

三次元NCデータ編集ソフトの開発に関する研究*

外山 真也*¹・佐藤 征臣*¹

Development of Editor for 3-Dimension NC Data

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25000行程度)を超えるデータとなることが多い。このようなNCデータの修正編集は、修正箇所の探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、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。

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、開発を進めている。

これまでの開発において、NCデータの編集機能の一部を開発できたので報告する。

キーワード : CG、NCデータ、CAD/CAM

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25000行程度)を超えるデータとなることが多い。このように約20000行を超えるようなNCデータの修正編集は、修正箇所の探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、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、開発を進めている。

これまでの開発において、NCデータの編集機能の一部を開発できたので報告する。

2 開発方法

NCデータリストを表示し、それらのデータの一部を指定すると、そのデータが示す工具軌跡部分を色を変更してグラフィック表示することを実現した。しかし、グラフィック表示部分をマウスで指定して、その指示されたNCデータを認識する機能を開発できていない。

この機能の開発にかなりの時間を費やしているが問題点も多いため、編集機能の開発を先に実施することとした。

ここで、NCデータをArrayListクラスとして定義し、編集においては、このクラスの複製を作成し、その複製データにおいて、作業を行うようにした。このようにすることで、作業途中での中断や、誤作業の対応も可能とした。

2-1 OpenGLによる開発

OpenGLを利用して開発したソフトにおいて、NCデータを読み込んだ後、工具軌跡をグラフィック表示させたものを図1に示す。また、読み込んだNCデータのリストを表示しているFormを図2に示す。

* 設計生産工程の高効率化に関する研究(第2報)

* 1 機械電子部

NCデータリストの一部をマウスで指示すると、その指示された部分のグラフィック表示部が青色に表示される。

この機能において、データが27000行程度の量になると、座標軸の回転などの操作において、動作が遅滞し、操作性が非常に悪くなった。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、その他の動作も反応が極めて遅くマウスの動きに追従できないなどの問題点がある。

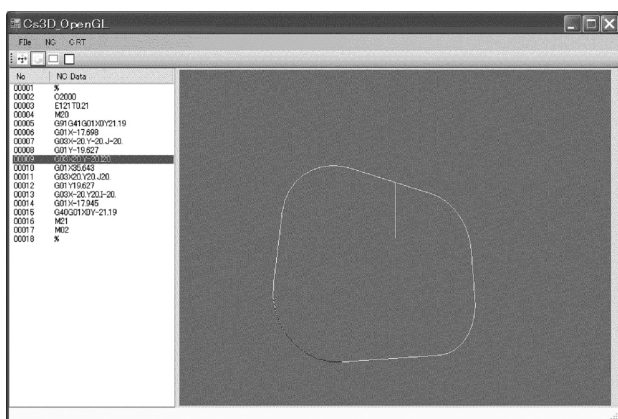


図1 OpenGLによる表示機能

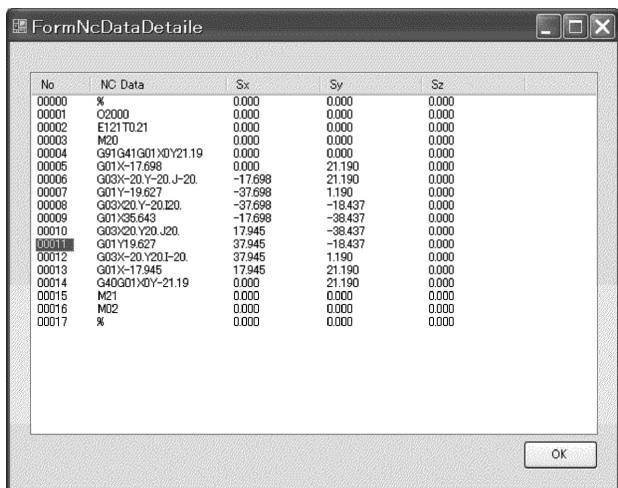


図2 NCデータのリスト表示

2-2 DirectXによる開発

DirectXを利用して開発したソフトにおいて、NCデータを読み込んだ後、工具軌跡をグラフィック表示させたものを図3に示す。また、読み込んだNCデータのリストを表示しているFormを図4に示す。

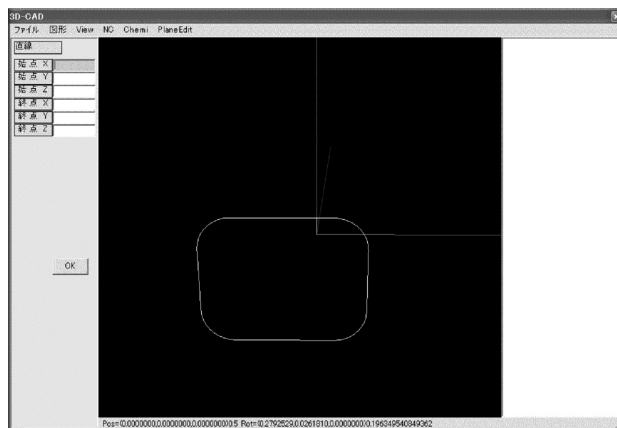


図3 DirectXによる表示機能

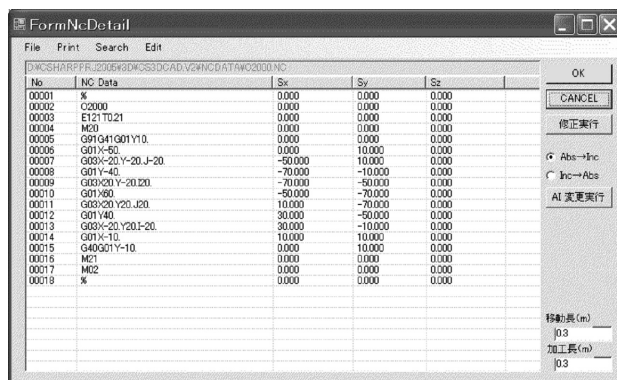


図4 NCデータのリスト表示

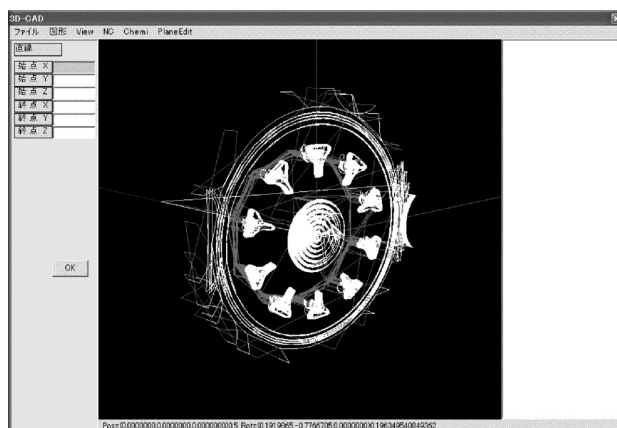


図5 DirectXを利用したプログラムでのNCデータ工具軌跡の表示

短いデータの場合は、NCデータを読み込んで表示するまでの時間において大差はないが、図5に示すような約27000行程度のデータ(573KB)の場合は、OpenGLソフトの場合で46秒、DirectXソフトの場合で20秒の、約2倍以上の差を生じた。このような差は、グラフィック表示機能の処理速度の相違により生じているものと推測するが、ソフトにおいて改善の手法が見出せない。

さらに、DirectXを利用して開発したソフトにおいて、図5に示すNCデータを読み込んだ後、そのデータを絶対座標系(ABS)から相対座標系(INC)へと変換する機能を開発し実施してみたところ、37秒を要した。

その変換前の状態を図6に、変換後の状態を図7に示す。

No	NC Data	Sx	Sy	Sz
00001	%	0.000	0.000	0.000
00002	G90	0.000	0.000	0.000
00003	G20(1) V0.250	0.000	0.000	0.000
00004	S100M03	0.000	0.000	50.000
00005	G01 Z25.684 Y6.52623 F250M08	0.000	0.000	50.000
00006	Z-0.5	25.684	6.526	3.000
00007	Z-0.05	25.684	6.526	2.950
00008	X-2.319 Y2.509	25.684	6.526	-0.050
00009	G02(1) 7.232 4.46 Y0.4571 Z-23.243 J-2.472 F1500	23.446	0.457	-0.050
00010	G01 Z-23.244 Y-3.2011 Z-22.101 J0.502	23.446	0.457	-0.050
00011	X-23.438 Y-0.659	23.447	0.457	-0.050
00012	Y-0.67	23.438	-0.659	-0.050
00013	G02(1) 23.234 Y-3.2011 Z-22.101 J0.502	23.234	-3.201	-0.050
00014	G01 Z-22.101 Y-3.219	23.234	-3.201	-0.050
00015	X-21.283 Y-9.963	23.219	-3.913	-0.050
00016	X-21.228 Y-9.976	21.283	-9.963	-0.050
00017	G02(1) 18.778 Y-17.3571 Z-19.769 J16.364	20.956	-11.387	-0.050
00018	G01 Z-19.769 Y-11.462	20.506	-11.387	-0.050
00019	X-17.872 Y-15.19	20.466	-11.462	-0.050
00020	X-17.824 Y-15.246	17.872	-15.190	-0.050
00021	G02(1) 16.943 Y-16.2211 Z-17.448 J14.881	16.943	-16.221	-0.050
00022	G01 X-16.942 Y-16.223	16.943	-16.221	-0.050
00023	X-16.929 Y-16.231	16.942	-16.223	-0.050
00024	X-16.932 Y-16.233	16.933	-16.231	-0.050
00025	G02(1) 15.778 Y-17.3571 Z-19.769 J19.113	15.778	-17.357	-0.050
00026	G01 X-15.746 Y-17.357	15.778	-17.357	-0.050

図6 DirectXを利用したソフトでのNCデータ編集(変換前)

No	NC Data	Sx	Sy	Sz
00001	%	0.000	0.000	0.000
00002	G91	0.000	0.000	0.000
00003	G92(1) V0.250	0.000	0.000	0.000
00004	S100M03	0.000	0.000	50.000
00005	G01 Z25.684 Y6.5262 Z-47 F250M08	0.000	0.000	50.000
00006	Z-0.5	25.684	6.526	3.000
00007	Z-0.05	25.684	6.526	2.950
00008	X-2.319 Y-4.016	25.684	6.526	-0.050
00009	G02(1) 7.191 Y-23.243 J-2.472 F1500	23.219	2.509	-0.050
00010	G01 X-0.001 Y-1.062	23.219	2.489	-0.050
00011	X-0.009 Y-1.066	23.220	2.489	-0.050
00012	Y-0.011	23.211	1.293	-0.050
00013	G02(1) 20.4 Y-2.5811 Z-22.101 J0.502	23.615	3.913	-0.050
00014	G01 X-0.015 Y-0.112	23.615	3.913	-0.050
00015	X-1.389 Y-6.85	23.600	3.907	-0.050
00016	X-0.055 Y-0.112	21.564	-2.749	-0.050
00017	G02(1) 7.231 Y-4.121 Z-33.669 J16.364	22.232	-1.449	-0.050
00018	G01 X-0.015 Y-0.076	22.232	-1.449	-0.050
00019	X-2.593 Y-3.728	22.192	-1.824	-0.050
00020	X-0.049 Y-0.056	19.899	-5.252	-0.050
00021	G02(1) 8.91 Y0.97631 Z-17.448 J14.881	20.432	-4.333	-0.050
00022	G01 X-0.001 Y-0.002	20.432	-4.333	-0.050
00023	X-0.009 Y-0.002	20.431	-4.333	-0.050
00024	X-0.001 Y-0.002	20.422	-4.343	-0.050
00025	G02(1) 15.4 Y-1.241 Z-19.769 J19.113	21.575	-3.221	-0.050
00026	G01 X-0.032 Y-0.03	21.575	-3.221	-0.050

図7 DirectXを利用したソフトでのNCデータ編集(変換後)

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27000行)程度のNCデータを読み込み、NCデータのリストと工具軌跡を表示し、座標軸の回転などの機能をDirectXやOpenGLを利用

して開発した。

さらに、NCデータの変換機能の一部を開発し、絶対座標系から相対座標系への変更修正、またその逆の変更修正を実施した。その結果、変更修正機能に問題がないことを確認し、データの読込と表示において処理速度と操作性の差異を明確にすることができた。

4 まとめ

今回の開発において、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、NCデータの編集機能の一部などを開発できた。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操作も容易であるため、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) 大川善邦他. DirectX9実践プログラミング. 工学社, 平成16年2月
- 2) 新藤義昭、阿部正平、OpenGLリアルタイムC#プログラミング. 秀和システム, 2001.02
- 3) I/O BOOKS[書籍版]DirectX9 実践プログラミング. 工学社