

Python プログラミングによる NC 加工支援

大本 憲一

岐阜大学工学部ものづくり技術教育支援センター

NC 加工支援とは, CAD/CAM で設計・デザインと加工経路から G コード生成して, マシニングセンタ等の NC 加工機による加工を, 利便性・効率化・安全対策等を多種多様な方法で支援をするものである. 今回 1 つの方法として, プログラム言語 Python のプログラムによって NC 加工支援を試み, 一定の成果を得られたので以下に報告する.

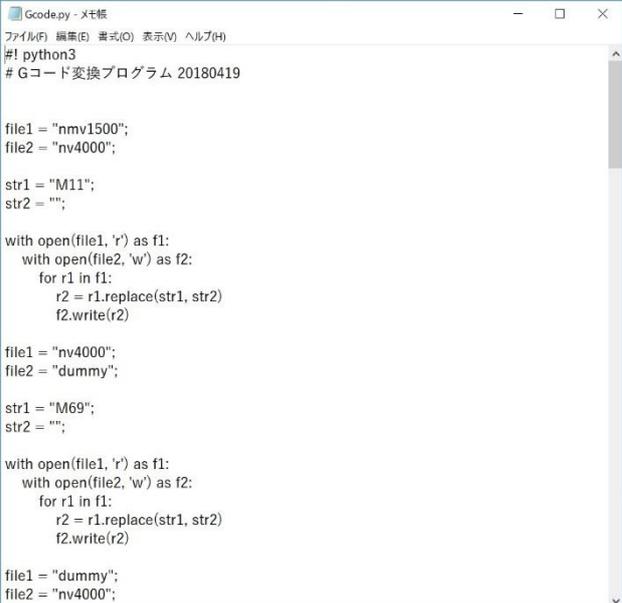
Key Words : Python

1. はじめに

Python プログラムによって, 3 種類の支援方法を考えた. はじめに, 「Python プログラミングI」として追加ポストプロセッサの機能を, メモ帳の機能でよく利用する置換を応用したプログラム. 2 つ目に, 「Python プログラミングII」として CAD/CAM のポストプロセッサを使用しないで, 対話式でドリル穴あけの G コードを生成するプログラム. 最後に 「Python プログラミングIII」として切削工具の必須加工条件を抽出する対話式のプログラムを作成した.

2. Python プログラミングI

ハイエンド CAD/CAM 「hyperMILL」は残念ながら 5 軸用マシニングセンタ (DMGMORI NMV1500DCG) のポストプロセッサ単独である. 高額な 3 軸用マシニングセンタ (DMGMORI NV4000) のポストプロセッサ購入の代わりに, 5 軸用 G コードを置換機能を応用した Python プログラムを作成し, 3 軸用 G コードに変換した. 抜粋した Python コードを図-1 に示す.



```
Gcode.py - 文字帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#! python3
# Gコード変換プログラム 20180419

file1 = "nmv1500";
file2 = "nv4000";

str1 = "M11";
str2 = "";

with open(file1, 'r') as f1:
    with open(file2, 'w') as f2:
        for r1 in f1:
            r2 = r1.replace(str1, str2)
            f2.write(r2)

file1 = "nv4000";
file2 = "dummy";

str1 = "M69";
str2 = "";

with open(file1, 'r') as f1:
    with open(file2, 'w') as f2:
        for r1 in f1:
            r2 = r1.replace(str1, str2)
            f2.write(r2)

file1 = "dummy";
file2 = "nv4000";
```

図-1 抜粋した G コードに変換 Python コード

3. Python プログラミングII

簡易的なドリルの穴あけに特化した G コードであれば, CAD/CAM 不要の G コード生成プログラムを Python プログラミングで作成可能と考え作成した. 抜粋した Python コードを図-2 に示す.

```

drill_5axis.py - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
dd = float(input('ドリル径<D>?'))
dsk = 10

if (dd < 2.0) or (dd > 12.0):
    exit()

elif (dd >= 2.0) and (dd <= 3.0):
    dn = 4600 #SGRドリル回転数
    df = 200 #ドリル送り
    dq=(dd / dsk) #ドリルステップ量
elif (dd >= 3.1) and (dd <= 4.0):
    dn = 3400
    df = 260
    dq=(dd / dsk)
elif (dd >= 4.1) and (dd <= 5.0):
    dn = 2700
    df = 250
    dq=(dd / dsk)
elif (dd >= 5.1) and (dd <= 6.0):
    dn = 2000
    df = 240
    dq=(dd / dsk)
elif (dd >= 6.1) and (dd <= 7.0):
    dn = 1767
    df = 230
    dq=(dd / dsk)
elif (dd >= 7.1) and (dd <= 8.0):
    dn = 1533
    df = 220
    dq=(dd / dsk)

```

図-2 抜粋した簡易的なドリルの穴あけ Python コード

4. Python プログラミングIII

従来は、切削工具のカタログからアナログ的に必須加工条件を調べて CAM に適応していたが、利便性の向上から使用頻度の高い切削工具を Python プログラミングで作成可能考え、対話式の必須加工条件抽出プログラムを作成した。抜粋した Python コードを図-3 に示す。Python の操作画面を図-4 に示す。

```

AE-VMS.py - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
ed = int(input('3, 4, 5, 6, 8, 10, 12 <AE-VMS ENDMILL直径>?'))

if (ed < 3) or (ed > 12) :
    exit()

cut = int(input('1:側面切削 2:溝切削 <切削タイプ>?'))
stock = int(input('1:SS400/S50C 2:SCM/SKD 3:NAK80/PX5 4:SUS <被削材番号>?'))

if cut == 1: # 側面切削
    if stock == 1: # SS400/S50C
        if ed == 3:
            en = 13800
            ef = 1660
            ap = ed
            ae = ed*0.1
            ae = round(ae,1)
            l1 = 8
        elif ed == 4:
            en = 10400
            ef = 1830
            ap = ed
            ae = ed*0.1
            ae = round(ae,1)
            l1 = 11
        elif ed == 5:
            en = 8300
            ef = 1990
            ap = ed
            ae = ed*0.1
            ae = round(ae,1)

```

図-3 抜粋した対話式加工条件抽出 Python コード

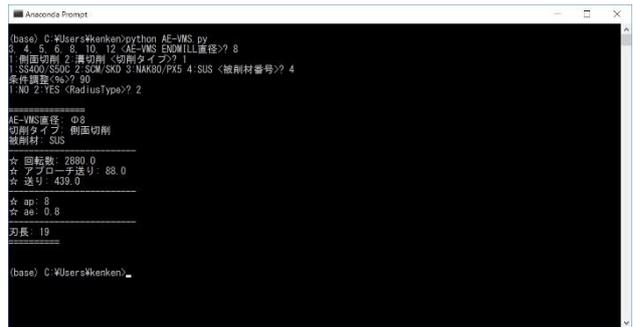


図-4 Python の操作画面

5. まとめ

Python プログラミングIIは、検証を済ませ実際に金型実習等で利便性と安全性（プログラムはミスをしな）に貢献した。Python プログラミングIIIは、実際の作業ではあまり支援効果はなかったが、Python プログラミングのスキルアップに効果があった。Python プログラミングIIIは、金型実習等で使用頻度が高く、使用者の評判も上々であった。以上から NC 加工支援は一定の成果が得られたといえる。