

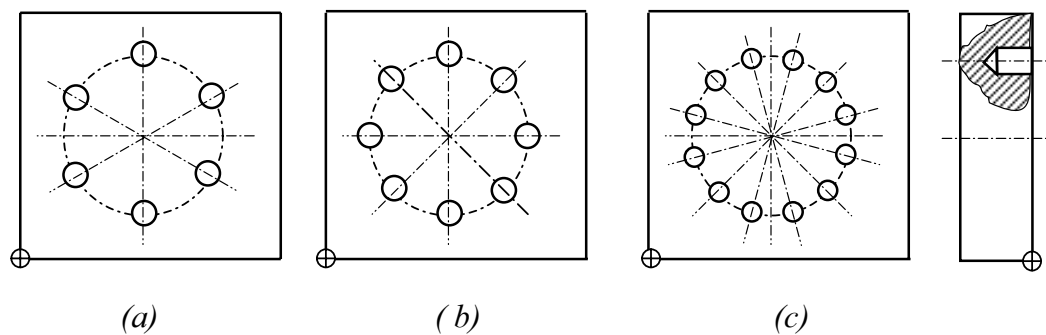
CHƯƠNG VII

CHƯƠNG TRÌNH THAM SỐ

7.1 Đặt vấn đề

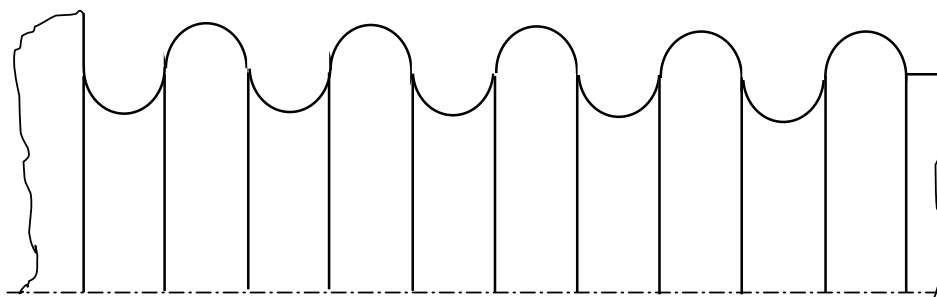
Một đặc điểm rất quan trọng trong khi gia công trên các máy điều khiển theo chương trình số CNC là đạt được hiệu quả kinh tế cao ngay cả trong điều kiện sản xuất hàng loạt vừa và nhỏ. Chính nhờ đặc tính linh hoạt cao của nó mà khi chuyển từ việc gia công kiểu chi tiết này sang gia công kiểu chi tiết khác người ta chỉ việc thay đổi chương trình. Điều đó lại càng phù hợp hơn trong thời đại sản xuất công nghiệp hiện đại mà yêu cầu về hình thức và mẫu mã không ngừng thay đổi nhằm tăng cường sức cạnh tranh và thoả mãn thị hiếu của người tiêu dùng. Tuy nhiên cũng cần phải thấy rằng các chi tiết gia công cơ khí rất phong phú và đa dạng về hình dáng, kích thước cũng như đặc điểm về công nghệ... Chính vì thế mà cần phải có một sự nghiên cứu về việc phân loại các chi tiết gia công cơ khí thành một số loại, kiểu hoặc nhóm chi tiết trên cơ sở có chung một số kiểu chu trình gia công trên máy CNC nhằm mục đích tổ chức quá trình gia công sao cho có hiệu quả nhất. Xuất phát từ ý tưởng trên, các nhà chế tạo máy CNC đã đưa vào trong hệ thống điều khiển số của máy CNC một kiểu lập chương trình mới được gọi là chương trình tham số. Với chương trình tham số, người ta sẽ lập một chương trình gia công trên cơ sở các kiểu hoặc nhóm chi tiết đại diện bằng việc đưa vào trong đó các tham số hình thức mà có thể gán các giá trị thực của nó theo từng chi tiết gia công cụ thể. Các tham số này có thể là các giá trị về toạ độ và các thông số về công nghệ như giá trị tốc độ cắt, lượng tiến dao, chiều sâu gia công; các thông số hình học của dao như bán kính và chiều dài dao...

Ví dụ:



Hình 7-1: Một số kiểu chi tiết trong lập trình tham số trên máy phay CNC

Ngày nay, người ta thường lập chương trình tham số cho các chi tiết có đặc điểm gần giống nhau về hình dáng hình học và phương pháp gia công. Các chi tiết này phải giống nhau cơ bản về chu kỳ gia công, ví dụ như khi gia công các lỗ được phân bố trên vòng tròn với các góc phân bố khác nhau hoặc là số lỗ khác nhau trên các tọa độ khác nhau hay khi gia công các rãnh được phân bố theo chiều dài trục như các loại trục cán...



Hình 7-2: Kiểu chi tiết gia công được lập trình trên máy tiện theo chương trình tham số

Ví dụ như trong các chu trình gia công trên máy phay và máy tiện được trình bày trong chương V, người ta đưa vào các tham số hình thức như S, E là số block đầu tiên và cuối cùng được thực hiện trong chương trình chính đối với hệ điều khiển FAGOR hoặc P, Q đối với hệ điều khiển FANUC; Các tham số L, M là giá trị lượng dư để lại cho gia công tinh theo phương X và Z đối với hệ điều khiển FAGOR hoặc U, W đối với hệ điều khiển FANUC; Các tham số C là chiều sâu cắt mỗi lần chạy dao đối với hệ điều khiển FAGOR hoặc U đối với hệ điều khiển FANUC...

Lập trình tham số cho phép sử dụng các biến trong một chương trình chính hay một chương trình con. Lúc mới khởi thảo ra kiểu lập trình này người ta chỉ quan niệm là lập với một số chu trình cứng để gia công một số bề mặt đặc trưng của chi tiết như khoan các lỗ sâu có bề phoi với lượng tiến dao không đổi hoặc thay đổi, chu trình gia công ăn dao theo biên dạng chi tiết...và sau này được mở rộng hơn cho các nhà chế tạo máy điều khiển theo chương trình số nhờ sự tích hợp từ nhiều cấu trúc khác nhau tổ chức vào trong hệ thống điều khiển số của máy CNC.

Chương trình tham số hiện nay cũng được sử dụng một cách rộng rãi hơn và thích hợp hơn trong các chương trình với các phép tính toán về lượng giác, các phép toán về số học và logic.

Hiện nay phần lớn các hệ điều khiển CNC cho khả năng xây dựng các chu trình đặc trưng chứa các chương trình con tham số. Kiểu lập trình tham số có thể

được xây dựng theo nhiều cách khác nhau và có thể bao gồm nhiều từ địa chỉ được sử dụng khác nhau. Các hệ điều khiển số của châu Âu thường sử dụng các biến tham số là địa chỉ E (MAHO), L (NUM), R (SINUMERIC)... và đi sau nó là một số chữ số để xác định các biến. Còn với hệ điều khiển của các nước như Mỹ, Nhật bản thường sử dụng các biến mà trong đó bao gồm các địa chỉ đã được mở rộng từ A đến Z.

Một chương trình *macro* được coi là một chương trình nhỏ có tính chất hoàn chỉnh mà khi được gọi ra thì nó được đưa vào chạy trong chương trình chính. Vì thế với các chương trình *macro* thì thường được thực hiện với các lệnh đã được tiêu chuẩn hoá.

Hiện nay, mỗi hệ điều khiển có chức năng lập trình tham số là khác nhau và các nước cũng như các hãng sản xuất cũng có những tiêu chuẩn khác nhau. Chính điều này sẽ gây trở ngại cho việc nghiên cứu và sử dụng chúng. Một số hệ điều khiển NUM 750T và 760T của Cộng hoà Pháp thường phân chương trình tham số ra làm 2 loại chính với kiểu biến lập trình là L và kiểu E. Còn với hệ điều khiển FANUC thì người ta sử dụng các ký hiệu # để làm các tham số hình thức; Với hệ điều khiển FAGOR thì người ta sử dụng tham số P...

7.2 Tổ chức lập trình

Chương trình tham số không những chỉ được áp dụng với chương trình con mà ngay cả với chương trình chính.

Ví dụ với hệ điều khiển NUM 750T-R và 760 T-TR-R của Pháp thì người ta sử dụng các biến chương trình là L. Nó được sử dụng với 2 kiểu biến L là:

- L0 đến L19.
- L100 đến L199 và L900 đến 939.

Về phạm vi sử dụng và mức độ khai thác của hai loại biến này là giống hệt nhau, tuy nhiên việc soạn thảo với các biến này trong chương trình sẽ dẫn đến việc thực hiện chúng sẽ khác nhau.

* Biến L0 đến L19 thường được đặt vào đầu chương trình trong khi mà chúng ta đặt thiết bị dưới điện áp và cũng được kết thúc với chức năng M02. Các biến này có thể được gán là các giá trị cố định như toạ độ, tốc độ cắt... hoặc cũng có thể là kết quả của các phép tính số học hoặc lượng giác sau: +; -; *; /; $\sqrt{\quad}$; sin; cosin; tg; artg... cũng như các phép tính logic, ví dụ như AND, OR...

Việc sử dụng nó vào trong chương trình không làm ảnh hưởng đến các chức năng khác trong khi soạn thảo chương trình chính và nó có thể được lập trình với 8 chữ số.

* Đối với biến L0 đến L99.

Ví dụ:

L8 =18

XL8 (giá trị tọa độ X lấy giá trị L8 tức là có tọa độ 18mm hoặc là X18).

FL8 (giá trị lượng chạy dao Fm lấy giá trị 18 mm/ph tức là F18).

Các biến của chương trình có thể được thiết lập cho tất cả mọi địa chỉ và với mọi chức năng. Với chức năng G79 nó còn cho phép thực hiện các phép nhảy chương trình có điều kiện.

Ví dụ:

L2=5

L1 =L2+ 5.3*3*S30 (L1=5+5,3*3*sin30°) = 15,45 tức L1= 15.45).

XL1 Z30 (giá trị L1 được đưa vào tọa độ X: tức là X15.45 Z30).

* Biến L100 đến L199 và L900 đến L939.

Dạng và phạm vi sử dụng loại biến này cũng giống như với biến L0 đến L19. Tuy nhiên việc soạn thảo chương trình với các biến này sẽ dẫn đến việc thực hiện khác nhau đối với chương trình gia công. Ví dụ việc đưa vào các biến từ L0 đến L19 thì sự hoạt động của chương trình không ảnh hưởng gì trong khi với các biến L100 đến L199 sẽ đình chỉ chức năng chuẩn bị của *block* có chứa các biến đó cho đến khi kết thúc việc thực hiện *block* sát trước. Một *block* được soạn thảo với biến L100 đến L199 không thể nằm trước một *block* buộc phải thực hiện đã được biết hay của các *block* tiếp sau về các chương trình biên dạng trên 2 hay 3 *block* hoặc chương trình hiệu chỉnh bán kính dao.

Các biến L100 đến L199 có thể được lập trình như biến L0 đến L19 bởi các chương trình của chức năng M99 và không được dùng các thao tác hay tác động tự động vào các chu trình tiếp theo sau các *block*.

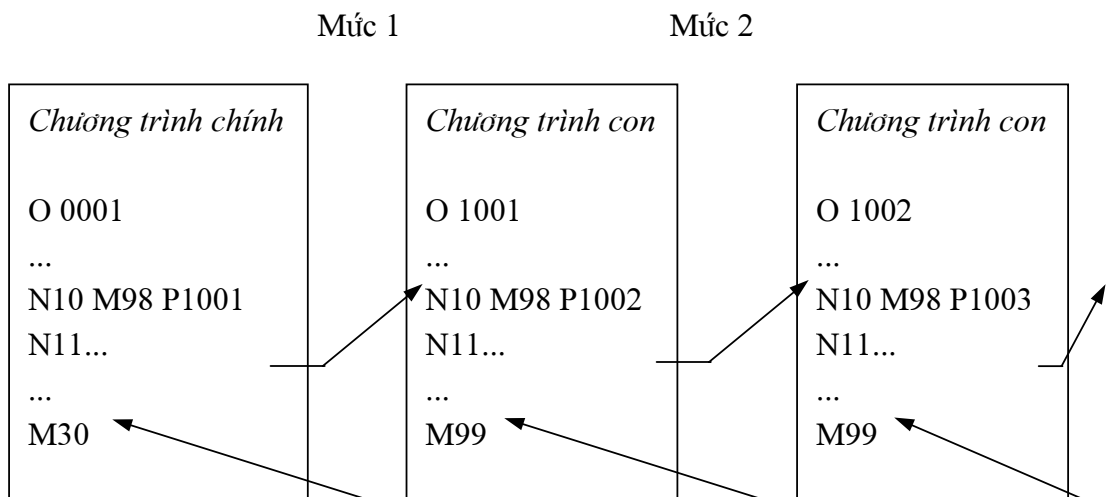
Việc viết các biến này hoặc truyền các biến thông thường trong chương trình gia công không được thực hiện khi lập trình thông thường cho đến khi kết thúc thực hiện các *block* trước đó. Chức năng M98 và M99 cho phép thực hiện trước các thao tác này:

Ví dụ:

%1	%2
....
....
....
N90 X_ Z_	N80 M99
N100 L110 =1+L110	N90 X_ Z_
N110 XL110...	N100 L110=1+L110
N120 XL120=L110+L10.	N110 XL110
N130 ZL120	N120 L120 = L110 +L10
....	N130 ZL120
...	N140 M98
....

Trong chương trình %1, N100 chỉ được chuẩn bị và thực hiện trong khi N90 đã kết thúc. Còn với chương trình %2 sẽ được chuẩn bị để thực hiện trước khi thực hiện N80 và nó sẽ không dừng chương trình khi kết thúc *block* N90.

Kỹ thuật lập trình với chương trình tham số có thể được thực hiện với các chương trình con hoặc chương trình chính và chúng có thể được bố trí với nhiều mức khác nhau mà khi được gọi đến để thực hiện thì nó sẽ được đọc và thực hiện từ mức cao nhất trở về mức thấp nhất của chương trình con hoặc trở về chương trình chính. Ví dụ:



Hình 7-3: Ví dụ về các chương trình chính và các mức chương trình con

Người ta phân biệt thành 3 kiểu biến khác nhau là biến địa phương, biến toàn cục và biến hệ thống.

Các biến địa phương (*Local*) chỉ sử dụng cho chương trình con mà không dùng được trong chương trình chính.

Biến toàn cục (*Global*) là các biến có thể được sử dụng trong cả trong chương trình con hoặc cả trong chương trình chính. Các biến này có thể được gán bằng các giá trị cụ thể như là tọa độ, tốc độ cắt, lượng tiến dao...hoặc có thể là kết quả của một phép tính số học, lượng giác hoặc logic. Số lượng biến của nó có thể ít (100 biến) hoặc nhiều (600 biến) [2] là tùy thuộc vào chức năng của cụm điều khiển CNC mà người đặt hàng yêu cầu.

Biến hệ thống (*System*) chính là các số ghi để dành cho các định dạng của máy hoặc của bộ ghi mà chứa các thông tin về giá trị xác định tọa độ vị trí các điểm gốc, điểm chuẩn trong khi chương trình hoạt động. Nó phụ thuộc vào nhà chế tạo máy quy định và khi có sự cố hoặc thật cần thiết thì phải kiểm tra ngay từ các chương trình nguồn của nó.

Giá trị các biến địa phương được truyền vào trong chương trình con với mục đích thực hiện các phép tính toán số học, lượng giác và các phép tính logic.

Bảng 7-1: Giá trị các biến địa phương và các Argument tương ứng theo các dạng I và II.

Kiểu I				Kiểu II:					
A	#1	Q	#17	A	#1	K3	#12	J7	#23
B	#2	R	#18	B	#2	I4	#13	K7	#24
C	#3	S	#19	C	#3	J4	#14	...	
D	#7	T	#20	I1	#4	K4	#15		
E	#8	U	#21	J1	#5	I5	#16		
F	#9	V	#22	K1	#6	J5	#17		
H	#11	W	#23	I2	#7	K5	#18		
I	#4	X	#24	J2	#8	I6	#19		
J	#5	Y	#25	K2	#9	J6	#20		
K	#6	Z	#26	I3	#10	K6	#21		
M	#13			J3	#11	I7	#22		

Các phép tính toán trên cơ sở các biến toàn cục được thực hiện ngay trong các *block* của chương trình chính hay của chương trình con. Sau đây trình bày một số các phép tính toán trong khi sử dụng các chương trình tham số.

Bảng 7-2: Các phép tính toán được sử dụng khi soạn thảo trong chương trình tham số.

Ký hiệu	Ý nghĩa	Ví dụ
()	Ngoặc đơn	Các lỗ trên vòng tròn
[]	Thực hiện tính toán ưu tiên	#100 = #4*2 - [#1]
#i =#j	Gán, thay thế	#100 = 1
#i =#j + #k	Cộng	#100 = #4 + #3
#i = #j - #k	Trừ	#100= #4 - #3
#i =#j * #k	Nhân	#100 = #4 * #3
#i = #j / #k	Chia	#100 = #4 / #3
#i = SIN [#j]	Sin	#100 = SIN [#4]
#i = COS [#j]	Cosin	#100 = COS [#4]
#i =TAN [#k]	Tang	#100 = TAN [#4]
#i =ASIN [#j]/[#k]	Arsin	#100 =ASIN [#4]/[#3]
#i = ACOS [#j]/[#k]	Arcosin	#100 ACOS [#4]/[#3]
#i = ATAN [#j]/[#k]	Artang	#100 =ATAN [#4]/[#3]
#i = SQRT [#j]	Căn bậc hai	#100 = SQRT [25]
#i = ABS [#j]	Giá trị tuyệt đối	#100 =ABS [#4]
#i = BIN [#j]	Biểu diễn trong hệ nhị phân (Binary)	#100 = BIN [#4]
#i = BCD [#j]	Biểu diễn trong hệ nhị thập phân (Binary Codel Decimal)	#4 = BCD [#100]
#i = ROUND [#j]	Làm tròn số	#100 = ROUND [#100]

Với các lệnh logic và các lệnh có cấu trúc thực hiện các lệnh tương tự như khi lập trình trong máy tính: Lệnh nhảy không điều kiện GOTO ...; các lệnh thực hiện có điều kiện IF... THEN; lệnh nhảy có điều kiện IF ... GOTO và vòng lặp WHILE... DO.

Bảng 7-3: Biểu thức các lệnh điều khiển logic

Biểu thức	Ý nghĩa	Ví dụ
#i EQ#k	Bằng nhau (<i>Equal</i>)	IF[#100 EQ #4] THEN
#i NE #k	Không bằng, khác (<i>Not Equal</i>)	IF[#100 NE #4] THEN
#i GT #k	Lớn hơn (<i>Greater Than</i>)	IF[#100 GT #4] THEN
#i LT #k	Nhỏ hơn (<i>Less Than</i>)	IF[#100 LT #4] THEN
#i GE #k	Lớn hơn hoặc bằng (<i>Greater then or Equal</i>)	IF[#100 GE #4] THEN
#i LE #k	Nhỏ hơn hoặc bằng (<i>Less then or Equal</i>)	IF[#100 LE #4] THEN

Các lệnh logic tiêu chuẩn hoá được dùng với các biểu thức thông thường của bộ điều khiển: Tương đương (EQ); khác (NE); lớn hơn (GT); bé hơn (LT) ; Lớn hơn hoặc bằng (GE); nhỏ hơn hoặc bằng (LE) và ứng dụng vào chương trình tham số như biểu diễn ở bảng trên.

Lệnh nhảy trực tiếp không điều kiện vào *block* n là GOTOn

Ví dụ:

GOTO40. {nhảy vào block N40}

Lệnh nhảy có điều kiện: IF... GOTO hoặc lệnh thực hiện có điều kiện: IF ... THEN.

Ví dụ:

IF [#100 EQ 0] GOTO40. {Xác định rằng nếu biến #100 = 0 thì hệ điều khiển sẽ nhảy vào thực hiện với block N40). Nếu không thì đầu đọc sẽ thực hiện một cách bình thường và vào ngay block tiếp theo sát ngay nó bao gồm từ IF}.

IF [#100 GT 0] THEN [#100 = #100 -1]. { Nếu biến #100 là lớn hơn 0 thì biến #100 được gán 1 giá trị mới bằng việc giảm đi 1 đơn vị #100 = #100 -1, cho đến khi giá trị của biến #100 giảm xuống bằng 0 thì kết thúc quá trình thực hiện}.

Lệnh cuối cùng là trình bày một vòng lặp của *block*. Sự lặp này được thực hiện trên các *block* nằm giữa các biểu thức WHILE và *block* thứ m.

Ví dụ: WHILE [#100 = ATAN [#5]/[#7]] DOm

...

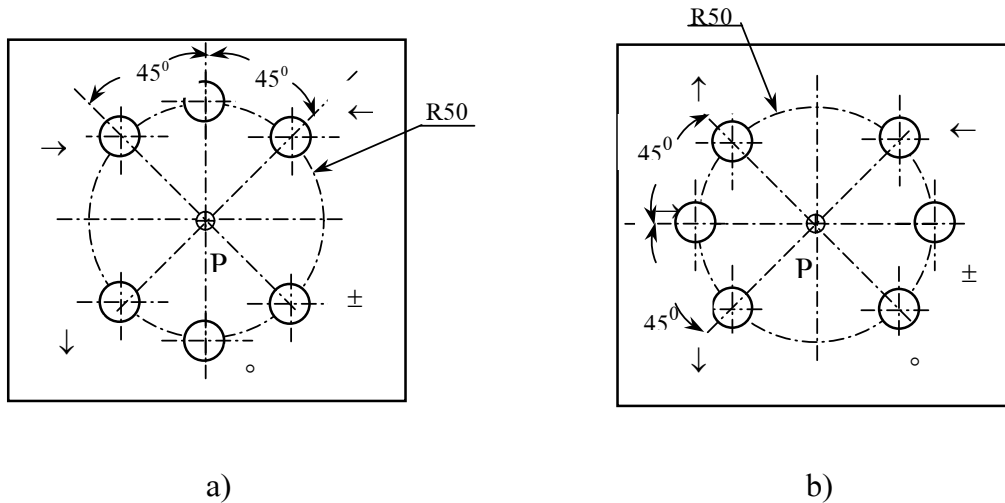
ENDm

{Xác định rằng nếu biến #100 bằng argt của biến #5 chia cho biến #7 thì nó sẽ thực hiện vòng lặp từ block bắt đầu có chức năng UHILE đến số block thứ m , ở đây block thứ m là block cuối cùng của đoạn chương trình END}.

7.4 Một số ví dụ về lập trình bằng chương trình tham số

Ví dụ 1:

Cho chi tiết gia công như hình vẽ



Hình 7-4: Bản vẽ chi tiết gia công

Chương trình gia công đối với chi tiết kiểu *b* trong hệ điều khiển FANUC

...
...

N5 T01 M06

N6 G16 S1500 M03

{Chương trình gia công trong hệ tọa độ cực}

N7 G90 G00 X50 Y45

{Lập trình trong hệ tọa độ tuyệt đối, chạy dao nhanh đến lỗ 1}

N8 G43 Z3 H01 M08

N9 G99 G81 Z-2.5 R3.0 F250

{Chu trình khoan, chiều sâu của lỗ -2.5, lượng ăn dao 250mm/ph}

/ N10 M99 P002

{Nhảy vào block N12 (P0012) để gia công lỗ #2}

N11 Y90

{Không thực hiện}

► N12 Y135

{Gia công lỗ #2}

N13 Y180

{Gia công lỗ #3}

N14 Y225

{Gia công lỗ #4}

<pre> / N15 M99 P0017 N16 Y270 N17 Y315 N18 Y360 N19 G15 G80 G00 </pre>	<pre> {Nhảy vào block N17 (P0017), gia công lỗ #5} { Không thực hiện} { Gia công lỗ #5} { Gia công lỗ #6} </pre>
---	--

...

Khi gia công chi tiết kiểu *a*, chỉ cần thay thế block N10 bằng câu lệnh:

```
/ N10 M99 P0011
```

Và tại block N13 được viết lại:

```
/ N13 M99 P0015.
```

Lúc này block N15 là :

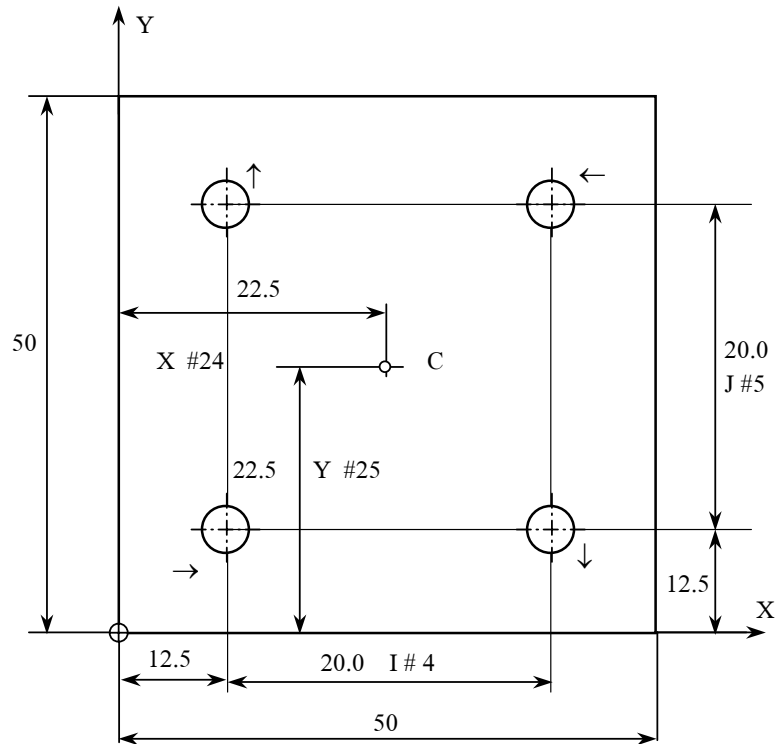
```
N15 Y225
```

Trong khi block N14 là: N14 Y180.v.v..

Như vậy có thể thấy rằng với việc lập chương trình tham số có thể thực hiện việc gia công các chi tiết có kiểu gần giống nhau chỉ bằng việc thay đổi một số câu lệnh tối thiểu.

Ví dụ 2:

Cho chi tiết gia công như hình vẽ



Hình 7-5: Bản vẽ chi tiết gia công

Chương trình gia công trong hệ điều khiển FANUC:

```
%  
O0001  
N1 M98 P9998 {Bắt đầu chương trình}  
N2 T01 M06 {Thay dao T01}  
N3 G90 G00 X12.7 Y12.7 S3000 M03 {Chương trình trong hệ tọa  
độ tuyệt đối; tốc độ cắt S và lượng  
chạy dao F; Chạy dao nhanh đến vị  
trí tọa độ X12,7 và Y12,7}  
N4 G43 Z25.0 H01 M08 {Chọn mặt phẳng tham chiếu; mở dụng  
dịch trôn nguội }  
N5 G99 G81 Z1.5 R3.0 L0 F20.0 {Chu trình khoan lỗ theo biến L0}  
N6 G65 P1001 X22.5 Y22.5 I20.0 J20.0 {Gọi chương trình con P1001, Tọa  
độ tâm C X22.5 Y22.5; Khoảng  
cách các lỗ I20 (theo phương X) và  
J20 (theo phương Y)}  
N7 M98 P9999 {Kết thúc chương trình}  
N8 M30  
O1001 {Chương trình con 1001}  
(I#4 (#4=20) {Khoảng cách giữa các lỗ theo phương  
trục X}  
(J#5 (#5=20.5) {Khoảng cách giữa các lỗ theo phương  
trục Y}  
  
(X#24 (#24=22.5) {Vị trí của tâm C theo phương trục X}  
(Y#25 (#25=22.5) {Vị trí của tâm C theo phương trục Y}  
N1 # 101 = # 4/20 = 20/2 = 10 {Khoảng cách các lỗ đến tâm C theo  
phương trục X}  
N2 # 102 = #5/20 = 20/2 = 10 {Khoảng cách các lỗ đến tâm C theo  
phương trục X}  
  
N3 X[#24+#101] Y[#25+#102] {Thực hiện tính toán các vị trí tâm lỗ, Tính  
tọa độ tâm lỗ 1 X( 22,5 +10) Y(22,5+10)}  
N4 X[#24-#101] Y[#25+#102] {Tính tọa độ tâm lỗ 2 X( 22,5 -10)  
Y(22,5+10)}  
  
N5 X[#24-#101] Y[#25-#102] {Tính tọa độ tâm lỗ 3 X( 22,5 -10)  
Y(22,5-10)}
```

N6 X[#24+#101] Y[#25-#102]

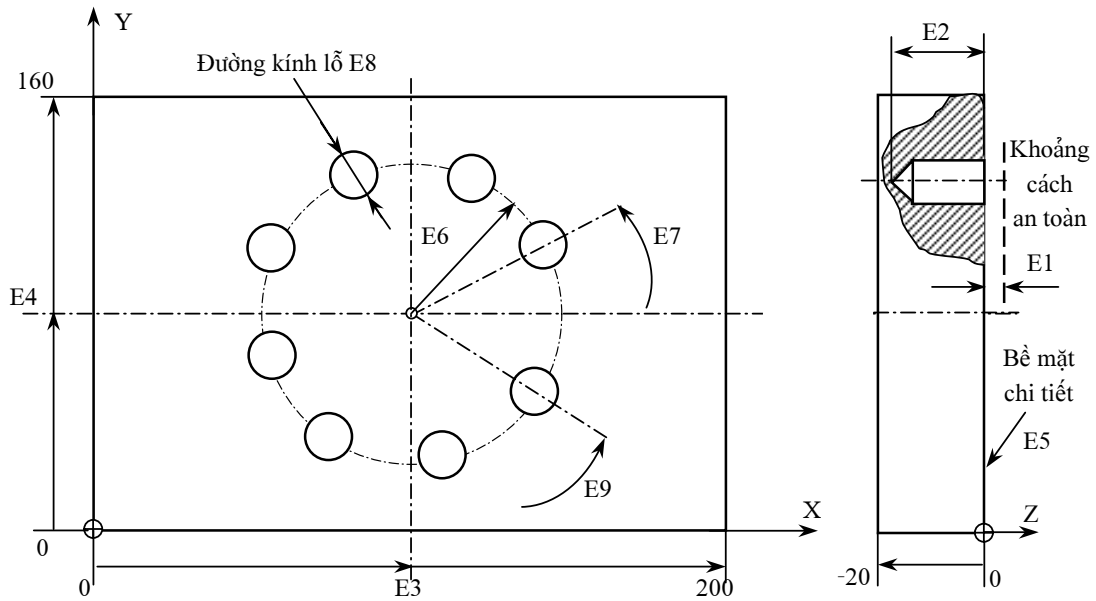
{*Tính tọa độ tâm lỗ 4 X(22,5 +10)
Y(22,5-10)*}

N7 M99

%

Ví dụ 3:

Cho bản vẽ chi tiết như hình vẽ



Hình 7-6: Bản vẽ chi tiết gia công

Chương trình tham số viết trong hệ điều khiển MAHO

% PM

N 9011

{ Số chương trình }

N1 G17 S800 T1 M06

{ Mặt phẳng nội suy XY; tốc độ 800 v/p;
thay dao tự động, dao số 01 có $\phi = E8$ }

N2 G54

{ Chọn số hiệu zero offset }

N3 G98 X-10 Y-10 Z-20 I220 J180 K30

{ Chu trình gia công }

N4 G99 X0 Y0 Z-20 I200 J160 K20

{ Chu trình gia công }

N5

E1 = 2

{ Gán giá trị các biến }

E2 = 15

{ Gán giá trị các biến }

E3 = 100

{ Gán giá trị các biến }

E4 = 80	{ Gán giá trị các biến}
E5 = 0	{ Gán giá trị các biến}
E6 = 60	{ Gán giá trị các biến}
E7 = 20.5	{ Gán giá trị các biến}
E8 = 8	{ Gán giá trị các biến}
E9 = 316	{ Gán giá trị các biến}
E10 = 50	{ Gán giá trị các biến}
N6 G81 Y = E1 Z =- E2 F = E10 M03	{ Gọi các biến vào trong chương trình}
N7 G77 X= E3 Y = E4 Z = E5 R= E6 I= E7 J= E8 K=E9	{ Gọi các biến vào trong chương trình }
N8 G0 Z50	{ Chạy nhanh về Z50}
N9 M30	{ Kết thúc}