

ŁUKASZ ŚLUSARCZYK\*

## BUDOWA KONWERTERA KODU G WSPÓŁPRACUJĄCEGO Z PROGRAMAMI CAD/CAM

### BUILDING OF G CODE CONVERTER COOPERATING WITH CAD/CAM SOFTWARE

#### Streszczenie

Artykuł przedstawia metodę konwersji pliku NC, w której za pomocą instrukcji kodu G można zapisać trajektorię ruchu narzędzia w formacie wymaganym przez określony układ sterowania maszyny CNC. Przyjęto, że format pliku po konwersji będzie zgodny z układem sterowania 2-osiowej elektroerozyjnej wycinarki drutowej BP-95d. Na podstawie stworzonych algorytmów opracowano aplikację komputerową *Konwerter Kodu G*. W dalszej części artykułu, wykorzystując stworzoną aplikację, oszacowano średni względny błąd interpolacji liniowej łuku ze względu na przyjęty krok interpolacji.

*Słowa kluczowe: obróbka, WEDM, CNC*

#### Abstract

The paper presents the method of the NC file conversion, in which the tool movement trajectory can be saved in the format, required by the CNC machine control system, with the help of G code instructions. It was assumed, the file format after conversion will be compatible with the control system of the 2-axis wire electrical discharge machine BP-95d. Basing on the created algorithms a computer application "G Code Converter" was worked out. In the farther part of the paper, using the created application, a mean relative error of the linear interpolation of the arc was estimated, in view of the interpolation step.

*Keywords: machining, WEDM, CNC*

\* Dr inż. Łukasz Ślusarczyk, Instytut Technologii Maszyn i Automatykacji Produkcji, Wydział Mechaniczny, Politechnika Krakowska.

## 1. Wstęp

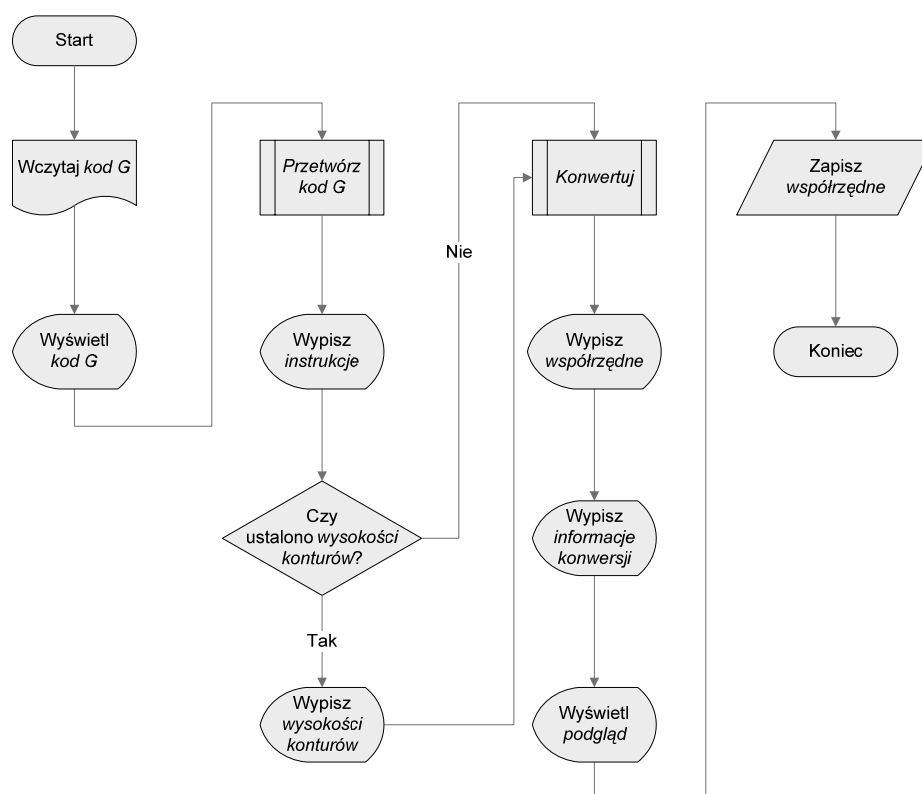
Mimo określenia standardu programowania maszyn CNC (norma ISO 6983), pojawiają się problemy niezgodności programów NC wygenerowanych przez różne systemy CAD/CAM z układami sterowania maszyn. Opracowano algorytmy umożliwiające konwersję pliku NC do formatu wymaganego przez określony układ sterowania maszyny. Przyjęto, że format pliku po konwersji będzie zgodny z układem sterowania 2-osiowej elektroerozyjnej wycinarki drutowej BP-95d [9]. Na podstawie stworzonych algorytmów zbudowano program komputerowy o nazwie Konwerter Kodu G. Działanie programu polega na przetworzeniu instrukcji kodu G, za pomocą których zapisano kontur elementu, do postaci ciągu współrzędnych punktów opisujących ten kontur. Poniżej na rys. 1 przedstawiono zastosowanie Konwertera Kodu G jako programu łączącego aplikację CAM z elektroerozyjną wycinarką drutową BP-95d.



Rys. 1. Konwerter Kodu G – program do konwersji pliku NC na ciąg współrzędnych

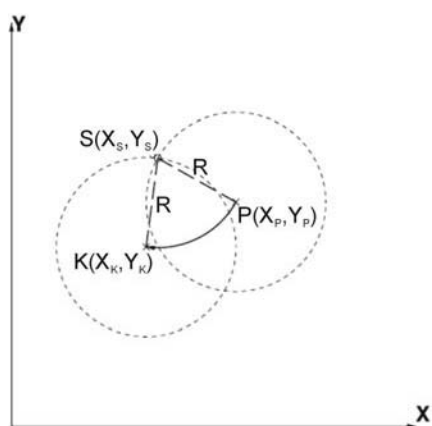
Fig. 1. G Code Converter – program for the NC file conversion to coordinates sequences

Ogólny algorytm działania Konwertera Kodu G przedstawiono na rys. 2 [6]. Po wczytaniu pliku NC, na etapie przetwarzania kodu G, zostają wyodrębnione słowa określające posuwu robocze. Kolejnym etapem działania algorytmu jest ustalenie wysokości wycinanego przedmiotu. Istotną cechą programu jest możliwość przetworzenia plików NC wygenerowanych dla 4-osiowego układu sterowania, gdzie kształt konturu górnego i dolnego jest różny. Po konwersji otrzymujemy ciąg współrzędnych punktów i podgląd konturu. W przypadku kształtów zbudowanych z odcinków prostych wyznaczenie par współrzędnych jest stosunkowo proste. Jeżeli natomiast kształt wycinanej figury jest bardziej złożony, posiada zaokrąglenia, łuki, krzywe, należy zastosować interpolację (np. liniową) w celu zdefiniowania punktów pośrednich między punktem początkowym a końcowym łuku. Aby wyznaczyć współrzędne punktów znajdujących się na łuku, należy wyznaczyć wartości pomocnicze. Pierwszymi wartościami są parametry występujące po instrukcji G2 lub G3. Punkt początkowy łuku jest punktem powstałym z zakończenia poprzedniej operacji, poprzedzającej instrukcję generującą łuk, tak więc nie jest podawany jako parametr ani do instrukcji G2, ani do instrukcji G3, a jest uwzględniany do obliczeń. Współrzędne punktu końcowego łuku są podane wprost po instrukcji G2 lub G3. W zależności od tego, czy podany jest promień łuku  $R$  (rys. 3), czy też współrzędne przyrostowe środka obrotu



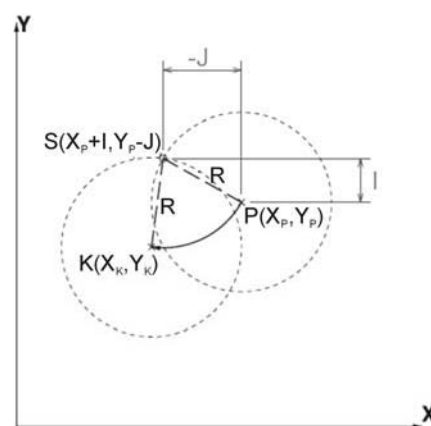
Rys. 2. Ogólny schemat algorytmu działania Konwertera Kodu G

Fig. 2. General algorithm of the G Code Converter



Rys. 3. Programowanie łuku z promieniem

Fig. 3. Arc programming with radius



Rys. 4. Programowanie łuku ze środkiem obrotu

Fig. 4. Arc programming with the center of rotation

łuku  $I$  i  $J$  (rys. 4), wyznaczane są dane uzupełniające te parametry. Jeżeli podano promień  $R$ , z odpowiednich wzorów (1) i (2) wyliczane są współrzędne środka obrotu. Jeżeli natomiast podano współrzędne przyrostowe  $I$  i  $J$ , wyliczany jest promień ze wzoru (3) [4].

$$X_S = \frac{X_p^3 + X_p Y_p^2 + X_p Y_K^2 - X_p^2 X_K - 2X_p Y_p Y_K}{2(X_p^2 + X_K^2 + Y_p^2 + Y_K^2 - 2X_p X_K - 2Y_p Y_K)} + \frac{X_K^3 + X_K Y_p^2 + X_K Y_K^2 - X_p X_K^2 - 2X_K Y_p Y_K}{2(X_p^2 + X_K^2 + Y_p^2 + Y_K^2 - 2X_p X_K - 2Y_p Y_K)} - \frac{\sqrt{(Y_p - Y_K)^2 (-X_p^2 - X_K^2 - Y_p^2 - Y_K^2 + 2X_p X_K + 2Y_p Y_K + 4R^2)}}{2\sqrt{(X_p^2 + X_K^2 + Y_p^2 + Y_K^2 - 2X_p X_K - 2Y_p Y_K)}} \cdot zn \quad (1)$$

$$Y_S = \frac{Y_p^3 + Y_p X_p^2 + Y_p X_K^2 - Y_p^2 Y_K - 2Y_p X_p X_K}{2(X_p^2 + X_K^2 + Y_p^2 + Y_K^2 - 2X_p X_K - 2Y_p Y_K)} + \frac{Y_K^3 + Y_K X_p^2 + Y_K X_K^2 - Y_p Y_K^2 - 2Y_K X_p X_K}{2(X_p^2 + X_K^2 + Y_p^2 + Y_K^2 - 2X_p X_K - 2Y_p Y_K)} + \frac{\sqrt{(X_p - X_K)^2 (-X_p^2 - X_K^2 - Y_p^2 - Y_K^2 + 2X_p X_K + 2Y_p Y_K + 4R^2)}}{2\sqrt{(X_p^2 + X_K^2 + Y_p^2 + Y_K^2 - 2X_p X_K - 2Y_p Y_K)}} \cdot zn \quad (2)$$

gdzie:

dla instrukcji G2:

$$zn = \begin{cases} 1 & \text{dla } R > 0 \\ -1 & \text{dla } R < 0 \end{cases}$$

dla instrukcji G3:

$$zn = \begin{cases} 1 & \text{dla } R < 0 \\ -1 & \text{dla } R > 0 \end{cases}$$

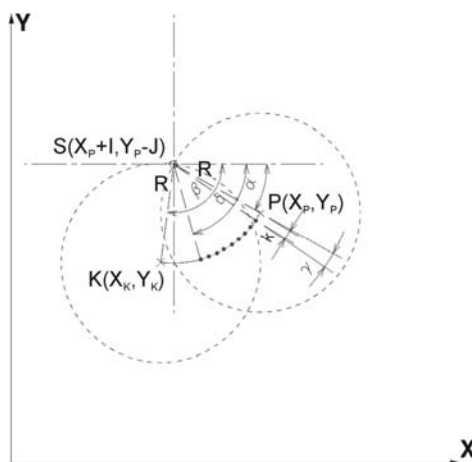
Wzór stosowany do wyliczenia promienia łuku

$$R = \sqrt{(X_S - X_p)^2 + (Y_S - Y_p)^2} \quad (3)$$

Znając wartość promienia, można wyznaczyć położenia aktualnie wyszukiwanego punktu na łuku. Na rysunku 5 przedstawiono algorytm wyznaczania położenia kolejnych punktów na łuku. Na wstępie zostają wyznaczone wartości pomocnicze, którymi się cztery kąty:  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ . Dla każdego łuku wartości trzech pierwszych kątów wylicza wartości tylko raz. Natomiast czwarty kąt  $\delta$ , o zmiennej wartości służy do wyznaczenia położenia aktualnie wyszukiwanego punktu na łuku [6].

Znając aktualną wartość kąta  $\delta$ , można wyznaczyć współrzędne nowego punktu na łuku powstałego z przecięcia ramienia kąta  $\delta$  z łukiem. Współrzędne tego punktu wyznacza wzór

$$\begin{aligned} X &= X_S + \cos(\delta) \cdot R \\ Y &= Y_S - \sin(\delta) \cdot R \end{aligned} \quad (4)$$



Rys. 5. Algorytm wyznaczania położenia punktów na łuku:  $k$  – krok interpolacji,  $\alpha$  – kąt o wierzchołku w punkcie środka obrotu między osią poziomą a ramieniem powstałym z punktu  $S$  oraz punktu początku łuku  $P$ ,  $\beta$  – kąt o wierzchołku w punkcie środka obrotu między osią poziomą a ramieniem powstałym z punktu  $S$  oraz punktu końca łuku  $K$ ,  $\gamma$  – kąt powstały w wyniku określenia kroku interpolacji,  $\delta$  – kąt o zmiennej wartości określający położenie aktualnie wyznaczonego punktu na łuku

Fig. 5. Algorithm of determination of the arc points position

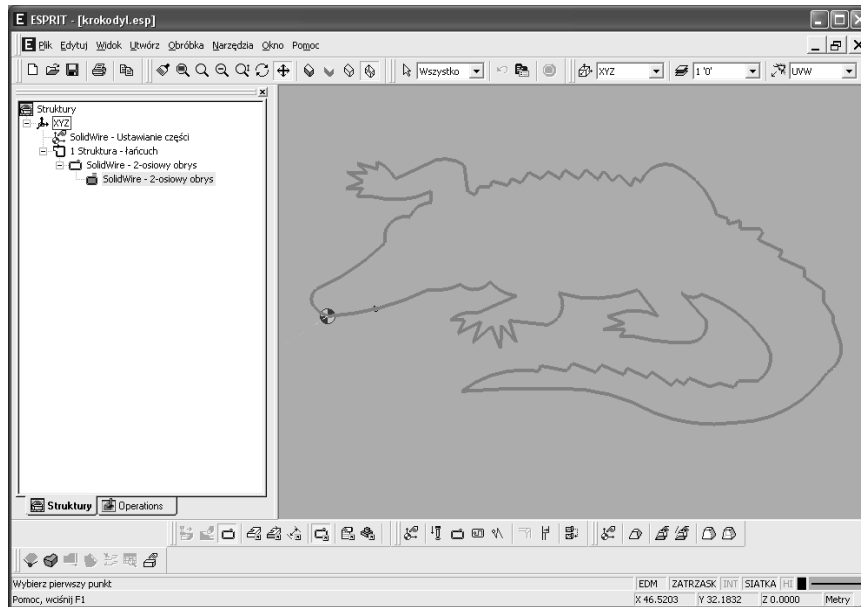
Dla instrukcji G2 rośnie z każdym powstałym punktem o stałą wartość  $\gamma$ , a dla instrukcji G3 maleje z każdym powstałym punktem o stałą wartość  $\gamma$ .

## 2. Konwerter kodu G

Rysunek 6 przedstawia kontur elementu o takim samym zarysie górnym i dolnym z zaznaczoną trajektorią cięcia, przygotowany w module do obróbki elektroerozyjnej SolidWire Gold programu CAD/CAM Esprit [7, 8]. Na podstawie utworzonego konturu wygenerowano plik NC.

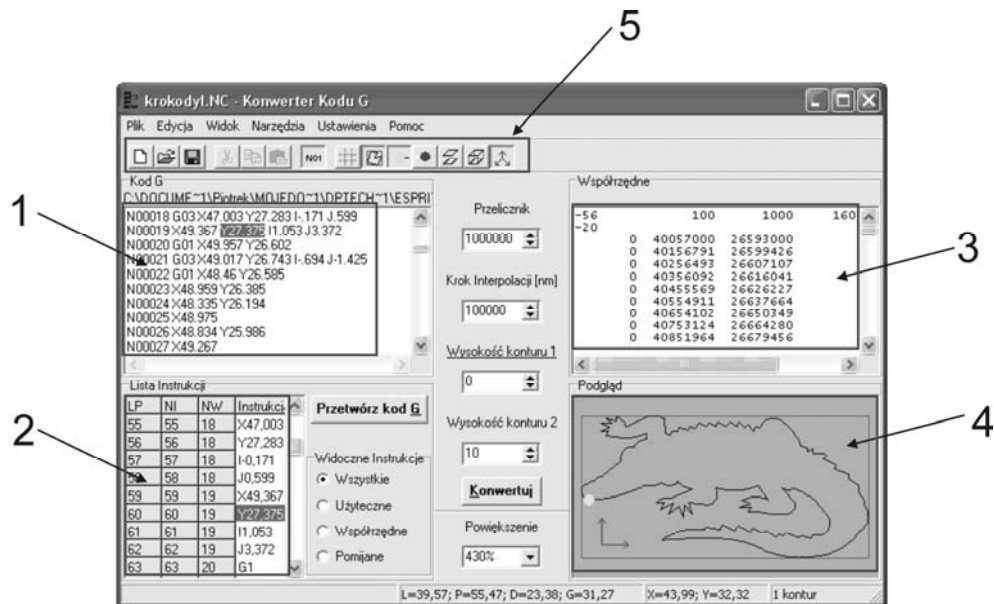
Na rysunku 7 przedstawiono interfejs programu Konwerter Kodu G. W oknie nr 1 widoczna jest zawartość wygenerowanego poprzednio pliku NC, opisującego kontur elementu. Okno nr 2 zawiera listę instrukcji kodu G po przetworzeniu.

Po konwersji w oknie nr 3 pojawia się ciąg współrzędnych punktów tworzących kontur oraz podgląd elementu, okno nr 4. W oknie nr 5 znajdują się ikony funkcyjne programu umożliwiające między innymi wygenerowanie podglądu 3D, zaznaczenie punktu startu lub punktu selekcji. Rysunek 8 przedstawia kontur elementu stworzony w aplikacji CAD/CAM Esprit. Na rysunku 9a i 9b przedstawiono efekt działania wybranych funkcji Konwertera Kodu G. Rysunek 9a przedstawia widok 3D elementu po konwersji, natomiast na rys. 9b zaznaczony został punkt startu – początek drążenia oraz punkt selekcji – fragment konturu wyodrębniony z instrukcji kodu G.



Rys. 6. Trajektoria cięcia stworzona w module SolidWire Gold programu CAD/CAM Esprit

Fig. 6. Cutting trajectory created in the module SolidWire Gold of the CAD/CAM Esprit software



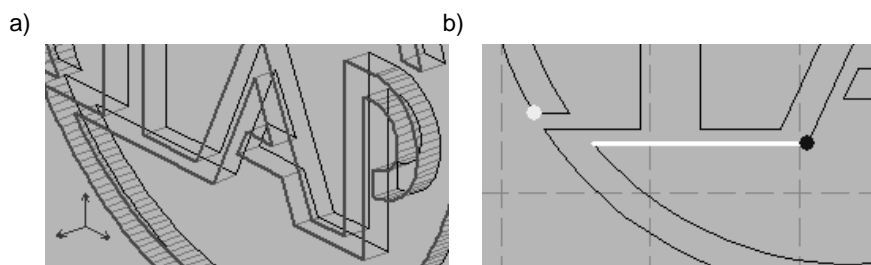
Rys. 7. Interfejs programu Konwerter Kodu G

Fig. 7. Program interface of the G Code Conversion



Rys. 8. Kontur elementu stworzony w aplikacji CAD/CAM Esprit

Fig. 8. Element outline created in the CAD/CAM Esprit software

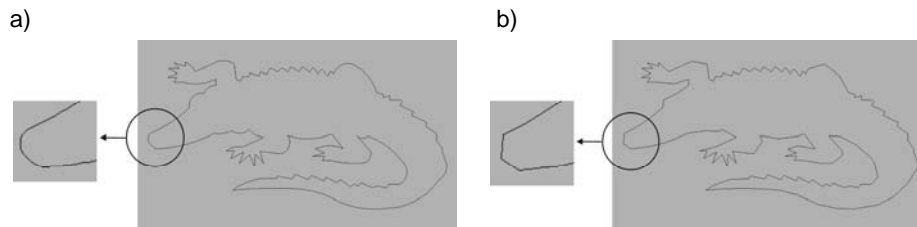


Rys. 9. Efekt działania wybranych funkcji Konwertera Kodu G: a) widok 3D elementu po konwersji, b) punkt startu oraz punkt selekcji

Fig. 9. Result of G Code Converter operation: a) 3D part view after conversion, b) starting point and selection point

### 3. Krok interpolacji

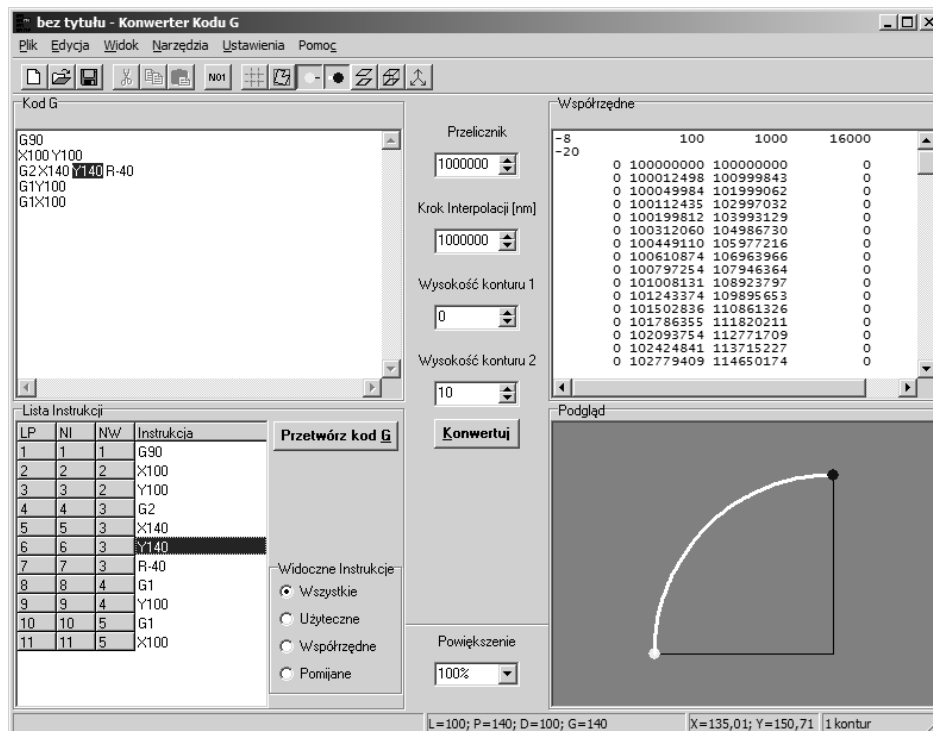
Na etapie konwersji pliku NC w Konwerterze Kodu G należy określić krok interpolacji. Przy jego doborze należy kierować się dwoma zasadami. Pierwsza zasada, mówiąca o zastosowaniu względnie małego kroku interpolacji, pozwala uzyskać lepszy efekt końcowy przez wygładzenie konturów w miejscach, gdzie użyto instrukcji tworzących łuki. Druga istotna zasada wynika z pojemności pamięci układu sterującego elektroerozyjnej wycinarki drutowej BP-95d. Dla małego kroku interpolacji mniejsze odległości między punktami pośrednimi łuku powodują zwiększenie ilości punktów w pliku wynikowym. Chcąc uzyskać optymalną ilość punktów przy zachowaniu powyższych ograniczeń, należy w Konwerterze Kodu G zadać niską wartość kroku interpolacji, a program zapyta nas o automatyczny dobór tego parametru. Poniżej na rys. 7 przedstawiono kontury wygenerowane dla różnych wartości kroku interpolacji. Rysunek 7a przedstawia kontur z krokiem interpolacji 0,1 mm, natomiast rys. 7b przedstawia kontur z krokiem interpolacji 1 mm.



Rys. 10. Kontury wygenerowane dla różnych wartości kroku interpolacji: a) krok interpolacji 0,1 mm, b) krok interpolacji 1 mm

Fig. 10. Outlines generated for different values of interpolation step: a) interpolation step 0,1 mm, b) interpolation step 1 mm

Podjęto próbę określenia wpływu kroku interpolacji na kształt powierzchni obrabianej. W tym celu za pomocą Konwertera Kodu G wygenerowano pliki określające trajektorię wycinka koła o kącie rozwarcia  $90^\circ$  i promieniu 40 mm – rys. 11, dla trzech kroków interpolacji łuku: 0,1 mm, 0,5 mm, 1 mm.

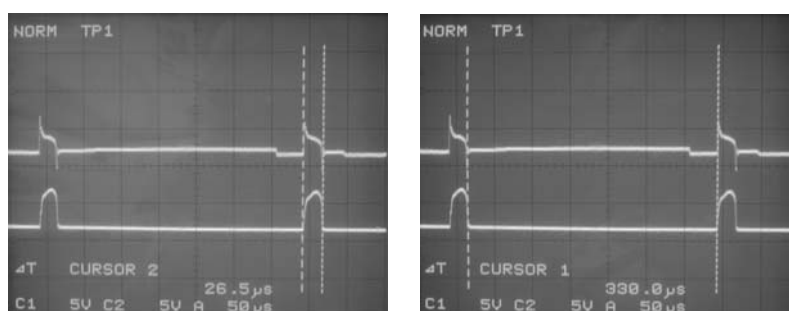


Rys. 11. Konwerter Kodu G z wygenerowanym plikiem określającym trajektorię wycinka koła o kącie rozwarcia  $90^\circ$  i promieniu 40 mm, dla kroku interpolacji łuku 0,1 mm

Fig. 11. G Code Converter with the generated file, determining the circular sector trajectory with obtuse angle  $90^\circ$  and radius 40 mm, for interpolation step 0,1 mm



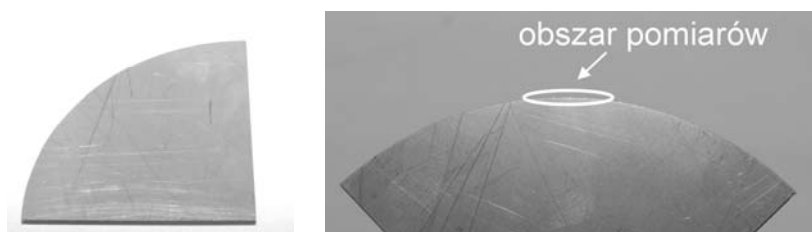
W dalszym etapie prac przygotowane kontury zostały wycięte na elektroerozyjnej wycinarce drutowej BP-95d. Materiałem obrabianym była blacha nierdzewna o grubości 1,5 mm. Wartość prądu drażenia dla każdej operacji wycinania wynosiła 72 A. Czas cięcia każdego z trzech elementów wynosił ok. 21 min. Prędkość posuwu głowicy została ustawiona na maksymalną wartość gwarantującą wysoką skuteczność obróbki i wynosiła 0,11 mm/s [1, 2, 3, 5]. Podczas procesu wycinania rejestrowano przebiegi prądu i napięcia na oscyloskopie. Przykładowe przebiegi przedstawiono na rys. 12. Czasy impulsów i czasy przerw uzyskiwane z generatora były zbliżone dla wszystkich trzech przypadków i wynosiły odpowiednio: czas impulsu ok. 26,5  $\mu$ s, natomiast czas przerwy ok. 330  $\mu$ s.



Rys. 12. Przykładowe przebiegi prądu i napięcia zarejestrowane podczas procesu obróbki

Fig. 12. Examples of voltage and electric current diagrams recorded during machining process

Na rysunku 13 przedstawiono element po wycięciu z zaznaczonym obszarem, na którym dokonywano pomiarów.

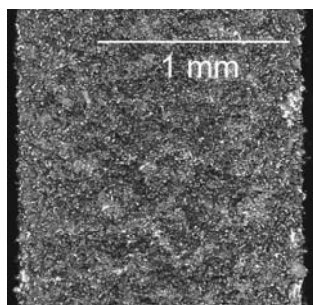


Rys. 13. Element z zaznaczonym obszarem na którym dokonywano pomiarów

Fig. 13. Element with marked measurement area

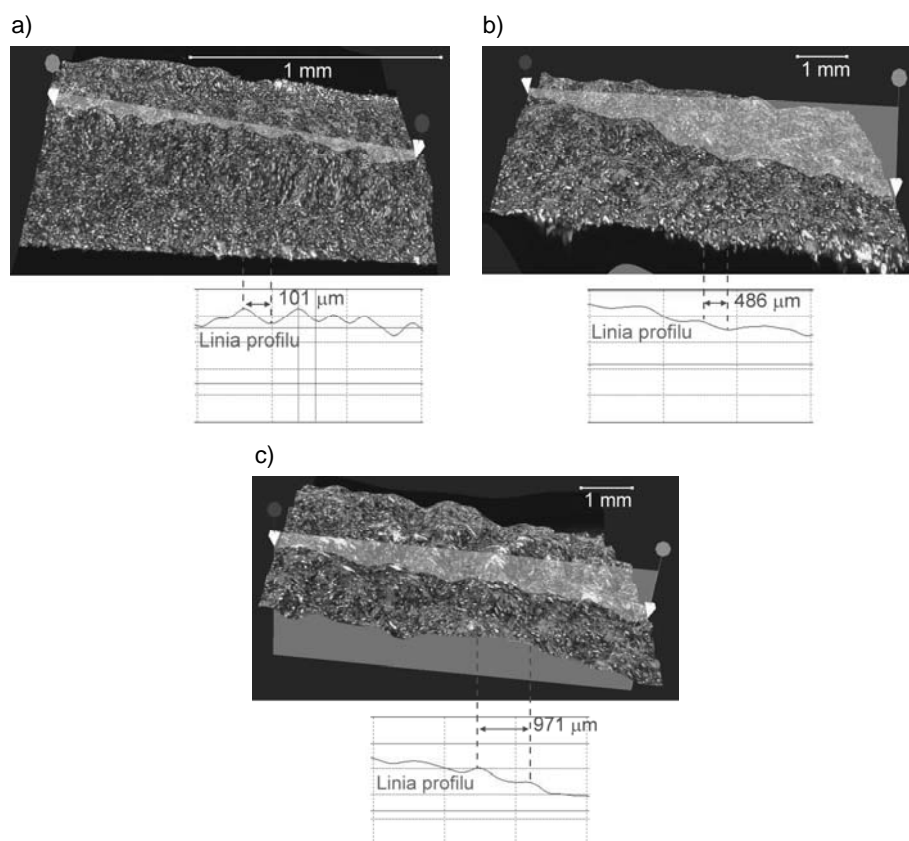
Analizę i pomiary otrzymanych powierzchni po obróbce realizowano mikroskopem cyfrowym Keyence VHX-600. Stosowano powiększenie 100 i 150 $\times$ . Przykładowa fotografia powierzchni obrobionej dla kroku interpolacji 0,1 mm w powiększeniu 100 $\times$  została przedstawiona na rys. 14.

Wykorzystując specjalistyczne oprogramowanie, dokonano serii pomiarów przekrojów powierzchni łuków. Przykładowe wyniki pomiarów dla trzech kroków interpolacji 0,1 mm, 0,5 mm oraz 1 mm przedstawiono na rys. 15a–c.



Rys. 14. Przykładowa fotografia powierzchni obrabianej, dla kroku interpolacji 0,1 mm w powiększeniu 100×

Fig. 14. Example photo of the machined surface, for interpolation step 0,1 mm; magnification 100×

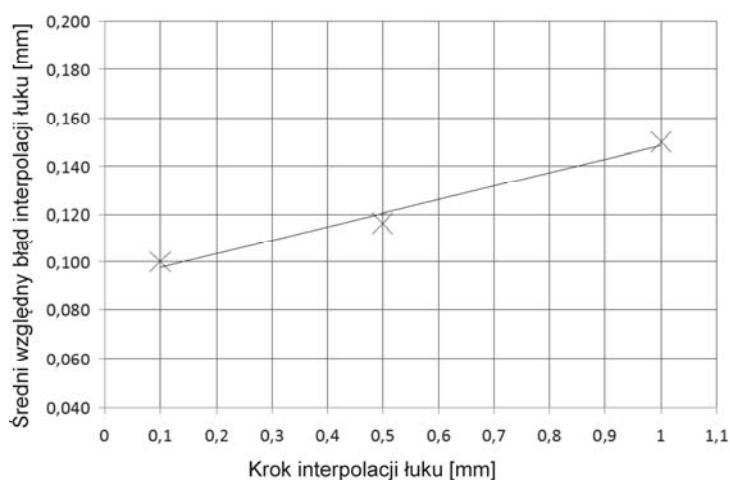


Rys. 15. Przykładowa fotografia przekroju powierzchni łuku, w powiększeniu 150× dla kroku interpolacji: a) 0,1 mm, b) 0,5 mm, c) 1 mm

Fig. 15. Example photo of the arc cross-section area, magnification 150× for interpolation step: a) 0,1 mm, b) 0,5 mm, c) 1mm

#### 4. Wnioski

Opracowano metodę konwersji pliku NC, w którym opisana jest trajektoria ruchu narzędzia, do postaci interpretowalnej przez układ sterowania 2-osiowej elektroerozyjnej wycinarki drutowej BP-95d. W wyniku wielu prób stwierdzono istotny wpływ wielkości kroku interpolacji na dokładność kształtu wycinanego przedmiotu. Oszacowano średni względny błąd interpolacji łuków wycinanych na elektroerozyjnej przecinarce drutowej BP-95d, w zależności od przyjętego kroku interpolacji rys. 16. Dobierając optymalne parametry procesu, wykonano przykładowe elementy.



Rys. 16. Zależność średniego względnego błędu interpolacji łuku od przyjętego kroku interpolacji

Fig. 16. Dependence of the mean relative error of the linear interpolation of the arc of the interpolation step

#### Literatura

- [1] Albinski K., Miernikiewicz A., Ruszaj A., Zimny J., *Laboratorium obróbki erozyjnej*, PWN, Warszawa 1980.
- [2] Miernikiewicz A., *Doświadczalno-teoretyczne podstawy obróbki elektroerozyjnej (EDM)*, Seria Mechanika, Monografia 274, Politechnika Krakowska, Kraków 2000.
- [3] Miernikiewicz A., *Instrukcje do ćwiczenia Obróbka Elektroerozyjna*, Wydawnictwo PK, Kraków 1990.
- [4] Snarska A., *Ćwiczenia z Delphi 3.0, 4.0, 5.0*, Mikom, 2000.
- [5] Siwczyk M., *Obróbka elektroerozyjna*, WNT, Warszawa 1981.
- [6] Ślusarczyk Ł., Śliwiński P., praca badawcza niepublikowana.
- [7] <http://www.dpotechnology.com/>.
- [8] <http://www.espricam.pl/>.
- [9] <http://www.zapbp.com.pl/>.