

要旨

旋削用NCプログラムのチェック作業において、2値化データを用いた干渉チェックと、数値データを用いた演算処理を組合せることにより、工具と要求形状との干渉チェック及び、プログラムされた座標値が要求形状通りの寸法になっているか否かを正確にチェックできるシステムを開発した。

1. 緒論

NC工作機械は加工の高精度化・高効率化及び省人化を図るために必要不可欠の基本設備となっている。NC工作機械を稼働するには、その動きを制御するNCデータを作成し、データが正確に作成されかつ干渉がないか否かのチェックを行う必要がある。NCプログラムの文法のチェックは比較的自動化は簡単であり、すでに実用化されている。しかし、座標値や干渉のチェックは人間による判断で行われており、この工程の自動化が望まれている。筆者らはこれらの目的に対し計算機メモリ空間を仮想加工空間モデルとする2値化データによる加工シミュレータの開発を行ってきた。この方法は極めて単純、正確であるがプログラムされた座標値と要求形状の関係(加工中の残し量等)を定量的に把握することは、困難であった。本報ではプログラムされた座標値及び工具と要求形状との干渉のチェックを主目的として、従来の方法に加え、工具のノーズR部と要求形状間の最短距離を求める数値演算処理を組み合わせることで、NCプログラムを正確にかつ高速に自動チェック可能なシステムを開発し、実用化したのでその内容について報告する。

2. 基本的な考え方

筆者らによって開発された、2値化データによる加工シミュレータは、パーソナルコンピュータの有する2組のV-RAMを仮想加工空間として利用し、チェック精度0.1mm一定で干渉チェックを可能とした。しかし、チェック精度をさらに向上させるためには膨大なメモリを必要とする欠点がある。またプログラミングされた座標値を、定量的にチェックすることは困難である。一般に旋削用NCプログラムは、工具先端のノーズR部における仮想刃先を基準にプログラミングされる。そこで座標値データを正確にチェックする方法として、工具先端のノーズR部と要求形状を構成する図形要素間の最短距離を求め、工具の移動中にそのデータをCRT画面上に表示する方法をとることとした。工具と要求形状との干渉チェックは従来通りの方法を用い、2値化データではチェックしきれない部分を、数値演算処理で求めた最短距離の表示で補完する方法により、工具と要求形状との干渉チェック以外に、残し加工時における残し量の把握、及び最終仕上加工における座標値の正確なチェック等が可能である。図1に開発したシステムの構成図を、図2に最短距離の算出方法を示す。

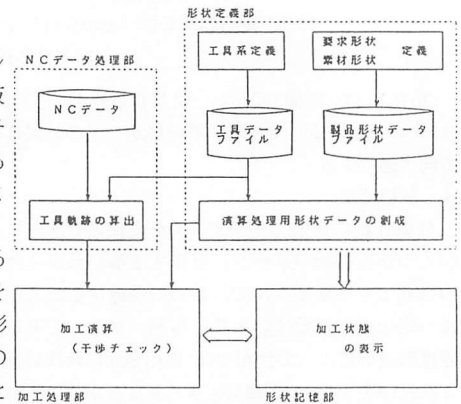


図1. システムの構成

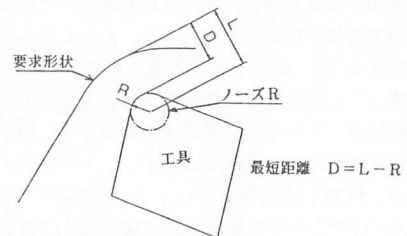


図2. 最短距離の算出方法

3. 工具形状定義

本シミュレータで定義可能な工具形状は工具先端のチップ形状のみで、図3に示す6種類のチップ形状に関し定義することができる。各々のチップ形状に対し1) ノーズ半径R、2) 基準寸法

A、3) 仮想刃先番号、4) チップ先端角TH1、5) 切刃角TH2、をパラメータとして入力し、形状を定義する。

工具データは工具番号をファイル名として登録され、NCプログラム内のTコード指令により、工具形状が自動的に生成され画面上に表示される。

4. NCデータの処理

NCデータの検証を行う場合、NCデータの指令にとまなう、工具移動軌跡の算出が必要となる。

工具移動の基本動作としては、1) G01, G00 の直線補間、2) G02, G03 の円弧補間、がある。直線補間の場合、始点及び終点の座標値からX方向とZ方向の増分値を計算し、増分値ずつX, Zの座標値を変化させ、工具の移動を行わせる。ここで、始点の座標値をXs, Zs, 終点の座標値をXe, ZeとするとX方向、Z方向の増分値Dx, Dzは次式で求まる。

$$Dx = (Xe - Xs) / L \quad Dz = (Ze - Zs) / L$$

$$L = \sqrt{(Xe - Xs)^2 + (Ze - Zs)^2}$$

円弧の動きに関しては、円弧を5°ピッチで分割して、その間を直線補間で移動させる方法を取った。

その他の動きとして、1) G41, G42の刃先R補正、2) M98のサブプログラム呼び出しでの移動、3) G65におけるマクロ命令による移動、の3つに関し工具移動軌跡を求めるサブルーチン関数を作成しこれらの動きに関しても、自動チェックを可能とした。図4に基本的なNCデータの処理の流れを示す。

5. 適用事例

タービンロータシャフト材の胴部加工用に作成されたNCデータを用い、本システムを用いたNCプログラムのチェックを実施した。写真1. にNCプログラムのチェック状況を示す。

工具の移動に関しては、刃先R補正、サブプログラム呼び出しでの移動、マクロ命令による移動とも問題なく動作した。また、工具形状に関しては、Tコードが変わるたびに、自動的にコンピュータ内で工具データを作り直し変更が行われた。要求形状と工具間の最短距離、回転数等の表示や干渉チェックも確実に行われた。NCプログラムのチェックに要する時間は、マニュアルで行う場合に対して40～60%短縮され、品質も大幅に向上した。

6. 結論

2値化データによる干渉チェックと、数値データによる工具と要求形状間の最短距離を求める演算を組み合わせた処理を行わせることにより、NCプログラムを正確かつ高速に自動チェック出来るシステムを開発、実用化し、チェック時間の短縮と品質の向上を可能とした。

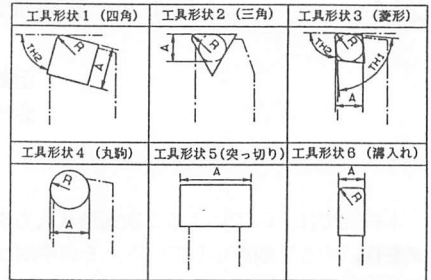


図3. 定義可能な工具形状

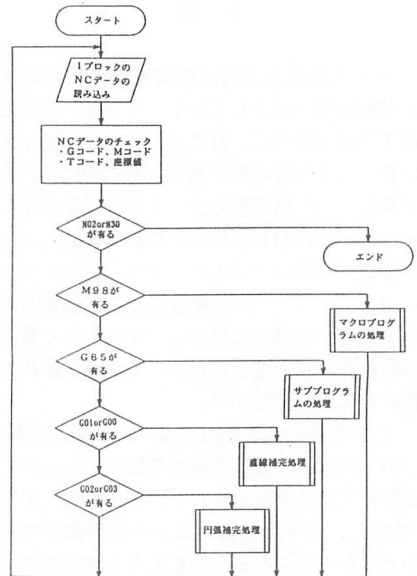


図4. 基本的なNCデータの処理の流れ

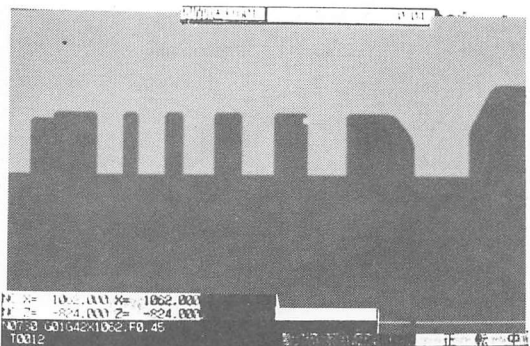


写真1. NCプログラムのチェック状況