【プログラム紹介】

パソコンによるNC孔明け機用シミュレーションプログラム

Application of Personal Computer to Simulation of Data for NC-Drill

関　秀明*　Hideaki SEKI
川崎義和*　Yoshikazu KAWASAKI
久留嶋一則*　Kazunori KURUSHIMA

1．まえがき

近年，生産部門においては，ＮＣ機器等の導入により作業効率の向上が図られ，孔明け作業においても，ＮＣ孔明け機が導入されている。

最近のＮＣ孔明け機は，従来のものに比べメモリー運転が可能になり，ドリルの改善による高速化と高精度化が計られているが，その稼動には多種多様な孔明けパターンのプログラムが必要である。そのためプログラム開発者は，机上におけるプログラム作成，ＮＣ機械内へのプログラム入力，孔明け機でのドライランチェックを，プログラムが完成するまで作業を繰り返さなければならない。また，ＮＣ機械へのプログラム入力は，手入力のため入力ミス等も発生する。この様にプログラム作成から実際の孔明け作業が始動するまでには，多くのロスタイムを生む要因があり，また誤作発生の原因ともなりかねない。

そこで今回，現場機械のロスタイムを無くし，さらにプログラムの効率的開発法として，パソコンによるＮＣ孔明け機用シミュレーションプログラムを開発したのでその概要を報告する。

2．概要

パソコンで行う作業は，次のようなものである。

1）ＮＣプログラムの編集を行う。
2）ＣＲＴ上でのプログラムシミュレーションにて，孔明け位置及び軌跡のチェックを行う。
3）完成したプログラムは，紙テープに出力し，ＮＣ機械に登録する。

本プログラムでプログラム開発の対象とする機械は，FANUC・SYSTEM 6Mの規格に基づいて作動する孔明け機である。

命令表及びプログラム仕様を表－１，２に示す。G機能・M機能・X，Y命令は工作機の制御を行い，WHILE＜式＞DO・GOTO・IF命令は，プログラムの流れの制御を行う。変数は，ローカル変数とコモン変数に分かれ，共に１０進８桁の精度である。

表－１　命令表

命　令	説　明
G機能	準備機能
M機能	補助機能
X，Y	ドリル移動命令
WHILE〈式〉DO	DO～END間を〈式〉が成り立っている間繰返す
END	
GOTO	分岐命令
IF GOTO	条件付分岐命令

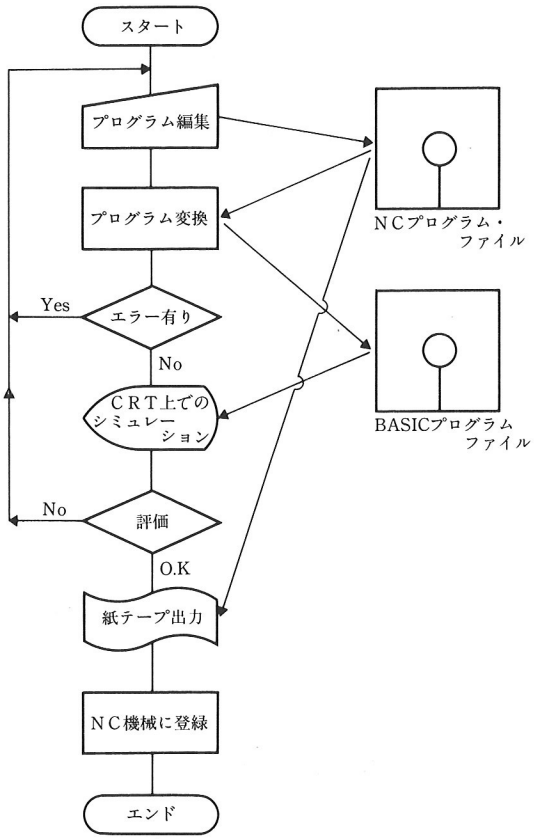
表－２　プログラム仕様

使用出来る変数	ローカル，コモン変数
演算精度	10進８桁
繰り返し識別番号	１～３
[　]の多重度	最大５重
サブプログラム呼び出し多重度	最大４重

＊川田工業㈱四国工場生産技術課

3．作業の流れ

プログラム編集からＮＣ機械に登録するまでの手順及びデータの流れを図－１に示す。



図－１　シミュレーションのフロー

開発者は，パソコンの編集プログラム（ＥＤ等）にてプログラム編集を行い，ＮＣプログラムファイルを作成する。

その後，プログラム変換を行う。この作業は，ＮＣプログラム（ユーザーマクロ）をパソコンプログラム（BASIC）に自動変換するものであり，同時にプログラムのエラーチェック（文法エラー）も行なっている。エラーメッセージ等が，ＣＲＴ上に出力されると，DEBUG（再編集）を行う。図－２，３は，プログラム変換例を示す。

プログラム変換にて，エラーが無くなれば，ＣＲＴ上に表示する座標の範囲の設定，各変数の初期値設定等のパラメータ入力を行う。そして，ＣＲＴ上でのシミュレーションに移り，ドリルのＸ・Ｙ座標を線分で，孔明け位置を円で画面表示する。又，変数の内容のプリントアウト，行番号のトレース等を用いて，DEBUGの効率を上げる事が出来る。図－４は，ＣＲＴ画面上でのドリル軌跡のシミュレーションの例である。

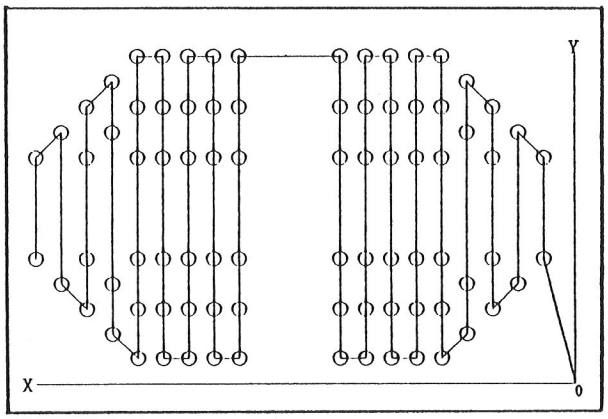
ＣＲＴ上で，軌跡の結果を判定し所定の動きが得られれば，ＮＣプログラムを紙テープに出力，さらにＮＣ孔明け機のメモリーに登録する。

```
M21
IF[#27EQ#28]GOTO6
#31=#103*#105
GOTO4
N6#31=#104*#105
N4#27=#27+1
END1
#31=0
#27=1
#105=-#105
IF[#28EQ1]GOTO5
X#7Y#103*#105
N5#28=#28-1
```

図－２　ユーザーマクロ例

```
100 AX#=HENSU#(1.SP%)
110 AY#=HENSU#(2.SP%)
120 LINE (32767-X#,32767-Y#)-(32767-(X#+AX#),32767-(Y#+AY#))
130 Y#=Y#+AY#:X#=X#+AX#
140 HENSU#(14.SP%)=1
150 HENSU#(15.SP%)=1
160 HENSU#(30.SP%)=0
170 HENSU#(31.SP%)=0
180 WHILE HENSU#(14.SP%)<=2*HENSU#(3.SP%)+2
190 WHILE HENSU#(15.SP%)<=HENSU#(11.SP%)+1
200 AX#=HENSU#(30.SP%)
210 AY#=HENSU#(31.SP%)
220 LINE (32767-X#,32767-Y#)-(32767-(X#+AX#),32767-(Y#+AY#))
230 Y#=Y#+AY#:X#=X#+AX#
240 CIRCLE (32767-X#,32767-Y#),15
250 HENSU#(31.SP%)=HENSU#(4.SP%)
260 HENSU#(15.SP%)=HENSU#(15.SP%)+1
```

図－３　変換後パソコンプログラム例



図－４　パソコン画面例

4．あとがき

今回は，ＮＣプログラムをパソコンによるシミュレーションでチェックするプログラムを開発し，有効に活用できた。この種のプログラムを適用できる分野は数多くあり，他の身近な問題にも色々波及できるものと思う。

今後は，時間的経過に伴なって，状況が変化する様な待ち行列モデルのシミュレーションプログラムの開発も行い，作業の円滑化を計って行きたい。