機械の動作シミュレーションによる多軸工作機械用 NC指令コード生成*

佐竹利文** 林 朗 弘*** 鈴木 裕***

Generation of NC Instruction Codes for Multi-axes Machine Tool by Simulation of Machine Tool Motion

Toshifumi SATAKE, Akihiro HAYASHI and Hiroshi SUZUKI

This paper presents a method of machine instruction code generation for multi-axis machine tool instead of the conventional manner. In the present method, each working part of multi-axis machine tool, cutting tool, and fixture are modeled as autonomous components and a mechanical simulation system is constructed by these autonomous components to simulate machining operation for multi-axis machine tool. NC instruction codes are generated by extracting motion of working parts made by performing mechanical simulation system model as the operative closure system served by mutual feedback among autonomous components under the constraints that are relation among working parts, cutting condition between cutting tool and workpiece, and interference with motion of cutting tool. This paper describes the concept of NC instruction code generation by performing mechanical simulation system as the operative closure system proposed in this study and details of evaluators and operators coded into the autonomous component for constructing such mechanical simulation system. Then effectiveness and feasibility of the proposed method has been estimated by performing the simulation and actual machining.

Key words : CAM, multi-axis machine tool, interference with cutting tool, mechanical simulation system, decentralized autonomous system, mutual feedback

1. はじめに

一般に、多軸工作機械用のNC指令コードは、CAMシステムによる工具位置及び姿勢データ（CLデータ）の生成、ポスト処理によるCLデータのNC指令コードへの変換という2つの過程を経ることで生成される。CAMシステムでは、加工物、工具、工具フォルダ及びジグの形状データを用いて干渉チェックや回避処理を行い工具先端が直線的に動作することを前提にCLデータを生成する。ポスト処理は、機械の構造や加工物の保持状態を考え、個々の機械に合わせてCLデータをNC指令コードに変換する。また、直線補間を前提に作られたCLデータを座標変換したのみでは、回転軸が動作する場合に工具の軌道は直線にはならないので、ポスト処理では、工具の軌道を評価して必要に応じてNC指令コードを追加／修正するリニアライゼーション処理が行われる。しかし、リニアライゼーション処理で追加されるNC指令コードは、工具、加工物、工具フォルダ、ジグ間の干渉チェックが行われていないため、工具干渉の危険性を持つ。この問題を解決するためには、これまでのような、CAMとポスト処理が分離した方法ではなく、CAMとポスト処理を一体として考え、形状データと機械の構造のデータを用いて機械の動作により得られる工具の軌道や干渉の問題を同時に評価する必要がある。

本研究では、多軸工作機械に工具、ジグ、加工物が取り付けられた状態を1つの機械系として計算機内にモデリングし、工具を加工面に沿って移動させることにより、機械系の状態を変化させその変化から直接NC指令コードを生成する方法を提案する。本論文では、機械、工具、加工物などを計算機内にモデル化する方法とこのモデルを用いてNC指令コードを生成する方法について述べる。

2. 多軸工作機械のモデル化と動作の生成

2.1 NC指令コード生成ソフトウェア

従来の多軸工作機械用NC指令コード生成処理の流れを図1に示す。多軸工作機械は、工具に任意の姿勢を与えることができるため、図中aに示すように、工具の位置データの生成に加えて工具をどのような姿勢で使用するかを考えなければならない。このとき、工具の姿勢や加工物の形状によっては工具干渉の危険性があるため、図中bに示すように、工具干渉のチェック及び干渉回避の処理が必要となる。干渉回避処理は、形状データを用いて工具、工具フォルダと加工物、ジグとの干渉をチェックし、干渉があった場合には干渉のない安全な姿勢に変更する処理である。干渉チェックを含めた工具姿勢の決定方法には、工具の形状表面上にチェックポイントを設け、すべてのチェックポイントにおいて干渉がないように姿勢を変化させる方法¹⁾²⁾、ロボットの分野で古くから用いられているコンフィギュレーション空間法を応用した方法³⁾などがある。

図中cに示すように、CAMシステムにより求められたCLデータを機械の構造を用いて座標変換すればNC指令コードに変換できる。しかし、回転軸が動作する場合、工具が直線的に動作しないため実際の工具軌道の評価と、修正を行う必要がある。図中dに示すように、この処理はリニアライゼーション処理と呼ばれ、機械の構造データを用いて工具の軌道を評価し、必要があればNC指令コードを追加する。リニアライゼーション処理を含めたポスト処理の方法には、多軸工作機械の構造ごとの特徴を考えた座標変換を行う方法⁴⁾、形状創成関数に基づいた座標変換により一般的な多軸工作機械に対して適用できるポスト処理の方法⁵⁾などがある。従来のNC指令コード生成ソフトウェアは、CLデータ生成処理とポスト処理が独立に構成されており、データは図中Aの矢印に示

* 原稿受付 平成10年5月21日
** 正会員 沼津工業高等専門学校（沼津市大岡 3600）
*** 正会員 九州工業大学情報工学部（飯塚市川津 680-4）

Fig.1 Process for generating NC instruction codes in the conventional system

Fig.2 Modeling of the multi-axis machine tool as a mechanical system

すような1方向のみの流れとなっている。そのため、リニアライゼーション処理での修正された NC 指令コードは、干渉チェックがなされず干渉の危険性を持つことになる。

提案する方法は、多軸工作機械に工具、ジグ、加工物が取り付けられた加工時の状態を1つの機械系として計算機内にモデル化し、このモデルの動作をシミュレートすることで工具干渉回避を含む工具姿勢決定から NC 指令コードの生成までを一体の処理として実現する。工具姿勢決定から NC 指令コードの生成までを一体のソフトウェアとして構成することによって、より完全な工具干渉のチェックや、加工物の形状データを用いて工具軌道を評価することで高精度な NC 指令コードが生成できるなどの効果が期待できる。

2.2 NC 指令コードの生成

図2は多軸工作機械、工具、ジグ、加工物から構成される機械系のモデル化の考え方を示したものである。(a)に示すように、加工時には、工具が工具フォルダにより主軸に取り付けられ、加工物はジグによりテーブルに固定される。さらに、工具と加工物は任意の状態接触し切削の状態がつくられ、全体として閉じた機械系を構成していると考えることができる。本方法では、回転軸や並進軸により連結している工作機械の構成部品までを含めて考え、この機械系をモデル化する。(b)に示すように、機械系を構成する各要素を、単純なルールを持ちそのルールにしたがって行動する自律的な要素により表す。この要素を動作要素と呼ぶ。

図3は、各動作要素のモデル化の方法を示したものである。(a)に示すように、テーブルやコラムなどの工作機械を構成する1つの部品は、1つの動作要素として表現する。この動作要素には、隣り合う動作要素との相互関係を維持するルールを与える。工具や工具フォルダは、(b)に示すように、形状を近似するように分布させた複数の動作要素の集合として表す。この動作要素には、構造を維持するルールに加えて、形状を近似するために他の動作要素との相対的な位置関係を維持するルールと干渉を回避するルール等を持たせる。(c)に示すように、加工物やジグは、市販のソリッドモデラを用いてモデリングする。これを空間に固定された動くことのない1

Fig.3 Modeling of the working parts

Fig.4 Data and action of the working parts

つの動作要素とする。これらの動作要素を、加工物やジグなどの空間に固定された動作要素を基準にシリアルに配置することで機械系を計算機内にモデル化する。

提案する方法では、このようなルールを持つ動作要素を工具と加工物との接触状態を条件に閉じた系として動作させ工具干渉のない機械の可能な状態を求める。求められた状態から各制御軸の変位を取り出し直接 NC 指令コードを生成する。

3. 動作要素の行動とルール

3.1 動作要素のデータと行動

図4は、動作要素の持つデータと動作の概要を示したものである。動作要素は、(a)に示すように、3つのベクトル v_i^1 , v_i^2 , v_i^3 で表されるローカル座標系と位置ベクトル P_i を持つ。動作要素の行動は、(b)に示すように、位置ベクトル P_i を更新することによる平行移動と、(c)に示すように、ローカル座標系を表すベクトル v_i^1 , v_i^2 , v_i^3 を変更する回転である。動作シミュレーションは、各動作要素のデータを、他の動作要素のデータと比較し、構造や形状を表す分布の矛盾を計算する。そして、矛盾を干渉するように動作要素が行動する。この処理をすべての動作要素について行い、構造の矛盾及び形状を表す分布の矛盾が解消されたとき、処理は終了する。

3.2 構造を表すためのルール

図5は、回転軸と並進軸で連結する構成部品を表す動作要素間の関係を示したものである。図中(a)に示すように、ベクトル v_i^1 を回転軸として結合された、2つの動作要素 i, j の間の関係は、 v_i^1 が一致することと接続している点で位置ベクトルが一致することである。また、図中(b)に示すように、 v_i^1 と平行な並進軸で結合する2つの動作要素間の関係は、3つの方向ベクトルがすべて一致することと、位置ベクトルが互いに他の動作要素の可動方向のベクトル v_i^1 の延長線上に存在することである。連結する他の動作要素との間の位置の矛盾 ΔP_i^j は、その動作要素を i , 連結すべき動作要素を j とすると、

$$\Delta P_i^j = V_i \cdot D_p \cdot (P_j - P_i) \quad (1)$$

Fig.5 Relation between the working parts which represents a mechanical system

Fig.6 Description of the cutting tool Fig.7 Procedure for calculating a conflict

$$\text{ただし, } V_i = [v_i^1, v_i^2, v_i^3], D_p = \begin{bmatrix} d_p^1 & 0 & 0 \\ 0 & d_p^2 & 0 \\ 0 & 0 & d_p^3 \end{bmatrix}, d_p^s = 1 \text{ or } 0$$

となる。ここで、 D_p は、移動方向を制限するためのパラメータであり、 $d_p^1 = 0$ であれば v_i^1 方向に移動できる機構を表す。動作要素 i の平行移動量は、連結する 2 つの動作要素それぞれとの間で求めた矛盾の平均とする。動作要素 i の方向ベクトルの矛盾 ΔV_i^j は、動作要素 i, j のローカル座標系ベクトルを V_i, V_j として、

$$\Delta V_i^j = (V_j - V_i) D_v \quad (2)$$

$$\text{ただし, } V_i = [v_i^1, v_i^2, v_i^3], D_v = \begin{bmatrix} d_v^1 & 0 & 0 \\ 0 & d_v^2 & 0 \\ 0 & 0 & d_v^3 \end{bmatrix}, d_v^s = 1 \text{ or } 0$$

となる。 D_v は、回転軸の回転方向を制限するためのパラメータであり、 $d_v^1 = 1$ であれば v_i^1 を回転軸として回転する機構を表す。動作要素 i の回転行動は、連結する 2 つの動作要素との間で求めた矛盾の平均を動作要素 i の方向ベクトルに加える。回転行動により方向ベクトルを変更した後、各軸を直交化する場合を行う。

3.3 工具形状を表すためのルール

工具及び工具フォルダは、図 6 に示すように、放射状に動作要素が分布したものとして表す。この分布形成の方法と動作要素に与えるルールは、これまで本研究で検討してきたセルの自己組織化手法を応用する⁶⁾⁷⁾。各動作要素には、位置ベクトルとローカル座標系を表すベクトルに加え、工具中心に向かうベクトル v_i^1 、長手方向ベクトル v_i^2 、工具中心軸回りの分割数 m 、干渉を検出するための判定領域の半径 r を持たせる。すべての動作要素は、工具中心軸から放射状にのびた線分上に等間隔で分布する。図 7 に示すように、動作要素 i と j の間の相対的な位置の矛盾は、動作要素 i と j のベクトル v_i^1 と v_j^1 のなす角 θ を求め分割数を用いてあるべき角度を

$$\theta' = \text{Round} \lfloor m \cdot \theta / 2\pi \rfloor \cdot 2\pi / m \quad (3)$$

で求める。次に工具中心軸回りに動作要素 j を θ' だけ回転し、この位置ベクトルを P_j' とする。このとき、 P_i と P_j' は v_i^1 と v_j^2 でできる平面上にある。次に、方向ベクトル v_i^1 と v_j^2 を使って、 P_j' を動作要素 i の近傍に移動する。移動後の位置と動作要素 i の位置ベクトル P_i の差

Fig.8 Procedure of conflict reduction

Fig.9 Method for driving a cutting tool

$$\Delta P_i^j = \sum_{n=1}^2 \{ \text{Round} \lfloor (P_j' - P_i) \cdot v_i^n / |v_i^n| \rfloor \cdot v_i^n \} + P_j' - P_i \quad (4)$$

が、位置の矛盾となる。動作要素 i の平行移動は、位置ベクトルをこの値を他のすべての動作要素について求め、その平均を移動量として、動作要素 i を移動させる。動作要素 i の位置ベクトルは、動作要素の総数を ℓ 個として、

$$P_i = P_i + \frac{1}{\ell - 1} \sum_{j=1}^{\ell-1} \Delta P_i^j \quad (5)$$

で求められる。方向ベクトルの矛盾は、式 (3) により動作要素 j の方向ベクトル v_j^1 を回転させたものと v_i^1 との差と、 v_j^2 と v_i^2 との差である。回転後の動作要素 j の方向ベクトルを $v_j^{1'}$ とすると、動作要素 i の方向ベクトルは、

$$(v_i^1, v_i^2) = \frac{1}{\ell - 1} \sum_{j=1}^{\ell-1} (v_j^{1'}, v_j^2) \quad (6)$$

で求められる。求めた方向ベクトルは、直交化を行う。このような処理をすべての動作要素が繰り返すことにより、矛盾は解消していき工具を表す分布が得られる。

3.4 工具干渉の回避

工具を表す動作要素が持つ干渉を検出するための判定領域の半径 r の領域に加工作やジグ、テーブルなどがある場合には干渉と見なし、動作要素は形状の法線ベクトル方向に位置ベクトルを変化させる。

図 8 は、干渉回避の状況を模式的に示したものである。(a) 及び (b) に示すように、干渉した動作要素は干渉回避ルールにより干渉を回避する方向に動き工具の形状を表す分布を乱す。この状態になると、動作要素間の相対関係に矛盾が生じるため、工具形状を表す分布を維持するルールが働き、各動作要素は分布の乱れを修正するように動く。それにより再び干渉が発生すると、干渉した動作要素は干渉回避のルールが働き干渉を回避する。干渉回避のルールと分布を維持するルールが状況に応じて交互に働いた結果、(c) に示すように、干渉のない空間に分布が形成されることになる。

3.5 加工面のトレース

工具を加工面に沿って動作させるために、図 9 に示すように、設定された加工開始位置から終了位置まで移動するルールと、加工面と接触するルールを工具刃先に位置する 1 つの動作要素に与える (以後、接触要素と呼ぶ)。また、工具底面の中心に位置する動作要素 (以降、中心要素と呼ぶ) には、加工開始位置と終了位置から求められる平面上を移動するルールを与え、干渉を判定する半径を大きく取る。これは、返り刃が加工面と接触しないように加工面に対してヒールをかけるためである。図 10 は、加工面に沿った動作の模式図である。

Fig.10 Procedure for determining a cutting tool attitude

Fig.11 Table tilt type 5-axis machine tool modeled as a mechanical system

Fig.12 Example workpieces and cutting line

最初、移動ベクトルを求める。接触要素の位置から加工終了位置に向かうベクトルを U_o ，加工面からの法線ベクトルを U_h ，微小移動距離を Δt とすると移動ベクトル U_m は、

$$U_m = \Delta t \cdot (U_o - U_h) / |U_o - U_h| \quad (7)$$

で求める。切削セルが加工面と離れている場合は、加工面と接するまで同様の計算を繰り返す。次に、移動した接触要素と中心要素の位置から求められるベクトルに接触要素の方向ベクトルを一致させる。最後に、工具を構成する他の動作要素を移動ベクトル U_m だけ平行移動させる。このとき、中心要素が加工面と離れている場合は、工具の長手方向ベクトルを微小量回転させる。これは、加工面に対しての工具姿勢をできるだけ一定にするための処理である。ボールエンドミルの場合、中心要素と接触要素は同一の動作要素となる。

このようなルールにより工具を加工面に沿って移動させたとき、接触要素の方向ベクトルの変化や加工面の状態によっては干渉が発生する。このような状態となった場合、形状を表すためのルールや干渉を回避するためのルールが働き、干渉のない空間に分布が作られる。そのため、接触要素は加工物の形状によって必ずしも加工面と接触しない。また、ヒールの角度も同様の理由により、必ずしも設定したものにならない。しかし、これらルールは、工具を表す動作要素の微小な状態変化と干渉回避及び形状の維持により、少なくとも切刃以外の部分での干渉のない連続的に変化する工具姿勢列を生成する。また、このような干渉のない場合には、想定された工具姿勢をできるだけ維持するため、工具姿勢を初期の姿勢に戻す方向に微小量変化させる処理を行う。

4. 多軸工作機械の動作生成シミュレーション

提案した方法の有効性を検証するために、コンピュータシミュレーションによる工具先端軌道の評価及び加工を行った。シミュレーション及び加工に用いる機械は、テーブルリルト型5軸工作機械である。図11に示すように、機械の主要

(a) A cross section of workpiece No.1

(b) Trajectory in the area A

(c) Trajectory in the area B

(d) Trajectory in the area C

Fig.13 A line graph of the trajectory of cutting tool tip

な構成部品は各々動作要素としてモデル化される。図12は、検証に用いた加工物の形状と工具経路を示したものである。(a)の加工物No.1は、直方体上に円柱を2つ載せた形状である。(b)の加工物No.2は、工具進行方向に対してオーバハンドした形状である。2つの加工物をそれぞれフラットエンドミル、ボールエンドミルを用いて加工を行うためのNC指令コードを生成した。

加工面をトレースする工具の移動量を一定の微小距離としてNC指令コードを生成し、その際の工具軌道についての検証をシミュレーションにより行った。図13は、加工物No.1をフラットエンドミルで加工する場合の工具先端の軌道を示したものである。3つのグラフは、形状急変部の工具先端の軌道を拡大表示したものである。点は、生成したNC指令コードで得られる工具先端の位置であり、点の間を結ぶ曲線は、制御軸の速度を一定と考え、求めたNC指令コード間の各制御軸の変位を計算して求めた工具先端の軌道を示す。(b)は、円筒面を上っていく際の工具軌道を示したものである。工具先端の軌道は、比較的加工面に食い込む量が少ないことが分かる。これは、本方法が工具姿勢をできるだけ一定に保つように工具姿勢を微小に変化させることに加え、工具の底面後方の干渉によって姿勢の変化がさらに小さくなるためである。(c)は、円筒面の乗り移りの部分の工具軌道である。工具前面と底面の干渉がある部分では工具先端が形状から離れた位置となっている。これは、工具底面での干渉、工具前方の干渉と工具先端と加工面との接触の条件などにより、その位置では加工面に接触する姿勢が求められなかったためである。(d)は、円筒面から下りる部分の工具軌道を示したものである。工具先端が加工面に食い込んでいることが分かる。これは、(a)の場合に比べ、工具の底面後方の干渉が少なく工具の姿勢変化が大きくなったためである。

図14は、加工物No.2をボールエンドミルで加工する場合の工具接触点の軌道を示したものである。図14(b)では、工具軌道が大きく加工面に食い込んでいることが分かる。これは、工具の干渉により工具の姿勢変化が大きくなったため

Fig.14 A line graph of the trajectory of cutting tool tip

ある。(c)では、逆に形状表面から工具軌道が離れている。これは、工具の姿勢を加工面に対して一定にする処理における姿勢変化量が大いいためである。(d)は、形状先端部の工具軌道を示したものである。形状凸部での食込みは少ない。

図15には、生成されたNC指令コードを用いて、2つの加工物の加工結果を示す。写真手前の部分が仕上げ加工まで行ったものであり、写真奥の部分は荒加工まで行った部分である。荒加工用のNC指令コードは、工具を表す動作要素の干渉領域を切込み深さに合わせて設定して生成した。2つの加工物の仕上げ加工のNCデータ生成に要した時間は、Pentium 133のパソコンを用いてNo.1が走査線1本当たり15分、No.2が走査線1本当たり30秒であった。

シミュレーション及び加工結果から、工具軌道を評価し得られる精度を検証できること、工具の切刃以外で干渉のない安全なNC指令コードを生成できることが分かる。本方法では工具姿勢変化をできるだけ小さくすることを基準として、工具の姿勢決定を行っているため、急激な工具姿勢変化は生じない。しかし、工具の移動量を一定量としてNC指令コードを生成する方法では要求精度は得られない場合がある。本方法は、形状データと機械構造データを用いて工具姿勢決定からNC指令コードの生成を一体の処理として実現している。そのため、工具の移動量を工具軌道を評価することにより決定しながらNC指令コードを生成する機能を追加することで、要求精度の加工面を得ることが期待できる。

5. おわりに

多軸工作機械用のNC指令コードを生成するシステムの開発を目的として、工具干渉回避を含む工具姿勢決定処理からNC指令コード生成処理を一体の処理として実現する方法を提案した。本論文の要点をまとめると以下ようになる。

- (1) 工作機械を構成する構成部品、工具フォルダ、工具、加工物及びジグを単純なルールにしたがって行動する動作

(a) Machining result of the Work No.1 by the flat end-milling cutter

(b) Machining result of the Work No.2 by the ball end-milling cutter

Fig.15 Actual machining result by using NC instruction codes generated

要素により表現し機械の構造にしたがって連結することで計算機内に表現する方法を提案した。

- (2) 動作要素に与える工作機械の構造を維持するためのルール、干渉を回避するルール、加工面をトレースするルールについて述べ、モデル化された機械系の動作をシミュレートすることで工具干渉回避を含む工具姿勢の決定からNC指令コードの生成までを一体の処理として構成する方法を示した。
- (3) フラットエンドミル及びボールエンドミル用のNC指令コードを生成し、シミュレーション及び加工実験を行い、本方法の有効性を示した。

本方法は、機械構造を表す動作要素のルールを機械の構造に合わせて設定することにより、様々な構造を持つ工作機械に適用可能である。また、加工面との接触状態を作る工具を表す動作要素に与えるルールを加工面の形状や工具切れ刃の形状に合わせて設定すれば、工具の特徴を生かした加工ができるNC指令コードを生成することもまた期待できる。

参考文献

- 1) 竹内芳美, 出村 健, 渡辺隆弘, 伊藤哲也: ソリッドモデルに基づく5軸制御加工, 精密工学会誌, **56**, 11 (1990) 111.
- 2) 趙曉明, 葛東方, 堤正臣: 同時5軸制御マシニングセンタ用CAMシステムに関する研究, 精密工学会誌, **61**, 12 (1995) 1745.
- 3) 森重功一, 加瀬 究, 竹内芳美: 2次元 Configuration 空間を用いた5軸制御加工における工具干渉回避法, 精密工学会誌, **62**, 1 (1996) 80.
- 4) 竹内芳美, 渡辺隆弘: 5軸制御マシニングセンタ用ポストプロセッサの研究, 精密工学会誌, **58**, 9 (1992) 1586.
- 5) 長坂 学, 竹内芳美: 形状創成関数に基づく5軸制御加工用一般化ポストプロセッサの研究, 精密工学会誌, **62**, 11 (1996) 1607.
- 6) 林 朗弘, 佐竹利文, 蘇 亦志, 鈴木 裕: セル集団の自己組織化によるロボットの形状モデリングと姿勢決定問題への適用, 精密工学会誌, **62**, 10 (1996) 1411.
- 7) 佐竹利文, 林 朗弘, 鬼塚昭一, 鈴木 裕: セル集団の自己組織化を適用したロボットアームの連続的な姿勢変化の生成, 精密工学会誌, **62**, 10 (1996) 1415.