



WYDZIAŁ INŻYNIERII ZARZĄDZANIA

Katedra Zarządzania Produkcją



## INSTRUKCJA DO ZAJĘĆ LABORATORYJNYCH

Laboratorium z przedmiotu:

**Procesy i techniki produkcyjne**

Kod przedmiotu:

**KS04202**

Temat:

**Programowanie zabiegów frezarskich z wykorzystaniem parametrów R**

Nr ćwiczenia:

**5**

Kierunek:

**Zarządzanie i Inżynieria Produkcji**

### Cel ćwiczenia

Zapoznanie studentów z możliwościami programowania parametrycznego w systemie Sinumerik. Zdobywanie umiejętności tworzenia programów obróbkowych z wykorzystaniem R parametrów, pętli i skoków w programie.

### Wyposażenie stanowiska

Frezarka numeryczna Skolar X3, komputer stacjonarny lub przenośny, suwmiarka, liniał, mikrometr.

### Zakres ćwiczenia

Planowanie obróbki części z wykorzystaniem parametryzacji. Tworzenie programów z obliczaniem wartości parametrów podczas przebiegu i przyporządkowywaniem ich innym adresom NC. Planowanie obróbki z wykorzystaniem skoków w programie.

### Zaliczenie ćwiczenia

Zaliczenie ćwiczenia odbywa się na podstawie sprawdzianu wstępnego, obserwacji pracy studenta w czasie zajęć i wykonanego sprawozdania sporządzonego zgodnie z protokołem dołączonym do niniejszej instrukcji.

### LITERATURA

1. Podręcznik programowania i obsługi SINUMERIK 808D. Frezowanie, część 2: Programowanie (instrukcje Siemens) 12/2012
2. Chlebus E.: Techniki komputerowe CAx w inżynierii produkcji, WNT, Warszawa, 2004
3. Przybylski W., Deja M.: Komputerowo wspomagane wytwarzanie maszyn: podstawy i zastosowanie, WNT, Warszawa, 2007.
4. Habrat W.: Obsługa i programowanie obrabiarek CNC: podręcznik operatora, Wyd. "KaBe", Krosno 2015.
5. Pytlak B., Zmienne w układzie sterowania CNC SINUMERIK Operate, Mechanik, Nr 3/2017

Opracował:

dr inż. Krzysztof Łukaszewicz

Sprawdził:

dr hab. inż. Jerzy Jaroszewicz, prof. nzw.

Politechnika Białostocka 2018

## 1. Wprowadzenie

W układzie sterowania SINUMERIK programowanie parametryczne opiera się na wykorzystaniu zmiennych, które w połączeniu z funkcjami obliczeniowymi i strukturami kontrolnymi zapewniają wysoką elastyczność tworzonych programów. Zmienne te można podzielić na systemowe i użytkownika. Ze zmiennych systemowych korzysta oprogramowanie systemowe, można je też odczytywać i zapisywać w programach obróbki użytkownika. Producent wstępnie definiuje znaczenie zmiennych systemowych. Istnieje jednak możliwość ich zmiany przez tzw. redefinicję. Zmienne systemowe zapewniają dostęp do aktualnego stanu sterowania, maszyny i procesu obróbki oraz pozwalają parametryzować układ sterowania. Standardowe oznaczenie zmiennych systemowych składa się ze znaku \$ + pierwsza litera (rodzaj danych) + druga litera (obszar obowiązywania) np.: \$M – dane maszynowe, \$S – dane ustawcze, \$T – dane narzędzi, \$P – wartości programowane. Zmienne systemowe przebiegu głównego, np.: \$V – dane Serwo, \$R – parametry obliczeniowe odróżnia dodatkowy znak \$. Jest on zarezerwowany do zmiennych systemowych i nie wolno go używać w zmiennych użytkownika. Szczegółowy wykaz i opis zmiennych systemowych znajduje się w dokumentacji układu sterowania SINUMERIK.

Zmienne użytkownika obejmują zmienne wstępnie zdefiniowane w systemie oraz zmienne definiowane przez samego użytkownika. Typ zmiennych użytkownika wstępnie zdefiniowanych jest z góry określony, natomiast ich liczbę określa się w odpowiednich danych maszynowych. Do zmiennych tych zaliczyć można: R-parametry, globalne R-parametry RG, zmienne typu Link. Zmienne definiowane przez użytkownika są tworzone w systemie na stałe lub tylko na czas wykonywania programu. Zmienne te można podzielić w zależności od zakresu obowiązywania na: globalne GUD (Global User Data), programowe PUD (Program User Data) i lokalne LUD (Local User Data).

Użytkownik może zdefiniować w programie własną zmienną z różnych typów danych (LUD = dane lokalne użytkownika). Zmienne te są dostępne tylko w programie, w którym zostały zdefiniowane. Definiowanie zachodzi natychmiast na początku programu i może zostać również skojarzone z jednoczesnym przydziałem wartości. W innym przypadku wartością początkową jest zero. Nazwa zmiennej może zostać zdefiniowana przez programistę. Nazewnictwo podlega następującym zasadom:

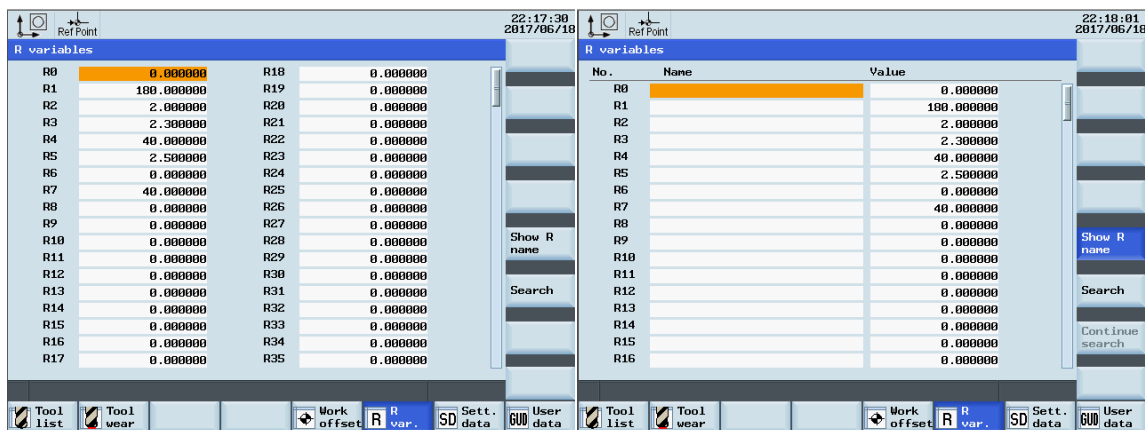
- można zastosować maksymalnie 32 znaki.
- pierwsze dwa znaki muszą być literami. Pozostałymi znakami mogą być litery, podkreślenie lub cyfry.
- nie należy stosować nazw wykorzystywanych już w systemie sterowania (adresy NC, słowa kluczowe, nazwy programów, podprogramów, itp.).

Zmienne pozwalają na przechowywanie liczb rzeczywistych, całkowitych, tablic, napisów tekstowych, itp. Aby wykorzystać zmienną należy ją najpierw zadeklarować, np.:

- DEF BOOL varname1 ;typ Boole'a, wartości: TRUE (=1), FALSE (=0)
- DEF CHAR varname2 ;typ znakowy, 1 znak z kodu ASCII: „a”, „b”, ...;wartość numeryczna kodu: 0 ... 255
- DEF INT varname3 ;typ całkowity, wartości całkowite, 32-bitowy zakres wartości,
- DEF REAL varname4 ;typ rzeczywisty, liczba naturalna, (8 miejsc dziesiętnych, znak arytmetyczny i separator dziesiętny) lub ;zapis wykładniczy: ± (10 do potęgi -300 ... 10 do potęgi +300)
- DEF STRING [string length] varname41; Typ STRING, [długość ciągu].

Każdy typ danych wymaga własnego wiersza w programie. Niemniej jednak w jednym wierszu można zdefiniować wiele zmiennych tego samego typu. Cechą charakterystyczną takich zmiennych jest ich lokalność (dostępne tylko w obrębie programu, w którym są zadeklarowane). Ich stosowanie zaleca się jednak doświadczonym programistom. Na poziomie średniozaawansowanym do dyspozycji programisty istnieje również wydajne narzędzie o nazwie parametry R.

R parametry to jednowymiarowa tabela w pamięci układu sterowania, indeksowana adresami R. W tabeli tej przechowywane są liczby rzeczywiste. Liczba komórek tabeli, a więc i zakres R parametrów zależy od konkretnego układu, w Sinumeriku 808D jest to 300 adresów od R0 do R299. Odwołując się do x-tej komórki tabeli przez adres Rx można zarówno pobierać z niej dane, jak i tam je wprowadzać. Zaletą R parametrów jest to, że podgląd ich wartości jest dostępny na pulpicie operatorskim. Możliwe jest zatem śledzenie zmian ich wartości w trakcie wykonywania programów sterujących, co pozwala m.in. na diagnostykę błędów. Dla polepszenia czytelności R parametrów w programie obróbki często umieszcza się dodatkowy opis znaczenia danego R parametru, np. w formie komentarza w programie obróbki:  $R10=200$ ; *współrzędna w osi X*. Obok kolumny R parametrów można wyświetlić dodatkową kolumnę z ich opisem (rys. 1).



**Rys. 1.** Okna R parametrów z wywołaniem kolumny opisów

Wartości R parametrów mogą być przypisywane do innych adresów. Ich przeznaczeniem może być przechowywanie informacji zbieranych w trakcie wykonywania programów sterujących. Jednak podstawowym celem stosowania R parametrów jest wykonywanie na nich obliczeń. Mogą to być zarówno proste operacje arytmetyczne, jak i skomplikowane funkcje trygonometryczne, logarytmiczne i inne (można na nich operować również wartościami stałymi). Najważniejsze operatory i funkcje dostępne w Sinumeriku 808D:

- + – operator dodawania;
- – operator odejmowania;
- \* – operator mnożenia;
- / – operator dzielenia;
- DIV() – dzielenie całkowitoliczbowe;
- MOD – dzielenie modulo;
- SIN() – funkcja sinus (argument w [°]);
- COS() – funkcja cosinus (argument w [°]);
- TAN() – funkcja tangens (argument w [°]);
- ASIN() – odwrotna funkcja sinus (wynik w [°]);
- ACOS(), – odwrotna funkcja cosinus (wynik w [°]);
- SQRT() – pierwiastek kwadratowy;
- ABS() – moduł liczby;
- POT() – kwadrat liczby (potęga o wykładniku 2);
- TRUNC() – część całkowita liczby;
- ROUND() – zaokrąglenie do liczby całkowitej;
- LN() – logarytm naturalny;
- EXP() – funkcja eksponentialna (ex).

Przykłady:

$R10=3$  przypisanie wartości 3 do R parametru R10

$R11=R16$  przepisanie wartości z R16 do R11

$R13=R14+2$  operacja dodawania wartości z R14 i stałej 2

$R8=R7+R2-R3$  obliczenia dodawania i odejmowania

$R1=R2*R9/R7$  obliczenia mnożenia i dzielenia

$R15=R14+R4*R5$  obliczenia mnożenia i dodawania

W ostatnim przykładzie najpierw system wykonuje mnożenie (operatory mnożenia i dzielenia mają wyższy priorytet niż dodawania i odejmowania) a potem dodawania. Jeżeli chcemy jasno sprecyzować kolejność wykonywania obliczeń można wykorzystać nawiasy, np.:  $R10=(R14+R12)*R13$ .

Programy NC przetwarzają swoje bloki w kolejności, w jakiej zostały ułożone podczas pisania. Kolejność przetwarzania można jednak zmienić, wprowadzając tzw. skoki programu. Punktem docelowym skoku może być blok zawierający etykietę lub numer bloku. Blok ten musi znajdować się w programie. Instrukcja bezwarunkowego skoku wymaga odrębnego bloku. W Sinumerik istnieją dwie funkcje skoków bezwarunkowych:

- **GOTOF LABEL** – skok bezwarunkowy w przód (ang. GOTO Forward) do etykiety LABEL.
- **GOTOB LABEL** – skok bezwarunkowy w tył (ang. GOTO Back) do etykiety LABEL.

Etykieta (LABEL) jest to ciąg znaków alfanumerycznych, zakończonych znakiem „:”. Umieszcza się ją na początku bloku, np.: N50 LABEL1: G0 X50 Y30. Jeśli obecny jest również numer bloku, etykieta jest umieszczana za numerem bloku.

Przy skoku w przód jest poszukiwana etykieta od bieżącego bloku w kierunku końca programu. Przy skoku w tył - w kierunku początku programu. Poszukiwanie kończy się z chwilą znalezienia bloku zawierającego podaną w instrukcji etykietę. Oznacza to, że w programie sterującym może być kilka bloków z tą samą etykietą, a mimo tego adres skoku jest wyznaczony w sposób jednoznaczny. Jeżeli przy przeszukiwaniu programu układ sterowania nie znajdzie podanej etykiety to przerywa wykonanie programu sygnalizując błąd.

W przypadku skoków warunkowych, warunki skoku są formułowane za instrukcją IF. Jeśli warunek skoku (wartość niezerowa) jest spełniony, skok zachodzi. Punktem docelowym skoku może być blok zawierający etykietę lub numer bloku. Blok ten musi znajdować się w programie. Instrukcje warunkowego skoku wymagają odrębnego bloku. W tym samym bloku można umieścić wiele instrukcji warunkowego skoku.

- **IF warunek GOTOF LABEL** ;Skok do przodu
- **IF warunek GOTOB LABEL** ;Skok wstecz

Ogólnie „warunek” - parametr arytmetyczny, najczęściej porównanie dwóch wartości liczbowych, np.:

- **=** = równe
- **<** > nie równe
- **>** większe niż
- **<** mniejsze niż
- **>=** większe lub równe
- **<=** mniejsze lub równe

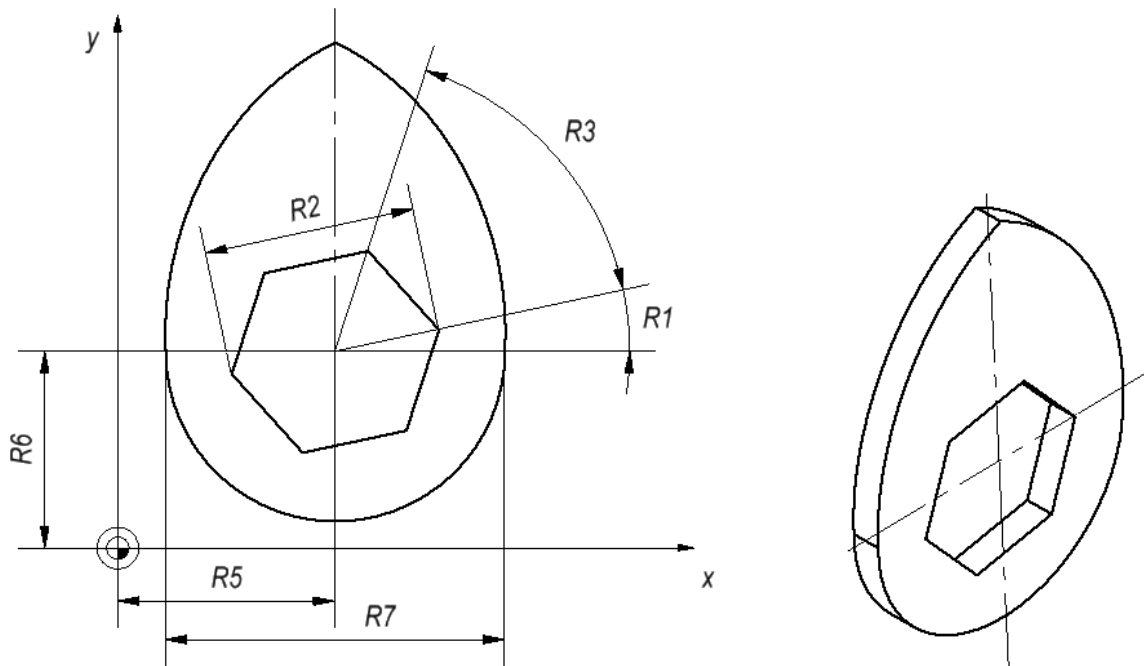
Przykład skoku warunkowego:

N100 IF R45==R7+1 GOTOB ETYKIETA3 - jeśli R45 jest równe R7 plus 1, przejście do bloku z ETYKIETA3

Jeżeli lista instrukcji do wykonania w przypadku pozytywnego lub negatywnego warunku logicznego jest długa, to może zawierać się w kilku blokach, jednak wtedy należy wykorzystać bardziej rozbudowaną postać instrukcji warunkowej: IF ENDIF, przy czym adresy IF i ENDIF muszą być jedynymi adresami w bloku.

## 2. PRZEBIEG ĆWICZENIA NR 5

Należy zaprogramować obróbkę konturów (rys. 2) krzywki której fragment ma postać spirali logarytmicznej, (obróbka od zewnątrz) oraz gniazda wielokątnego (obróbka od wewnątrz). Oba kontury są opisane wymiarami z użyciem parametrów. Przyjąć wartości wymiarów podane przez prowadzącego, a nieznanymi wymiary obliczyć korzystając z R parametrów. Przyjąć powierzchnię materiału  $Z=0$ , głębokość obróbki  $Z=-4$ . Obróbkę wykonać frezem palcowym o średnicy 6 mm.



**Rys. 2.** Kontury krzywki i wieloboku z zaznaczonymi nazwami R parametrów

Rozwiązanie:

- Ustalić przebieg obróbki, np: jakie narzędzia i kiedy będą używane do obróbki którego konturu, w jakiej kolejności będą wykonywane poszczególne elementy obrabianego przedmiotu.
- Sporządzić plan pracy. Ustalić kolejność kroków wszystkich procesów obróbkowych w maszynie, np.: przesuw szybko przy pozycjonowaniu, zmiana narzędzia, włączenie/wyłączenie wrzeciona, korekcja toru ruchu, dosunięcie/odsunięcie od konturu.
- Zdefiniować w programie parametry, opatrzyć komentarzami i przypisać im wartości podane przez prowadzącego.
  - R1 - kąt początkowy wieloboku,
  - R2 - przekątna wieloboku,
  - R3 - rozstaw kąta wieloboku (można również obliczyć na podstawie R4),
  - R4 - ilość boków,
  - R5 - położenie środka wieloboku w osi x
  - R6 - położenie środka wieloboku w osi y
  - R7 - średnica części cylindrycznej krzywki,
  - R8 - wykładnik równania spirali logarytmicznej,
  - R9 - przyrost kąta
- Przetłumaczyć kroki robocze na język programowania. Zapisać każdy krok jako blok (wzgl. bloki) NC.

- Zdefiniować w programie i wykorzystać etykiety i skoki warunkowe do stworzenia pętli realizującej zarys wieloboku i krzywki

*Wskazówka:*

Równanie spirali logarytmicznej we współrzędnych biegunowych ma następującą postać:

$$R(\varphi) = e^{\lambda \cdot \varphi}$$

gdzie:  $\lambda$  – wykładnik spirali;

$R, \varphi$  [rad] – współrzędne biegunowe (promień i kąt).

W przedstawionej krzywej występują dwa symetryczne odcinki spirali logarytmicznej. Aby wykonać ruch po krzywej danej równaniem (zakładając brak odpowiedniej interpolacji w układzie sterowania) należy interpolować go odcinkami linii prostej o małej długości, co z wystarczającą dokładnością pozwoli odwzorować dowolny kształt. Kąt ten dla pierwszej części odcinka spirali zmienia się od wartości początkowej  $0^\circ$  do  $90^\circ$ . Wartość o jaką zmienia się kąt zależy od dokładności, z jaką potrzebujemy odwzorować kształt spirali – im mniejsza wartość przyrostu kąta tym większa dokładność. Drugą część spirali można uzyskać przy użyciu analogicznego sposobu z tą różnicą, że wartość kąta we wzorze na promień  $R$  musi być równa  $(180 - \varphi)$ .

- W edytorze tekstowym stworzyć pełny program obróbki, zapisać plik (\*.mpf)
- Przenieść plik na maszynę CNC, przeprowadzić symulację obróbki i wykonać przedmiot.
- Dokonać pomiarów uzyskanego elementu, notując wymiary i inne spostrzeżenia, np. dotyczące wyglądu obrabianej powierzchni.

Przykładowy program:

<pre>;%_N_KL_MPF N10 G40 G55 G71 G90 G94 G64 N20 T1 D1 S3000 F500 M3 M6 N30 R5=.....;SRODEK X N40 R6= .....;SRODEK Y N50 G0 X=R5 Y=R6 N60 Z3 N70 G1 Z-4 N80 R2=.....;PRZEKATNA WIELOBOKU N100 R1=0 R2=R2/2 N110 R4=..... R3=360/R4 N120 IMB: G1 X=R2*cos(R1)+R5 Y=R2*sin(R1)+R6 N130 R1=R1+R3 R4=R4-1 N140 IF R4&gt;-1 GOTOB IMB N150 Z5 N160 R7=..... N170 R10=R7/2 ;PROMIEN KRZYWKI N170 G0 G40 X=R5-(R10+10) Y=R6 N180 G1 Z-4 N190 G42 X=R5-R10 Y=R6 N200 G3 X=R5+R10 CR=R10 N210 MSG("RUCH PO SPIRALI") N220 G111 X=R5 Y=R6 ;BIEGUN ; DANE WEJSCIOWE N230 R11=0 ;KAT FI POCZATEK N240 R8=..... ;WYKLADNIK N250 R12=0 ;PROMIEN POCZATEK</pre>	<pre>; POCZATEK LEWEJ SPIRALI N270 SPIR1: N280 R11=R11+R9 ;NOWY KAT N290 R13=R11*3.1415/180 ;RAD N300 R12=R10-1+EXP(R8*R13) ;NOWY PR N310 G1 RP=R12 AP=R11 ;RUCH N320 IF R11&lt;90 GOTOB SPIR1 ; START PRAWYJ SPIRALI N330 R11=90 N340 SPIR2: N350 R11=R11+R9 ;NOWY KAT N360 R13=(180-R11)*3.1415/180 ;RAD N370 R12=(R10+EXP(R8*R13)-1) ;NOWY PR N380 G1 RP=R12 AP=R11 ;RUCH N390 IF R11&lt;180 GOTOB SPIR2 N400 Z5 N410 G40 NORM N420 G0 SUPA Z0 D0 N430 M30</pre>
--	---

- Ogólna dyskusja w grupie nad otrzymanymi wynikami i zapis plików.

### **3. SPRAWOZDANIE**

Sprawozdanie winno zawierać:

- stronę tytułową,
- cel i zakres ćwiczenia laboratoryjnego,
- wypełniony protokół laboratoryjny, kopia pliku programu NC (\*.mpf)
- wnioski.

### **4. BHP**

W celu minimalizacji zagrożeń podczas testów pracownicy i studenci zobowiązani są do przestrzegania ogólnych zasad BHP oraz do przestrzegania przepisów porządkowych i organizacyjnych obowiązujących w laboratoriach PiTP. O przepisach tych studenci poinformowani zostali na zajęciach wstępnych.

### **4. PROTOKÓŁ**



## PROTOKÓŁ LABORATORYJNY

### Programowanie zabiegów frezarskich z wykorzystaniem parametrów R

Szkic konturu z wymiarami

Zadane parametry R	Fotografia wykonanego elementu
R1= ..... - kąt początkowy wieloboku R2= ..... - przekątna wieloboku, R3= ..... - rozstaw kąta wieloboku, R4= ..... - ilość boków, R5=..... - położenie środka wieloboku w osi x R6= ..... - położenie środka wieloboku w osi y R7=..... - średnica części cylindrycznej krzywki, R8= ..... - wykładnik równania spirali logarytmicznej, R9=..... - przyrost kąta	



Listing programu NC

--	--

.....  
data wykonania ćwiczenia

.....  
podpis prowadzącego