

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4574580号
(P4574580)

(45) 発行日 平成22年11月4日 (2010. 11. 4)

(24) 登録日 平成22年8月27日 (2010. 8. 27)

(51) Int. Cl.	F 1
G 0 5 B 19/4069 (2006. 01)	G O 5 B 19/4069
B 2 5 J 9/22 (2006. 01)	B 2 5 J 9/22 A
G 0 5 B 19/42 (2006. 01)	G O 5 B 19/42 W

請求項の数 3 (全 16 頁)

(21) 出願番号	特願2006-94103 (P2006-94103)	(73) 特許権者	000001236
(22) 出願日	平成18年3月30日 (2006. 3. 30)		株式会社小松製作所
(65) 公開番号	特開2007-272309 (P2007-272309A)		東京都港区赤坂二丁目3番6号
(43) 公開日	平成19年10月18日 (2007. 10. 18)	(74) 代理人	100071054
審査請求日	平成20年10月29日 (2008. 10. 29)		弁理士 木村 高久
		(74) 代理人	100106068
			弁理士 小幡 義之
		(72) 発明者	山中 伸好
			大阪府枚方市上野3-1-1 株式会社小松製作所 大阪工場内
		審査官	植村 森平

最終頁に続く

(54) 【発明の名称】 作業ロボットのオフラインティーチング装置

(57) 【特許請求の範囲】

【請求項 1】

作業ロボットの3次元モデルと、ワークおよび周辺装置の3次元モデルとを画面上に表示させ、ロボットプログラムに従って作業ロボットの3次元モデルを模擬的に再生動作させて、再生動作の結果に応じてロボットプログラムを修正するようにした作業ロボットのオフラインティーチング装置であって、

作業ロボットの3次元モデルの表面の位置若しくは表面から離れた外側の位置に、干渉チェック用の線分を、表面に沿って複数、設定し、

かつ、作業ロボットの3次元モデルの各部位のうち、ワークおよび周辺装置の3次元モデルと干渉する可能性の高い部位ほど、干渉チェック用線分の設定密度を高く設定し、

この干渉チェック用の複数の線分それぞれが、ワークおよび周辺装置の3次元モデルと交差することを判定することによって、作業ロボットがワークおよび周辺装置に干渉することを判断する干渉チェック処理が行われること

を特徴とする作業ロボットのオフラインティーチング装置。

【請求項 2】

作業ロボットの3次元モデルが、実際の作業ロボットと同じ速度で再生動作するように、表示画面を更新する画面更新処理が行われる請求項1記載の作業ロボットのオフラインティーチング装置であって、

画面が更新される時間内に、干渉チェック処理が終了するように、干渉チェック処理に用いられる干渉チェック用線分の本数を調整すること

10

20

を特徴とする作業ロボットのオフラインティーチング装置。

【請求項3】

作業ロボットの3次元モデルの各部位のうち、ワークおよび周辺装置の3次元モデルと干渉する可能性の高い部位は、溶接トーチおよび／またはアームであること

を特徴とする請求項1記載の作業ロボットのオフラインティーチング装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、作業ロボットとワークおよび周辺装置をパーソナルコンピュータの画面上に表示させて、ロボットプログラムに従って作業ロボットを再生動作させて、再生動作の結果に応じてロボットプログラムを修正するようにした作業ロボットのオフラインティーチング装置に関するものである。

10

【背景技術】

【0002】

(従来技術1)

アーク溶接等を行う溶接ロボットは、最適溶接位置を確保するポジショナやスライダ（走行台車）や治具と組み合わせてワークを溶接する作業を行う。なお、これらポジショナ、スライダ、治具などを、以下ロボットの「周辺装置」という。

【0003】

溶接ロボットなどの作業ロボットでは、作業ロボットの各軸が駆動制御されることにより、溶接トーチの先端が溶接線に沿って移動され、溶接作業が行われる。ここで溶接作業を行わせるためには、作業ロボットの作業内容を教示して、作業ロボットの制御プログラムとしてのロボットプログラムを作成する必要がある。

20

【0004】

作業ロボットに作業内容を教示する方法には、作業者が作業ロボットを手動動作させることにより所望の位置、姿勢角度を教示するティーチングプレイバック方式と、作業ロボットのコントローラとは独立したコンピュータ上に作業ロボットやワークのモデルを構築し、そのモデルを使用してロボットの位置、姿勢角度を教示するオフラインティーチング方式とがある。

【0005】

ティーチングプレイバック方式によるときは、生産設備や治具、対象ワークが現実には揃っている必要があり、ティーチング作業をしているときは設備等を占有する。

30

【0006】

そこで溶接分野などにおけるティーチングは、外部から数値データやプログラム形式で情報を入力することにより生産ラインから離れてティーチングできるオフラインティーチング方式が主流となっている。

【0007】

作業ロボットの3次元モデルと、ワークおよびポジショナなどの周辺装置の3次元モデルとを、パーソナルコンピュータ（以下、パソコンという）の画面上に表示させ、ロボットプログラムに従って、作業ロボットをパソコンの画面上で模擬的に再生動作させて、再生動作の結果に応じて、ロボットプログラムを修正するという作業ロボットのオフラインティーチング装置は、後掲する特許文献1にみられるように、既に公知となっている。

40

【0008】

オフラインティーチング装置では、作業ロボットの3次元モデルを、実際の作業ロボットと同じ速度でパソコンの画面上で再生動作させる必要がある。これは、作業ロボットが実際に作業を行うときの作業時間を知るためと、実際の作業ロボットと同じ速度で再生動作させなければ検出できないエラーをチェックするためである。たとえば特異点付近での作業ロボットの動きにエラーが発生していないかなどがチェックされる。

【0009】

こうしたオフラインティーチングにおけるエラーチェックの中でも、再生動作時に、作

50

業ロボットの３次元モデルが、ワークおよび周辺装置に干渉するかどうかを判断する干渉チェックは、特に重要である。

【００１０】

（従来技術２）

一般に３次元モデルの表面は、ポリゴン（三角形の面）を最小単位として構成されている。

【００１１】

従来にあっては、図４（ａ）に示すように、再生動作時に、作業ロボット１の３次元モデル１Ｍの表面を構成するポリゴン９０が、ワーク１０および周辺装置８０の３次元モデル１０Ｍ、８０Ｍの表面を構成するポリゴン９１に交差したか否かを判定することをもって、作業ロボット１０が、ワーク１０および周辺装置８０に干渉したか否かを判断するようにしている。

10

【００１２】

（従来技術３）

特許文献２、３には、干渉チェック対象となる両３次元モデルを、これに近似する単純な直方体形状等のモデルに置換した上で、両３次元モデルを構成する面同士の交差を判定することで、干渉チェックを行うという発明が記載されている。

【特許文献１】特開２００３－１２７０７７号公報

【特許文献２】特開平１－１７３２０５号公報

【特許文献３】特開平１１－２５４３７９号公報

20

【発明の開示】

【発明が解決しようとする課題】

【００１３】

（第１の課題）

オフラインティーチングによるエラーチェックを精度よく行い、実機で使用に耐えるロボットプログラムに修正するには、その前提として、作業ロボットなどの３次元モデルを、実機に忠実に作成して、画面上に表示させなければならない。このように３次元モデルを実機に近く精密に表示させようとする、３次元モデルを構成するポリゴンの数は、必然的に増加する。このため従来技術２をそのまま適用すると、膨大な組み合わせのポリゴン９０とポリゴン９１の交差を逐一判定しなければならず、干渉チェック処理に多大な時間を要することになる。

30

【００１４】

確かに従来技術３を適用すれば、３次元モデルが単純化されることで、３次元モデルを構成するポリゴンの数が減り、干渉チェック処理のための計算量が減り、干渉チェック処理の時間を減らすことができる。しかし、作業ロボット等を単純な形状に近似してしまうと、エラーチェックの精度が低下してしまい、実機で使用に耐えるロボットプログラムに修正することはできない。また、単純な３次元モデルで得られた干渉チェックの結果は、信頼性に乏しくそのまま使うことができないため、元の実機に忠実な３次元モデルに戻した上で、再度、干渉チェックを行って詳細に検討する必要がある。このように干渉チェックを２重に行わざるを得なく、結果として、干渉チェックの処理には、時間を要することになってしまう。本発明はこうした実状に鑑みてなされたものであり、オフラインティーチングで、作業ロボットとワークおよび周辺装置の干渉チェックを行うに際して、干渉チェックの精度、信頼性を高く維持しつつも、干渉チェックの処理時間を減らすことを解決課題とするものである。

40

【００１５】

（第２の課題）

従来技術１で述べたように、オフラインティーチング装置では、作業ロボットの３次元モデルを、実際の作業ロボットと同じ速度でパソコンの画面上で再生動作させる必要がある。作業ロボットの再生動作中に、パソコンの画面上で作業ロボットの動きを滑らかに表示させるには、たとえば１秒間に数十回（たとえば３０フレーム）の画面の更新が必要と

50

される。そして、画面が更新される毎に、作業ロボットの動きが変化することから、画面が更新される間隔の時間（たとえば1/30秒）毎に干渉チェック処理を行い、その結果をリアルタイムに画面上に表示する必要がある。

【0016】

しかし、上述の第1の課題で述べたように、作業ロボットの3次元モデルを、実機に忠実に再生動作させて、干渉チェックを行うことにすると、干渉チェックの処理時間が必然的に増加してしまう。

【0017】

このため非常に短い画面更新の間隔の時間毎に、計算量が多く、多大な時間を要する干渉チェックを行わなければならない。この結果、画面更新時間内に、干渉チェック処理が終了できずに、画面が「コマ落ち」して、作業ロボットが画面上で滑らかに動かなくなってしまうおそれがある。これによりオペレータが画面上における作業ロボットの細かな動きを確認できなかったり、エラーを見逃してしまうおそれがある。たとえば、作業ロボットのツール先端が溶接線に沿って忠実にトレースしているか否かを、オペレータが画面上で確認できなくなるおそれがある。

10

【0018】

本発明は、こうした実状に鑑みてなされたものであり、作業ロボットの3次元モデルを画面上で実際の作業ロボットと同じ速度で再生動作させるに際して、画面更新間隔の時間内に干渉チェックを確実に終了させるようにして、コマ落ちすることなく画面上で作業ロボットを滑らかに再生動作させることを解決課題とするものである。

20

【課題を解決するための手段】

【0019】

第1発明は、

作業ロボットの3次元モデルと、ワークおよび周辺装置の3次元モデルとを画面上に表示させ、ロボットプログラムに従い作業ロボットの3次元モデルを模擬的に再生動作させて、再生動作結果にロボットプログラムを修正するようにした作業ロボットのオフラインティーチング装置であって、

作業ロボットの3次元モデルの表面の位置若しくは表面から離れた外側の位置に、干渉チェック用の線分を、表面に沿って複数、設定し、

この干渉チェック用の複数の線分それぞれが、ワークおよび周辺装置の3次元モデルと交差することを判定することによって、作業ロボットがワークおよび周辺装置に干渉することを判断する干渉チェック処理が行われること

30

を特徴とする。

【0020】

第2発明は、第1発明において、

作業ロボットの3次元モデルが、実際の作業ロボットと同じ速度で再生動作するように、表示画面を更新する画面更新処理が行われる請求項1記載の作業ロボットのオフラインティーチング装置であって、

画面が更新される時間内に、干渉チェック処理が終了するように、干渉チェック処理に用いられる干渉チェック用線分の本数を調整すること

40

を特徴とする。

【0021】

第3発明は、第1発明において、

作業ロボットの3次元モデルの各部位のうち、ワークおよび周辺装置の3次元モデルと干渉する可能性の高い部位ほど、干渉チェック用線分の設定密度を高く設定すること

を特徴とする。

【0022】

第1発明によれば、図3、図4(b)に示すように、作業ロボット1の3次元モデル1M（ポリゴン90）とは別に干渉チェック用線分20が設定され、線分20と、ワーク10および周辺装置80の3次元モデル10M、80Mを構成するポリゴン91とが交差し

50

ていることを判定することで、両者の干渉を判定するようにしたので、従来（図４（a））のように、作業ロボット１の３次元モデル１Ｍを構成するポリゴン（面）９０と、ワーク１０および周辺装置８０の３次元モデル１０Ｍ、８０Ｍを構成するポリゴン９１とが交差していることを判定する場合に比して、干渉チェック処理に要する時間を短縮することができる。このため３次元モデルを単純化せずとも短時間で干渉チェックを行うことができる。これにより本発明によれば、オフラインティーチングで、作業ロボット１とワーク１０および周辺装置８０の干渉チェックを行うに際して、干渉チェックの精度、信頼性を高く維持しつつも、干渉チェックに要する時間を減らすことができる。

【００２３】

第３発明では、作業ロボット１の３次元モデル１Ｍの各部位のうち、ワーク１０および周辺装置８０の３次元モデル１０Ｍ、８０Ｍと干渉する可能性の高い部位（たとえば溶接トーチ２、アーム３）ほど、干渉チェック用線分２０の設定密度が高くなるように設定される（図３）。このため、干渉チェックの精度向上と時間短縮を両立させることができる。

【００２４】

また、第２発明によれば、図７に示すように、干渉チェックサブルーチン処理に使用する干渉チェック用線分２０の本数Ｃを調整することで、干渉チェックサブルーチン処理に要する時間を調整して、画面更新間隔の時間 t を標準更新時間 t_{std} 付近に調整するようにしたので、画面更新間隔の時間内に、干渉チェックを確実に終了させることができる。このため３次元モデルを単純化せずとも確実に画面更新間隔の時間内に干渉チェックを終了させることができる。これにより本発明によれば、作業ロボット１の３次元モデル１Ｍを画面７ａ上で実際の作業ロボット１と同じ速度で再生動作させるに際して、画面更新間隔の時間内に干渉チェックを確実に終了させることができるようになり、コマ落ちすることなく画面７ａ上で作業ロボット１を滑らかに再生動作させることができる。

【発明を実施するための最良の形態】

【００２５】

以下、図面を参照して本発明の実施形態について説明する。

【００２６】

なお、実施形態では、作業ロボットとして、溶接作業を行う溶接ロボットを想定し、その溶接ロボットの各軸を駆動制御するためのロボットプログラムを修正するための装置を想定している。しかし、本発明としては、ツール先端をワーク上の移動経路に沿って移動させて所定の作業を行うものであれば、任意の作業ロボットに適用可能であり、たとえばシーリングを行うシーリング作業用ロボットにも適用することができる。

【００２７】

図１は、実施形態の装置の全体構成を示している。

【００２８】

同図１に示すように作業ロボット１は、アーム３を有しており、このアーム３の先端には、ツールである溶接トーチ２が取り付けられている。

【００２９】

作業ロボット１は、たとえば６軸作業ロボットであり、各軸 X 、 Y 、 Z 、 A 、 B 、 C が駆動されることにより作業ロボット座標系 $X-Y-Z$ 上で、トーチ先端２ａの座標位置およびトーチ姿勢角（ X 、 Y 、 Z 、 A 、 B 、 C ）が変化される。なおトーチ姿勢角（ A 、 B 、 C ）はオイラ角で定義される。

【００３０】

作業ロボット１の外部には、周辺装置８０が設けられている。周辺装置８０は、スライダ４、ポジショナ９からなる。

【００３１】

作業ロボット１の各軸、スライダ４の各軸、ポジショナ９の各軸は、ロボットコントローラで５によって駆動制御され、これによりトーチ先端２ａが、溶接対象であるワーク１０上の所定の移動経路に沿って移動される。なおスライダ４、ポジショナ９以外の治具な

どを設けて、周辺装置 80 を構成してもよい。

【0032】

ロボットコントローラ 5 には、「プログラミングペンダント」と呼ばれる携行自在の操作盤 50 が設けられている。この操作盤 50 を操作することによりロボットコントローラ 5 を介して作業ロボット 1 を実際に動作させることができる。また、この操作盤 50 を操作することにより、後述するようにパソコン 7 の表示画面 7a 上で作業ロボット 1 の 3 次元モデル 1M を動作させることができる。そして、作業ロボット 1 が動作したロボット各軸を記憶することにより、作業の各動作点ごとに、作業ロボット 1 の各軸位置、スライダ 4 の各軸位置、ポジショナ 9 の各軸位置を教示することができる。操作盤 50 で教示された教示データに基づいてロボットプログラム 30 (図 5) が作成され、また修正される。そして、修正されたロボットプログラム 30 にしたがって作業ロボット 1 の各軸、スライダ 4 の各軸、ポジショナ 9 の各軸が動作する。

10

【0033】

ワーク 10 は、ポジショナ 9 に装着されている。ポジショナ 9 は、各軸 U、V を有しており、各軸 U、V の駆動位置が変更されることにより、作業ロボット 1 に対するワーク 10 の相対位置関係が変化する。なお、本実施例では、ワーク 10 として、建設機械のバケットを想定している。

【0034】

作業ロボット 1 は、走行台車であるスライダ 4 上に移動自在に載置されている。スライダ 4 は各軸 R、S を有しており、各軸 R、S の駆動位置が変更されることにより、作業ロ

20

【0035】

これらポジショナ 9、スライダ 4 の各軸は、作業ロボット 1 の軸に対して外部 (EX) 軸と呼ばれる。

【0036】

したがって作業ロボット 1 の軸が駆動されることにより作業ロボット座標系におけるツール先端 2a の位置 P が変化されるとともに、外部軸が駆動されることにより、外部座標系におけるタッチ先端 2a の位置 (外部軸位置) E が変化する。作業ロボット軸と外部軸が駆動されたときのタッチ先端 2a の位置は、作業ロボット座標系上の位置 P と外部座標系上の位置 E とを合成した位置 $P + E$ として表される。

30

【0037】

パソコン 7 は、オフラインティーチング用端末を構成している。作業ロボット 1 の 3 次元モデル 1M のデータ、ワーク 10 の 3 次元モデル 10M のデータ、周辺装置 80 の 3 次元モデル 80M のデータは、図示しない 3 次元 CAD 装置で作成され、パソコン 7 に送られ、パソコン 7 に取り込まれる。また、操作盤 50 の操作によって教示された教示データは、パソコン 7 に取り込まれる。

【0038】

パソコン 7 では、教示データに基づいてロボットプログラム 30 が作成される。なお、教示データに基づき、ロボットプログラム 30 を予め作成しておき、ロボットプログラム 30 をパソコン 7 に取り込むようにしてもよい。

40

【0039】

図 2 は、パソコン 7 とロボットコントローラ 5 と操作盤 50 の構成をブロック図にて示している。

【0040】

パソコン 7 は、パソコン本体 700 と、ディスプレイ 750 からなる。ディスプレイ 750 の表示画面 7a に画像が表示される。パソコン本体 700 は、CPU 701 と、画像表示専用ボードとしてのビデオカード 702 と、ハードディスク 706 と、入出力部 (I/O) 705 を含んで構成されている。パソコン本体 700 には、入力装置としてのキーボード 703、マウス 704 が接続されている。ビデオカード 702 は、画面表示データを記憶するメモリを搭載した拡張カードであり、本実施例では、後述する画面表示機能、

50

干渉チェック用線分 20 と、ワーク 10 および周辺装置 80 の 3 次元モデル 10 M、80 M との交差を検出する機能を有している。

【0041】

パソコン本体 700 の入出力部 705 と、ロボットコントローラ 5 の入出力部 501 は、通信線 790 によって接続されている。

【0042】

操作盤 50 の操作に応じて、作業ロボット 1、スライダ 4、ポジショナ 9 が動き、それらの位置は通信線 790 を介してパソコン本体 700 に送られ、表示している 3 次元モデルが動作するソフトウェアがインストールされている。

操作盤 50 の操作に応じて、ロボットコントローラ 5 内で、ロボットプログラム 30 を作成し、また修正する処理が実行され、ロボットコントローラ 5 内のメモリ 502 に記憶される。

10

【0043】

また、パソコン本体 700 には、図 6、図 7 で後述するように、作業ロボット 1 とワーク 10、スライダ 4、ポジショナ 9 との干渉チェックを行うソフトウェアがインストールされている。

【0044】

図 3 は、作業ロボット 1 の 3 次元モデル 1 M を斜視図にて示している。

【0045】

同図 3 に示すように、作業ロボット 1 の 3 次元モデル 1 M の表面から離れた外側の位置には、干渉チェック用の線分 20 が、3 次元モデル 1 M の表面に沿って複数、設定されている。作業ロボット 1 の 3 次元モデル 1 M の各部位のうち、ワーク 10 および周辺装置 80 の 3 次元モデル 10 M、80 M と干渉する可能性の高い部位ほど、干渉チェック用線分 20 の設定密度が高くなるように設定されている。本実施例では、アーム 3、溶接トーチ 2 以外の箇所 に 比 して、アーム 3、溶接トーチ 2 の箇所の方が干渉チェック用線分 20 の設定密度が高くなるように、干渉チェック用線分 20 が設定されている。Nmax 本の干渉チェック用線分 20 のそれぞれには、#1、#2、…#N…#Nmax と、各線分を識別する識別番号が付与されている。干渉チェック用線分 20 を、作業ロボット 1 の 3 次元モデル 1 M の表面から離れた外側の位置に設定しているのは、3 次元モデル同士が接触する前の接近している段階で警告をするためである。

20

30

【0046】

干渉チェック用線分 20 の作成は、3 次元 CAD 装置で、作業ロボット 1 の 3 次元モデル 1 M を作成する際に行ってもよく、パソコン 7 で行ってもよい。

【0047】

図 8、図 9 は、作業ロボット 1 が行う作業の一例を斜視図にて示している。図 5 は、ロボットプログラム 30 の一例を示している。

【0048】

ロボットプログラム 30 は、各ステップ 33、34…43 からなり、各ステップ毎に、移動命令などのコマンド、補間条件、溶接条件データ、速度データ、作業ロボット 1 各軸の位置、姿勢角データ、外部軸の位置などが対応づけられている。

40

【0049】

ステップ 33 からステップ 43 には、作業ロボット 1 のトーチ先端 2a が溶接開始点まで直線補間で移動し、溶接開始点から溶接終了点まで直線補間で多層盛の溶接を行うという動作が記述されている。

【0050】

ステップ 33～43 のうち、移動命令のコマンドを示すステップ 37、38、40、41、43 には、教示データ、つまりツール先端 2a の座標位置およびトーチ姿勢角 (X、Y、Z、A、B、C)、ポジショナ 9、スライダ 4 の各軸位置 (U、V、R、S) が対応づけられている。作業ロボット 1 のトーチ先端 2a は、移動命令に従って各教示点 P1、P2、P3、P4、P5 に移動する。

50

【0051】

教示データが適切であれば、図8に示すように、作業ロボット1がワーク10などに干渉することはないが、教示データが不適切である場合には、図9に斜線にて示すように、作業ロボット1がワーク10などに干渉する。

【0052】

操作盤50を操作することで教示データが、通信線790を介してパソコン7に取り込まれ、ロボットプログラム30が作成される。なお、キーボード703、マウス704を操作して、教示データをパソコン7に取り込みロボットプログラム30を作成してもよい。

【0053】

10

図6、図7は、ロボットコントローラ5でのプログラム実行処理とパソコン7で行われる干渉チェック処理を示している。

【0054】

まず、ロボットコントローラ5において、ロボットプログラム30を1ステップずつ読み取り（ステップ101）、そのステップの内容が「移動命令」のコマンドであるか否かが判断される（ステップ102）。この結果「移動命令」のコマンドであると判断された場合には（ステップ102の判断Yes）、教示点間（たとえばP1～P2）の軌跡を補間して各目標位置Pを算出する。

【0055】

すなわち、作業ロボット1のトーチ先端2aの位置および姿勢角度（トーチ姿勢オイラ角）のデータ（X、Y、Z、A、B、C）は、各移動点P1、P2、P3、P4、P5毎に教示されている。なおポジショナ9、スライダ4の各軸のデータ（U、V、R、S）についても各移動点毎に教示されている。

20

【0056】

教示されたP1、P2、P3、P4点以外の中間の位置は、補間演算によって求められる。

【0057】

具体的には、始点（たとえばP1）をPs（Xs、Ys、Zs、As、Bs、Cs）とし、終点（たとえばP2）をPe（Xe、Ye、Ze、Ae、Be、Ce）とすると、軌跡補間データLpは下記（1）式の演算式により補間演算される。

【0058】

30

$$Lp = f(Ps, Pe, Vr) \quad \cdots (1)$$

上記（1）式において、Vrは予め設定されたトーチ先端2aの移動速度であり、fは予め定められた所定の関数である。

【0059】

各時刻tにおける補間位置P（X、Y、Z、A、B、C）は下記（2）式で与えられる。

【0060】

$$P = g(Lp, t) \quad \cdots (2)$$

上記（2）式において、gは予め定められた所定の関数である。このように、（1）、（2）式に従って軌跡が補間されて各目標位置Pが算出される（ステップ103）。

40

【0061】

つぎに、逆変換によって目標位置Pから作業ロボット1の各軸角度 θ_i が算出される。

【0062】

すなわち、作業ロボット1の各軸角度 θ_i （ $i = 1 \sim 6$ ）は、下記（3）式にて与えられる。

【0063】

$$\theta_i = \text{inv}K(P) \quad \cdots (3)$$

上記（3）式で関数 $\text{inv}K$ は、各軸角度 θ_i からトーチ先端位置Pを求める関数Kの逆関数である。トーチ先端位置Pを逆変換することによって作業ロボット1の各軸角度 θ_i を求めることができる。よって、上記（3）式にしたがい逆変換によって目標位置Pか

50

ら作業ロボット1の各軸角度 θ_i が算出される(ステップ104)。

【0064】

つぎに、上記ステップ104で演算された各軸角度 θ_i のデータがパソコン7に送られて、パソコン7に表示されている作業ロボット1の3次元モデル1Mが作成されて、作業ロボット1の3次元モデル1Mが更新される。なお、スライダ4、ポジショナ9の外部軸Eについても、各軸角度U、V、R、Sの変化に応じた姿勢のスライダ4、ポジショナ9の3次元モデル4M、9Mが作成されて、同3次元モデル4M、9Mが更新される(ステップ105)。

【0065】

上記ステップ105で、作業ロボット1の3次元モデル1Mの姿勢が変化する毎に、パソコン7側で干渉チェックサブルーチン処理が行われる。干渉チェックサブルーチン処理は、後述するように図7に示される(ステップ106)。

10

【0066】

干渉チェックサブルーチン処理が終了すると、ステップ105で更新された作業ロボット1の3次元モデル1Mを、パソコン7の表示画面7aに表示するように、画面7aの内容が更新される。なお、スライダ4、ポジショナ9の外部軸Eについてもスライダ4、ポジショナ9の3次元モデル4M、9Mの姿勢が更新される毎に、画面7aの内容が更新されることになる(ステップ107)。

【0067】

ステップ108では、教示点間(たとえばP1~P2)の軌跡の終わりまで各移動位置Pごとに、同様のステップ103~ステップ107の処理が繰り返し実行される。

20

【0068】

1つの「移動命令」のコマンドについて、上記の処理が終了すると、ロボットプログラム30が終了するまで、各移動命令のコマンド毎にステップ101~108の処理が繰り返される。

【0069】

このように、各時刻における作業ロボット1の各軸角度 θ_i が求められ、各時刻における各軸角度 θ_i の姿勢の作業ロボット1の3次元モデル1Mをパソコン7の表示画面7a上で順次更新させていけば、作業ロボット1を画面7a上で擬似的に再生動作させることができる。

30

【0070】

つぎに、図7に示す干渉チェックサブルーチン処理について説明する。

【0071】

なお、この干渉チェックサブルーチン処理の前提として、干渉チェックサブルーチン処理が行われる毎に、干渉チェックサブルーチン処理を実施した時刻が記憶されておかれるものとする

図6のステップ105で、作業ロボット1の3次元モデル1Mが更新されると、手順は、図7のステップ201に移行されて、現在の時刻 t_p から、前回に干渉チェックサブルーチン処理を実施した時刻 t_c を減算して、前回から今回までに処理に費やした処理時間 t が求められる。処理時間 t は、表示画面7aを更新する間隔の時間に対応する(ステップ201)。そして、現在時刻 t_p を、前回実施時刻 t_c に更新する(ステップ202)。

40

【0072】

以下では、表示画面7aが更新される時間内に、干渉チェックサブルーチン処理が終了するように、干渉チェックサブルーチン処理に用いられる干渉チェック用線分20の本数Cを調整する処理が実行される(ステップ203~205)。

【0073】

すなわち、表示画面7aを更新する間隔の時間は、作業ロボット1の3次元モデル1Mを、実際の作業ロボット1と同じ速度で動作させて画面上で滑らかに表示させるために短い時間(たとえば、1/30秒)に設定されている。一方で、画面更新毎に、作業ロボット1の3次元モデル1Mの姿勢が変化することから、画面更新毎に干渉チェックサブルー

50

チン処理を行い、リアルタイムに干渉チェック結果を画面 7 a 上に表示させる必要がある。そこで、標準更新時間 t_{std} が予め設定され、この標準更新時間 t_{std} で干渉チェックサブルーチン処理が行われて画面更新が行われるように、干渉チェック用線分 20 の本数 C が調整される。

【0074】

まず、ステップ 201 で求められた処理時間 t が、標準更新時間 t_{std} よりも大きいかな否かが判断される（ステップ 203）。

【0075】

ここで、複数本の干渉チェック用線分 20 を干渉チェック（ステップ 208）に使用する順序は、 $\#1 \rightarrow \#2 \rightarrow \dots \#N \rightarrow \dots \#N_{max}$ と、識別番号が順次大きくなる順序で行い、最大の識別番号 N_{max} に達すると、最小の識別番号 $\#1$ に戻り、以下同様に、識別番号を順次大きくする順序で干渉チェック用線分 20 を選択して干渉チェック（ステップ 208）に使用するものとする。初期状態では、干渉チェックサブルーチン処理に使用する干渉チェック用線分 20 の本数 C は、予め所定の本数 C_0 に設定されておかれるものとする。

【0076】

ステップ 203 の判断の結果、処理時間 t が、標準更新時間 t_{std} よりも大きいと判断された場合には（ステップ 203 の判断 YES）、干渉チェックサブルーチン処理に使用する干渉チェック用線分 20 の本数を減らさなければ画面更新間隔の時間内に干渉チェックサブルーチン処理が終了しないおそれがあるものと判断し、干渉チェック用線分 20 の本数 C が、現在の本数 C から 1 本を引いた数 $C - 1$ 本に調整される。ただし、この演算の結果、本数 C が、1 よりも小さくなった場合には、本数 C は 1 に設定される（ステップ 204）。

【0077】

一方、ステップ 203 の判断の結果、処理時間 t が、標準更新時間 t_{std} 以下であると判断された場合には（ステップ 203 の判断 NO）、干渉チェックサブルーチン処理に使用する干渉チェック用線分 20 の本数を増やしても画面更新間隔の時間内に干渉チェックサブルーチン処理が終了する余裕があるものと判断し、干渉チェック用線分 20 の本数 C が、現在の本数 C に 1 本を加えた数 $C + 1$ 本に調整される。ただし、この演算の結果、本数 C が、全本数 N_{max} よりも大きくなった場合には、本数 C は、干渉チェック用線分 20 の全本数 N_{max} に設定される（ステップ 205）。

【0078】

以上のように、干渉チェックサブルーチン処理に使用する干渉チェック用線分 20 の本数 C を調整することで、干渉チェックサブルーチン処理に要する時間が調整されて、画面更新間隔の時間 t が標準更新時間 t_{std} 付近に調整される。

【0079】

以下のステップ 206 からステップ 210 では、干渉チェック用線分 20 が、ワーク 10、周辺装置 80（ポジショナ 9、スライダ 4）の 3 次元モデル 10M、80M（9M、4M）を構成するポリゴン 91 に交差したか否かを判定することをもって、作業ロボット 10 が、ワーク 10 および周辺装置 80（ポジショナ 9、スライダ 4）に干渉したか否かが判断される。

【0080】

すなわち、初回の干渉チェックサブルーチン処理では、識別番号 $\#1$ の干渉チェック用線分 20 から始まり、識別番号 $\#N$ を順次 +1 インクリメントして（ステップ 207）、干渉チェック（ステップ 208）を、予め設定された本数 C_0 まで行う（ステップ 206、210）。次回の干渉チェックサブルーチン処理では、前回の干渉チェックサブルーチン処理で最後に使用した識別番号（ C_0 番目の番号）の干渉チェック用線分 20 のつぎの識別番号の干渉チェック用線分 20 から始まり（ステップ 207）、干渉チェック（ステップ 208）が行われる毎に、識別番号 N を順次 +1 インクリメントして（ステップ 207）、ステップ 204 あるいはステップ 205 で演算された本数 C まで繰り返し行う。な

お、識別番号#Nが最大数N_{max}に達すると、最初の識別番号N₁に戻る（ステップ207）。以下、同様にして、毎回の干渉チェックサブルーチン処理を、その回の干渉チェックサブルーチン処理実行時に演算された本数C（ステップ204、205）だけ繰り返し行う（ステップ206、210）。

【0081】

図7の干渉チェックサブルーチン処理が終了すると、図6のステップ107にリターンされる。

【0082】

上記ステップ208における干渉チェックは、以下のように行う。

【0083】

すなわち、図4（b）に示すように、干渉チェック用線分20は、作業ロボット1の3次元モデル1Mの表面に対して所定の位置関係で設定されている。ここで、現在の作業ロボット1の3次元モデル1Mの表面の座標位置は、ステップ105で得られている。よって、ある識別番号Nの干渉チェック用線分20の現在の座標位置は、ステップ105で得られた現在の作業ロボット1の3次元モデル1Mの表面の座標位置に基づいて、求めることができる。一方、ワーク10、スライダ4、ポジショナ9の3次元モデル10M、4M、9Mの表面を構成するポリゴン（平面）91の座標位置についても、ステップ105で得られている。

【0084】

そこで、ある識別番号Nの干渉チェック用線分20の座標位置と、ワーク10、スライダ4、ポジショナ9の3次元モデル10M、4M、9Mを構成する各ポリゴン91、91…の座標位置とが比較され、両座標位置が一致しているか否か、つまり、ある識別番号Nの干渉用チェック用線分20が、各ポリゴン（平面）91、91…に交差しているかが判定される（ステップ208）。その結果両者の座標位置が一致しており、ある識別番号Nの干渉用チェック用線分20が、ポリゴン（平面）91、91…に交差している場合には、「干渉あり」と判定され（ステップ208の判断YES）、オペレータに警告を与えるべく、図9に斜線にて示すように、表示画面7a上において、作業ロボット1の3次元モデル1Mのうち、干渉している部位が、他の部位と識別できる警告用の特別の色（たとえば赤色）で表示される。これをもって、ロボットプログラム30の教示データが不適切であったことをオペレータは、認識することができる（ステップ209）。ステップ208の判断の結果、両者の座標位置が一致しておらず、ある識別番号Nの干渉用チェック用線分20が、ポリゴン（平面）91、91…に交差していない場合には、「干渉していない」と判定され（ステップ208の判断NO）、オペレータに上述の警告を与えることなく、つぎの処理に移行される（ステップ210）。すなわち、図8に示すように、表示画面7a上において、作業ロボット1の3次元モデル1が、警告用の色（図9では斜線にて示している）に変化しなければ、ロボットプログラム30の教示データは適切であることをオペレータは、認識することができる。

【0085】

本実施例では、干渉チェック用線分20は、作業ロボット1の3次元モデル1Mの表面から離れた外部位置に設定されているため、作業ロボット1が、ワーク10または周辺装置80（ポジショナ9、スライダ4）に接触する段階ではなく、接近している段階で警告を表示することができる。

【0086】

このように3次元モデル同士が接触する段階ではなく、3次元モデル同士が接近している段階で「干渉あり」として警告を表示させているのは、実際の作業ロボット1には誤差があるため、3次元モデル同士では接近している段階といえども、実際の機器では接触している可能性が高いからである。

【0087】

たとえば図9に示すように、トーチ先端2aが溶接線に沿って移動中に、作業ロボット1がワーク10に干渉（接近）していることを警告する表示（斜線にて示す）が画面7a

10

20

30

40

50

上でなされた場合には、ロボットプログラム30のうち、多層盛溶接を行う前のステップ41における移動点P4の教示データ（ツール先端2aの座標位置およびトーチ姿勢角（X、Y、Z、A、B、C）、ポジション9、スライダ4の各軸位置（U、S、R、S））が不適切であると判断して、このステップ41における教示データが修正される。

【0088】

教示データの修正は、操作盤50を操作して、作業ロボット1および周辺装置80の姿勢を、ステップ41の姿勢に戻した上で行われる。なお、操作盤50を操作する代わりに、キーボード703やマウス804を操作することで、教示データを修正してもよい。

【0089】

こうして、操作盤50の操作に応じて、ロボットコントローラ5で、ロボットプログラム30を修正する処理が実行されると、修正されたロボットプログラム30は、ロボットコントローラ5内のメモリ502に記憶される。ロボットコントローラ5は、修正されたロボットプログラム30をメモリ502から読み出して、ロボットプログラム30に従った駆動制御指令を、実機の作業ロボット1および周辺装置80（ポジション9、スライダ4）に与える。これにより実機の作業ロボット1は、ワーク10や周辺装置80（ポジション9、スライダ4）と干渉することなく、溶接作業を行うことができる。

【0090】

以上の図6、図7における各ステップのうち、3次元モデルの画面表示、3次元モデル同士の交差の検出処理に係るステップ105、106、107、ステップ208に関しては、ビデオカード702で、行われる。よって、既存のビデオカードを利用して、安価にシステムを構築することができる。

【0091】

なお、本実施例では、図3に示すように、作業ロボット1の3次元モデル1Mの表面から離れた外側の位置に、干渉チェック用の線分20を、設定しているが、干渉チェック用線分20を、作業ロボット1の3次元モデル1Mの表面と同一面上に設定してもよい。

【0092】

以上のように、本実施例によれば、図4（b）に示すように、作業ロボット1の3次元モデル1M（ポリゴン90）とは別に干渉チェック用線分20を設定し、線分20と、ワーク10および周辺装置80の3次元モデル10M、80Mを構成するポリゴン91とが交差していることを判定することで、両者の干渉を判定するようにしたので、従来のように、作業ロボット1の3次元モデル1Mを構成するポリゴン（面）90と、ワーク10および周辺装置80の3次元モデル10M、80Mを構成するポリゴン91とが交差していることを判定する場合（図4（a））に比して、干渉チェック処理に要する時間を短縮することができる。このため3次元モデルを単純化せずとも短時間で干渉チェックを行うことができる。これにより本実施例によれば、オフラインティーチングで、作業ロボット1とワーク10および周辺装置80の干渉チェックを行うに際して、干渉チェックの精度、信頼性を高く維持しつつも、干渉チェックに要する時間を減らすことができる。

【0093】

特に本実施例によれば、図3に示すように、作業ロボット1の3次元モデル1Mの各部位のうち、ワーク10および周辺装置80の3次元モデル10M、80Mと干渉する可能性の高い部位ほど、干渉チェック用線分20の設定密度が高くなるように設定したので、干渉チェックの精度向上と時間短縮を両立させることができる。

【0094】

また、本実施例によれば、図7に示すように、干渉チェックサブルーチン処理に使用する干渉チェック用線分20の本数Cを調整することで、干渉チェックサブルーチン処理に要する時間を調整して、画面更新間隔の時間tを標準更新時間tstd付近に調整するようにしたので、画面更新間隔の時間内に、干渉チェックを確実に終了させることができる。このため3次元モデルを単純化せずとも確実に画面更新間隔の時間内に干渉チェックを終了させることができる。これにより本実施例によれば、作業ロボット1の3次元モデル1Mを画面7a上で実際の作業ロボット1と同じ速度で再生動作させるに際して、画面更新

間隔の時間内に干渉チェックを確実に終了させることができるようになり、コマ落ちさせることなく画面 7 a 上で作業ロボット 1 を滑らかに再生動作させることができる。

【図面の簡単な説明】

【0095】

【図 1】図 1 は、実施形態の装置の全体構成を示す図である。

【図 2】図 2 は、実施例の装置のブロック図である。

【図 3】図 3 は、作業ロボットの 3 次元モデルと干渉チェック用線分の位置関係を示す斜視図である。

【図 4】図 4 (a)、(b) はそれぞれ、従来技術の干渉チェック処理を説明する図、実施形態の干渉チェック処理を説明する図である。

【図 5】図 5 は、実施形態のロボットプログラムを例示する図である。

【図 6】図 6 は、実施形態の処理の手順を示すフローチャートである。

【図 7】図 7 は、実施形態の処理の手順を示すフローチャートである。

【図 8】図 8 は、表示画面に擬似的に表示される作業ロボットの 3 次元モデルの斜視図で、干渉がない場合の姿勢を示した図である。

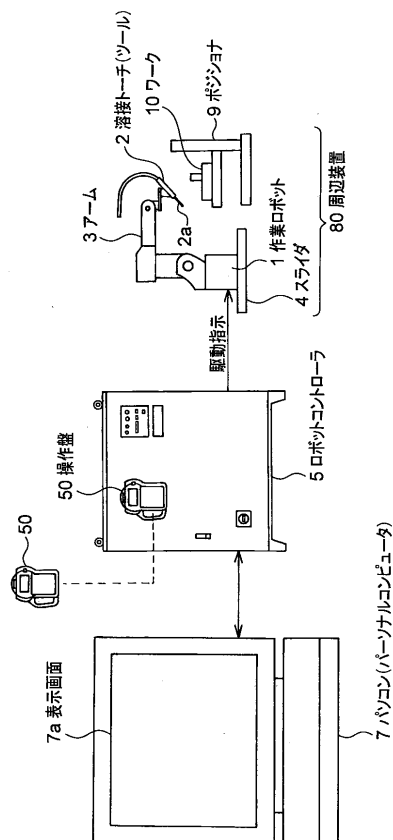
【図 9】図 9 は、表示画面に擬似的に表示される作業ロボットの 3 次元モデルの斜視図で、干渉がある場合の姿勢を示した図である。

【符号の説明】

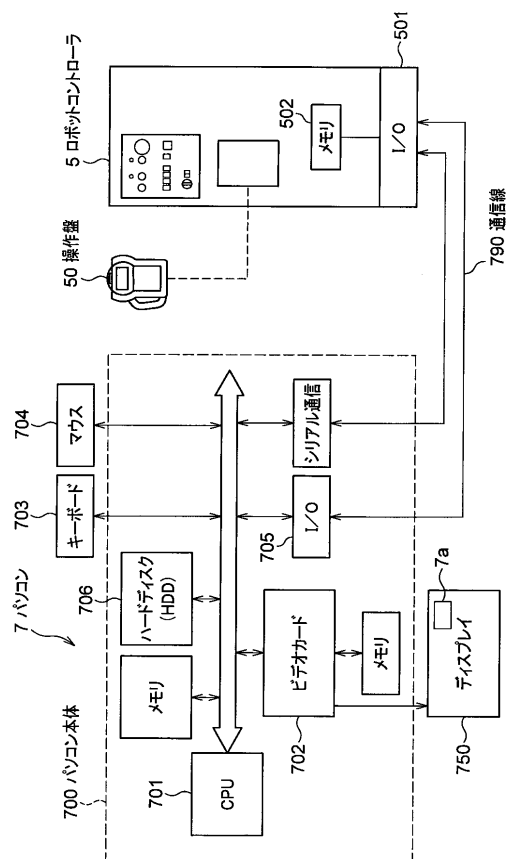
【0096】

1 作業ロボット、4 スライダ、5 ロボットコントローラ、7 パソコン、9 ポジショナ、10 ワーク、20 干渉チェック用線分、80 周辺装置

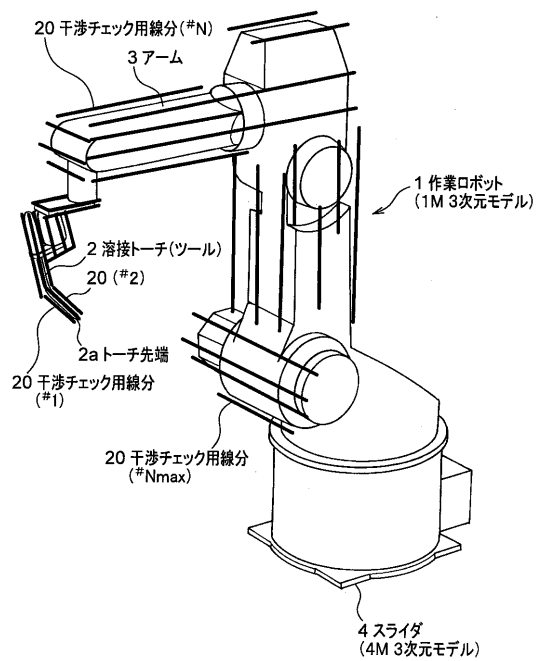
【図 1】



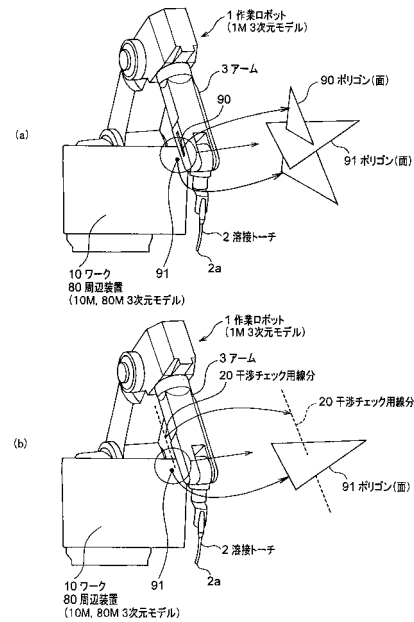
【図 2】



【図 3】



【図 4】

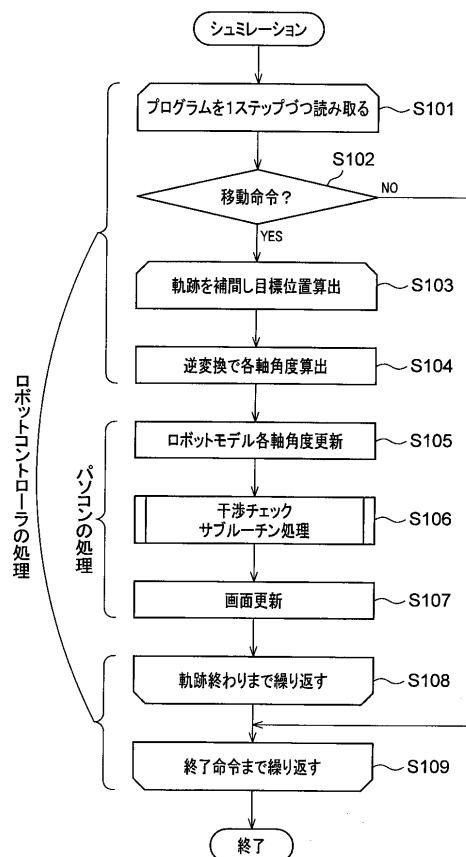


【図 5】

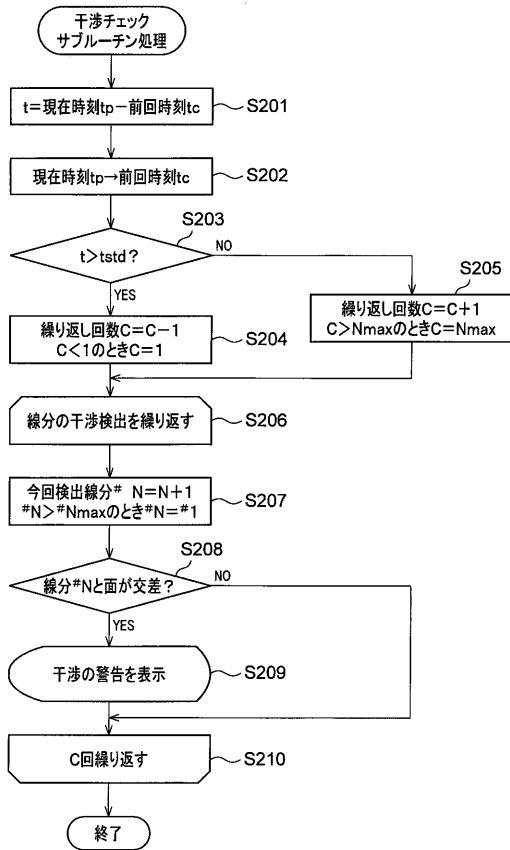
30 ロボットプログラム

33 ラベル	54 [板プラ左内前]	
34 シフト復帰	レジスタ 1	
35 出力	接点 27: オン	
36 出力	接点 31: オフ	
37 EX直線	位置	速度 F2 パス2 P1
38 サーチシフト	下↓→平行	位置 方向 P2
	サーチ速度 20 戻り 5 最大 60	
39 ジョブコール	ジョブ 930	
40 EX直線	位置	速度 F2 パス2 P3
41 EX直線	ポイント 5401	速度 200 パス2 P4
42 溶接	多層盛 条件 170 電源 1	
	記録 54 再生 0 クリア	
	倍率 1.00 スキップ 0.00	
43 EX直線	ポイント 5402	速度 100 パス2 P5

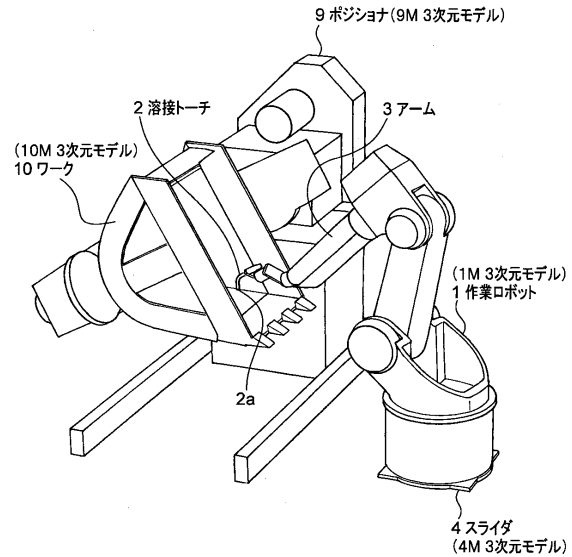
【図 6】



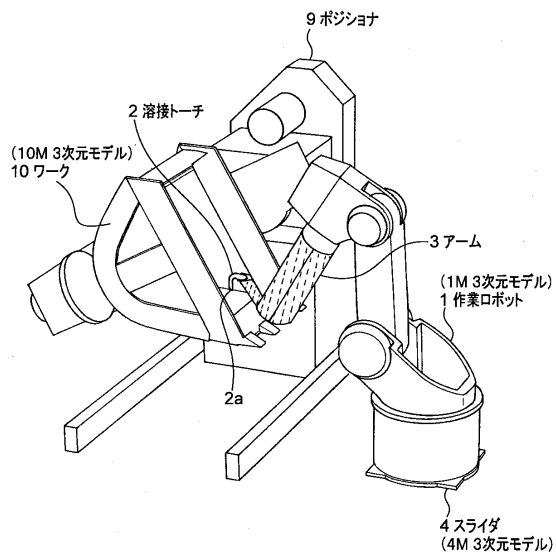
【図 7】



【図 8】



【図 9】



フロントページの続き

(56)参考文献 特開2003-127077 (JP, A)
特開昭64-084310 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G05B 19/18-19/416

G05B 19/42-19/46

B23Q 15/00-15/28

B25J 1/00-21/02