

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6548830号
(P6548830)

(45) 発行日 令和1年7月24日(2019.7.24)

(24) 登録日 令和1年7月5日(2019.7.5)

(51) Int.Cl. F I
G O 5 B 19/4093 (2006.01) G O 5 B 19/4093 H
G O 5 B 19/4103 (2006.01) G O 5 B 19/4103 A

請求項の数 10 (全 28 頁)

(21) 出願番号	特願2018-536908 (P2018-536908)	(73) 特許権者	000006013 三菱電機株式会社 東京都千代田区丸の内二丁目7番3号
(86) (22) 出願日	平成29年2月16日(2017.2.16)	(74) 代理人	100118762 弁理士 高村 順
(86) 国際出願番号	PCT/JP2017/005770	(72) 発明者	西野 慎哉 東京都千代田区丸の内二丁目7番3号 三 菱電機株式会社内
(87) 国際公開番号	W02018/042704	(72) 発明者	魚住 誠二 東京都千代田区丸の内二丁目7番3号 三 菱電機株式会社内
(87) 国際公開日	平成30年3月8日(2018.3.8)	(72) 発明者	津田 剛志 東京都千代田区丸の内二丁目7番3号 三 菱電機株式会社内
審査請求日	平成30年8月14日(2018.8.14)		
(31) 優先権主張番号	特願2016-172161 (P2016-172161)		
(32) 優先日	平成28年9月2日(2016.9.2)		
(33) 優先権主張国	日本国(JP)		

最終頁に続く

(54) 【発明の名称】 指令値生成装置

(57) 【特許請求の範囲】

【請求項1】

加工プログラムに基づいて工具経路を演算し、工具経路上の単位時間毎の補間点群である工具移動指令を生成する指令値生成装置において、

前記加工プログラムを複数の実行単位に分割して分割加工プログラムを生成する分割部と、

複数の演算装置を有し、前記実行単位毎に前記分割加工プログラムを並列して実行し、前記実行単位毎の工具経路上の補間点群である分割指令を生成する並列演算部と、

前記実行単位毎の前記分割指令から工具移動指令を生成する指令値生成部とを備え、

前記加工プログラムの開始から終了へ向かう方向を後、前記加工プログラムの終了から開始へ向かう方向を前とするとき、

前記分割部は、少なくとも1つの分割点前後の前記分割加工プログラムにおいて、前記分割点より前にある1つの前記分割加工プログラムの終了部分、及び前記分割点より後にある1つの前記分割加工プログラムの開始部分に対して、予め設定された加速度に基づいて指令速度に到達するまでに移動する加速距離を演算し、前記分割加工プログラムの前記分割点におけるオーバーラップ量が前記加速距離以上となるように演算し、前記分割加工プログラムの開始及び終了部分を前記オーバーラップ量だけ互いにオーバーラップさせることを特徴とする指令値生成装置。

【請求項2】

前記指令値生成部は、前記実行単位毎の前記分割指令からオーバーラップした部分を削除

することを特徴とする請求項 1 に記載の指令値生成装置。

【請求項 3】

前記分割部は、前記加工プログラムを前記並列演算部の演算装置の設置数以上の実行単位に分割することを特徴とする請求項 1 に記載の指令値生成装置。

【請求項 4】

前記分割部は、前記実行単位毎の前記分割指令の演算時間又は前記加工プログラムの前記実行単位毎の経路長を等しくすることを特徴とする請求項 1 に記載の指令値生成装置。

【請求項 5】

前記工具移動指令に基づいて被加工物の加工形状をシミュレーションする加工シミュレーション部を備えることを特徴とする請求項 1 に記載の指令値生成装置。

10

【請求項 6】

予め設定されたサーボモデルに基づいて、前記工具移動指令からサーボ応答を推定するサーボシミュレーション部を備えることを特徴とする請求項 1 に記載の指令値生成装置。

【請求項 7】

前記並列演算部は、マルチコア CPU、複数の CPU を備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されていることを特徴とする請求項 1 に記載の指令値生成装置。

【請求項 8】

前記指令値生成部は、前記分割指令が切り替わる箇所で生じた補間点間隔を修正するために、補間点の位置を分割指令の経路上で調整することを特徴とする請求項 1 に記載の指令値生成装置。

20

【請求項 9】

前記指令値生成部は、前記実行単位毎に並列して前記分割指令からオーバーラップした部分を削除することを特徴とする請求項 1 に記載の指令値生成装置。

【請求項 10】

前記並列演算部は、前記分割指令を細かい単位で複数回に分けて出力することを特徴とする請求項 1 に記載の指令値生成装置。

【発明の詳細な説明】

【技術分野】

【0001】

30

本発明は、複数の演算装置を有し、複数の演算装置を並列実行することで工具移動指令を生成する指令値生成装置に関する。

【背景技術】

【0002】

従来、工作機械を用いて被加工物を加工する場合には、CAM (Computer Aided Manufacturing) で加工プログラムを作成することが一般的である。加工プログラムには加工形状の情報、工具の送り速度及び工具の回転数が記述されている。指令値生成装置は、この加工プログラムを読み込み、座標変換、工具長補正、工具径補正及び機械誤差補正といった補正を行うことで工具経路の演算を行う。更に、指令値生成装置は、加減速をはじめとする処理を行い、単位時間毎の工具経路上の指令点である補間点の演算を行う。以下の説明では、単位時間毎の工具経路上の指令点を補間点と称呼し、一連の補間点群を工具移動指令と称呼する。多くの場合、指令値生成装置には数値制御装置 (NC: Numerical Control) が用いられる。加工形状が自由曲面を有する場合には、連続した複数の微小な線分で自由曲面を近似した加工プログラムを生成する。加工プログラムを生成する際には、自由曲面をできるだけ正確に表現するために加工プログラムの線分長を短くする必要があるため、加工プログラムの線分長は短くなる傾向にある。加工プログラムの線分長が短くなると、指令値生成装置で処理する線分数が増加するため、工具移動指令の生成に要する時間が増大する。この問題は、自由曲面を有する金型加工及び座標変換を伴う 5 軸加工において特に顕著である。

40

【0003】

50

そこで、複数の演算装置により並列して工具移動指令の演算を行うことで高速に工具移動指令を生成する方法が検討されている。従来技術の一例である特許文献1には、加工工程毎に加工プログラムを分割し、複数のプロセッサコアにより並列して工具移動指令の演算を行う技術が開示されている。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特表2014-522529号公報

【発明の概要】

【発明が解決しようとする課題】

10

【0005】

しかしながら、上記従来の技術によれば、加工が一旦中断する箇所では加工プログラムを分割することができず、複数の演算装置の処理量を均等に分散することができない。そのため、複数の演算装置の処理性能を活用して工具移動指令を高速に生成することができない、という問題があった。

【0006】

本発明は、上記に鑑みてなされたものであって、複数の演算装置を活用して、従来よりも高速に工具移動指令を生成する指令値生成装置を得ることを目的とする。

【課題を解決するための手段】

【0007】

20

上述した課題を解決し、目的を達成するために、本発明は、加工プログラムに基づいて工具経路を演算し、工具経路上の単位時間毎の補間点群である工具移動指令を生成する指令値生成装置において、前記加工プログラムを複数の実行単位に分割して分割加工プログラムを生成する分割部と、複数の演算装置を有し、前記実行単位毎に前記分割加工プログラムを並列して実行し、前記実行単位毎の工具経路上の補間点群である分割指令を生成する並列演算部と、前記実行単位毎の前記分割指令から工具移動指令を生成する指令値生成部とを備え、前記加工プログラムの開始から終了へ向かう方向を後、前記加工プログラムの終了から開始へ向かう方向を前とするとき、前記分割部は、少なくとも1つの分割点前後の前記分割加工プログラムにおいて、前記分割点より前にある1つの前記分割加工プログラムの終了部分、及び前記分割点より後にある1つの前記分割加工プログラムの開始部分

30

に対して、予め設定された加速度に基づいて指令速度に到達するまでに移動する加速距離を演算し、前記分割加工プログラムの前記分割点におけるオーバーラップ量が前記加速距離以上となるように演算し、前記分割加工プログラムの開始及び終了部分を前記オーバーラップ量だけ互いにオーバーラップさせることを特徴とする。

【発明の効果】

【0008】

本発明によれば、複数の演算装置を活用して、従来よりも高速に工具移動指令を生成することができるという効果を奏する。

【図面の簡単な説明】

【0009】

40

【図1】実施の形態1に係る指令値生成装置の構成を示す図

【図2】実施の形態1において、図1に示す指令値生成装置におけるデータフローを示す第1図

【図3】実施の形態1において、図1に示す指令値生成装置におけるデータフローを示す第2図

【図4】実施の形態1において、図1に示す分割部の処理フローを示す図

【図5】実施の形態1において、時間を横軸とし、速度を縦軸とした、オーバーラップさせない場合の実行単位1及び実行単位2の分割点の前後における挙動を示す図

【図6】実施の形態1において、時間を横軸とし、速度を縦軸とした、オーバーラップさせる場合の実行単位1及び実行単位2の分割点の前後における挙動を示す図

50

【図 7】実施の形態 1 において、時間を横軸とし、速度を縦軸とした、加速距離 L_a を示す図

【図 8】実施の形態 1 において、時間を横軸とし、速度を縦軸とした、オーバーラップ量 L_o を示す図

【図 9】実施の形態 1 において、加工プログラムにおける工具の移動経路の一例を示す図

【図 10】実施の形態 1 において、オーバーラップ量を考慮した、加工プログラムにおける工具の移動経路の一例を示す第 1 図

【図 11】実施の形態 1 において、オーバーラップ量を考慮した、加工プログラムにおける工具の移動経路の一例を示す第 2 図

【図 12】実施の形態 1 において、加工プログラムの一例を示す図

10

【図 13】実施の形態 1 において、分割加工プログラム 1 を示す図

【図 14】実施の形態 1 において、加工モードに関する指令を追記した分割加工プログラム 2 を示す図

【図 15】実施の形態 1 において、加工モードに関する指令を追記した分割加工プログラム 3 を示す図

【図 16】実施の形態 1 において、内回りが生じる、加工プログラムにおける工具の移動経路の一例を示す第 1 図

【図 17】実施の形態 1 において、内回りが生じる、加工プログラムにおける工具の移動経路の一例を示す第 2 図

【図 18】実施の形態 1 において、図 1 に示す並列演算部における分割加工プログラムの実行タイミングと、図 1 に示す指令値生成部における一連の工具移動指令の生成タイミングとを示すタイミングチャート

20

【図 19】実施の形態 1 において、分割加工プログラム 1 及び分割加工プログラム 2 と、分割指令 1 及び分割指令 2 の一例を示す図

【図 20】実施の形態 1 において、分割指令が切り替わる点における補間点の間隔の一例を示す図

【図 21】実施の形態 1 において、分割指令が切り替わる点において調整された補間点の間隔の一例を示す図

【図 22】実施の形態 1 において、加工プログラム形状の一例を示す第 1 図

【図 23】実施の形態 1 において、図 22 の加工プログラム形状の実行時について、時間を横軸とし、速度を縦軸とし、X 軸速度及び Y 軸速度を示す図

30

【図 24】実施の形態 1 において、図 22 の加工プログラム形状の実行時について、生成される工具移動指令の一例を示す図

【図 25】実施の形態 1 において、図 22 の加工プログラム形状の実行時における速度波形と、工具経路とを示す図

【図 26】実施の形態 1 において、図 22 の加工プログラム形状の実行時における工具移動指令の一例を示す図

【図 27】実施の形態 1 において、加工プログラム形状の一例を示す第 2 図

【図 28】実施の形態 1 において、図 27 の加工プログラム形状の実行時について、時間を横軸とし、速度を縦軸とし、X 軸速度及び Y 軸速度を示す図

40

【図 29】実施の形態 1 において、図 27 の加工プログラム形状の実行時について、生成される工具移動指令の一例を示す図

【図 30】実施の形態 1 において、図 27 の加工プログラム形状の実行時における速度波形と、工具経路とを示す図

【図 31】実施の形態 1 において、図 27 の加工プログラム形状の実行時における工具移動指令の一例を示す図

【図 32】実施の形態 2 に係る指令値生成装置の構成を示す図

【図 33】実施の形態 2 における図 32 に示す並列演算部における分割加工プログラムの実行タイミングと、図 32 に示す指令値生成部における一連の工具移動指令の生成タイミングとを示すタイミングチャート

50

【図 3 4】実施の形態 3 に係る指令値生成装置の構成を示す図

【図 3 5】実施の形態 1 における図 1 に示す演算装置における分割加工プログラムの実行タイミングと、図 1 に示す指令値生成部における一連の工具移動指令の生成タイミングとを示すタイミングチャート

【図 3 6】実施の形態 3 における図 3 4 に示す演算装置における分割加工プログラムの実行タイミングと、図 3 4 に示す指令値生成部における一連の工具移動指令の生成タイミングとを示すタイミングチャート

【図 3 7】実施の形態 3 における図 3 4 に示す指令値生成装置におけるデータフローを示す第 1 図

【図 3 8】実施の形態 3 における図 3 4 に示す指令値生成装置におけるデータフローを示す第 2 図

10

【発明を実施するための形態】

【0010】

以下に、本発明の実施の形態に係る指令値生成装置を図面に基づいて詳細に説明する。なお、この実施の形態によりこの発明が限定されるものではない。

【0011】

実施の形態 1 .

図 1 は、本発明の実施の形態 1 に係る指令値生成装置の構成を示す図である。図 1 に示す指令値生成装置 100 は、図示しない NC 工作機械を制御する数値制御装置である。図 1 に示す指令値生成装置 100 は、入力された加工プログラム 10 に従って、工具経路上の単位時間毎の補間点群である工具移動指令 11 を出力することで、NC 工作機械の被加工物に対する工具の移動を数値制御する。

20

【0012】

指令値生成装置 100 は、分割部 110、並列演算部 120 及び指令値生成部 130 を備える。分割部 110 は、加工プログラム 10 を実行単位に分割して分割加工プログラムを生成する。並列演算部 120 は、実行単位毎の加工プログラムに従って実行単位毎の工具移動指令である分割指令を生成する。指令値生成部 130 は、分割指令から一連の工具移動指令 11 を生成する。

【0013】

ここで、並列演算部 120 には演算装置 121、演算装置 122 及び演算装置 123 が設けられている。並列演算部 120 は、マルチコア CPU (Central Processing Unit)、複数の CPU を備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されている。なお、通信回線には、ローカルエリアネットワーク及びインターネットを例示することができる。また、図 1 において、並列演算部 120 は 3 つの演算装置を備えるが、本発明はこれに限定されず、並列演算部 120 が備える演算装置の設置数は 1 つ以上であればよい。並列演算部 120 が備える演算装置の設置数を多くすると、工具移動指令 11 を高速に生成することができるため、高速化の目標値に応じて演算装置の設置数を決めればよい。

30

【0014】

分割部 110 は、入力された加工プログラム 10 を複数の実行単位に分割し、実行単位毎に分割加工プログラムを並列演算部 120 に出力する。

40

【0015】

並列演算部 120 は、実行単位毎に分割加工プログラムを演算装置 121、122、123 に割り振る。演算装置 121、演算装置 122 及び演算装置 123 は、加減速及び座標変換をはじめとする処理を行い、実行単位毎の工具の移動指令である分割指令を順次生成して指令値生成部 130 に出力する。ここで、演算装置 121、演算装置 122 及び演算装置 123 は、並列実行されており、1 つの実行単位における分割指令の生成が完了すると、次の実行単位における分割指令の生成を開始し、未処理の実行単位がなくなるまで処理を繰り返す。

【0016】

50

指令値生成部 130 は、実行単位毎の分割指令から一連の工具移動指令 11 を生成して出力する。なお、指令値生成部 130 は、単一の CPU、単一の CPU を備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されている。又は、指令値生成部 130 は、並列演算部 120 と同様に、マルチコア CPU、複数の CPU を備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されていてもよい。

【0017】

次に、並列演算の実行順序について更に詳細に説明する。図 2 は、実施の形態 1 において、図 1 に示す指令値生成装置 100 におけるデータフローを示す第 1 図である。図 2 は、並列演算部 120 の初回実行時のデータフローを示している。

10

【0018】

図 3 は、実施の形態 1 において、図 1 に示す指令値生成装置 100 におけるデータフローを示す第 2 図である。図 3 は、並列演算部 120 の初回実行が完了し、次の実行単位における加工プログラムの実行時のデータフローを示している。なお、図 2 及び図 3 では、分割部 110 は、加工プログラム 10 を 6 個の実行単位、すなわち実行単位 1、実行単位 2、実行単位 3、実行単位 4、実行単位 5 及び実行単位 6 に分割しているが、本発明はこれに限定されず、分割数は適宜設定することができる。

【0019】

図 2 に示すように、分割部 110 が加工プログラム 10 を実行単位 1、実行単位 2、実行単位 3、実行単位 4、実行単位 5 及び実行単位 6 に分割した場合、並列演算部 120 は 3 個の演算装置を備えるため、並列演算部 120 は、実行単位 1、実行単位 2、実行単位 3 を並列実行して分割指令 1、分割指令 2 及び分割指令 3 を生成する。そして、指令値生成部 130 は、分割指令 1、分割指令 2 及び分割指令 3 から実行単位 1、実行単位 2 及び実行単位 3 に対する工具移動指令を生成する。

20

【0020】

図 3 に示すように、並列演算部 120 が実行単位 1 から実行単位 3 の実行を完了すると、並列演算部 120 は、次の実行単位である実行単位 4 から実行単位 6 を実行し、分割指令 4 から分割指令 6 を生成する。そして、指令値生成部 130 は、実行単位 4 から 6 に対する工具移動指令を生成し、実行単位 1 から実行単位 3 に対する工具移動指令と併せて工具移動指令 11 を生成して出力する。

30

【0021】

なお、未実行の実行単位がある場合には、未実行の実行単位がなくなるまで同様の処理を繰り返す。また、ここで、分割指令 1 は実行単位 1 を実行するときの分割指令であり、これは、分割指令 2, 3, 4, 5, 6 についても同様である。

【0022】

次に、各構成要素の処理内容について詳細に説明する。図 4 は、実施の形態 1 において、図 1 に示す分割部 110 の処理フローを示す図である。まず、分割部 110 は、処理をスタートし、分割数 N を決定する (S1)。ここで、分割数 N は、並列演算部 120 が備える演算装置の設置数以上とするが、加工プログラムの行数が少ない場合には分割数 N を演算装置の設置数未満としてもよい。実施の形態 1 では、一例として分割数 $N = 6$ としている。

40

【0023】

次に、分割部 110 は、加工プログラム 10 を分割する分割点を決定する (S2)。加工プログラム 10 を N 個の実行単位に分割する場合には、全ての実行単位の並列演算部 120 での演算時間を可能な限り均一にする。加工プログラム 10 の 1 ブロックあたりの演算時間 $T(i)$ はブロックの長さ $L(i)$ を指令速度 $F(i)$ で割ることにより求められる。従って、加工プログラム 10 の全体の演算時間 T は下記の式 (1) に示すように、1 ブロックあたりの演算時間 $T(i)$ の全ブロックの総和で表わされる。

【0024】

【数 1】

$$T = \sum_{i=1}^M T(i) = \sum_{i=1}^M \left(\frac{L(i)}{F(i)} \right) \quad \dots(1)$$

【0025】

ここで、上記の式(1)中のMは、加工プログラム10のブロック数、すなわち加工プログラム10の行数である。つまり、実行単位1つあたりの演算時間が可能な限り均一となるようにT/Nとなる点を分割点とする。ここで、分割点は加工プログラムのブロック終点と一致しないことが多い。しかしながら、分割点は必ずしもブロック終点に一致する必要はないため、上記の式(1)で演算した点を分割点としてもよいし、上記の式(1)で演算した点に最近接のブロック終点を分割点としてもよい。

10

【0026】

なお、上記の式(1)では、1ブロックあたりの演算時間T(i)は、ブロックの長さL(i)と指令速度F(i)とによって演算されているが、更にパラメータで設定されている機械の許容加速度を用いて加減速及びクランプ速度の影響も考慮して演算時間T(i)を演算することで、分割点を決定してもよい。また、実行単位1つあたりの経路長が可能な限り均一となるように全ブロックの長さの総和Lを分割数Nで除算した長さとなる点を分割点としてもよい。

【0027】

次に、分割部110は、実行単位に分割する際のオーバーラップ量Loを計算する(S3)。(図5は、実施の形態1において、時間を横軸とし、速度を縦軸とした、オーバーラップさせない場合の実行単位1及び実行単位2の分割点の前後における挙動を示す図である。図5に示すように、分割点では速度ゼロまで減速するため、分割点の前後において、分割した場合の工具移動指令の経路と分割しなかった場合の工具移動指令の経路とが異なってしまう。図6は、実施の形態1において、時間を横軸とし、速度を縦軸とした、オーバーラップさせる場合の実行単位1及び実行単位2の分割点の前後における挙動を示す図である。そこで、分割部110は、図6に示すように、分割単位をオーバーラップさせることで、分割点における減速をなくし、分割した場合と分割しなかった場合とで、同一の工具移動指令を生成する。そのため、オーバーラップ量Loは、速度ゼロから指令速度Fに到達するまでに移動する距離、すなわち加速距離La以上とすればよい。

20

30

【0028】

図7は、実施の形態1において、時間を横軸とし、速度を縦軸とした、加速距離Laを示す図である。図8は、実施の形態1において、時間を横軸とし、速度を縦軸とした、オーバーラップ量Loを示す図である。図7及び図8において、加速度Aで指令速度Fまで加速し、更に時定数Tsの移動平均フィルタにより滑らかな速度波形を生成する場合には、加速距離Laは、時刻0から時刻(F/A + Ts)までに移動する距離である。図8に示すように、オーバーラップ量Loは、加速に必要な加速距離La以上にするため、加速度Aで指令速度Fまで加速した後に、指令速度Fで移動平均フィルタ時定数Ts分だけ移動したときの距離とする。すなわち、オーバーラップ量Loは、図8に示す台形の面積であり、下記の式(2)により演算することができる。この台形の上辺の長さはTsであり、下辺の長さは(F/A + Ts)であり、高さはFである。

40

【0029】

【数 2】

$$L_o = \frac{F}{2} \left(\frac{F}{A} + 2Ts \right) \quad \dots(2)$$

【0030】

次に、分割部110は、S2で決定した分割点で加工プログラム10を実行単位に分割する(S4)。ここで、加工プログラム10の分割の際に、S3で演算したオーバーラップ量Loだけオーバーラップさせる。

50

【 0 0 3 1 】

図 9 は、実施の形態 1 において、加工プログラム 1 0 における工具の移動経路の一例を示す図である。図 9 においては、分割点 1 で 2 つの実行単位、実行単位 1 及び実行単位 2 に分割されている。

【 0 0 3 2 】

図 1 0 は、実施の形態 1 において、オーバーラップ量 L_o を考慮した、加工プログラムにおける工具の移動経路の一例を示す第 1 図である。図 1 0 に示すように、実行単位 1 の始点は「加工プログラムの始点」であるが、実行単位 1 の終点は「分割点 1 からオーバーラップ量 L_o だけ終点側に移動した点」である。

【 0 0 3 3 】

図 1 1 は、実施の形態 1 において、オーバーラップ量 L_o を考慮した、加工プログラムにおける工具の移動経路の一例を示す第 2 図である。図 1 1 に示すように、実行単位 2 の始点は「分割点 1 からオーバーラップ量 L_o だけ始点側に移動した点」であり、実行単位 2 の終点は「加工プログラムの終点」である。

【 0 0 3 4 】

なお、図 9 から図 1 1 では、説明を簡単にするために、分割点はブロック終点と一致させているが、分割点は、必ずしもブロック終点と一致する必要はなく、ブロックの途中であってもよい。

【 0 0 3 5 】

また、実行単位の始点及び終点もブロックの途中であってもよいし、前述の方法で求めた始点又は終点に最近接のブロック終点を改めて実行単位の始点又は終点としてもよい。

【 0 0 3 6 】

また、図 9 から図 1 1 では、分割点は 1 点のみ記載されているが、分割点が複数存在する場合にも同様の方法で分割することができる。

【 0 0 3 7 】

ここで、実行単位への具体的な分割方法について説明する。実施の形態 1 においては、加工プログラム 1 0 を分割して新たに分割加工プログラムを生成することで実行単位に分割する。ただし、必ずしも分割加工プログラムを生成することを要しない。一例として、加工プログラム 1 0 のブロック番号である加工プログラムの 1 行毎のシリアル番号を指定することで実行単位に分割してもよいし、加工プログラムの分割点に分割の目印となる指令である G コード又は M コードを挿入してもよい。

【 0 0 3 8 】

次に、分割部 1 1 0 は、分割加工プログラムの先頭に加工モード指令を追記する (S 5) 。

【 0 0 3 9 】

図 1 2 は、実施の形態 1 において、加工プログラムの一例を示す図である。図 1 2 に示す加工プログラムには、加工を行うための座標値が 1 ブロック、すなわち 1 行毎に記載されている。その他、指令速度である F 指令、早送りである G 0、切削モードである G 1 及び工具長補正である G 4 3 といった加工モードに関する指令も加工プログラムに記載されている。図 1 2 に示す加工プログラムでは、加工プログラムの先頭で加工モードに関する指令が記述されており、その後、加工を行うための座標値が記述されている。この座標値群を座標値群 1 と称呼する。更にその後、再度加工モードに関する指令があり、加工を行うための座標値が記述されている。この座標値群を座標値群 2 と称呼する。

【 0 0 4 0 】

ここで、座標値群 1 の途中に分割点 1 が存在し、座標値群 2 の途中に分割点 2 が存在する場合の分割加工プログラムについて説明する。この場合には、3 つの分割加工プログラム、すなわち加工プログラムの先頭から分割点 1 までの分割加工プログラム 1 と、分割点 1 から分割点 2 までの分割加工プログラム 2 と、分割点 2 から加工プログラム末尾までの分割加工プログラム 3 とが生成される。ただし、各分割加工プログラムにはオーバーラップを付加する。

10

20

30

40

50

【 0 0 4 1 】

図 1 3 は、実施の形態 1 において、分割加工プログラム 1 を示す図である。図 1 3 に示す分割加工プログラム 1 は、分割点 1 にオーバーラップが付加されて分割されたものである。ここで、分割前の加工プログラムの先頭で加工モードに関する指令が記述されているため、分割加工プログラム 1 には加工モードに関する指令を追記する必要はない。

【 0 0 4 2 】

図 1 4 は、実施の形態 1 において、加工モードに関する指令を追記した分割加工プログラム 2 を示す図である。図 1 4 に示す分割加工プログラム 2 は、分割加工プログラム 2 の先頭が加工を行うための座標値であり、加工モードに関する指令が記述されていないので、分割加工プログラム 2 の先頭に必要な加工モードに関する指令が追記されている。ここで、分割加工プログラム 2 に追記される加工モードに関する指令は、分割前の加工プログラムにおいて、加工プログラムの先頭から分割加工プログラム 2 の先頭までの間に指令された加工モードに関する指令である。

10

【 0 0 4 3 】

図 1 5 は、実施の形態 1 において、加工モードに関する指令を追記した分割加工プログラム 3 を示す図である。図 1 5 に示すように、分割加工プログラム 3 の先頭にも、分割加工プログラム 2 と同様に、加工モードに関する指令が追記される。ただし、同一種類の加工モードに関する指令が複数回指令された場合には、最後に指令された指令のみが追記されていけばよい。一例として、図 1 2 における速度指令、すなわち F 指令では、F 3 0 0 0 mm / m i n の指令がされた後で、F 5 0 0 0 mm / m i n の指令があり、その後、分割加工プログラム 3 の先頭である分割点 2 からオーバーラップ分遡った箇所が存在する。そのため、分割加工プログラム 3 の先頭以前に指令された最後の速度指令である F 5 0 0 0 mm / m i n のみ指令すればよい。

20

【 0 0 4 4 】

このような加工モードに関する指令の追記は、他にも絶対値指令と相対値指令、工具長補正及び工具径補正といった、一度指令するとそれ以降のブロックでも有効となる指令、すなわちモーダル指令の全てに対して行われる。

【 0 0 4 5 】

実施の形態 1 においては分割数 $N = 6$ であるので、同様に分割加工プログラム 4 から分割加工プログラム 6 を生成し、生成された分割加工プログラムを並列演算部 1 2 0 に出力する。

30

【 0 0 4 6 】

次に、複数の演算装置を備える並列演算部 1 2 0 の処理内容について説明する。上記したように、図 1 に示す並列演算部 1 2 0 は、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 を備える。なお、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 は、具体的には、プロセッサコア、CPU、ネットワークによって接続されたパーソナルコンピュータ若しくはサーバ又はクラウド上の仮想コンピュータを例示することができる。

【 0 0 4 7 】

並列演算部 1 2 0 は、分割部 1 1 0 から分割数 N で分割加工プログラムが出力されると、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 の各々に分割加工プログラムを割り振る。図 1 に示す並列演算部 1 2 0 は、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 を備えるため、まず、演算装置 1 2 1 には分割加工プログラム 1 が割り振られ、演算装置 1 2 2 には分割加工プログラム 2 が割り振られ、演算装置 1 2 3 には分割加工プログラム 3 が割り振られる。

40

【 0 0 4 8 】

演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 の各々では、数値制御装置と同一の処理を並列して実行し、分割加工プログラムから工具の指令経路である分割指令 1、分割指令 2 及び分割指令 3 を生成する。ここで、具体的な処理には、予め設定された加速度で加減速するための速度波形を生成する加減速処理、加減速処理により生成した速度波形を滑らかにするスムージング処理及びスムージング処理後速度で移動したときの単位時間

50

毎の工具位置である補間点を演算する補間処理を例示することができる。なお、スムージング処理は、移動平均フィルタ処理とも称される。ここでは、単位時間毎の工具位置の点の各々を補間点と称し、分割加工プログラムにおける補間点全体を分割指令と称する。

【 0 0 4 9 】

このようにして滑らかに生成された分割指令の経路は、一般的に分割加工プログラムに対して内回りが生じる。なお、内回りが生じる理由は後述する。図 1 6 は、実施の形態 1 において、内回りが生じる、加工プログラムにおける工具の移動経路の一例を示す第 1 図である。図 1 7 は、実施の形態 1 において、内回りが生じる、加工プログラムにおける工具の移動経路の一例を示す第 2 図である。そのため、図 1 0 , 1 1 で示す分割加工プログラムを実行した場合の分割指令は、図 1 6 , 1 7 に示すように分割加工プログラムの始点及び終点を通り、始点と終点との間は、分割加工プログラムの形状に対して内回りが生じた形状となる。

10

【 0 0 5 0 】

なお、始点又は終点の近くでは、加工プログラムを分割したことによる分割指令の内回り量は小さくなる。この内回り量が小さくなる領域を内回り減少領域と称する。ここで、内回り量は指令速度によって決まり、内回り減少領域の長さは図 7 に示す加速距離 L_a となる。

【 0 0 5 1 】

なお、分割部 1 1 0 は、加速距離 L_a 以上のオーバーラップ量 L_o を付加して分割している。そのため、分割加工プログラムの各々における内回り減少領域が重なることはなく、加工プログラム形状の全域において、内回り減少領域でない補間点を少なくとも 1 点は得ることができる。なお、図 1 6 , 1 7 は、説明のために加工プログラムを 2 つに分割した場合を示しているが、更に多くに分割した場合も同様である。

20

【 0 0 5 2 】

そして、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 の各々は、分割加工プログラム 1、分割加工プログラム 2 及び分割加工プログラム 3 を実行し、並列演算部 1 2 0 は、分割指令 1、分割指令 2 及び分割指令 3 を生成して、分割指令 1、分割指令 2 及び分割指令 3 を指令値生成部 1 3 0 に出力する。そして、同様に、並列演算部 1 2 0 は、分割加工プログラム 4、分割加工プログラム 5 及び分割加工プログラム 6 を順次実行し、分割指令 4、分割指令 5 及び分割指令 6 を指令値生成部 1 3 0 に出力する。この処理を未実行の分割加工プログラムがなくなるまで繰り返す。

30

【 0 0 5 3 】

次に、指令値生成部 1 3 0 の処理内容について説明する。指令値生成部 1 3 0 は、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 の各々から出力される分割指令に基づいて、一連の工具移動指令 1 1 を生成する。

【 0 0 5 4 】

分割部 1 1 0 では分割加工プログラムの各々をオーバーラップさせているため、分割指令の各々もオーバーラップしている。実施の形態 1 においては、分割指令のオーバーラップしている領域を後述するように除去し、その後、結合することで一連の工具移動指令 1 1 を生成する。

40

【 0 0 5 5 】

図 1 8 は、実施の形態 1 において図 1 に示す並列演算部 1 2 0 における分割加工プログラムの実行タイミングと、実施の形態 1 において図 1 に示す指令値生成部 1 3 0 における一連の工具移動指令 1 1 の生成タイミングとを示すタイミングチャートである。

【 0 0 5 6 】

上記したように、並列演算部 1 2 0 は、演算装置 1 2 1 で分割加工プログラム 1 を実行し、演算装置 1 2 2 で分割加工プログラム 2 を実行し、演算装置 1 2 3 で分割加工プログラム 3 を実行しつつ、分割指令 1、分割指令 2 及び分割指令 3 の生成及び出力を開始する。厳密には、分割加工プログラムの実行中には分割指令は完成していないため、分割指令

50

を構成する補間点である単位時間あたりの工具の位置を示す点を順次出力する。

【0057】

指令値生成部130は、並列演算部120から分割指令1、分割指令2及び分割指令3が入力されると、分割指令1を出力する。このとき、分割指令2及び分割指令3は、指令値生成装置100内のいずれかの保存領域に保存される。指令値生成部130は、分割指令1の出力が完了すると、後述する、オーバーラップしている領域の除去及び分割指令が切り替わる点で生じる急激な補間点間隔変化の調整を行い、分割指令2を出力する。この時、演算装置121では分割加工プログラム1の実行が完了しているため、分割加工プログラム4の実行を開始し、分割指令4を指令値生成部130に出力する。また、並列演算部120は、演算装置122における分割加工プログラム2の実行が完了次第、分割加工プログラム5の実行を開始し、分割指令5を指令値生成部130に出力する。また、並列演算部120は、演算装置123における分割加工プログラム3の実行が完了次第、分割加工プログラム6の実行を開始し、分割指令6を指令値生成部130に出力する。図18では、分割加工プログラム1、分割加工プログラム2及び分割加工プログラム3の実行が同時に完了した状態としているが、本発明はこれに限定されるものではない。

10

【0058】

次に、各分割指令のオーバーラップしている領域の除去について、図19を参照して説明する。図19は、実施の形態1において、分割加工プログラム1及び分割加工プログラム2と、分割指令1及び分割指令2の一例を示す図である。図19に示すように、分割加工プログラム1と分割加工プログラム2とは分割点1で分割され、分割加工プログラム1及び分割加工プログラム2は、分割点1を中心にオーバーラップ量 L_0 だけオーバーラップしている。

20

【0059】

指令値生成部130は、並列演算部120からの分割指令1が入力されると、まず分割指令1を工具移動指令11として出力するが、分割加工プログラム1の分割点1に相当する補間点まで出力を完了すると、分割指令2の出力を開始する。すなわち、分割指令1に対しては、分割加工プログラム1の分割点1から終点に相当する補間点は出力しない。

【0060】

また、分割指令2については、分割点1に相当する補間点から出力が開始される。すなわち、分割指令2に対しては、分割加工プログラム2の始点から分割点1に相当する補間点は出力しない。分割点1に相当する補間点から順次出力し、分割点2に相当する補間点まで出力を完了すると、分割指令3を出力し始める。なお、分割点2は、分割加工プログラム2と分割加工プログラム3とを分割する点である。すなわち、分割指令2に対しては、分割加工プログラム2の分割点2から終点に相当する補間点も出力しない。

30

【0061】

また、図示しない分割指令3については、分割点2に相当する補間点から出力が開始される。すなわち、分割指令3に対しては、分割加工プログラム3の始点から分割点2に相当する補間点は出力しない。

【0062】

このような処理を順次繰り返し、分割指令6まで出力される。なお、図示していないが分割指令4については、始点及び終点の双方がオーバーラップしているので、分割指令4に対しては、分割加工プログラム4の始点から分割点3に相当する補間点は出力されず、分割加工プログラム4の分割点4から終点に相当する補間点も出力されない。分割指令5についても同様であるが、分割指令6については、終点側がオーバーラップしていないため、終点まで出力される。

40

【0063】

次に、各分割点に相当する補間点の求め方について説明する。数値制御装置では、滑らかな経路及び速度波形を生成するため、図19に示されるように、各分割指令の経路は各分割加工プログラムの経路を内回る形状となる。すなわち、分割指令の経路の長さ、分割加工プログラムの経路の長さとは異なる。

50

【 0 0 6 4 】

移動平均フィルタにより経路を滑らかにする場合、移動平均フィルタ後経路である工具移動指令の経路上の補間点 $x(n)$ は、移動平均フィルタ前経路である加工プログラムの経路上の点 X の平均値で表わされるため、下記の式 (3) で表わされる。

【 0 0 6 5 】

【数 3】

$$x(n) = \frac{X(n-m+1) + X(n-m+2) + \dots + X(n)}{m} \quad \dots(3)$$

【 0 0 6 6 】

10

ここで、上記の式 (3) の n は、始点から終点までの補間点の番号である。また、上記の式 (3) の m は、移動平均フィルタの時定数であり、パラメータにより設定される。上記の式 (3) から移動平均フィルタ前経路上の点 X を求めると、下記の式 (4) のようになる。

【 0 0 6 7 】

【数 4】

$$X(n) = m(x(n) - x(n-1)) + X(m-n) \quad \dots(4)$$

【 0 0 6 8 】

上記の式 (4) を用いると、移動平均フィルタ後経路上の補間点 x から移動平均フィルタ前経路上の点 X を演算することができる。これにより、移動平均フィルタ前経路における距離を演算することができる。移動平均フィルタ前経路における始点又は終点からの距離がオーバーラップ量 L_0 となる補間点 x を、分割点に相当する補間点と定義する。

20

【 0 0 6 9 】

なお、実施の形態 1 では、移動平均フィルタ後経路上の補間点から移動平均フィルタ前経路での距離を演算することで、分割点に相当する補間点を演算しているが、本発明はこれに限定されるものではなく、演算装置 1 2 1、演算装置 1 2 2 及び演算装置 1 2 3 の各々が、分割点に相当する補間点に適切なフラグを付けて出力する構成としてもよい。

【 0 0 7 0 】

なお、一般的に、分割点に相当する補間点位置、すなわち分割指令が切り替わる点では補間点の間隔が急峻に変化してしまう。図 2 0 は、実施の形態 1 において、分割指令が切り替わる点における補間点の間隔の一例を示す図である。図 2 0 に示すように、分割指令 1 から分割指令 2 に切り替わる補間点位置における補間点の間隔は、他の補間点の間隔とは異なる。補間点は単位時間あたりの工具位置を示しており、補間点の間隔は工具速度を示している。すなわち、補間点の間隔が変化すると工具速度が変化してしまい、加工不良の原因となる。

30

【 0 0 7 1 】

図 2 1 は、実施の形態 1 において、分割指令が切り替わる点において調整された補間点の間隔の一例を示す図である。指令値生成部 1 3 0 は、図 2 1 に示すように、分割指令が切り替わる点においても補間点の間隔が変化しないように補間点の位置を分割指令の経路上で調整する。

40

【 0 0 7 2 】

図 2 0 に示すように、指令値生成部 1 3 0 が出力する分割指令 1 の最後の補間点、すなわち、オーバーラップした領域でない最後の補間点を補間点 1 E、指令値生成部 1 3 0 が出力する分割指令 2 の最初の補間点、すなわち、オーバーラップした領域でない最初の補間点を補間点 2 S とする。指令値生成部 1 3 0 は、補間点 1 E と補間点 2 S との間隔が補間点 1 E とその前の補間点との間隔 L_1 と等しくなるように、補間点 2 S の位置を分割指令 2 の経路に沿って調整する。又は、指令値生成部 1 3 0 は、補間点 1 E と補間点 2 S との間隔が、補間点 1 E と、指令値生成部 1 3 0 から出力されない次の補間点との間隔と等しくなるように補間点 2 S の位置を分割指令 2 の経路に沿って調整してもよい。また、指令値

50

生成部 130 は、調整後の補間点 2S とその次の補間点との間隔が、調整前の補間点 2S とその次の補間点との間隔 L2 と等しくなるように補間点 2S の次の補間点の位置を分割指令 2 の経路に沿って調整する。この処理を順次繰り返すことで、工具移動指令全域において、工具速度の変化のない指令を生成することができるので、加工不良を抑えることができる。

【0073】

なお、前述の方法で補間点の位置を調整する場合、分割指令 2 以降の補間点は全て調整を要する。そこで、許容速度の変化を予め設定しておき、許容速度変化以下の速度変化を許容して補間点の位置を調整することで、徐々に補間点位置の調整を小さくしてもよい。このようにすることで、部分的な調整のみで分割指令の切り替わり点における速度変化を抑制することが可能である。

10

【0074】

以上説明したように、オーバーラップした領域を除いて工具移動指令を出力することで、加工プログラムを分割しなかった場合と同一の工具移動指令を出力することが可能である。

【0075】

実施の形態 1 で説明した処理を行うことで、分割に起因する工具移動指令の変化が抑制されるため、あらゆる位置で加工プログラムを分割することが可能となる。これにより、演算装置の計算リソースを効率よく使用することが可能である。なお、実施の形態 1 は、クラウドをはじめとする演算装置を多数使用できる環境で特に効果大きい。

20

【0076】

実施の形態 1 によれば、複数の演算装置を用いて先行して分割指令を生成することができる。従って、線分長が短く処理負荷が重い箇所も先行して分割指令を生成することができるため、適切な演算周期で工具移動指令を演算でき、不要な減速を防止することができる。

【0077】

また、実施の形態 1 では簡単のために分割プログラム数を 6 個としたが、更に細かく分割することで、複数の演算装置の各々の処理負荷をより分散して不要な減速を防止することが可能である。

【0078】

ここで、比較例である従来技術の構成と実施の形態 1 の構成とを対比して説明する。この比較例によっても、工具移動指令の演算を並列実行することで工具移動指令を生成することは可能である。

30

【0079】

指令値生成装置 100 は、加工プログラムの複数の線分を先読みし、工作機械が振動しないように経路、速度及び加速度を滑らかにした工具移動指令を生成している。図 22 は、実施の形態 1 において、加工プログラム形状の一例を示す第 1 図である。加工プログラム形状が図 22 に示すように、XY 平面における X 軸のみの移動から Y 軸のみの移動となる 90° のコーナである場合について説明する。

【0080】

図 23 は、実施の形態 1 において、図 22 の加工プログラム形状の実行時について、時間を横軸とし、速度を縦軸とし、X 軸速度及び Y 軸速度を示す図である。指令値生成装置 100 は、加工プログラムの線分を先読みすることで、図 23 に示すようにコーナ部において X 軸速度と Y 軸速度とが重なった滑らかな X 軸速度と Y 軸速度とを演算する。ここで、図 23 においては加速度までを滑らかにした速度波形を示しているが、本発明はこれに限定されず、速度までを滑らかにした各軸速度としてもよいし、又は加加速度までを滑らかにした各軸速度としてもよい。

40

【0081】

図 24 は、実施の形態 1 において、図 22 の加工プログラム形状の実行時について、生成される工具移動指令の一例を示す図である。指令値生成装置 100 の先読みによってコ

50

ーナ部においてX軸速度とY軸速度とが重なり、X軸及びY軸の両方が動作するため、図24に示すように生成される工具移動指令はコーナ部が滑らかな形状となる。

【0082】

他方で、図22に示すXY平面における90°のコーナを有する加工プログラムをコーナ部で2つに分割し、分割した加工プログラムについて各々異なる演算装置で工具移動指令を生成した場合について説明する。この場合には、X軸のみの移動と、Y軸のみの移動とを各々異なる演算装置で演算することになる。そのため、X軸速度とY軸速度とは異なる演算装置で演算され、X軸速度とY軸速度とが重なる部分はなく、分割後の加工プログラム形状と工具経路である工具移動指令とは、図25に示すように同一形状となる。図25は、実施の形態1において、図22の加工プログラム形状の実行時における速度波形と、工具経路とを示す図である。これは、本来ならば分割した点の前後の形状が互いに影響し合った滑らかな経路を生成するはずであるところ、加工プログラムを分割することで、分割した加工プログラムの各々が別々に実行され、分割した点を通る工具移動指令を生成してしまうためである。図26は、実施の形態1において、図22の加工プログラム形状の実行時における工具移動指令の一例を示す図である。この図26に示す工具移動指令に基づいて加工を行うと、コーナ部は滑らかにならず、工具移動指令と加工プログラム形状とは同一形状となり、コーナ部を滑らかにした工具移動指令は生成されない。

10

【0083】

図27は、実施の形態1において、加工プログラム形状の一例を示す第2図である。図27では、加工プログラム形状はXY平面における半円である。円弧形状の場合には、各軸速度は基本的に三角関数に従うが、始点において速度ゼロから加速し、終点に向けて減速して速度ゼロとなる。図28は、実施の形態1において、図27の加工プログラム形状の実行時について、時間を横軸とし、速度を縦軸とし、X軸速度及びY軸速度を示す図である。移動平均フィルタをかけると、図28に示すような滑らかな各軸速度波形を生成することができる。

20

【0084】

図29は、実施の形態1において、図27の加工プログラム形状の実行時について、生成される工具移動指令の一例を示す図である。移動平均フィルタで各軸速度波形に遅れが生じるため、工具移動指令は、図29に示すように、加工プログラムの始点と終点とを通り、始点と終点との間の経路は加工プログラム形状を内回る曲線形状となる。

30

【0085】

他方で、図27に示すXY平面における半円の加工プログラムを半分に分け、分割した加工プログラムについて各々異なる演算装置で工具移動指令を生成した場合について説明する。この場合も、分割した加工プログラムは別々に実行されるため、分割した点において各軸速度がゼロとなる。そのため、工具経路である工具移動指令は、図30に示すように、始点、終点及び分割した点を通り、各点の間の経路は加工プログラム形状を内回る曲線形状となる。図30は、実施の形態1において、図27の加工プログラム形状の実行時における速度波形と、工具経路とを示す図である。図31は、実施の形態1において、図27の加工プログラム形状の実行時における工具移動指令の一例を示す図である。この工具移動指令に基づいて加工を行うと、図31に示すように、始点、終点及び分割した点を通り、各点の間の経路は加工プログラム形状を内回る曲線形状となる。つまり、分割した点において突起を有する形状に加工されてしまう。

40

【0086】

以上説明したように、加工の途中で分割を行うと、分割した点を通る工具移動指令が生成され、分割しなかった場合の加工プログラムにおける工具移動指令とは異なる工具移動指令が生成されてしまう。このような工具移動指令が生成されると、分割した点で工具移動指令の経路が大きく変化し、加工面に傷が発生するといった加工不良が生じる。

【0087】

従って、比較例では、加工工程、例えば、粗加工、中仕上げ及び仕上げ加工の区切りといった加工が一旦中断する箇所においてのみ加工プログラムを分割することができる。

50

【0088】

そのため、加工工程数により分割数が制限されるので、複数の演算装置の活用には改善の余地がある。更に、一連の加工形状を中断することなく加工するような場合には、加工プログラムを分割することができず、工具移動指令は1つの演算装置で生成せざるを得ず、高速化できないという問題があった。

【0089】

実施の形態1によれば、一連の加工形状を連続して加工する場合又は加工工程数よりも演算装置の設置数が多い場合においても、複数の演算装置を活用して高速に工具移動指令を生成することができる。

【0090】

実施の形態2 .

図32は、本発明の実施の形態2に係る指令値生成装置の構成を示す図である。図32に示す指令値生成装置200は、入力された加工プログラム20に従って工具移動指令を生成した場合の被加工物の加工後形状をシミュレーションすることができる。なお、実施の形態2において、実施の形態1と同様の構成については実施の形態1の説明を援用し、重複する説明を省略するものとする。

【0091】

従来の加工シミュレーションの多くは加工プログラム形状に基づいて被加工物の加工後形状をシミュレーションしている。しかしながら、実際の加工では、加工プログラムに様々な補正を加えた工具移動指令に基づいて加工を行う。ここで、加工プログラムに加える補正は、加減速、工具補正又は座標変換である。この補正により被加工物に傷が発生するといった加工不良が発生することもあるにも関わらず、加工プログラム形状に基づいて加工後形状をシミュレーションしても、このような加工不良はシミュレーションされない。実施の形態2では、工具移動指令から得られたサーボ応答に基づいて被加工物の加工後形状をシミュレーションするため、加減速、工具補正又は座標変換の補正に起因する傷をシミュレーションすることも可能である。

【0092】

指令値生成装置200は、分割部210、並列演算部220、指令値生成部230、サーボシミュレーション部240及び加工シミュレーション部250を備える。分割部210は、加工プログラム20を実行単位に分割する。並列演算部220は、実行単位毎の加工プログラムに従って実行単位毎の工具移動指令である分割指令を生成する。指令値生成部230は、分割指令から一連の工具移動指令を生成する。サーボシミュレーション部240は、工具移動指令が入力された場合の図示しない工作機械のサーボ系の動作をシミュレーションしてサーボ応答を生成する。加工シミュレーション部250は、サーボ応答に基づいて被加工物の加工形状をシミュレーションする。また、サーボシミュレーション部240を省略し、工具移動指令又は分割指令に基づいて加工シミュレーションを行ってもよい。

【0093】

ここで、並列演算部220には演算装置221, 222, 223が設けられている。並列演算部220は、マルチコアCPU、複数のCPUを備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されている。また、図32において、並列演算部220は3つの演算装置を備えるが、本発明はこれに限定されず、並列演算部220が備える演算装置の設置数は1つ以上であればよい。並列演算部220が備える演算装置の設置数を多くすると、工具移動指令を高速に生成することができ、高速化の目標値に応じて演算装置の設置数を決めればよい。

【0094】

分割部210は、入力された加工プログラム20を複数の実行単位に分割し、実行単位毎に分割加工プログラムを並列演算部220に出力する。

【0095】

並列演算部220は、実行単位毎に分割加工プログラムを演算装置221, 222, 2

10

20

30

40

50

23に割り振る。演算装置221、演算装置222及び演算装置223は、加減速及び座標変換の処理を行い、実行単位毎の工具の移動指令である分割指令を順次生成して指令値生成部230に出力する。ここで、演算装置221、演算装置222及び演算装置223は、並列実行されており、1つの実行単位における分割指令の生成が完了すると、次の実行単位における分割指令の生成を開始し、未処理の実行単位がなくなるまで処理を繰り返す。

【0096】

指令値生成部230は、実行単位毎の分割指令から一連の工具移動指令を生成してサーボシミュレーション部240に出力する。なお、指令値生成部230は、単一のCPU、単一のCPUを備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されている。又は、指令値生成部230は、並列演算部220と同様に、マルチコアCPU、複数のCPUを備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されていてもよい。

10

【0097】

サーボシミュレーション部240は、予め設定されたサーボモデルを用いて工具移動指令からサーボ応答を推定し、加工シミュレーション部250に出力する。加工シミュレーション部250は、サーボシミュレーション部240からのサーボ応答に基づいて工作機械が動作した場合の被加工物の加工形状をシミュレーションし、図示しない出力装置に出力する。ここで、出力装置にはディスプレイを例示することができる。

20

【0098】

次に、各構成要素の処理内容について詳細に説明する。なお、分割部210及び並列演算部220の処理内容は、実施の形態1の分割部110及び並列演算部120と同様であるため省略する。

【0099】

指令値生成部230の処理内容も実施の形態1の指令値生成部130と類似している。すなわち、各分割指令のオーバーラップした領域を除いて、先頭の分割指令から順に出力することで、一連の工具移動指令を生成する。ただし、指令値生成部230の出力より後段の処理が、工具移動指令の順序を問わない用途の場合には、図33に示すように、各分割指令の生成が開始され次第、各分割指令のオーバーラップした領域を削除して並列して分割指令を出力してもよい。図33は、実施の形態2における図32に示す並列演算部220における分割加工プログラムの実行タイミングと、図32に示す指令値生成部230における一連の工具移動指令の生成タイミングとを示すタイミングチャートである。

30

【0100】

一例として、指令値生成部230の出力より後段の処理が、加工シミュレーション部250のみである場合について説明する。加工シミュレーション部250は、工具移動指令の順序を問わない。すなわち、並列演算部220で生成した分割指令を他の順番で加工シミュレーションしても最終的な加工シミュレーション結果は同じ結果となる。この場合には、指令値生成部230では、分割指令を分割した順序に整列させるのではなく、並列して出力し、後の処理である加工シミュレーションも並列して処理した方が演算効率がよい。

40

【0101】

サーボシミュレーション部240は、制御対象である図示しない工作機械の挙動を計算機上で模擬する。サーボシミュレーション部240は、具体的には、工作機械の慣性、粘性及び弾性のパラメータと、慣性、粘性及び弾性に起因する共振周波数又は反共振周波数と、軸反転時のバックラッシュ又はロストモーションのパラメータと、熱変位のパラメータと、加工時の反力に起因する変位量のパラメータとを予め設定し、制御対象の挙動を模擬するためのサーボモデルを構築する。サーボシミュレーション部240は、構築したサーボモデルを用いて工具移動指令から実際の工作機械の動作であるサーボ応答を生成し、加工シミュレーション部250に出力する。なお、サーボ応答の経路及び速度波形を出力装

50

置であるディスプレイに表示してもよい。

【0102】

加工シミュレーション部250は、サーボシミュレーション部240により生成されたサーボ応答に基づいて工作機械の工具を動作させたときの被加工物の加工形状をシミュレーションし、加工シミュレーション結果を出力装置に出力する。すなわち、加工シミュレーション部250は、加工シミュレーション結果をディスプレイに表示する。なお、実施の形態2の指令値生成装置200は、サーボシミュレーション部240を備え、サーボ応答に基づいて加工シミュレーション部250が加工シミュレーションを行うが、サーボシミュレーション部240を省略し、工具移動指令又は分割指令に基づいて加工シミュレーションを行ってもよい。

10

【0103】

加工シミュレーション部250では、工具がサーボ応答に沿って動作するときの容積である掃引容積が演算される。この掃引容積によって横切られる被加工物の部分を除去することで被加工物の最終的な加工結果をシミュレーションすることができる。

【0104】

実施の形態2で説明した処理を行うことで、工具移動指令に基づいたサーボシミュレーション又は加工シミュレーションを高速に行うことが可能となる。これにより、1つの演算装置を用いたシステムでは実加工に匹敵する時間を要していたシミュレーションを短時間で行うことができ、工作機械の調整作業であるパラメータ調整に要する工数を削減することが可能である。特に、加工シミュレーションの工具移動指令の順序を問わない場合には、並列演算部220で分割指令の生成の完了と工具移動指令の生成の完了との間にタイムラグが生じにくい。そのため、1つの演算装置で工具移動指令を生成した場合と比較すると「1/演算装置の設置数」の時間で工具移動指令を生成することができる。

20

【0105】

実施の形態3.

図34は、本発明の実施の形態3に係る指令値生成装置の構成を示す図である。図34に示す指令値生成装置300は、図1に示す指令値生成装置100と同様に高速で指令値を生成することが可能な構成であり、図34に示す指令値生成装置300は、図1に示す指令値生成装置100と2つの点で相違する。第1点目は、指令値生成部130に代えて、並列指令値生成部330を備え、複数の演算装置の各々に対して指令値生成部が各々設けられており、指令値生成処理を並列で行う点である。並列指令値生成部330は、指令値生成部331, 332, 333を備える。なお、実施の形態3において、実施の形態1, 2と同様の構成については実施の形態1, 2の説明を援用し、重複する説明を省略するものとする。

30

【0106】

図35は、実施の形態1における図1に示す演算装置121, 122, 123における分割加工プログラムの実行タイミングと、図1に示す指令値生成部130における一連の工具移動指令の生成タイミングとを示すタイミングチャートである。図36は、実施の形態3における図34に示す演算装置321, 322, 323における分割加工プログラムの実行タイミングと、図34に示す指令値生成部331, 332, 333における一連の工具移動指令の生成タイミングとを示すタイミングチャートである。図35及び図36には、各演算装置の演算時間にばらつきがある場合のタイミングチャートが示されており、図35は比較のために示すものである。

40

【0107】

図35においては、演算装置121, 122, 123の各々における処理が終わり次第、分割指令1, 2, 3, 4, 5, 6を指令値生成部130に出力する。しかしながら、指令値生成部130では、分割指令1, 2, 3, 4, 5, 6をこの順で処理するため、演算装置121の演算時間が演算装置122, 123の演算時間に比してばらつきがある場合には、演算装置122, 123の演算が完了しているにもかかわらず、演算装置121の処理が完了するまで、指令値生成部130は、分割指令2, 3の処理を開始することがで

50

きない。また、分割指令 4, 5, 6 の処理についても同様に、分割指令 4 の処理が完了するまで、指令値生成部 130 において分割指令 5, 6 の処理を開始することができない。その結果、工具移動指令の生成が遅れるため、指令値生成装置 100 の全体の処理速度が演算装置の演算時間のばらつきに起因して低下することになる。

【0108】

一方、図 34 に示す指令値生成装置 300 では、演算装置ごとに独立した指令値生成処理が可能であるため、演算装置 321 の分割指令 1 の処理が完了していなくても、演算装置 322, 323 の処理が完了していれば、分割指令 2, 3 の処理を開始することができる。

【0109】

図 34 に示す構成とすることで、並列演算部 320 内の各演算装置の演算時間にばらつきがあっても、分割指令の生成の完了と工具移動指令の生成の完了との間にタイムラグが生じにくく、演算装置の演算時間のばらつきによる速度低下が小さくなる。

【0110】

第 2 点目は、各演算装置における処理が途中の段階であっても、その時点までに生成された各分割指令を出力する点、言い換えると分割指令を細かい単位で複数回に分けて出力する点である。

【0111】

図 34 に示す構成とすることで、並列演算部 320 の処理と同時並行で並列指令値生成部 330 の処理、図示しない NC 工作機械の処理又は加工シミュレーションの処理を行うことができる。そのため、リアルタイム性を重視する場合、又は NC 工作機械若しくは加工シミュレーションをより高速に行いたい場合には最適である。

【0112】

図 34 に示す指令値生成装置 300 は、分割部 310、並列演算部 320 及び並列指令値生成部 330 を備え、入力された加工プログラム 30 に従って、工具経路上の単位時間毎の補間点群である工具移動指令 31 を出力することで、NC 工作機械の被加工物に対する工具の移動を数値制御する。分割部 310 は、加工プログラム 30 を実行単位に分割して分割加工プログラムを生成する。並列演算部 320 は、実行単位毎の加工プログラムに従って実行単位毎の工具移動指令である分割指令を生成する。並列指令値生成部 330 は、分割指令から工具移動指令を生成する。

【0113】

ここで、並列演算部 320 には、演算装置 321, 322, 323 が設けられている。また、並列指令値生成部 330 には、指令値生成部 331, 332, 333 が設けられている。並列演算部 320 及び並列指令値生成部 330 は、マルチコア CPU、複数の CPU を備えたパーソナルコンピュータ又は通信回線によって接続されたサーバ若しくはクラウドシステムによって構成されている。また、図 34 において、並列演算部 320 は 3 つの演算装置を備えるが、本発明はこれに限定されず、並列演算部 320 が備える演算装置の設置数は 1 つ以上であればよい。並列指令値生成部 330 においても指令値生成部の数は、並列演算部 320 が備える演算装置の設置数と等しくすればよい。並列演算部 320 が備える演算装置の設置数を多くすると、工具移動指令を高速に生成することができ、高速化の目標値に応じて演算装置の設置数を決めればよい。

【0114】

次に、各構成要素の処理内容について詳細に説明する。なお、分割部 310 の処理内容は、図 1 に示す分割部 110 と同じであるため説明を省略する。並列演算部 320 の処理内容は、実施の形態 1 と類似し、数値制御装置と同一の処理を並列して実行し、分割加工プログラムから工具の指令経路である分割指令を生成する。実施の形態 3 では、各演算装置における処理が途中の段階であっても、その時点までに生成された各分割指令を、細かい単位で複数回に分けて出力することができる。この、分割指令よりも細かい単位を出力単位数と称呼する。なお、出力単位数は適宜設定することが可能である。

【0115】

ここで、分割数 $N = 6$ とし、出力単位数を各分割指令の 3 等分となるように設定したとする。図 3 7 は、実施の形態 3 における図 3 4 に示す指令値生成装置 3 0 0 におけるデータフローを示す第 1 図である。図 3 8 は、実施の形態 3 における図 3 4 に示す指令値生成装置 3 0 0 におけるデータフローを示す第 2 図である。図 3 7 は、並列演算部 3 2 0 の初回実行時のデータフローを示し、図 3 8 は、並列演算部 3 2 0 の初回実行が完了し、次の実行単位における加工プログラムの実行時のデータフローを示している。

【 0 1 1 6 】

なお、図 3 7 及び図 3 8 では、分割部 3 1 0 は、加工プログラム 3 0 を 6 個の実行単位、すなわち実行単位 1, 2, 3, 4, 5, 6 に分割し、さらに実行単位毎の分割指令を 3 等分に細かく分けて出力しているが、本発明はこれに限定されず、分割数及び出力単位数は適宜設定することができる。

10

【 0 1 1 7 】

図 3 7 に示すように、分割部 3 1 0 が加工プログラム 3 0 を実行単位 1, 2, 3, 4, 5, 6 に分割した場合には、3 個の演算装置を備える並列演算部 3 2 0 は、実行単位 1, 2, 3 を並列実行して、分割指令 1 - 1, 1 - 2, 1 - 3、分割指令 2 - 1, 2 - 2, 2 - 3 及び分割指令 3 - 1, 3 - 2, 3 - 3 を生成する。そして、並列指令値生成部 3 3 0 は、分割指令 1 - 1, 1 - 2, 1 - 3、分割指令 2 - 1, 2 - 2, 2 - 3 及び分割指令 3 - 1, 3 - 2, 3 - 3 から実行単位 1 から実行単位 3 に対する工具移動指令を生成する。

【 0 1 1 8 】

まず、数値制御装置と同一の処理を実行し、単位時間毎の補間点群の集合である分割指令 1 - 1, 2 - 1, 3 - 1 が生成されていく。ここで、具体的な処理には、予め設定された加速度で加減速するための速度波形を生成する加減速処理、加減速処理により生成した速度波形を滑らかにするスムージング処理及びスムージング処理後速度で移動したときの単位時間毎の工具位置である補間点を演算する補間処理を例示することができる。これと同時に、分割指令 1 - 1, 2 - 1, 3 - 1 の補間点数を計算していく。

20

【 0 1 1 9 】

その後、分割指令 1 - 1, 2 - 1, 3 - 1 の各々が、出力単位数に達したか否か及びオーバーラップ領域が含まれているか否かを判定する。これは、分割指令 1 - 1, 2 - 1, 3 - 1 の各々の開始部分にオーバーラップ領域が含まれているか、又は終了部分にオーバーラップ領域が含まれているか否かにより判定手法が異なる。

30

【 0 1 2 0 】

まず、開始部分にオーバーラップ領域が含まれているか否かは、分割指令を出力した際の累積距離とオーバーラップ量とを比較して、その大小で判定することができる。前者の方が大きい場合、分割指令にはオーバーラップ領域は含まれていないと判定し、一方で、後者の方が大きい場合、分割指令にはオーバーラップ領域が含まれていると判定する。

【 0 1 2 1 】

次に、終了部分にオーバーラップ領域が含まれているか否かは、分割指令を出力した際の残距離とオーバーラップ量とを比較して、その大小で判定することができる。前者の方が大きい場合、分割指令にはオーバーラップ領域は含まれていないと判定し、一方で、後者の方が大きい場合、分割指令にはオーバーラップ領域が含まれていると判定する。

40

【 0 1 2 2 】

上記の条件を満たしたとき、分割指令 1 - 1, 2 - 1, 3 - 1 の各々が並列指令値生成部 3 3 0 へ出力される。なお、上記の条件を満たすまで、分割指令 1 - 1, 2 - 1, 3 - 1 は、並列演算部 3 2 0 内で保存されるが、その間、分割指令 1 - 2, 2 - 2, 3 - 2 の生成は逐次行われていく。

【 0 1 2 3 】

これと同時に、分割指令 1 - 2, 2 - 2, 3 - 2 の各々の補間点数を計算していき、分割指令 1 - 1, 2 - 1, 3 - 1 の各々に対して出力単位数に達したか否か及びオーバーラップ領域が含まれているか否かを再び判定した後、分割指令 1 - 2, 2 - 2, 3 - 2 の各々を並列指令値生成部 3 3 0 へ出力する。この作業を分割指令 1 - 3, 2 - 3, 3 - 3 が並

50

列指令値生成部 330 へ出力されるまで行い、実行単位 1 から実行単位 3 に対する工具移動指令を生成する。

【0124】

図 38 に示すように、並列演算部 320 が実行単位 1 から実行単位 3 の実行を完了すると、並列演算部 320 は、次の実行単位である実行単位 4 から実行単位 6 を実行し、分割指令 4 - 1, 4 - 2, 4 - 3、分割指令 5 - 1, 5 - 2, 5 - 3 及び分割指令 6 - 1, 6 - 2, 6 - 3 を生成する。

【0125】

次に、並列演算部 320 の演算装置の設置数と同数の指令値生成部を備える並列指令値生成部 330 の処理内容について説明する。指令値生成部 330 の処理内容も、図 1 に示す指令値生成部 130 と類似する。すなわち、各分割指令のオーバーラップした領域を除いて出力することで、工具移動指令を生成する。

【0126】

ただし、並列演算部 320 において、オーバーラップ領域が含まれていないと判定された分割指令に対してはそのまま出力される。また、上記処理は、各演算装置に対応した指令値生成部にて行う。このようにして、分割指令が生成された演算装置から順次、指令値生成の処理を行うことができるため、各演算装置の処理時間にばらつきがあっても、指令値生成装置全体の速度低下を抑えることができる。

【0127】

以上、実施の形態 3 で説明した処理を行うことで、各演算装置の処理時間のばらつきに起因する指令値生成装置全体の速度低下の影響を最小限に抑えることができる。また、並列演算部 320 で分割指令の生成の完了と工具移動指令の生成の完了との間にタイムラグが生じにくい。そのため、実施の形態 3 に係る指令値生成装置は、実施の形態 1 に係る指令値生成装置よりも演算装置の処理時間のばらつきの影響を受けにくく、また、高速で工具移動指令を生成することが可能である。なお、前述の方法を用いて、分割指令を細かい単位で複数回に分けて出力する場合、出力単位数を設定する必要がある。しかしながら、指令値生成部からの出力単位数変更要求信号を受け取ったタイミングで出力単位数を変更することもできる。又は、指令値生成部からの出力要求信号を受け取ったタイミングで出力することで、出力単位数を設けずに、分割指令を細かい単位で複数回に分けて出力することも可能である。

【0128】

以上、実施の形態 3 にて説明した構成は実施の形態 2 にて説明した構成と組み合わせることも可能である。すなわち、実施の形態 3 にて説明したように並列指令値生成部を備える指令値生成装置が、実施の形態 2 にて説明したサーボシミュレーション部及び加工シミュレーション部を備えていてもよい。

【0129】

以上の実施の形態に示した構成は、本発明の内容の一例を示すものであり、別の公知の技術と組み合わせることも可能であるし、本発明の要旨を逸脱しない範囲で、構成の一部を省略、変更することも可能である。

【符号の説明】

【0130】

10, 20, 30 加工プログラム、11, 31 工具移動指令、100, 200, 300 指令値生成装置、110, 210, 310 分割部、120, 220, 320 並列演算部、121, 122, 123, 221, 222, 223, 321, 322, 323 演算装置、130, 230, 331, 332, 333 指令値生成部、240 サーボシミュレーション部、250 加工シミュレーション部、330 並列指令値生成部。

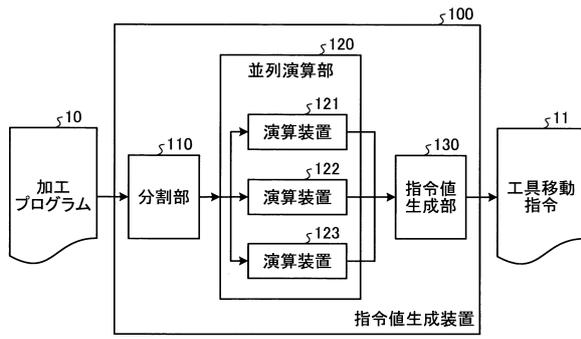
10

20

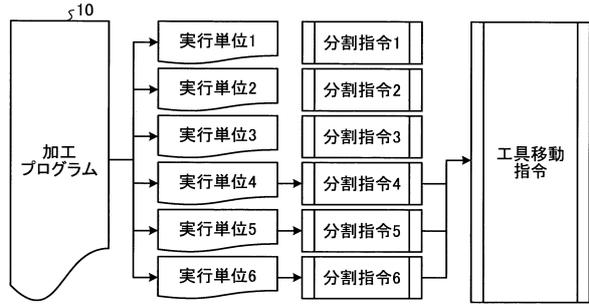
30

40

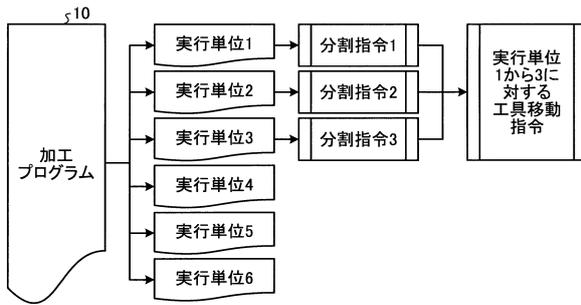
【図1】



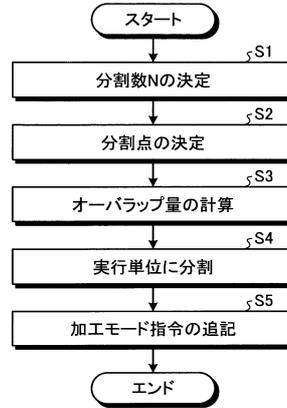
【図3】



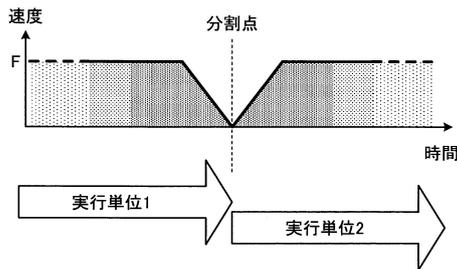
【図2】



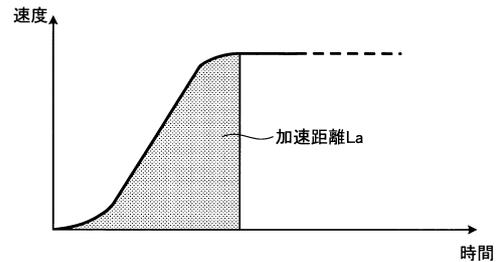
【図4】



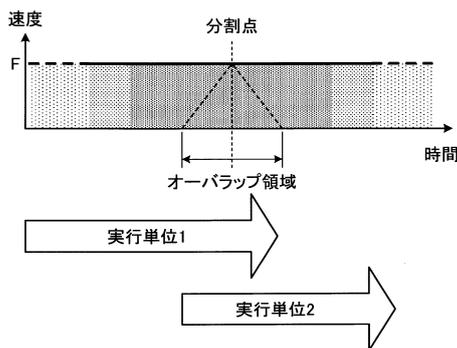
【図5】



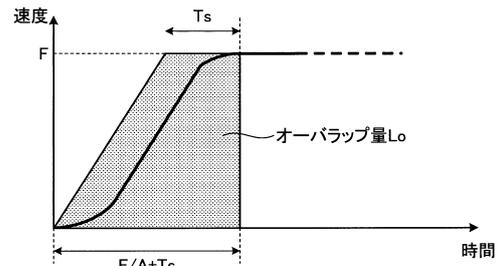
【図7】



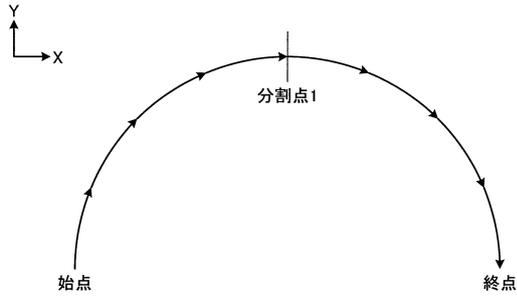
【図6】



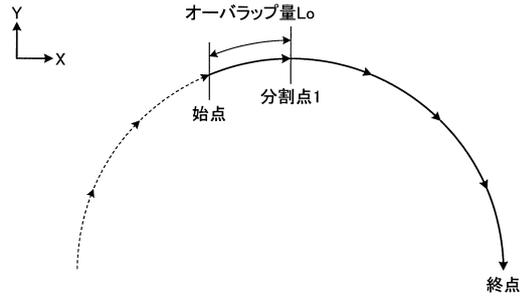
【図8】



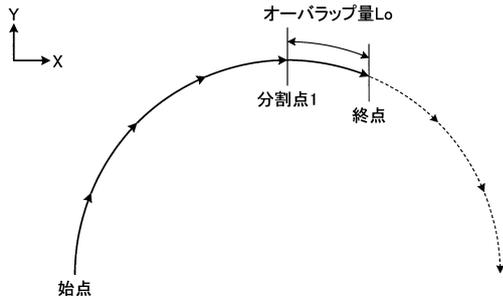
【 図 9 】



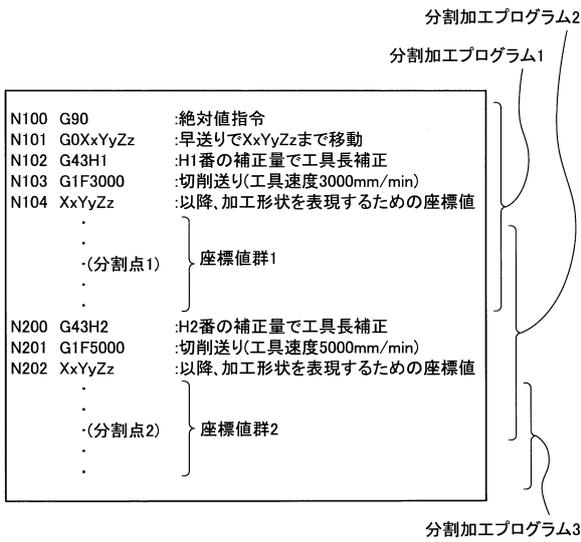
【 図 1 1 】



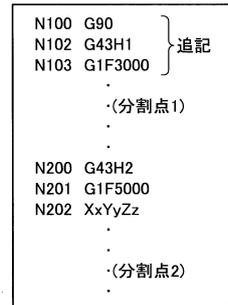
【 図 1 0 】



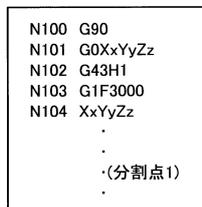
【 図 1 2 】



【 図 1 4 】



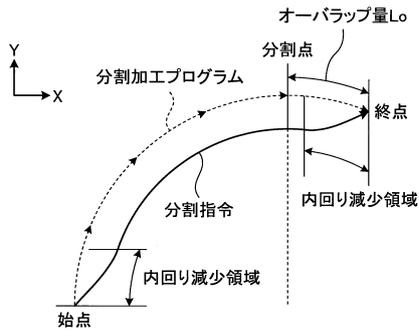
【 図 1 3 】



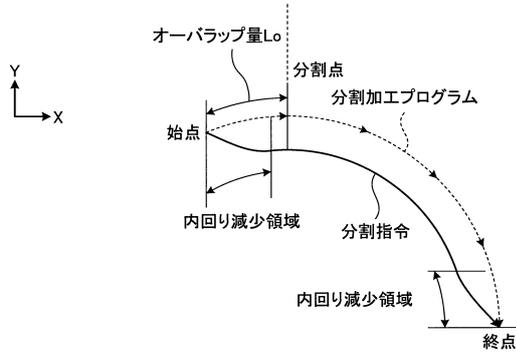
【 図 1 5 】



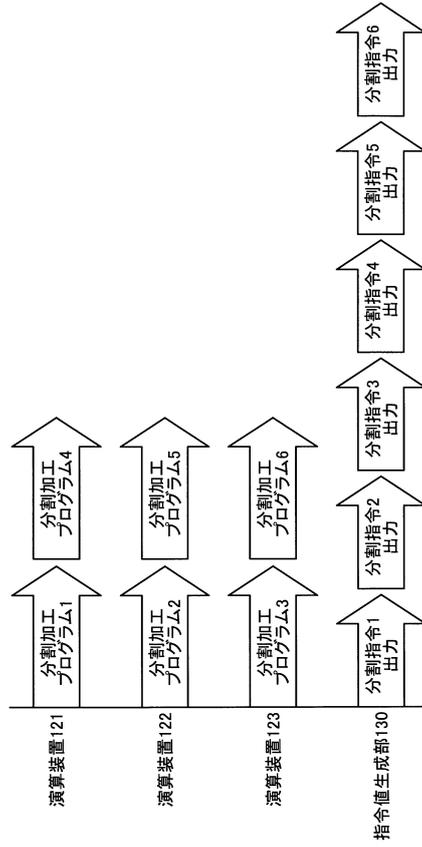
【図16】



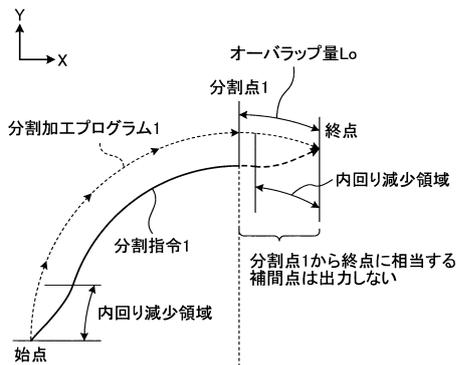
【図17】



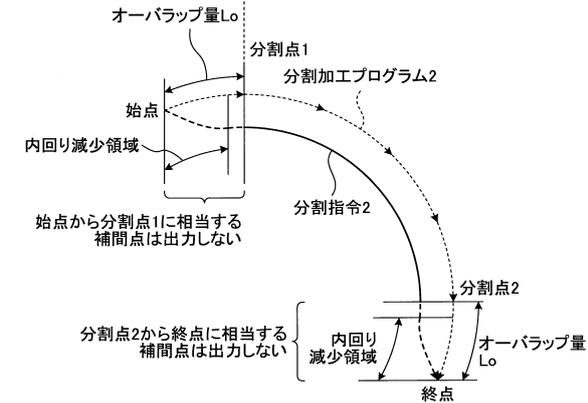
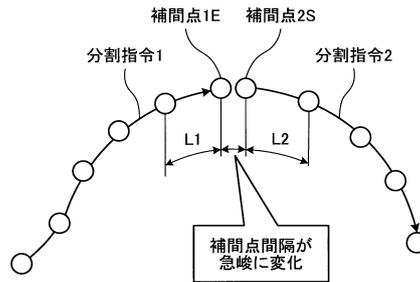
【図18】



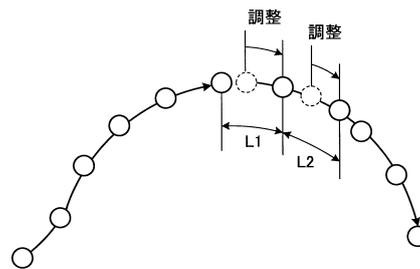
【図19】



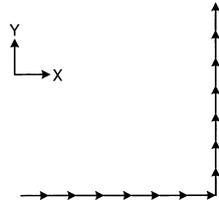
【図20】



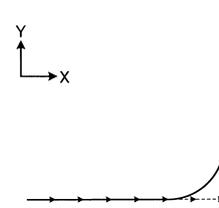
【図21】



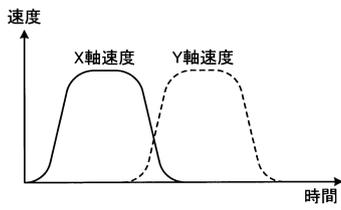
【 図 2 2 】



【 図 2 4 】



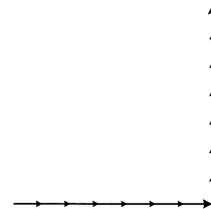
【 図 2 3 】



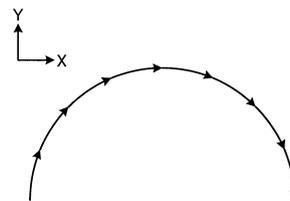
【 図 2 5 】

加工プログラム形状		
速度波形		
工具経路		

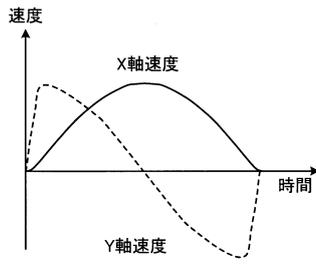
【 図 2 6 】



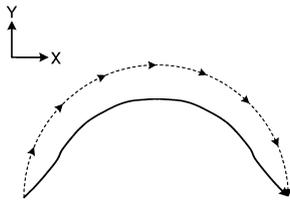
【 図 2 7 】



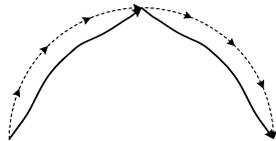
【図28】



【図29】



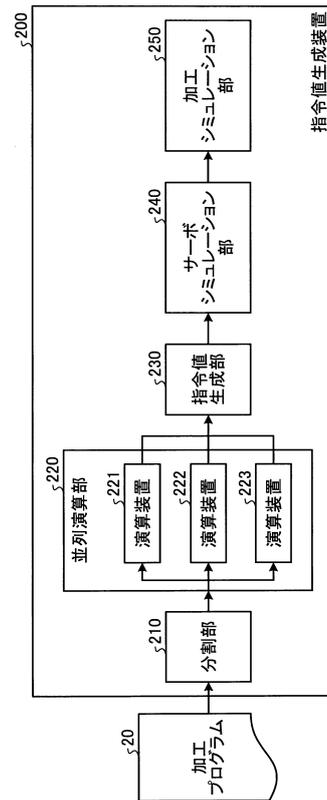
【図31】



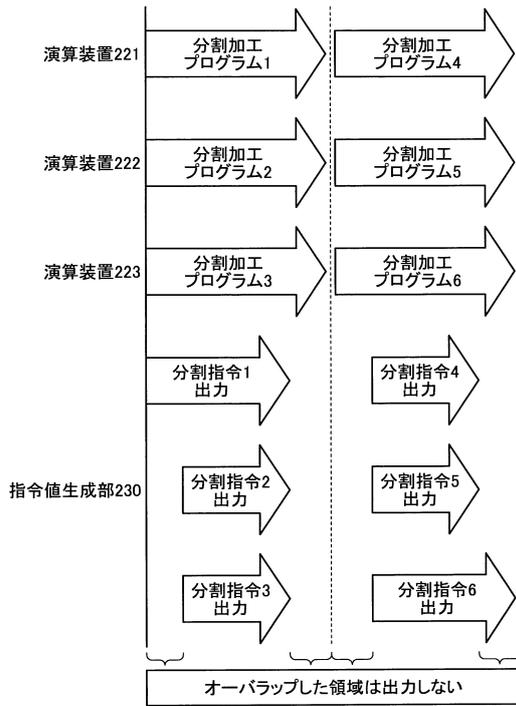
【図30】

加工プログラム形状	速度波形	工具経路

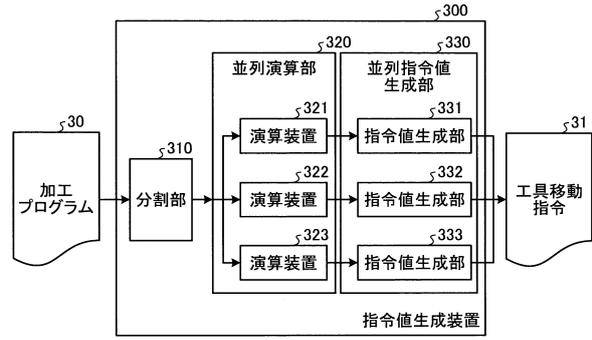
【図32】



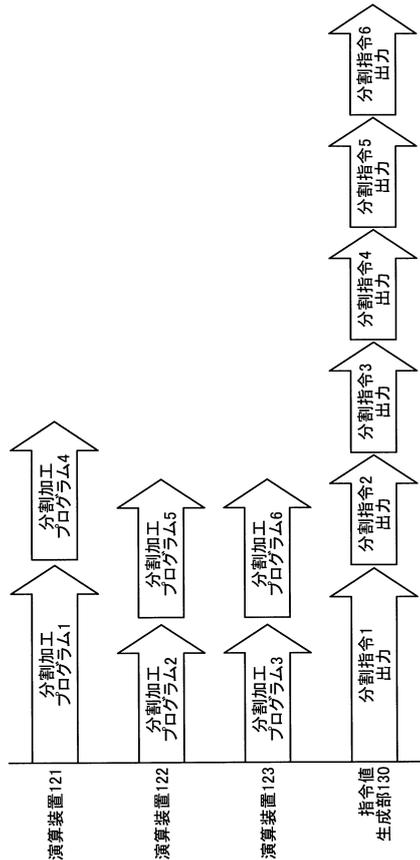
【図 3 3】



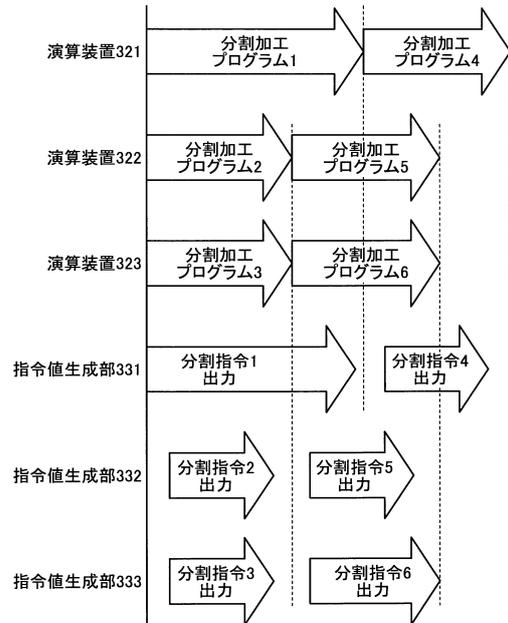
【図 3 4】



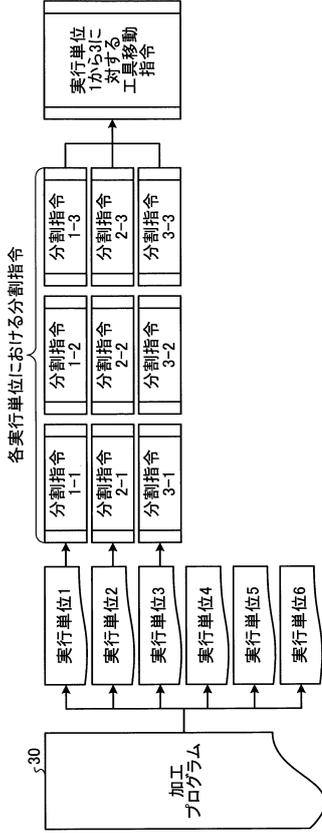
【図 3 5】



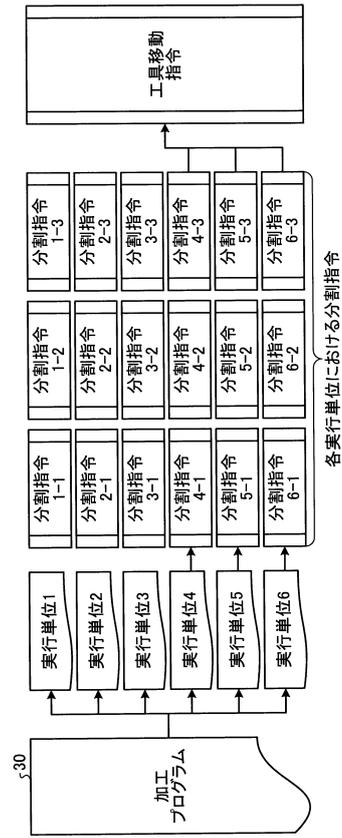
【図 3 6】



【 図 3 7 】



【 図 3 8 】



フロントページの続き

審査官 武市 匡紘

(56)参考文献 特開平11-114858(JP,A)
特許第5165159(JP,B2)
特開2001-117617(JP,A)

(58)調査した分野(Int.Cl., DB名)
G05B 19/18 - 19/46
B23Q 15/00 - 15/28