

切削加工条件検索システムの構築

大本 憲一・宮川 翔太・生田 智大

岐阜大学工学部ものづくり技術教育支援センター

金型実習（射出班・プレス班）における金型設計製作時間は限られている．金型設計製作時間の短縮の中で，金型製作時間の短縮は必須である．金型製作時間に大きく関わる要素として「マシニングセンタの操作」「ワークの段取り」「CAM の操作」「工具メーカーのカatalogから切削加工条件を導き出す」がある．この中で，一番安全面を犠牲にせず効率化による時間短縮を行えるのは「工具メーカーのカatalogから切削加工条件を導き出す」であると考える．また，利便性の向上も考慮し，プログラム言語 Python を駆使して切削加工条件検索のシステム構築を行なった．以下に報告する．

Key Words： 金型製作時間の短縮, 利便性の向上, Python

1. はじめに

金型製作時間の短縮と利便性の向上を可能にする方法として，プログラム言語 Python を駆使して実現する．

2. Python プログラミング

切削加工条件検索システムのプログラミングの考え方は，必須入力項目と変数型の決定・条件分岐と計算式・加工条件調整・追加条件・加工条件検索結果の表示である．

2.1 必須入力項目と変数型の決定

必須入力項目として，工具（ドリル・タップ・リーマ・エンドミル）の直径を整数型又は浮動小数点数型，被削材・エンドミル切削タイプを整数型，タップのピッチ・ヘリカル穴加工径を浮動小数点数型の変数に設定した．図-1 に抜粋したプログラム先頭の Python コードを示す．



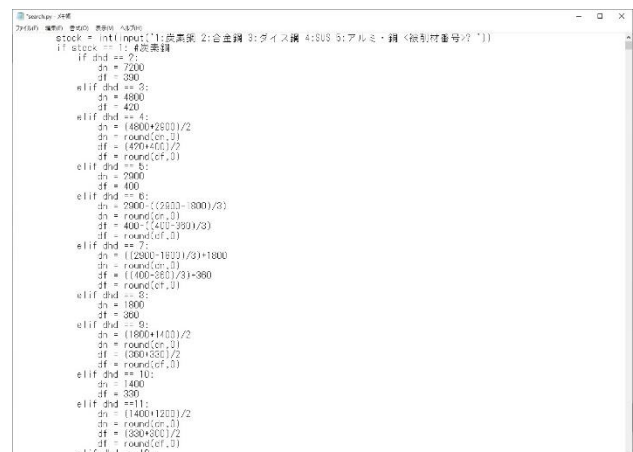
```
pfile = 'intinput(1:ドリル 2:タップ 3:リーマ 4:ヘリカルス 5:エンドミル 6:WireCut <切削タイプ?> '))

if cfile == 1:
    cfile = int(input(1:AGES 2:VP+GDR 3:ADDEXR 4:ADDEXZ 5:ADF 6:SDSR 7:ZPB 8:NEXUS+GDR 9:AGBUSR <ドリル種類?> '))
    if dfile == 1: #NACHI AS-2 URILL 切削条件 20180907
        dnd = float(input(4:FS ドリル 標準寸法-122? '))
```

図-1 プログラム先頭の Python コード

2.2 条件分岐と計算式

条件分岐の基本構造は，if 文と elif をネストした制御文に，シンプルな計算式を導き出して if・elif の条件に融合した．図-2 に抜粋したドリル（AGES）の条件分岐と計算式の Python コードを示す．



```
stock = int(input(1:品番表 2:金型鋼 3:ダイス鋼 4:SUS 5:アルミ・銅 <被削材番号?> '))

if stock == 1:
    dnd = 7200
    if dnd == 2:
        dnd = 360
    elif dnd == 3:
        dnd = 4800
    elif dnd == 4:
        dnd = 1600+2500/2
        dnd = round(dnd,0)
        dnd = (400+400)/2
        dnd = round(dnd,0)
    elif dnd == 5:
        dnd = 2000
        dnd = 400
    elif dnd == 6:
        dnd = 2000*((2000-800)/3)
        dnd = round(dnd,0)
        dnd = 400*((400-300)/3)
        dnd = round(dnd,0)
    elif dnd == 7:
        dnd = ((2000-1800)/3)*1800
        dnd = round(dnd,0)
        dnd = ((1800-300)/3)*300
        dnd = round(dnd,0)
    elif dnd == 8:
        dnd = 1800
        dnd = 360
    elif dnd == 9:
        dnd = 1600+1400/2
        dnd = round(dnd,0)
        dnd = (300+300)/2
        dnd = round(dnd,0)
    elif dnd == 10:
        dnd = 1400
        dnd = 300
    elif dnd == 11:
        dnd = ((1400+1200)/2)
        dnd = round(dnd,0)
        dnd = (300+300)/2
        dnd = round(dnd,0)
    elif dnd == 12:
        dnd = 1200
```

図-2 条件分岐と計算式の Python コード

2.3 加工条件調整

実際の加工では,加工条件の調整を行うことが頻繁にある.図-3 にエンドミル (AE-VMS) の条件調整の Python コードを示す.

```
search.py - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#条件調整
kn = float(input('条件調整<4>? ')) #カタログ条件の何%で加工するか入力,変数に代入
kf = kn/100
en = kn
en = round(en,0)
ef = ef*kf
ef = round(ef,0)
aef = ef/4
aef = round(aef,0)
```

図-3 エンドミル条件調整の Python コード

2.4 追加条件

スクエアタイプに,ラジastypeの条件を追加した.図-4 にエンドミル (AE-VMS) の追加条件の Python コードを示す.

```
search.py - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#追加エンドミルタイプ
radiusType = int(input('1:NO 2:YES <RadiusType>? ')) #スクエアからラジastype選択入力,変数に代入
if radiusType == 1: #条件分岐
    ef = ef
elif radiusType == 2: #条件分岐
    ef = ef*0.8
    aef = round(aef,0)
    aef = (ef/4)*0.8
    aef = round(aef,0)

    hef = (aef+ef)/2
    hef = round(hef,0)
    hp = cd/25
    hp = round(hp,2)
    lowef = ef*0.7
    lowef = round(lowef,0)
```

図-4 エンドミル追加条件の Python コード

2.5 加工条件検索結果の表示

加工条件検索結果は,必須項目と利便性の高い追加情報を表示した. 図-5 にエンドミル (AE-VMS) の加工条件検索をした結果を Python の表示画面を示す.

```
1:ドリル 2:タップ 3:リーマ 4:ヘリックス 5:エンドミル 6:WireOut <切削タイプ>? 5
1:AE-VMS 2:VQ-JHV 3:VQ-MHV 4:VQ-MHZV 5:EMXA 6:RH2P 7:ALZ345 8:Epoch SUS Multi 9:E141 <エンドミル種類>? 1
3, 4, 5, 6, 8, 10, 12 <AE-VMS ENDMILL直径>? 8
1:側面切削 2:溝切削 <切削タイプ>? 2
1:SS400/SS60 2:SCM/SKD 3:NAK80/PM5 4:SUS <被削材番号>? 1
条件調整<4>? 80
1:NO 2:YES <RadiusType>? 1

AE-VMS直径: 8
切削タイプ: 溝切削
被削材: SS400/SS60

★ 回転数: 3200.0
★ アプローチ送り: 182.0
★ 送り: 728.0

低切削抵抗送り: 510.0

★ ヘリカル送り: 455.0
★ ヘリカルピッチ: 0.32

★ ap: 2.0
★ ae: 4.0

刃長: 19
```

図-5 エンドミル加工条件検索結果 Python 画面

3. ステップアップ

切削加工条件検索システムを,利便性の向上のため改良を複数行った. 図-6 に示す.

ステップアップ

ドリル	タップ	リーマ	ヘリックス	エンドミル	ワイヤカット
ADF	A-TAP	CRM	HD11	AE-VMS	FA10S
AGES	CoroTap		HD13	ALZ345	
AGSUSR	EX-SFT		HD17	RH2P	
AQDEXR	V-SFT/VP-SFT		HD22	VQ-JHV	
AQDEXZ			HD27	VQ-MHV	
NEXUS-GDR				VQ-MHZV	
SGESR				EpochSUSMulti	
ZPB				E141	
VP-GDR				EMXA	

一括検索用ファイル名: search.py

Python不要

図-6 切削加工条件システムのステップアップ

4. 参考：検索時間

参考に検索時間を計測した. 図-7 に示す.

参考/検索時間

工具種類	型番等	サイズ(Φ)	被削材	条件調整	カタログ(秒)	Python(秒)	比較
ドリル	AGES	5.1mm	炭素鋼	80%	71	15	4.8
ドリル	AQDEXR	7.0mm	炭素鋼	80%	117	14	8.5
エンドミル	Epoch	5.0mm	炭素鋼	80%	94	22	4.3
エンドミル	AE-VMS	6.0mm	炭素鋼	80%	230	15	15.7
エンドミル	EXMA	6.0mm	炭素鋼	80%	240	28	8.6
タップ	A-TAP	6.0mm	炭素鋼	80%	177	21	8.5
リーマ	CRM	6.0mm	SKD	80%	199	22	9.1
				平均(秒)	161	19	8.3
				平均(分)	2.7	0.3	8.3

図-7 参考検索時間

5. まとめ

金型実習(射出班・プレス班)の金型製作時間と利便性の面から貢献できた.また,実習を行った学生の評判も非常に良かった.

Python のインストールが不要のメリットは大きく,多数のパソコンで使用できるため利便性は大きく向上した.

機械工場の委託依頼作業にも利便性の面で貢献している.