金型設計における機械加工検証システムの開発（第二報）*

－加工困難解消のための形状修正範囲の導出－

乾 正 知**

Development of Manufacturability Verification System for Assisting Mold Design
- Computation of Shape Modification Range to Resolve Unmanufacturable Features -

Masatomo INUI

In designing a holder part of a large stamping die, designers must consider not only the functional property of the holder, but also its manufacturability. The holder part is produced by cutting and engraving table, wall, slot and pocket features into the raw cast shape. Since designers do not always have enough knowledge about the manufacturing method, they sometimes design a part with some unmanufacturable features. The author develops a new system for suggesting a design modification method to eliminate such unmanufacturable ones. This system automatically determines some optimum positions of a cutter for completely machining the unmanufacturable feature. Such cutters in the optimum positions have interferences with the holder part. Our system visualizes the interference volume between the cutter and the holder part. The designer can determine the design modification method of the holder part based on the interference volume.

Key words: CAD, CAM, manufacturability evaluation, optimization, interference detection.

1. はじめに

自動車のプレス成形用金型を保持するためのホルダ部品には、金型を取り付ける座面や側面、溝、ポケット、穴など、数多くのフィーチャが定義されている。図1にはそのようなフィーチャの例を示した。水平面を意味する座面は特に多く用いられており、図2に示した大型のホルダ部品の場合、図中に白く示した座面の数は百を超えている。ホルダ部品の加工は、予め概略形状まで成型された鋳物から、これらのフィーチャを削り出す作業となる。

ホルダ部品の設計者は、フィーチャの形状や配置を決める際に、金型の位置決めや固定といった機能的な要件だけでなく、切削に用いる工具の形状や工具の動きなどの加工要件を十分に理解して作業する必要がある。しかし誤解や見落としを完全に無くすことは難しく、工具が干渉するなどの理由で、しばしば加工できない形状を設計してしまう。このような加工困難形状が、製造下流のNCデータ作成や機械加工の段階で判明した場合には、大きな手戻りになり加工コストの増大を招いてしまう。

ある自動車メーカーでは、このような手戻りを防止するために、設計直後に加工の専門家がホルダ部品のCADモデルを加工性の観点からチェックし、必要があれば設計者に修正を求めている。しかし大型部品の場合、修正を要する箇所が数十個見つかることもあり、この作業が加工の専門家と設計者の双方にとって大きな負担になっている。著者は、この問題を解決するために、設計者自身による問題箇所の検出と修正が可能で、機械加工検証システムの研究を進めている。本システムを用いることで、加工要件を熟知していない設計者でも、加工上の問題点を設計中に随時検出でき、早い段階で設計修正することが可能になる。

Fig. 1 Form features appearing in holder parts

Fig. 2 Table features defined on a holder part for supporting a large stamping die

われわれは論文¹⁾において、加工困難な形状を自動検出するアルゴリズムを提案した。このアルゴリズムは、指定された工具でフィーチャが切削可能か否かを幾何的にチェックし、加工困難な場合には削り残る形状を画面に表示す

* 原稿受付 平成20年10月6日

** 正会員 茨城大学（茨城県日立市中成沢 4-12-1）

Fig. 3 Unmanufacturable region remained on a table feature (a), and possible cutter interferences to remove the region (b)

Fig. 4 Cutter positions for machining a point p on a feature. Ball-end cutter for milling a curved feature (a), flat-end cutter for milling a vertical wall feature (b), and flat-end cutter for milling a horizontal table feature (c)

る。提案した手法を用いて検出した加工困難形状の例を図3(a)に示した。図中の白い部分は、切削加工が必要な座面を表している。円形の座面は長方形の座面より少し上に配置されている。そのためフラットエンドミルが縦壁と円形の座面の間を通過できず、座面の上に図中に緑色で示すような削り残り（加工困難形状）が生じてしまう。このままではホルダ部品は機能を果たすことができないので、設計者は削り残りが生じないように設計変更する必要がある。しかし加工困難形状の情報だけでは、ホルダ部品にどのような修正を加えるべきか決めることが難しい。図3の例では、設計者は円形の座面の位置や形状を変更し問題の解消を図ることが一般的だが、削り残り形状の情報だけでは、座面をどの程度修正すべきか判断できない。

この修正作業を支援するために、本研究では加工困難な形状を除去できる適切な工具配置を自動決定するアルゴリズムを開発した。このアルゴリズムにより配置された工具は、ホルダ部品と干渉する。この干渉範囲を設計者に提示すれば、設計変更の際に有用な情報を与えることができる。図3(b)には、今回開発したアルゴリズムによる処理結果を示した。加工困難形状を除去するために配置された工具が、

ホルダ部品のどの部分と干渉するかが、赤く色分け表示されている。この図を見れば、例えば座面の位置を変更して問題を解決する場合には、どの程度右へ動かせば工具干渉が回避でき、結果として加工困難が解消できるか容易に判断できる。

2. 過去の研究例

加工困難な形状を抽出し、製品の加工性を評価する研究は過去に多数行われている^{2)~7)}。しかし本研究が対象としている、加工困難形状を解消するための設計修正支援を扱った研究は例がない。指定された形状を加工する工具経路を計算することは、CAMソフトウェアの最も基本的な機能である。工具経路の計算技術については膨大な研究成果があるが（例えば⁸⁾）、その多くは工具干渉の生じない経路を得ることを目的としており、そのままでは今回の目的に使えない。

工具干渉を許容する経路の生成も可能だが、今回の目的で使う場合には以下の問題がある。フィーチャの大部分を占める座面は水平なので、加工にはフラットエンドミルを用いる。ボールエンドミルを用いた曲面加工では、図4(a)に示すように、指定された点 p を加工する際の工具の位置 q は一意に決まる。フラットエンドミルを用いて垂直な側面を加工する場合も、同図(b)に示すように、側面の p から p' の範囲を加工できる工具位置 q は、ほぼ一意に決めることができる。しかし水平な面をフラットエンドミルで加工する場合には、点 p を加工するための工具位置を一意に決めることができない。同図(c)に示した2つの工具位置 q と q' の間のどこに工具を配置しても p を加工できる。このような曖昧さがあるため、加工困難形状を除去するための工具配置には様々なパターンが考えられる。ホルダ部品の加工に用いる工具の直径は数十ミリもあるため、用いるパターンによっては工具干渉の範囲が必要以上に大きくなり、誤った情報を設計者に与えてしまう。

本研究ではこの問題を解決するために、加工困難形状が完全に除去でき、しかも周囲の他の形状と最も干渉にくい位置に工具を自動配置する新しいアルゴリズムを開発した。なお、上述した理由から、水平面を有するフィーチャをフラットエンドミル加工する場合に対象を限定する。このようなフィーチャとしては、座面以外に図1(c)と(d)に示したポケットや溝が考えられる。

3. アルゴリズムの概要

3.1 入力と出力

開発した、加工困難形状除去のための工具配置アルゴリズムは、入力として以下の3つの情報を想定している。

ホルダ部品のCADモデル：本研究では、ホルダ部品の形状が三角形ポリゴンの集合として表されていることを仮定する。座面や側面などのフィーチャは、フィーチャを構成するポリゴンに、フィーチャのタイプなどを属性として付記することで表現する。自動車メーカでは、フィーチャ記述を有するCADシステムの利用が普及しており、上述した属性情報を伴う三角形ポリゴン群を、STLやOBJ形式で容易に得ることができる。本研究では、フィーチャ属性を有するOBJ形式のデータを入力として用いることにした。

加工困難形状の定義：本研究では、水平な平面上に検出された加工困難形状のみを対象としている。加工困難形状の輪郭を表す折れ線の点列が、入力データとして与えられる。

工具形状の定義：加工用の工具は、切刃、シャンク、工具ホルダの三つの部分からなる。切刃としてフラットエンド

Fig. 5 Typical cutter motions to machine some table features

Fig. 6 A region where the cutter center point should locate to remove given unmanufacturable region (a), and the region after removing the cutter center points which are too close to the wall feature (b)

ミルのみを想定している。シャンクや工具ホルダの形状は、円錐台の連続として定義されているものとする。

本アルゴリズムは、これらのデータに基づいて加工困難形状の除去が可能な最適な工具配置を決定し、配置された工具とホルダ部品の干渉範囲をグラフィックス出力する。

3.2 処理の基本方針と流れ

本研究では、水平な面上の加工困難形状を除去するためのフラットエンドミルの配置法が重要な役割を果たす。適切な工具配置をアルゴリズム化するために、ある自動車メーカーで、実際に行なわれている座面や溝、ポケット加工の手法を調査した。調査結果をまとめたものの一部を図5に示す。同図(a)には、周囲に障害物がない座面を加工する際の工具移動を赤線で示した。工具は基本的には座面の内側に沿って移動するが、このままでは角の部分が削り残るので、その部分だけ最小限外にはみ出すように移動する。一方、座面の横に側面フィーチャが存在する場合には、同図(b)に示すように、角の部分での工具の移動量と移動方向が、側面に削り込まないように変化する。ただしこの場合でも、工具のはみ出す量は最小になるように制御される。工具径よりも幅の狭い座面を加工する際には、同図(c)に示すように工具は狭い部分の中央を移動するが、これも工具のはみ出しが最小となるように制御された結果と判断できる。

これらの結果から、加工困難形状を除去する際には、工具は以下の2条件を満たすように配置されるべきと考えた。
条件1：工具は近接する側面フィーチャに削り込まないように配置される。

条件2：工具は加工困難形状の各部分を加工できる位置の中で、外にはみ出す量が最小の場所、すなわち工具とホルダ部品の干渉が最も起きにくい場所に配置される。

これらを踏まえて、本研究では以下に示す3ステップか

Fig. 7 An unmanufacturable region where some modifications of the wall features is required to achieve its removal

らなる処理により、加工困難形状を除去するための工具配置を決定することにした。

Step1：加工困難形状を一部でも除去可能な工具配置の範囲を求める。

Step2：得られた範囲を細かな格子に分割し、格子点ごとに、そこに配置された工具とホルダ部品のCADモデルの干渉量を計算し記録する。

Step3：記録に基づいて、加工困難形状の各部分を除去するための最適な工具位置を、工具とホルダ部品の干渉量が最小の位置として求める。

最後に、求められた位置に配置された工具とホルダ部品の干渉範囲を、赤く色分け表示する。次章でStep1～Step3の処理の詳細について述べる。

4. 加工困難形状除去のための工具配置アルゴリズム

4.1 Step1：工具の取り得る位置の範囲の取得

ホルダ部品は、主軸方向が垂直な3軸制御の工作機械で加工される。加工困難形状は水平な平面上的閉じた図形なので、その輪郭が加工できれば図形の内部は必ず加工できる。輪郭線を加工するには、フラットエンドミルの工具中心（工具先端の円の中心）が、輪郭から工具半径 r 以内に存在する必要がある。そこで輪郭線を構成する頂点と線分を工具半径分膨張させ、工具中心の存在すべき範囲を得る。図6(a)では、緑色の領域が加工困難形状を、そして周囲の灰色の領域が工具中心の取り得る範囲を表している。

座面に近接して側面フィーチャが与えられている場合には、前述の条件1を考慮して、工具は側面に削り込まないように配置される。この条件を取り込むために、上で得られた工具中心の存在すべき範囲から、側面からの距離が r 以内の領域を削除する。その様子を図6(b)に示した。設計の際に座面や側面フィーチャの配置を誤ると、座面や側面を大幅に設計変更しない限り加工できない形状が生じてしまう。図7にはそのような形状を示した。この例では、二つの側面が座面の角に非常に接近して配置されているため、側面への削り込みを禁止すると、工具をどのように配置しても座面の角の部分が加工できない。ホルダ部品の設計ではこのような設計ミスも多く見られるため、「側面に削り込まない」という条件は必要に応じて解除できるものとする。

4.2 Step2：工具とホルダ部品の干渉量の算出

本ステップでは、Step1で得られた範囲を細かな直交格子に分割する。そして工具中心を各格子点に配置したときの工具とホルダ部品の干渉量を調べ、各格子点に記録する。

工具は加工困難な形状を一部でも除去できる位置に配置されるので、配置された工具とホルダ部品は必ず干渉を持

Fig. 8 Computation of the interference volume between the obstacle table and a cutter being positioned on a grid point in the region given in Fig.6

Fig. 9 Z-map representation of a holder part and a cutter

つ。図 8 の例では、格子点 **a** に置かれた工具は、図中の円形の座面と **A** と示した干渉を起こすので、この干渉部の体積を **a** 点に記録する。同様に **b** に置かれた工具は **B** と示した干渉を起こすので、その体積を **b** 点に記録する。

格子のサイズを細かなものにするほど、後述する工具配置決定の際に高精度な処理が可能になるが、格子サイズを 1/2 にすると、格子の総数が 4 倍に増えるので、干渉量の計算回数と計算時間もそれに比例して増加してしまう。本研究では、格子サイズは一辺が 0.1~0.5 mm 程度になるように調整した。配置された工具とホルダ部品の干渉の状況をビジュアルに示し、設計者に設計変更の目安を与えるためには、この程度の精度で十分と考えている。

干渉量の計算では、格子点に配置された工具形状とホルダ部品の積形状を計算し、その体積を得ることになるが、この計算には大きな手間を要する。処理を高速化するために、ホルダ部品の CAD モデルと工具の形状モデルを、ともに図 9 に示すような Z マップ形式に変換した。その際には、ホルダ部品の Z マップは、下側に与えられた格子から上に伸びる直方体群としてモデル化し、逆に工具の Z マップは、十分に上側に与えられた格子から下に向かう直方体群としてモデル化した。Z マップへの変換には、コンピュータ・グラフィックスのデプスバッファ機能を利用した⁹⁾。

Fig. 10 A part of a CAD model (a) and its transformation result into Z-map (b)

これら二つの Z マップの定義の基になっている直交格子の解像度などは、上述した工具を配置する領域の直行格子と同じになっている。同じ格子位置に定義されたホルダ部品の Z マップの直方体の上端の高さを zh 、工具の Z マップの直方体の下端の高さを zt とする。 $zh - zt = h$ が正ならば二つの直方体は交わりを有することになる。直方体の断面積を a とすれば、交わりの体積は ha となるので、ホルダ部品の直方体と工具の直方体の対応する全てのペアについて交差している部分の体積を計算し積算することで、ホルダ部品と工具の干渉量を高速に得ることができる。

ホルダ部品の Z マップは、一度取得すれば同じものを何度も使いまわすことができる。工具の Z マップは、工具を領域内の各格子点に配置するたびに計算しなおす必要があるが、今回は工具を配置する領域の格子と工具の Z マップ化の際に利用する格子が同一なので、一度得られた Z マップをそのまま平行移動させるだけで、工具の位置変更に対応できる¹⁰⁾。

Z マップモデルでは横に挟れた形状が表現できない。そのためホルダ部品に図 10(a)に示すような形状があった場合には、Z マップ化の際にそれが同図(b)に示すように変化して記録されてしまう。その結果、干渉量の計算に誤差が生じるが、その影響はごくわずかなので、本研究では特別な扱いはしない。

4.3 Step3 : 最適な工具位置の決定

Step2 により、加工困難形状を一部でも除去可能な全ての工具位置（格子点）には、そこに工具を配置した時の、工具とホルダ部品の干渉量が記録されている。Step3 では、この情報に基づいて、最適な工具位置を決定する。

本研究では、加工困難形状の中の点 **p** を除去するための最適な工具位置を、この点を除去可能な工具配置の中で工具とホルダ部品間の干渉量が最小の位置と定めた。前述のように、加工困難形状を囲む輪郭線が除去できれば内部は必ず除去できる。そこで加工困難形状の輪郭線を、格子サイズとほぼ同じ長さの細かな折れ線に分割し、得られた（細分された）輪郭線上の各点を除去するための最適な工具位置を、以下の手順で決定する。

細分された輪郭線上の各点 **p** について、図 11 に示した破線の円のように、その点を中心とする工具半径に等しい半径の円を描く。この円内に工具中心が存在するように工具を配置すれば、

Fig. 11 Selection method of the optimum cutter position for milling a point p on the boundary of the unmanufacturable region

Fig. 12 Sample holder part with some unmanufacturable regions

Fig. 13 An unmanufacturable region remained on the holder part (a), and an interference region between the holder part and cutters being placed to remove the unmanufacturable region (b)

p は必ず除去できる。そこで円内の全ての格子点について、記録してある工具とホルダ部品間の干渉量を調べ、その値が最小の格子点を、 p を除去するための最適な工具位置と考え、そこに工具を配置する。図 11 の例では、白丸で表示した点に工具を配置すると干渉量（図中に斜線で示した部分の体積）が最小となるので、ここが p を除去するための最適な工具位置となる。この処理を、輪郭線上の全ての点について繰り返すと、加工困難形状を解消するための最適な工具配置が全て得られる。最後に、配置された工

Fig. 14 Two unmanufacturable regions remained on the holder part (a), and interference regions between the holder part and cutters being placed to remove the unmanufacturable regions (b)

具モデルとホルダ部品の集合積を計算し、その上面を赤く染めることで、工具とホルダ部品の干渉範囲を設計者に表示する。集合積の計算では、Step2 で作成した Z マップモデルを再度利用する。

5. プログラムの実装と処理結果

上述した、加工困難形状を解消する最適な工具配置の決定アルゴリズムを Visual C++ を用いてプログラム化し、手法の有効性を検証した。

論文¹⁾で報告した加工困難形状の検出プログラムを、図 12 に示したホルダ部品の CAD モデル（サイズ：2232mm×1882mm×937mm）に適用し、幾つかの加工困難な形状（図中の緑色の部分）を検出した。検出された加工困難形状の一つを図 13(a)に示した。この形状に今回報告したプログラムを適用し、加工困難形状を解消するための工具配置を求め、それとホルダ部品モデルの干渉する範囲を表示した結果を、同図 (b)に示した。この計算では、円筒形のホルダ（φ300mm）に取り付けられた φ80mm のフラットエンドミルを工具として指定した。刃先とシャンクの長さの合計は 490mm とした。図中で赤く示された部分が干渉部なので、これがなくなるようにフィーチャの大きさや位置を変更すれば、加工困難が解消できることになる。

同様の処理を図 14(a)の二箇所の加工困難形状に適用した結果を、同図(b)に示した。これらの例では、隣接する側面への工具の削り込みを禁止して処理をおこなった。処理に用いた Z マップモデルの格子の解像度は、5000×5000 に設定した。各格子のサイズ（＝計算精度）は 0.28mm であった。計算には、CPU が AMD Opteron 3.2GHz、メモリが 16GB の Windows Xp x64 ワークステーションを用いた。最適な工具配置と干渉部分の表示に要した計算時間は、図 13 の例では 43.4CPU 秒、図 14 の例では 10.2CPU 秒であった。これらの計算時間のほとんどは、Step2 の工具とホルダ部品の干渉量の計算に費やされている。

図 15(a)に示した加工困難形状は、座面に隣接する 2 枚の側面に起因している。そのため側面への削り込みを禁止すると、工具をどのように配置してもこの加工困難形状を解消できない。同図(b)には、側面への削り込みを禁止した場合の計算結果を示す。側面との干渉を回避するように工具を配置しているため、加工困難形状の一部が解消できず、緑色の領域が隅に残ってしまっている。また配置された工

Fig. 15 Unmanufacturable region (a), and some cutter interference regions computed by the system if interferences between the cutter and wall feature are not allowed (b), and interferences between them are allowed (c).

Fig. 16 An unmanufacturable region detected in a slot feature (a), and cutter interference shape computed by our program (b)

具が干渉すると判定された領域も、非常に不自然な位置に発生している。(c)には、側面への削り込みを許容した場合の結果を示した。こちらでは、全ての加工困難形状が解消されていることが分かる。設計者はこれらの表示結果から、側面フィーチャの形状や位置を修正して干渉を回避するか、または座面の側の形状や位置を修正して削り残りが生じないようにするか判断することになる。

側面への削り込みを許容した場合には、Step1において工具の取り得る位置の範囲が広がるため、計算にはより多くの時間を要する。図15(b)の計算に要した時間は8.4CPU秒だったのに対し、(c)の計算には23.1CPU秒を必要としており、この例では処理時間が3倍ほど増加している。

提案した手法では、期待した結果が得られないケースを図16に示す。この例では、水平な平面の左右に側面が定義されていて、細長い溝フィーチャを構成している。溝の幅が工具径よりも狭いため、同図(a)に緑色で示すように、溝全体が加工困難となっている。本手法を適用した結果、工具が配置され、同図(b)に赤色で示した部分が工具干渉として検出された。溝の幅が一定であるにもかかわらず、溝の出口の部分で干渉の幅が広がっている。これは図(c)に示したように、溝の出口付近の点Pを除去する際には、工具が溝の中心に配置された状態(図中の破線の円A)よりも、左右に少しずれた位置に配置された状態(実線の円B)のほうが、工具とホルダ部品(図中に灰色で示した形状)の干渉量が小さくなるためである。このような問題はあるものの、提案したアルゴリズムはほとんどの場合において望ましい結果を得ており、実務への適用は十分に可能と考えている。

6. まとめと展望

本論文では、ホルダ部品の設計支援を目的に、加工困難な形状を解消するための工具位置を計算する手法について述べた。具体的には以下の2つの技術について述べた。

- (1) 加工困難形状の除去が可能な工具位置の範囲を計算し、その内部に工具を配置した時の工具とホルダ部品の干渉量の効率的な計算法。
- (2) 干渉量が最小の位置に工具を配置し、その時の工具とホルダ部品の干渉形状をグラフィックス出力することで、設計変更に必要な情報を設計者に示す手法。

実際にプログラムを開発し幾つかの例題に適用することで、本手法の有効性を検証した。

今後は以下の方針で研究を進展させたい。

- (1) 計算時間の短縮：現状では計算に数十秒を要することがある。これを数秒に短縮するため、アルゴリズムの見直しを進めたい。
- (2) 工具配置の最適化手法の改善：今回提案した手法は、溝フィーチャで望ましい結果が得られない場合があった。このような問題を解消するために、工具配置の選択方針の見直しを進めたい。
- (3) システムの知能化：配置された工具とホルダ部品の干渉形状から、具体的にどのような設計変更をすべきかアドバイスするシステムへと発展させたい。

参考文献

- 1) M. Inui and T. Miyashita: Hollow Shape Extraction: Geometric Method for Assisting Process Planning of Mold Machining, Proc. 2003 IEEE ISATP 2003 (2003) 30.
- 2) S. Gupta, D. Das, W.C. Regli and D.S. Nau: Automated manufacturability analysis: A survey, Research in Engineering Design, 9, 3 (1997) 168.
- 3) S. Subramanyan and S. Lu: The impact of an AI-based design environment for simultaneous engineering on process planning, International Journal of Computer Integrated Manufacturing, 4, 2 (1991) 71.
- 4) M.R. Cutkosky and J.M. Tenenbaum: Toward a framework for concurrent design, International Journal of Systems Automation: Research and Applications, 1, 3 (1992) 239.
- 5) D. Hsiao: Feature Mapping and Manufacturability Evaluation with an Open Set Feature Modeler, Ph.D Thesis, Mechanical Engineering, Arizona State University (1991).
- 6) S.K. Gupta and D.S. Nau: Systematic approach to analyzing the manufacturability of machined parts, Computer-Aided Design, 27, 5 (1995) 323.
- 7) S. Miyakawa: Simultaneous engineering and producibility evaluation method, Proc. of the SME International Conference on Application of Manufacturing Technologies (1991).
- 8) B. K. Choi and R. B. Jerard: Sculptured Surface Machining, Theory and Applications, Kluwer Academic Publishers, Dordrecht, 1998.
- 9) M. Inui and A. Ohta: Using GPU to Accelerate Die and Mold Fabrication, IEEE CG&A Magazine, 27, 1 (2007) 82.
- 10) T. Saito, T. Takahashi: NC Machining with G-buffer Method, Computer Graphics, 25, 4 (1991) 207.