三次元NCデータ編集ソフトの開発*

外山　真也*1・佐藤　征亜*1

Development of the Editor for NC Data of 3D

Masaya TOYAMA and Masatsugu SATO

三次元CAD/CAMシステムを利用して作成したNCデータの容量は，一般的に500KB（約25,000行程度）を超えるデータとなることが多く，その修正編集は，修正箇所の検索だけでも容易ではない。また，NCデータを直接編集しようとすると，相対指令により設定された座標の計算などがあり，容易ではない。

そこで，NCデータの変更修正，あるいはグラフィック表示された工具軌跡を指示して，NCデータを変更修正できるソフトの開発を検討することとした。今回の開発では，NCデータのリスト表示と，そのデータによる工具軌跡のグラフィック表示を，一対一の関係を保持して実現できたので紹介する。

キーワード：NCデータ，CAM，Gコード，検索，編集

1　はじめに

三次元CAD/CAMシステムを利用して作成したNCデータの容量は，一般的に500KB（約25,000行程度）を超えるデータとなることが多い。このように約20,000行を超えるようなNCデータの修正編集は,修正箇所の検索だけでも容易ではなく，かつNCデータを直接編集しようとすると，相対指令により設定された座標の計算，G02やG03による円弧加工に対応した円弧の中心位置の座標計算など容易ではない。

さらに，NCデータを基に工具軌跡がグラフィック表示されている場合には，NCデータの変更修正に伴い工具軌跡も変更して表示されることが望ましい。このような機能を有するNCデータの編集が可能なソフトは見あたらない。

そこで，NCデータの変更修正，あるいはグラフィック表示された工具軌跡を指示して，NCデータを変更修正できる，両方向からの変更修正が可能なソフトの開発を検討することとした。

今回の開発では，NCデータのリスト表示と，そのデータによる工具軌跡のグラフィック表示を，一対一の関係を保持して実現したので紹介する。

2　開発方法

最初に，NCデータとグラフィック表示のためのデータとを一対一の関係で取り扱うためのデータ構造について，図１のようなクラス（ClsNcData）を定義した。そして，このクラスに文字列のNCデータを与えて，Gコード，移動先の座標などを求める機能を有する関数を定義した。

さらに，このデータを配列（ArrayList）として扱うクラス（ClsVectorNcDataList）を作成した。

```
public class ClsNcData  {
  String StrNc =""; // Nc Data String
  String StrFc =""; // Nc FCode String
  int    Code  = 0; // Code
  int    Gtype = 0; // G Type
  int    NcAI  = 0; // Nc Abs or Inc
  Object obj;       // Object Line Arc
}
```

図１　ClsNcDataの定義

＊　生産システムの高能率化・高精度化に関する研究（第１報）

＊１　機械電子・デザイン部

ここで，XYZの座標を有するデータについて定義するクラス（ClsXYZ）を図２に示す。そして，このClsXYZを利用して，直線と円弧の図形のクラスを作成した。直線（G01）に対応するクラスの定義を図３に，円弧（G02，G03）に対応するクラスの定義を図４に示す。

```
class ClsXYZ {
  double x;
  double y;
  double z;
  public ClsXYZ Set( ClsXYZ p ) {
       x = p.x;
       y = p.y;
       z = p.z;
       return this;
  }
}
```

図２　ClsXYZの定義

```
class ClsGLine  {
  int GCode = 0;
  ClsXYZ Sp = new ClsXYZ();// Start
  ClsXYZ Ep = new ClsXYZ();// End
  Col = ClsColor.LColor_BLACK;
}
```

図３　ClsGLineの定義

```
class ClsGArc  {
  int GCode = 0;
  ClsXYZ Sp = new ClsXYZ();//Start
  ClsXYZ Ep = new ClsXYZ();//End
  ClsXYZ Cp = new ClsXYZ();//Center
  double Rr = 0.0;// Radius
  int  Col= ClsColor.LColor_BLACK;
}
```

図４　ClsGArcの定義

２－１　データの読み込み

指定されたNCデータを読み込み，文字列データとしてのNCデータを保持し，そのデータで求められたGコードと座標データにより，直線または円弧のデータを作成する。例えば「G01 X 10.0 Y 2.35」のNCデータが与えられた場合は，図１で示したObject objには直線データのクラス（ClsGLine）が設定される。

このような手法で開発を行い，NCデータを読み込んで配列リストにNCデータと工具軌跡表示用のデータを作成するようにした。

２－２　グラフィック表示

グラフィック表示については，DirectXとOpenGLを利用し，開発を実施してみた。その各手法について述べる。

１）DirectXを利用した場合

DirectXを利用して開発したソフトで，NCデータを読み込み工具軌跡をグラフィック表示した例を図５に示す。また，その一部分を拡大して表示したものを図６に示す。

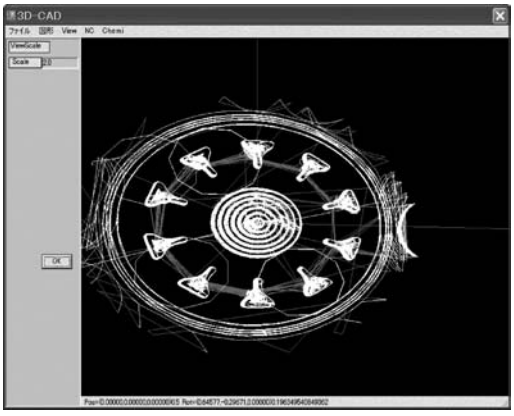

図５　DirectXを利用したソフトでのNCデータ工具軌跡の表示

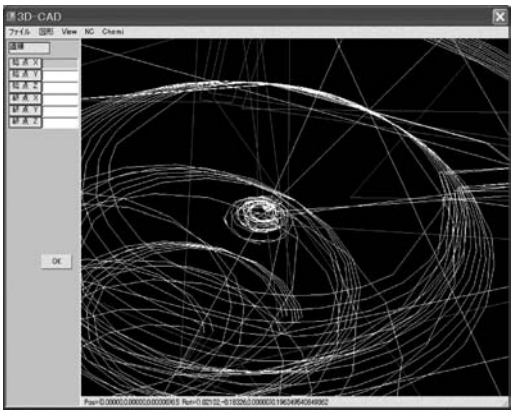

図６　図５の拡大表示

マウスにより，座標軸を回転させるなどの動作

を実現できたが，円弧の形状の表示は直線分割により表現しているために，拡大した場合の形状の表示に課題が残る。

2）OpenGL を利用した場合

OpenGLを利用して開発したソフトで，NCデータを読み込み工具軌跡をグラフィック表示した例を図７に示す。また，それを回転させて表示したものを図８に示す。

OpenGLを利用して開発したものは，拡大すると視点位置に形状が近づくようになり，ある程度以上になると視野から消失する。また，座標軸回転の場合も，デバッグ状態では反応が極めて遅く，マウスの動きに追従できない。

図７　OpenGLを利用したプログラムでのNCデータ工具軌跡の表示

図８　図７の拡大表示

３　結果および考察

プログラム開発はMicrosoft Visual Studio. NET C#を利用した。DirectXによるプログラムでは，デバッグ状態においてもマウスに追従した動作が可能であるが，OpenGLを利用したプログラムでは，マウスに追従した動作が容易ではなく，画面のちらつきも発生した。ただし，実行プログラムを起動した場合には，このような問題は解決される。

今回は，約570KB（約27,000行）程度のNCデータを読み込み，NCデータのリストとその工具軌跡を表示し，座標軸の回転を行うなどの機能をDirectXやOpenGLを利用して開発した。

その結果，各カーネルにおいて，開発手法は異なるが，ほぼ同様の機能を開発できた。

今回のデータでは，グラフィック表示された直線と円弧の図形データは，約25,000個となっている。

４　まとめ

今回の開発において，図形データは25,000個程度になっている。デバッグ状態において，座標軸回転などの動作や操作性にやや難点があるものの，基本機能の動作を確認できた。機能の点からはOpenGLを利用しているプログラムの方が，コマンドも充実しているために優れているが，動作の観点からは，DirectXを利用したプログラムの方がデバックなども容易であるため，今後の開発はDirectXによる開発を進めることにする。

５　参考文献

1）新藤義昭，阿部正平，“OpenGLリアルタイム3Dプログラミング”，株式会社秀和システム

2）“I/O BOOKS［書籍版］DirectX9 実践プログラミング”，工学社