EAUTO*) - bez deklaracji sposobu rozpoczynania interpolacji,

BNAT (*ENAT*) - deklaracja rozpoczynania interpolacji z zerową krzywizną w punkcie początkowym (końcowym),

BTAN (ETAN) - deklaracja rozpoczęcia (kończenia) interpolacji stycznie do następnego punktu.

Poniżej przedstawiono przykład bloku programowego z interpolacją C-spline, a na rys. 18 krzywą uzyskaną w wyniku realizacji powyższego bloku programowego.

N10 G1 X0 Y0 F300

N15 X10

N20 BNAT ENAT {deklaracja początku i końca interpolacji z zerową krzywizną}

N30 CSPLINE X20 Y10 {deklaracja interpolacji C-spline}

N40 X30

N50 X40 Y5

N60 X50 Y15

N70 X55 Y7

N80 X60 Y20

N90 X65 Y20

N100 X70 Y0

N110 X80 Y10

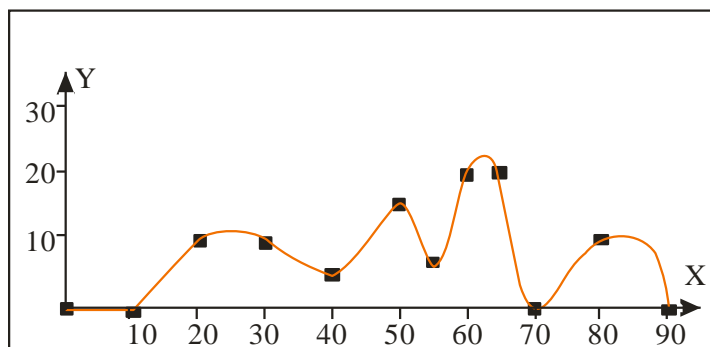
N120 X90 Y0

N130 M30.

Firma Heidenhain [7] dla celów interpolacji programowej stosuje parametryczne krzywe wielomianowe trzeciego stopnia w postaci:

$$\begin{aligned} X(t) &= K_{3X} \cdot t^3 + K_{2X} \cdot t^2 + K_{1X} \cdot t + X \\ Y(t) &= K_{3Y} \cdot t^3 + K_{2Y} \cdot t^2 + K_{1Y} \cdot t + Y \\ Z(t) &= K_{3Z} \cdot t^3 + K_{2Z} \cdot t^2 + K_{1Z} \cdot t + Z \end{aligned} \quad (26)$$

gdzie: t - parametr o zakresie $[0,1]$, którego przyrost zależy od zaprogramowanego posuwu i długości krzywej.



Rys. 18. Przykład interpolacji C-spline [9]
Fig. 18. An example of C-spline interpolation [9]

Heidenhain stosuje tzw. bezpośrednie obliczanie wartości funkcji (współrzędnych $X(t)$, $Y(t)$...) i na ich podstawie generowanie sygnałów sterujących serwonapędami. Jest to więc interpolacja zbliżona do klasy A-spline. Programista definiując współczynniki K_i musi zadbać o warunki styczności na granicach przedziałów. Każdy odcinek krzywej

interpolowanej jest programowany oddzielnie, tzn. występuje oddzielny blok programowy, zawierający polecenie interpolacji splajnowej (SPL), współrzędne punktu końcowego i wartości współczynników K_i . Sklejanie takich odcinków krzywej odbywa się "ręcznie".

Poniżej przedstawiono blok programu dla dwuosiowej interpolacji wielomianowej w układach sterowania Heidenhain [8].

Tabela 3. Przykład bloku programowego dla dwuosiowej interpolacji wielomianowej
Table 3. Example of a program block for two dimension polynomial interpolation

6	
7	<i>L X+33,909 Z+75,107 F MAX</i>	Punkt startowy interpolacji
8	<i>SPL X+39,824 Z+77,425</i> <i>K3X+0,0983 K2X_0,441 K1X_5,5724</i> <i>K3Z+0,0015 K2Z_0,9549 K1Z+3,0875 F10000</i>	Wywołanie interpolacji +punkt końcowy interpolacji Parametry interpolacji dla osi X Parametry interpolacji dla osi Z
9	<i>SPL X+44,862 Z+73,44</i> <i>K3X+0,0934 K2X_0,7211 K1X_4,4102</i> <i>K3Z_0,0576 K2Z_0,7822 K1Z+4,8246</i>	Punkt końcowy interpolacji Parametry interpolacji dla osi X Parametry interpolacji dla osi Z
10	

W wierszu siódmym zdefiniowany zostaje punkt początkowy interpolacji. Natomiast w wierszu ósmym zawarto formalne wywołanie interpolacji *SPL* oraz zdefiniowane zostały współrzędne punktu końcowego interpolacji. Następne wiersze zawierają współczynniki K_{xi} , K_{yi} i K_{zi} , występujące w równaniach parametrycznych (26) oraz wielkość posuwu.

Interpolacja w sterownikach Heidenhaina nie wykorzystuje krzywych Bezierra czy krzywych NURBS. Można ją stosować dla takich krzywych, dla których zdefiniowane zostały współczynniki w równaniach parametrycznych. Można skonstruować bazę współczynników K_i dla zdefiniowanej grupy linii krzywych. Wówczas programista ma ułatwione zadanie programowania krzywej splajnowej.

6. PODSUMOWANIE

W artykule omówiono zagadnienia interpolacji w obrabiarkach sterowanych numerycznie. Głównym celem artykułu było uświadomienie czytelnikom możliwości nadawania wymiarów i kształtów przedmiotom na obrabiarkach sterowanych numerycznie CNC. Wydaje się, że wiedza o potencjalnie dużych możliwościach kształtowania przedmiotów na obrabiarkach CNC jest jeszcze stosunkowo mała. Dotyczy to zwłaszcza zaawansowanych metod sterowania nadawaniem kształtów i wymiarów przedmiotom za pomocą interpolatorów programowych, wykorzystujących złożony zapis zarysów przedmiotu za pomocą funkcji wielomianowych i krzywych sklepanych.

Dla pokazania zasadniczych różnic i korzyści ze stosowania interpolatorów w sterowaniach I generacji (sterowniki NC) i II generacji (sterowniki CNC) omówiono

także powszechnie znane zasady interpolacji z wykorzystaniem rozwiązań sprzętowych dla interpolacji liniowej i kołowej. Następnie przedstawiono metody programowe i matematyczne zasady opisu zarysów przedmiotu za pomocą parametrycznych funkcji wielomianowych trzeciego stopnia, a także za pomocą krzywych Beziera i krzywych NURBS.

Na koniec przedstawiono przykłady praktycznych zastosowań metod programowych, bazując na informacjach zawartych w podręcznikach programowania sterowników CNC firm Siemens i Heidenhain.

LITERATURA

- [1] ADAMSKI W., 2005, *Krzywe stosowane w definiowaniu kształtu części w systemach CAD/CAM*, Część I, *Mechanik*, 3, 196-199.
- [2] ADAMSKI W., 2005, *Krzywe stosowane w definiowaniu kształtu części w systemach CAD/CAM*, Część II, *Mechanik*, 5-6, 534-537.
- [3] LEWANDOWSKI T., 1967, *Schematy strukturalne interpolatorów sterowanych programowo*, *Przegląd Mechaniczny*, 16, 495-498.
- [4] KOSMOL J., 2000, *Automatyzacja obrabiarek i obróbki skrawaniem*, wydanie II, WNT, Warszawa, 444.
- [5] MIERZEJEWSKI J., 1977, *Serwomechanizmy obrabiarek sterowanych numerycznie*, WNT, Warszawa, 281.
- [6] PRITSCHOW G., 1995, *Technika sterowania obrabiarkami i robotami przemysłowymi*, Oficyna wydawnicza Politechniki Wrocławskiej, Wrocław, 524.
- [7] HEIDENHAIN, *iTNC 530 Manual*.
- [8] HEIDENHAIN, *360-994-82 iTNC530 Manual*.
- [9] SIEMENS, *Programming Guide 11/2002 Edition*.
- [10] WIAK S. i inni, 2010, *Mechatronika*, t. 2, Akademicka Oficyna Wydawnicza EXIT, Politechnika Łódzka, Łódź, 746.
- [11] www.zobaczycmatematyke.krak.pl/025-Zolkos-Krakow/bezier.html
- [12] www.ics.p.lodz.pl/~piet/11_krzywe_i_powierzchnie, *Reprezentacja krzywych i powierzchni*.
- [13] https://pl.wikipedia.org/wiki/Wymierna_krzywa_B%C3%A9ziera

INTERPOLATION IN COMPUTERIZED NUMERICAL CONTROL

The paper presents advantages and disadvantages of interpolators used in numerically controlled machine tools. Interpolators used in I generation NC systems, and in CNC (systems of II generation) were discussed. In the first group, linear and circular interpolators based on hardware solutions, and in the second one, based on software solutions were presented. The hardware and software interpolators were compared taking into account accuracy of interpolation and the highest obtainable speed of feed, which has the biggest influence on productivity.

Keywords: *numerical control, NC, CNC, interpolation, spline, polynomial*