

MC機による不規則な加工プログラムの作成

ポリテクカレッジ新潟 渡辺 寛厚
(新潟職業能力開発短期大学校)

1. はじめに

三条木製品協同組合は、会員企業数76社で、従業員数470名のそれほど大きな団体ではないが、元工具に使用されている木製部分を製造していた企業の集まりで、建築施工の変化や木製部分がプラスチック等の新素材に代用されてきたことなどから、現在では、今までの伝統的な木材加工技術を生かした商品開発が行われ、贈答品や装飾品、学校関係の工作用キット等を製造している。

今回、協同組合が加工方法について検討してきた中で、MC機を使って不規則な加工ができないものかという依頼が当校にあった。

三条木製品協同組合では、この不規則な加工方法を応用して、小物類、家具類等の表面に彫刻刀跡の加工を施した物や模様や色・形のデザイン等に意外性のある新しい商品開発に期待していた。

最近の消費者ニーズは、個人志向化し、多種多様化してきていることと、加工方法も高速化傾向にあるため、将来、消費者のニーズに対応した街工房的な産業に発展する可能性も考えている。

しかし、マシニングセンタは、プログラムの指示どおり加工する精密工作機械であるため、正確に移動するための機能や機構は装備されているが、不規則に変化する機能は皆無である。

そこで、マシニングセンタで使用されているカスタムマクロの「システム変数」を応用して、加工直前の時間を読み取り、その時間の変化を利用して不

規則な数字を作り出し、それを加工に取り入れる研究を開始した。

2. 乱数の作成

2.1 時間の変化を読み取る

システム変数のクロック1は、「#3001」で1ミリ秒単位で常時カウントしている（クロックの精度は、16ミリ秒で、65536ミリ秒でオーバーフローし、ゼロに戻る）。

例 64172

加工直前の時間を読み取り、そこから1つの数を取り出す方法を考える。

読み取った時間の1桁目の数を利用する。

(1) 加工直前の読み取った時間が「64172」であった場合10で割る。

$$64172 \div 10 = 6417.2$$

(2) 小数点以下を切り捨てる。

$$6417$$

(3) その数を10倍する。

$$64170$$

(4) 最初に取り出した数から引く。

$$64172 - 64170 = 2$$

ここでは、「2」という数を取り出された。

この方法で行うと、取り出される数は、「0」から「9」までとなる。

この数をそのまま利用すると、「0」から「-9」mmまでとなり大きすぎる場合があるので、加工する目的に応じて、その数を修正させる必要がある。

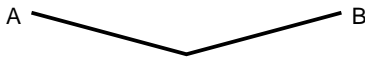
2.2 表札の加工に活用

この不規則な数をZ軸に取り入れ、刃物の深さをランダムに動かし、実際に表札（材料：朴）を加工しながら現実的な問題点を拾い出し、実用的なプログラムを完成させることにする。

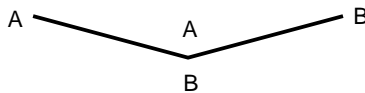
表札は、文字の周りを彫刻刀で彫ったような、手作的な味わいを持たせるようにする。

彫刻刀で実際に加工する場合、刃物を材料に差し込み押し上げるという動作を一動作として繰り返しているが、ここではプログラムを簡単にするため、刃物を差し込んで一動作、押し上げて一動作とすることにした。

彫刻刀の動きは、AからBに、



ABで1つの動作とする。



次の点を考慮してZ軸の調整を行うこととする。

Z軸方向の上下動差は、「-2」mm以内とする。

Z軸は、「-2」mmを基準にして上下動させる。

始点終点間のZ軸方向の差は、最小「0.2」最大「1.6」とする。

このような条件を盛り込み、取り出した数を変形する。

(1) 「0」から「0.8」までの数を作る

「0」から「9」までの取り出した数に「0.2」を掛け、「0」から「0.8」の数にする。

$$0 \times 0.2 = 0.0$$

$$1 \times 0.2 = 0.2$$

$$2 \times 0.2 = 0.4$$

$$3 \times 0.2 = 0.6$$

$$4 \times 0.2 = 0.8$$

$$5 \times 0.2 = 1.0$$

5以上になると「1」を越えるので「5」からは、その数から「5」を引いてから「0.2」を掛ける。

$$(5 - 5) \times 0.2 = 0.0$$

$$(6 - 5) \times 0.2 = 0.2$$

$$(7 - 5) \times 0.2 = 0.4$$

$$(8 - 5) \times 0.2 = 0.6$$

$$(9 - 5) \times 0.2 = 0.8$$

(2) Z軸を「2」mm切り込んで置き、そこを基準にして「-1.2」から「-2.8」まで変化するようにする

取り出した数が「0」から「4」までのときは、基準値「-2」に加える。

$$-2 + 0.0 = -2.0$$

$$-2 + 0.2 = -1.8$$

$$-2 + 0.4 = -1.6$$

$$-2 + 0.6 = -1.4$$

$$-2 + 0.8 = -1.2$$

取り出した数が「5」から「9」までのときは、基準値「-2」から引く。

$$-2 - 0.0 = -2.0$$

$$-2 - 0.2 = -2.2$$

$$-2 - 0.4 = -2.4$$

$$-2 - 0.6 = -2.6$$

$$-2 - 0.8 = -2.8$$

これにより、「-1.2」mmから「0.2」刻みで「-2.8」mmまでの加工が可能となる。

(3) マクロプログラムの作成

以上のことを考慮して、マクロでプログラムを作成する。

#29=0.2 Z軸の切り込みを調整

#18=-2.0 Z軸の基準切り込み値

#32=#3001

加工直前の時間を読み取る。

#30=FIX[#32/10]*10

#32を10で割って下1桁目を切り捨て、10倍する。

#28=#32-#30

その差を計算する（0～9までの数字が出てくる）

#110=#28*#29

0～9までの数字に0.2を掛ける。

IF[#28GT5]GOTO2
 #28が5以上のときはN2へ移動。
 #111=#18+#110
 基準 - 2より浅くなるように足す。
 G#1X#16Y#25Z#111F#9
 GOTO3
 N2
 #112=#28-5
 #28から5を引き, 5以上にならないようにする。
 #113=#112*#29
 さらにそこに0.2を掛け, 1以上にならないようにする。
 #114=#18-#113
 基準 - 2より深くなるように引く。
 G#1X#16Y#25Z#114F#9
 N3

2.3 NCプログラム作成方法

- (1) 工具は, 荒加工用 6のボールエンドミルを使用する。
- (2) 荒加工の区域切削は, ジグザグでステップ量を3mmとする。
- (3) Z軸の切り込み量は, ゼロでNCデータを作成する。
- (4) マクロプログラムは, サブプロとして作成する。
- (5) CAMで作成したNCプログラムに, マクロプログラムを呼び出すG66.1を使用する。
 これにより, 移動指令のあるすべてのブロックで呼び出すことになり, 次の箇所の問題が発生する。

CAMで作成されたNCデータ

G59G90G00X96.521Y-58.024S6000T02
 G43Z100.H21
 Z20.M03
 G01Z2.0F1000
 Z-1.0F20
 G66.1P7023X106.F20
 ここからマクロプログラムが開始する
 Y-61.

X-106.
 Y-58.024
 ステップのための移動 2.4(1)
 X106.
 ・
 ・
 G02G17X6.681Y-31.51-2.783J8.074
 円弧補間 2.4(2)
 X5.974Y-31.244I0.661J2.926
 G01X-6.19
 G02X-9.096I-1.453J9.372
 G01X-106.
 ・
 ・
 G00Z2.54
 早送りのブロック 2.4(3)
 X-85.207Y-25.293
 早送りのブロック 2.4(3)
 G01Z0.
 Z軸だけの移動 2.4(4)
 X-87.747F20
 G03X-84.902Y-22.317I-20.565J22.508
 ・
 ・
 G01X-1.217
 G02X-7.587Y19.181I2.299J13.836
 G01X-10.377
 G67
 マクロの呼び出しをキャンセル
 G00Z100.M05
 G91G28Z0
 G28X0Y0M19
 M06
 M30

2.4 NCプログラムの問題点(横型の表札の場合)

サブプロとして作成したマクロプログラムを, CAMで作成したNCプログラムが実行するたびに呼び込む場合, 上記のブロックでは, Z軸の不規則な

動きを実行させないようにしなければならない。

(1) ステップのための移動のブロック

```
IF[#24EQ#0]GOTO20
```

X値がないときは、ステップ移動なので通過させる（横方向に移動するジグザグ加工の場合）

```
N20
```

```
G01Y#25F#9
```

(2) 円弧補間のあるブロック

マクロプログラムは、直進するプログラムに対し、Z軸を上下させ加工するプログラムのため、円弧補間のカッターパスは通過させなければならない。

```
#1=#4001
```

#4001は、グループ「1」のモダル情報（先読みブロック）で、G01、G02、G03の1、2、3の数を読み取る。

```
IF[#1GT1]GOTO10
```

もし、#1が2、3であったら、N10へ移動させる。

```
N10
```

```
G#1X#120Y#25I#4J#5F#9
```

G02、03のときは、プログラムどおり動かし、メインに戻る。

```
GOTO30
```

```
N30
```

```
M99
```

(3) G00のブロックは、加工箇所移動のためなのでN10に飛ばす

```
IF[#1GT0]GOTO10
```

G00のときN10に飛ばす。

```
N10
```

```
G#1X#120Y#25Z#26I#4J#5F#9
```

プログラムどおり移動させる。

(4) Z軸がアプローチや逃げで、移動するブロック

このときも、そのまま通過させる。

```
IF[#24+#25]EQ#0]GOTO25
```

X、Yに値がないときN25へ移動させる。

```
N25
```

```
G01Z#26F10
```

Z軸だけの移動なので、プログラムどおり移動させる。

2.5 X方向、Y方向に移動する距離を20mm間隔とする（表札が横型のため）

X方向に移動する距離を20mm間隔とした場合、プログラムのXの指令値を読み込み20mm刻みで距離を計算し移動させる必要がある。

移動量が20mmより小さいときは、N10に飛ばす。

また、移動方向が負の方向に動く場合は、N4に飛ばし、プラス方向と同じように移動させる。

```
#120=#24
```

Xの指令値を入れる。

```
#15=#5001
```

ブロック直前のX軸値を読み込む。

```
#16=#15
```

ブロック直前の値を変化させるので#16に入れる。

```
#6=#16-#120
```

移動する距離の計算をする。

```
IF[ABS[#6]LE20]GOTO10
```

移動量が20より小さいときは、N10に飛ばす。

```
IF[#120LT#16]GOTO4
```

カッターがXの負の方向に動く場合は、N4に飛ばす。

```
#16=#16+20
```

```
WHILE[#120GE#16]DO1
```

```
#32=#3001
```

```
#30=FIX[#32/10]*10
```

ここは、カッターがX軸の正の方向に移動するとき。

```
#28=#32-#30
```

```
#110=#28*#29
```

```
IF[#28GT5]GOTO2
```

```
#111=#26+#110
```

```
G#1X#16Y#25Z#111F#9
```

```

GOTO3
N2
#112=#28-5
#113=#112*#29
#114=#26-#113
G#1X#16Y#25Z#114F#9
N3
#16=#16+20
    1 つ動くたびに20mmずつ増やす。
END1
GOTO10
N4
#16=#16-20
WHILE[#120LE#16]DO2
#32=#3001
#30=FIX[#32/10]*10
#28=#32-#30
#110=#28*#29
    ここは、カッターがX軸の負の方向に移動するとき。
IF[#28GT5]GOTO5
#111=#26+#110
G#1X#16Y#25Z#111F#9
GOTO6
N5
#112=#28-5
#113=#112*#29
#114=#26-#113
G#1X#16Y#25Z#114F#9
N6
#16=#16-20
END2
N10

```

2.6 横型表札用完成プログラム

```

%
O7023
N101
#1=#4001
#120=#24

```

```

IF[#1EQ0]GOTO10
IF[#24+#25]EQ#0]GOTO25
#29=0.2
#18=-2.0
IF[#24EQ#0]GOTO20
#15=#5001
#16=#15
IF[#1GT1]GOTO10
#6=#16-#120
IF[ABS[#6]LE20]GOTO10
IF[#120LT#16]GOTO4
#16=#16+20
WHILE[#120GE#16]DO1
#32=#3001
#30=FIX[#32/10]*10
#28=#32-#30
#110=#28*#29
IF[#28GT5]GOTO2
#111=#18+#110
G#1X#16Y#25Z#111F#9
GOTO3
N2
#112=#28-5
#113=#112*#29
#114=#18-#113
G#1X#16Y#25Z#114F#9
N3
#16=#16+20
END1
GOTO10
N4
#16=#16-20
WHILE[#120LE#16]DO2
#32=#3001
#30=FIX[#32/10]*10
#28=#32-#30
#110=#28*#29
IF[#28GT5]GOTO5
#111=#18+#110
G#1X#16Y#25Z#111F#9

```



不規則な加工方法による表札

```
GOTO6
N5
#112=#28-5
#113=#112*#29
#114=#18-#113
G#1X#16Y#25Z#114F#9
N6
#16=#16-20
END2
N10
G#1X#120Y#25Z#26I#4J#5F#9
GOTO30
N20
G01Y#25F#9
GOTO30
N25
G01Z#26F10
N30
M99
%
```

2.7 仕上げ加工

表札の文字の周りを 6 のボールエンドミルでジグザグ加工のみだと、文字がハッキリ浮き出ないため、1 フラットエンドミルで文字の周囲 4 mm 間を Z - 2.0mm 切り込んで、区域切削を行う。ステッ

プ量は、0.5 とする。

さらに、荒加工と文字周囲加工の境のバリを除去するため、もう一度 6 のボールエンドミルで輪郭加工を行う。

3. おわりに

カスタムマクロを使って「0」から「9」までの数を、ランダムに取り出すことが可能となり、表札の作成を通して実用的なプログラム開発に取り組んできた。

表札加工用 NC データは、文字を、スキャナーで CAD データとして読み込み、CAM を使ってジグザグ加工の区域切削データを作成し、カスタムマクロで 20mm 移動するごとに Z 軸をランダムに上下させ、彫刻刀で彫った感じを表現させることができた。

製品によってプログラムを改良しなければならないが、マシニングセンタでランダムな加工が可能となったことで、表札以外の製品においても、不規則に切り込む深さによって作り出す模様や色の変化・形状により意外性のある商品開発が可能となり、今後新しい商品に取り入れられていくことを大いに期待している。

今回は、平面のみの加工であるが、今後立体形状にも応用できるプログラム開発に発展させていきたいと考えている。