

金型の加工誤差補償システムの開発 (第4報)

カッターパス補正ソフトウェアの改善と加工実験

門藤至宏, 中濱久雄^{*1}, 西川隆敏^{*2}, 菊田敬一^{*2}, 古本浩章^{*3}, 山下弘之^{*3}

Compensation System for Machining Error of Die and Mold IV

Upgrading tool-path revision software and process experiment

MONDO Munehiro, NAKAHAMA Hisao^{*1}, NISHIKAWA Takatoshi^{*2}, KIKUTA Keiichi^{*2},
FURUMOTO Hiroaki^{*3} and YAMASHITA Hiroyuki^{*3}

Practical software of cutting-tool path revisions was upgraded for arc command corrections. The software reduces cutting error of die and mold derived from tool deflections and tool-shape errors. In the part of arc revision, unnecessary least square approximation for short path and necessary revised direction perpendicular to the arc face are recognized and fixed up by adding algorithm. According to experimental results, another problem appeared in algorithm. Algorithmic correction of cutting command is implemented and is analyzed the effectiveness using simulation procedure.

切削加工中の工具たわみと工具形状に起因する加工誤差を補正したカッターパスを作成するソフトウェアについて、前報で未実施であった円弧補正部分の実装を行った。その中で、円弧指令のアルゴリズムについて、最小自乗近似の不要な短いパス用のものや、単一平面のみならず移動平面に垂直方向の移動成分が必要なものが判明し、その対応を行った。また、開発したソフトウェアによるパス修正結果を用いた切削実験において、補正アルゴリズムに問題が見つかったため、その修正を行い、シミュレーションにより問題点の解消を確認した。

キーワード：金型，切削加工，加工誤差，カッターパス，CAM

1. 緒 言

プレス金型作製工程では、切削加工とクリアランス(上下の金型間の隙間)確認が繰り返し行われるため、加工時間やコストの増大を招いている。高精度な切削加工のためには、加工現象を正確に知る必要がある。切削加工中の工具は、被削物から切削力を受けるため、たわみながら加工していて、そのたわみ量は工具の材質や直径などに依存する。また、工具自身の製造誤差や取付精度などの形状誤差によっても、加工精度は影響を受ける。

こうした切削中の工具の動的たわみや工具の形状誤差は、市販のCAMソフトから出力されるカッターパスには反映されていない。そのため、通常のCAMソフトによるカッターパスは、実加工における誤差要因を含んでいると考えられる。我々は、その点に着目し、市販のCAMソフトによるカッターパスに、それらの加工誤差分を加算

した新しいカッターパスを出力するソフトウェアを試作している。採用している手法は、元のカッターパスを、誤差補正を含めたカッターパスに変換するため、その前後の設計ソフトや工作機械に変更を加えることなく加工誤差補正を行うことが可能である。前回の報告において、アルゴリズムの提示のみで、実装出来ていなかった円弧指令を追加した。また、円弧指令のカッターパス出力部分の検討の中で、パスの短い場合や誤差補正により通常の加工面に垂直方向の成分を持つ場合への対応が必要なが判明した。さらに、誤差補正アルゴリズムの一部に、過剰切削現象を起こす場合があることが判明した。今回は、これらの問題への対処について報告する。

2. 加工誤差補正

加工誤差補正(工具経路修正)の処理の流れは、前報同様(図1)である。グラフィックスハードウェアの描画機能¹⁾により、被削材の初期形状、直前までの工具軌跡(掃引体)、工具の現在位置(予測位置)を、同一画面上に多重描画し、切削領域を判定する。次に工具刃先を微小切刃に分割し、各切刃での加工を判定し、分力を求め、その総和から1回転分の切削力予測値を得る²⁾。更に、切削力による工具のたわみや工具の形状誤差による加工誤差を求める^{3) 4)}。この加工誤差を元のカッターパ

2010. 5. 14 受理 加工技術研究部

*1 東部工業技術センター材料技術研究部

*2 西部工業技術センター生産技術アカデミー
金型加工プロジェクトチーム

*3 西部工業技術センター生産技術アカデミー
生産システム研究部

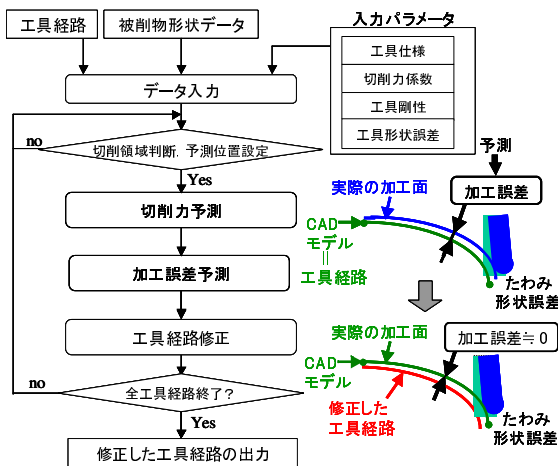


図1 本システムの処理フロー

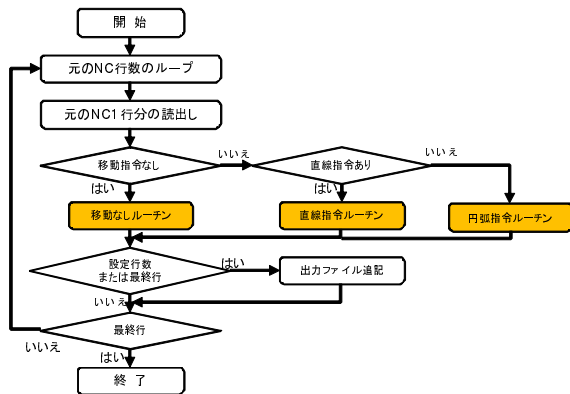


図2 カッターパス補正フロー

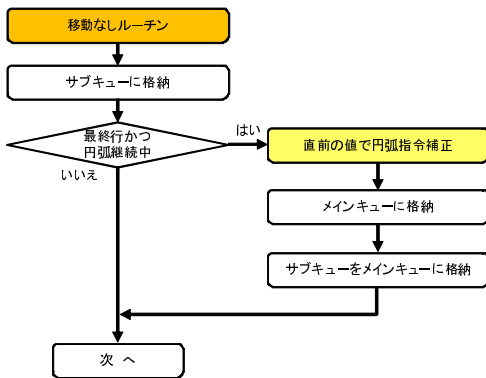


図3 移動なしルーチン

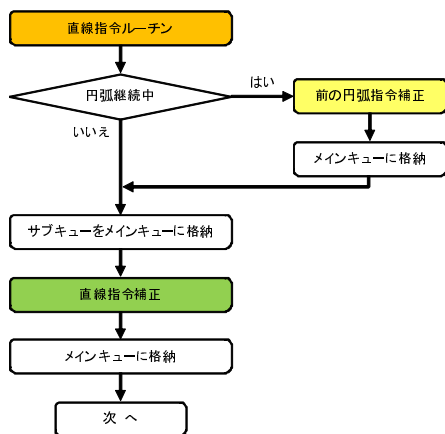


図4 直線指令ルーチン

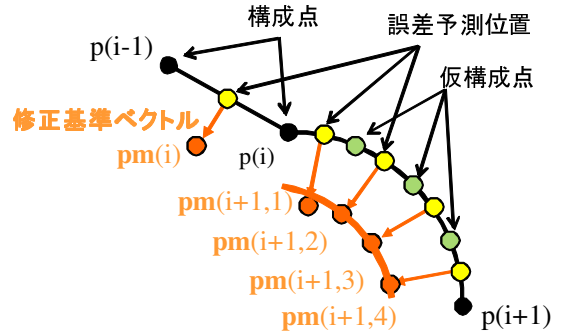


図5 仮構成点・修正基準ベクトル pm

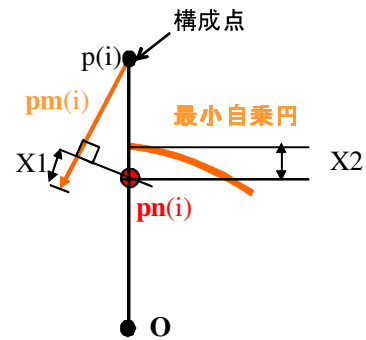


図6 最小自乗円・修正ベクトル pn

に適用し、工具たわみや工具形状誤差を織り込んだ新しいパスを作成する⁵⁾事が本研究の目的であり、その補正フローを図2に示す。このフローは、大きく3つのルーチン(移動なし, 直線指令, 円弧指令)に分けられ、元のNC加工データを行単位で読み出す毎に分類され、行内での混在はないとした。また行数においても、補正前後で変えていない。

2.1 移動なしルーチン

1番目は、移動なし(XYZ指令等を含まない)ルーチン(図3)で、任意の行に任意の行数が発生する。このルーチンでは、元の行をそのままの形で残す処理を行っている。実際の金型のNC加工データは、かなりのメモリを使用するため、一時的な先入れ先出し配列(キュー)を利用し、補正結果を逐次ファイル出力する事でメモリを節約している。補正処理の都合上、2種類のキュー(メイン, サブ)を用いた。ランダムに出現する移動なしの指令内容を、一時的にサブキューに格納し、適切なタイミングでメインキューに組み入れる事で、移動指令の補正処理を妨げない処理を可能とした。

2.2 直線指令ルーチン

2番目は、直線指令を補正するルーチン(図4)である。

まず、円弧指令が継続中(直前の移動指令が円弧の場合)ならば、この直線指令により円弧終点が決まるため、直前の円弧指令の補正を実施できる。その後、サブキューにデータがあればメインキューに格納し、移動なし指令を統合する。さらにその後で、本来の直線指令の誤差

補正処理を行い、メインキューに格納する。

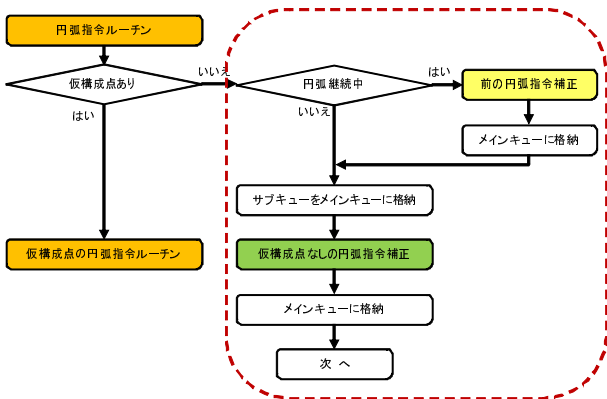


図7 円弧指令ルーチン

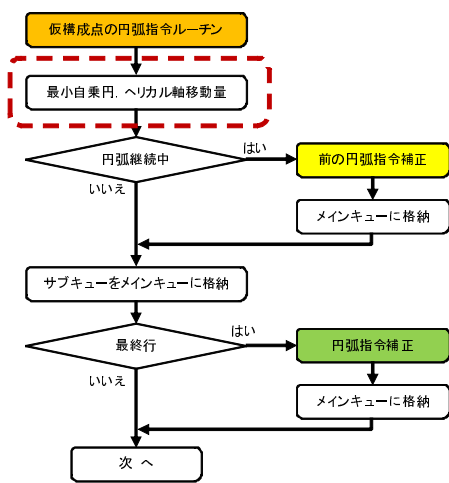
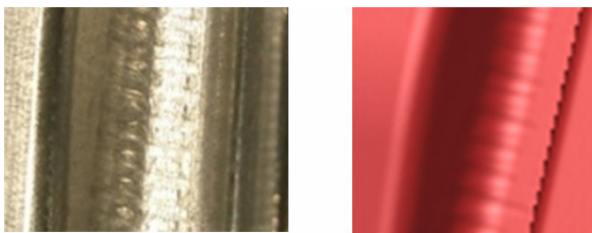


図8 仮構成点の円弧指令ルーチン



加工面 NCデータ

図9 切削実験結果と補正 NC データ

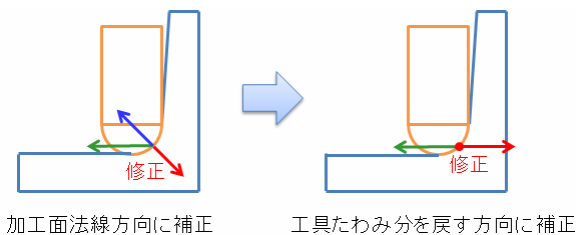


図10 補正方向の修正

2.3 円弧指令ルーチン

円弧指令の補正ルーチンは、仮構成点の有無により処理が分岐する。ここで、加工誤差計算において、必要に応じてパスの始点と終点の間の円弧上に複数の構成点を追加する場合、それらの追加構成点を仮構成点（図5参照）と呼ぶ。円弧始点 $p(i)$ と終点 $p(i+1)$ の間に一定間隔で設定した仮構成点の間の切削領域中点において誤差予測を行い、修正基準ベクトル p_m を求め、さらに最小自乗円近似により始点側・終点側の修正ベクトル（図6参照）や新円弧中心座標を算出する。これらの処理により、円弧指令を短い断片に分解し、各断片での加工誤差を求め、それらを集約して新円弧指令に置換する事が可能となる。

円弧が短く、仮構成点がない場合（図7の点線内）は、図5～6の処理は実行できず、円弧指令でありながら直線指令と類似した補正となる。直前の移動指令が円弧の場合であれば、その補正を先に実施し、続いてサブキュー、それから自身の仮構成点なしの円弧指令の補正処理を行う。直線指令と同様に、次の移動指令処理を待たずに補正処理が可能で、結果をメインキューに格納する。

仮構成点があり（図8）、補正処理を単純化するため、指定平面以外に誤差補正する必要がある場合は、ヘリカル補間（指定平面で回転移動しながら、その垂直方向に直線移動する）とし、最小自乗円を求める過程で、ヘリカル軸方向の移動量も加算（図8の点線内）する。ヘリカル補間を許容する事で、補正処理は単純化されるが、一部のNC機械では対応できない可能性がある。ここで、円弧指令が継続中（直前の移動指令が円弧の場合）ならば、この時点で前の円弧終点が決まるため、直前の円弧指令の補正を行い、メインキューに格納する。その後、サブキューにデータがあればメインキューに格納し、移動なし指令を統合する。さらに最終行の場合は、注目している円弧指令の終点が決まるため、その補正を行い、メインキューに格納するが、最終行でない場合は、次の移動指令が出現した際に補正を行う。これら一連のアルゴリズムにより、指定平面以外の補正に対処可能となった。

開発環境は、前報⁵⁾同様に Microsoft Visual Studio 2008 において .NET Framework 3.5 を利用し、開発言語は Visual C++ を用いて作成している。

3. 加工実験

3.1 過剰切削の問題

上述したアルゴリズムを用いて加工実験を行った際に、凹形状部において過剰切削現象（食込み）が発生した。図9左側が実際の加工面である。その現象は、補正後のNCデータでの掃引体（工具軌跡）描画においても確認され（図9右側）、補正処理において発生した問題であることが判明した。

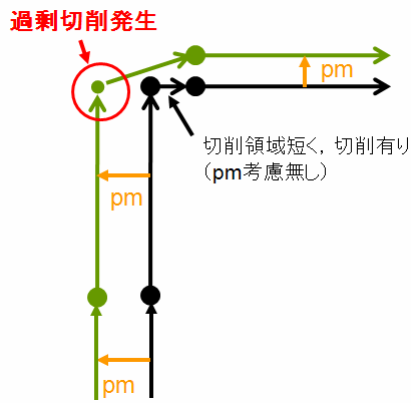


図 11 見直し前のアルゴリズム

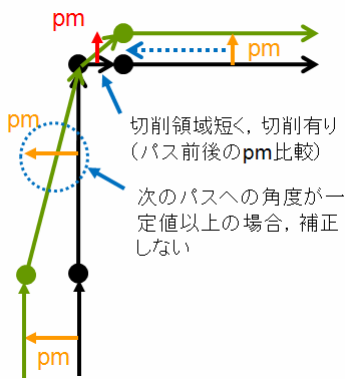


図 12 見直し後のアルゴリズム

3.2 誤差補正の修正

過剰切削現象の原因として、誤差補正方向を加工面法線方向に取っている事が考えられ、本来の工具たわみ補正方向に修正した結果、この現象が解消された(図10)。この場合、工具は長さ方向ではなく、横方向にたわむため、長さ方向に補正すると過剰切削となる。

図11は見直し前のアルゴリズムを示し、黒線が補正前のカッターパスを、また黒点は補正前の構成点を表している。パスは90°に曲がり、角度変化後の切削領域(黒点の間隔)が短くなっている。見直し前のアルゴリズムでは、切削領域の短いパスについては、修正基準ベクトル(pm)を考慮しないため、その他の切削領域の修正基準ベクトルに基づき、緑線のカッターパスに補正される。ここで、修正基準ベクトルは3次元方向成分をもち、紙面垂直方向成分も存在する。図9で問題となった過剰切削は、この垂直成分である。

そこで、見直した内容は2点あり、図12に示す。

1点目は、パスの方向が一定角度以上変化する場合、その変化点でパスの送り速度がほぼゼロになるため、加工誤差は発生しないとみなせる。具体的には、図12の黒線矢印方向に加工が進行する場合、角度変化点の手前の修正基準ベクトルをゼロとみなす。

2点目は、見直し前のアルゴリズムで考慮しなかった切削領域の短いパスについても、パス前後の角度比較から、送り方向角度差の小さい方のpmで代用して補正処理を行う事とした。図12の場合は、短いパスと次のパスの角度差がゼロなので、その修正ベクトルで代用する。

これらの見直しにより、補正後のカッターパスは、図12の緑線となり、過剰切削現象を解決した。

4. 結 言

前報⁵⁾において、アルゴリズムの提示のみで、実装出来ていなかった円弧指令処理コードを、加工誤差補償ソフトウェアに追加した。円弧指令のカッターパス出力部分において、パスの短い場合(仮構成点の無い場合)や誤差補正により通常の加工面に垂直方向の成分を持つ場合への対応が必要となり、ソフトウェアの修正を行った。

さらに、誤差補正アルゴリズムの一部に、過剰切削現象を起こす場合があることが判明し、送り速度がほぼゼロになる場合や、1刃送り以下の短いパス(切削有り)の修正基準ベクトルの見直しを行い、解決した。

今後は、実用化に向け、高速計算処理技術が重要になると考えられる。

文 献

- 1) 金子順一, 堀尾健一郎: 精密工学会誌, 74, 480-485 (2008).
- 2) 菊田敬一ほか6名: 広島県立総合技術研究所西部工業技術センター研究報告, 51, 26-29 (2008).
- 3) 西川隆敏ほか6名: 広島県立総合技術研究所西部工業技術センター研究報告, 52, 40-43 (2009).
- 4) 西川隆敏ほか5名: 型技術, 24(8), 70-71 (2009).
- 5) 門藤至宏ほか5名: 広島県立総合技術研究所東部工業技術センター研究報告, 22, 10-13 (2009).