

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6700669号
(P6700669)

(45) 発行日 令和2年5月27日 (2020.5.27)

(24) 登録日 令和2年5月8日 (2020.5.8)

(51) Int.Cl.

B25J 9/10 (2006.01)

F I

B25J 9/10

A

請求項の数 22 (全 32 頁)

(21) 出願番号 特願2015-78716 (P2015-78716)
 (22) 出願日 平成27年4月7日 (2015.4.7)
 (65) 公開番号 特開2016-198828 (P2016-198828A)
 (43) 公開日 平成28年12月1日 (2016.12.1)
 審査請求日 平成30年3月27日 (2018.3.27)

(73) 特許権者 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 110003133
 特許業務法人近島国際特許事務所
 (72) 発明者 根岸 真人
 東京都大田区下丸子3丁目30番2号 キ
 ヤノン株式会社内

審査官 臼井 卓巳

最終頁に続く

(54) 【発明の名称】 制御方法、ロボット装置、プログラム、記録媒体、及び物品の製造方法

(57) 【特許請求の範囲】

【請求項1】

関節のアクチュエータを、関節指令値に従って動作させる制御を行う制御部を備えたロボットの制御方法であって、

前記制御部が、前記関節の位置姿勢に関する誤差である関節誤差を求める関節誤差算出工程と、

前記制御部が、前記アクチュエータを所定方向に動作させた際に、前記アクチュエータの弾性により生じる誤差である動作方向誤差を求める動作方向誤差算出工程と、

前記制御部が、前記関節誤差から前記動作方向誤差を除いた残差を求める残差算出工程と、

前記制御部が、前記残差に基づいて前記関節の動作により移動する部位の位置姿勢に関する誤差である移動部位誤差を求める移動部位誤差算出工程と、

前記制御部が、前記動作方向誤差と、前記移動部位誤差と、に基づいて前記アクチュエータを動作させる動作工程と、

を備えたことを特徴とする制御方法。

【請求項2】

前記動作工程において、

前記制御部が、前記動作方向誤差に基づいて前記関節指令値を補正するための第1補正量を求める第1補正量算出工程と、

前記制御部が、前記移動部位誤差に基づいて前記関節指令値を補正するための第2補正

量を求める第 2 補正量算出工程と、

前記制御部が、前記第 1 補正量及び前記第 2 補正量を用いて前記関節指令値を補正する補正工程と、

を備えたことを特徴とする請求項 1 に記載の制御方法。

【請求項 3】

前記移動部位誤差算出工程において、前記制御部が、前記残差に基づいて順機構計算を行うことで前記移動部位誤差を求め、

前記第 2 補正量算出工程において、前記制御部が、前記移動部位誤差に基づいて逆機構計算を行うことで前記第 2 補正量を求める、

ことを特徴とする請求項 2 に記載の制御方法。

10

【請求項 4】

前記補正工程では、前記制御部が、前記第 1 補正量と前記第 2 補正量とを前記関節指令値から減算する補正を行う、

ことを特徴とする請求項 3 に記載の制御方法。

【請求項 5】

前記制御部は、前記関節指令値を補正する補正モードと、前記関節指令値を補正しない非補正モードと、前記非補正モードから前記補正モードに切り替わる際の遷移モードと、を有しており、

前記補正工程では、前記制御部が、前記遷移モードにおいて、前記第 1 補正量及び前記第 2 補正量のうち少なくとも一方に、時間が経過するに連れて 0 から 1 に増加する係数を掛け算して、前記第 1 補正量及び前記第 2 補正量のうち少なくとも一方を補正し、少なくとも一方が補正された前記第 1 補正量及び前記第 2 補正量を、前記関節指令値から減算する補正を行う、

20

ことを特徴とする請求項 2 から 4 のいずれか 1 項に記載の制御方法。

【請求項 6】

前記制御部は、前記関節指令値を補正する補正モードと、前記関節指令値を補正しない非補正モードと、前記非補正モードから前記補正モードに切り替わる際の遷移モードと、を有しており、

前記補正工程では、前記制御部が、前記遷移モードにおいて、前記第 1 補正量と前記第 2 補正量とを足し合わせた総補正量に、時間が経過するに連れて 0 から 1 に増加する係数を掛け算して前記総補正量を補正し、補正された前記総補正量を前記関節指令値から減算する補正を行う、

30

ことを特徴とする請求項 2 から 4 のいずれか 1 項に記載の制御方法。

【請求項 7】

前記制御部は、前記関節指令値を補正する補正モードと、前記関節指令値を補正しない非補正モードと、前記補正モードから前記非補正モードに切り替わる際の遷移モードと、を有しており、

前記補正工程では、前記制御部が、前記遷移モードにおいて、前記第 1 補正量及び前記第 2 補正量のうち少なくとも一方に、時間が経過するに連れて 1 から 0 に減少する係数を掛け算して、前記第 1 補正量及び前記第 2 補正量のうち少なくとも一方を補正し、少なくとも一方が補正された前記第 1 補正量及び前記第 2 補正量を、前記関節指令値から減算する補正を行う、

40

ことを特徴とする請求項 2 から 4 のいずれか 1 項に記載の制御方法。

【請求項 8】

前記制御部は、前記関節指令値を補正する補正モードと、前記関節指令値を補正しない非補正モードと、前記補正モードから前記非補正モードに切り替わる際の遷移モードと、を有しており、

前記補正工程では、前記制御部が、前記遷移モードにおいて、前記第 1 補正量と前記第 2 補正量とを足し合わせた総補正量に、時間が経過するに連れて 1 から 0 に減少する係数を掛け算して前記総補正量を補正し、補正された前記総補正量を前記関節指令値から減算

50

する補正を行う、

ことを特徴とする請求項 2 から 4 のいずれか 1 項に記載の制御方法。

【請求項 9】

前記残差算出工程では、前記制御部が、前記ロボットの前記部位におけるヤコビ行列の行列式の絶対値に応じて、0 から 1 までの間の重み係数を、前記残差に掛け算して前記残差を補正する、

ことを特徴とする請求項 1 から 8 のいずれか 1 項に記載の制御方法。

【請求項 10】

前記残差算出工程では、前記制御部が、前記絶対値が小さくなるほど、前記重み係数が小さくなるようにする、

ことを特徴とする請求項 9 に記載の制御方法。

【請求項 11】

前記関節誤差算出工程では、前記制御部が、前記関節の変形による前記関節の第 1 関節誤差と、前記ロボットの製作誤差による前記関節の第 2 関節誤差から、前記関節誤差を求める、

ことを特徴とする請求項 1 から 10 のいずれか 1 項に記載の制御方法。

【請求項 12】

前記関節誤差算出工程では、前記制御部が、前記関節の変形による前記関節の第 1 関節誤差と、前記ロボットの製作誤差による前記関節の第 2 関節誤差から、前記関節誤差を求め、

前記第 1 補正量算出工程では、前記制御部が、前記関節誤差算出工程で求めた前記関節誤差に含まれる前記動作方向誤差として、前記第 1 関節誤差に含まれる前記動作方向誤差を用いる、

ことを特徴とする請求項 2 から 8 のいずれか 1 項に記載の制御方法。

【請求項 13】

前記製作誤差のパラメータが記憶部に記憶されており、

前記関節誤差算出工程では、前記制御部が、前記第 2 関節誤差を求める際に、前記記憶部から前記製作誤差のパラメータを読み出す、

ことを特徴とする請求項 11 又は 12 に記載の制御方法。

【請求項 14】

前記関節誤差算出工程では、前記制御部が、前記ロボットの温度に応じて前記製作誤差のパラメータを補正する、

ことを特徴とする請求項 13 に記載の制御方法。

【請求項 15】

前記ロボットの前記部位が嵌合する嵌合部材が、前記ロボットの動作領域に配置されており、

前記制御部が、前記ロボットの前記部位を前記嵌合部材に嵌合させたときの前記ロボットの前記部位の実際の位置姿勢と、目標の位置姿勢との誤差に基づき、前記第 2 関節誤差を求める、

ことを特徴とする請求項 11 から 14 のいずれか 1 項に記載の制御方法。

【請求項 16】

前記アクチュエータは、減速機を備えており、

前記動作方向誤差は、前記減速機における前記所定方向に基づく変形により生じる誤差である、

ことを特徴とする請求項 1 から 15 のいずれか 1 項に記載の制御方法。

【請求項 17】

前記部位は、前記ロボットの先端であり、

前記移動部位誤差は、前記先端の位置姿勢に関する誤差である、

ことを特徴とする請求項 1 から 16 のいずれか 1 項に記載の制御方法。

【請求項 18】

前記制御部が、前記関節にかかる力を算出する力算出工程をさらに備え、
前記動作方向誤差算出工程において、前記制御部が、前記力算出工程の算出結果に基づ
いて、前記動作方向誤差を求める、
ことを特徴とする請求項 1 から 17 のいずれか 1 項に記載の制御方法。

【請求項 19】

制御部が、ロボットの関節のアクチュエータを、関節指令値に従って動作させる制御を行うロボット装置であって、

前記制御部が、前記関節の位置姿勢に関する誤差である関節誤差を求め、

前記制御部が、前記アクチュエータを所定の方に動作させた際に、前記アクチュエータの弾性により生じる誤差である動作方向誤差を求め、

前記制御部が、前記関節誤差から前記動作方向誤差を除いた残差を求め、

前記制御部が、前記残差に基づいて前記関節の動作により移動する部位の位置姿勢に関する誤差である移動部位誤差を求め、

前記制御部が、前記動作方向誤差と、前記移動部位誤差と、に基づいて前記アクチュエータを動作させる、

ことを特徴とするロボット装置。

【請求項 20】

コンピュータに、請求項 1 から 18 のいずれか 1 項に記載の制御方法の各工程を実行させるためのプログラム。

【請求項 21】

請求項 20 に記載のプログラムを記録した、コンピュータ読み取り可能な記録媒体。

【請求項 22】

請求項 1 から 18 のいずれか 1 項に記載の制御方法により前記ロボットを制御して物品を製造する物品の製造方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、関節指令値に従ってロボットの関節の動作を制御する制御方法、ロボット装置、プログラム、記録媒体、及び物品の製造方法に関する。

【背景技術】

【0002】

近年、組立等の生産に従事するロボット装置の開発事例が増えている。このようなロボット装置の究極の目的は、人間の手のような複雑で高速な組立作業をロボットに実現させることである。ロボットは、指令軌道、即ち各関節の関節指令値（回転関節の場合は角度を示す関節指令値、直動関節の場合は長さを示す関節指令値）に従って制御される。しかし、ロボットの指令軌道とロボットの実際の軌道との間には、差（軌道誤差）が存在する。この軌道誤差がロボットに精密な作業を行わせる際の問題となっていた。例えば、精密な部品の組み立てには 0.1 ミリメートル以下の誤差に抑える必要があった。

【0003】

これに対し、多関節のロボットにおけるたわみ分を補正し、位置姿勢の精度を向上させ、高精度な動作を可能とする技術が提案されている（特許文献 1）。

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】特許第 3808321 号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

従来技術では、各関節のたわみによるロボット先端の誤差を計算し、その誤差を補正する各関節指令値の補正量を計算していた。ここで、一般に多関節のロボットは、片持ち構

10

20

30

40

50

造なので剛性が低く、たわみが大きい。そして、各関節の位置姿勢誤差には、関節の駆動方向の誤差のほか、別の方向の誤差も含まれている。各関節の位置姿勢誤差の影響として、ロボット先端の誤差は、数ミリメートルになる。全長が1[m]を超える大きいアームの場合は、10ミリメートルを超える場合も考えられる。このロボット先端の誤差から、各関節の関節指令値の補正量を計算するには、一般的に逆機構計算（逆運動学計算）を用いる。逆機構計算では、関節のたわみなど、関節の位置姿勢に誤差が無い場合を想定して数式を用いて解が導かれる。このため、逆機構計算によって計算した、各関節の関節指令値を補正する補正量には、かならず計算誤差が含まれる。そして、ロボット先端の誤差が大きい場合、その誤差を補正する時に発生する補正量の補正誤差も大きい。

【0006】

10

そのため、従来技術では、一回の計算で各関節の関節指令値の補正量を十分な精度で求めることができないので、収束計算、即ちループ計算を必要としていた。しかし、この補正計算をロボットの動作中に計算する、例えば各関節の関節指令値を出力する時間間隔を2[ms]としたときには、2[ms]といった短い時間内に収束計算をする必要があった。収束計算に要する計算時間を短くするために、収束判定の制限を緩和すると、補正誤差が増大してしまい、高精度な補正が困難となっていた。

【0007】

そこで、本発明は、計算量を低減し、かつロボット先端の位置姿勢誤差を低減することを目的とする。

【課題を解決するための手段】

20

【0008】

本発明の制御方法は、関節のアクチュエータを、関節指令値に従って動作させる制御を行う制御部を備えたロボットの制御方法であって、前記制御部が、前記関節の位置姿勢に関する誤差である関節誤差を求める関節誤差算出工程と、前記制御部が、前記アクチュエータを所定方向に動作させた際に、前記アクチュエータの弾性により生じる誤差である動作方向誤差を求める動作方向誤差算出工程と、前記制御部が、前記関節誤差から前記動作方向誤差を除いた残差を求める残差算出工程と、前記制御部が、前記残差に基づいて前記関節の動作により移動する部位の位置姿勢に関する誤差である移動部位誤差を求める移動部位誤差算出工程と、前記制御部が、前記動作方向誤差と、前記移動部位誤差と、に基づいて前記アクチュエータを動作させる動作工程と、を備えたことを特徴とする。

30

【発明の効果】

【0009】

各関節の関節指令値を補正するための第1補正量は、各関節のアクチュエータの駆動方向の誤差成分を用いて直接求められるので、第1補正量に含まれる補正誤差を小さくすることができる。また、各関節の位置姿勢誤差に含まれるアクチュエータの駆動方向の誤差成分を除いた残差は、アクチュエータの駆動方向の誤差成分よりも小さいので、各関節の残差に基づくロボットの先端の位置姿勢誤差から求まる第2補正量は、第1補正量よりも小さい。したがって、ロボットの先端の位置姿勢誤差から各関節の第2補正量を求めた際に第2補正量に補正誤差が含まれていたとしても、第2補正量は小さいので第2補正量に含まれる補正誤差も小さくなる。したがって、第1補正量及び第2補正量のいずれも補正誤差が小さいので、収束計算を行う必要がなく、補正量の算出に要する計算量、即ち計算時間も低減することができる。また、補正誤差の小さい第1補正量及び第2補正量を用いて関節指令値が補正されるので、ロボットの先端の位置姿勢誤差が低減され、高精度にロボットを動作させることができる。

40

【図面の簡単な説明】

【0010】

【図1】第1実施形態に係るロボット装置の概略構成を示す斜視図である。

【図2】第1実施形態に係るロボット装置のロボット制御装置を示すブロック図である。

【図3】第1実施形態に係るロボット制御方法を示すフローチャートである。

50

【図 4】各関節の位置姿勢誤差を求めるための説明図である。

【図 5】(a) は、式 (2 2) の重み係数を示す関数を示すグラフである。(b) は、式 (2 9) の係数を示す関数を示すグラフである。(c) は、式 (3 0) の係数を示す関数を示すグラフである。

【図 6】シミュレーションに用いたロボットの軸構成を示す模式図である。

【図 7】シミュレーションに用いたロボットを示す斜視図である。

【図 8】評価したロボットの姿勢を示す斜視図である。

【図 9】第 2 実施形態に係るロボット制御方法を示すフローチャートである。

【図 1 0】第 3 実施形態に係るロボット装置のロボットの斜視図である。

【図 1 1】第 3 実施形態に係るロボット装置のロボット制御装置を示すブロック図である

10

。【図 1 2】第 3 実施形態に係るロボット制御方法を示すフローチャートである。

【図 1 3】(a) は、第 3 実施形態におけるリンクの模式図である。(b) は、温度によって変形したリンクを示す模式図である。

【図 1 4】第 4 実施形態に係るロボット装置におけるロボット及びその周辺部材を示す斜視図である。

【図 1 5】(a) は、ロボットハンドを位置マーク部材に嵌合する前の状態を示す模式図である。(b) は、ロボットハンドを位置マーク部材に嵌合した状態を示す模式図である

。【発明を実施するための形態】

20

【 0 0 1 1 】

以下、本発明を実施するための形態を、図面を参照しながら詳細に説明する。ここで、関節は、1 自由度の回転関節又は直動関節を意味する。関節指令値とは回転関節の場合は角度の指令値、直動関節の場合は長さの指令値である。この 2 種類の関節の動作は異なるが、数式上は両者を区別せずに表現できる。そこで、以下の説明では、主に回転関節について説明するが、直動関節の場合についても同じである。

【 0 0 1 2 】

まず、以下の実施形態において数式で使う記号を示す。3 方向の平行移動と 3 つのオイラー角で表現した回転移動からなる、6 次元のベクトル v を 4×4 の同次座標変換行列 T に変換する式を次のように記す。

30

【 0 0 1 3 】

【数 1】

$$T = \text{CoordTrans}(v) \quad (1)$$

【 0 0 1 4 】

より具体的に表現する。 $\text{Cos}(v_3)$ を C_3 、 $\text{Sin}(v_3)$ を S_3 などと略記すると次式となる。

【 0 0 1 5 】

【数 2】

40

$$T = \begin{bmatrix} C_5 C_6 & S_4 S_5 C_6 - C_4 S_6 & C_4 S_5 C_6 + S_4 S_6 & v_1 \\ C_5 S_6 & S_4 S_5 S_6 + C_4 C_6 & C_4 S_5 S_6 - S_4 C_6 & v_2 \\ -S_5 & S_4 C_5 & C_4 C_5 & v_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

【 0 0 1 6 】

また、その逆変換を次のように記す。

【 0 0 1 7 】

【数 3】

$$v = \text{CoordTrans}^{-1}(T) \quad (3)$$

【0018】

以下の実施形態において数式で使う主な記号と用語をまとめる。

【0019】

【数 4】

q_j	各関節の動作量。 j は関節番号で6関節のロボットの場合は1から6の整数である。 回転関節の場合は回転角度である。 直動関節の場合は移動距離である。	10
Σ_j	各関節に固定した座標系。 j は関節番号、また、 $j=0$ はグローバル座標系を表す。 $j=e$ はロボットの先端に固定した座標系を表す。 例 Σ_0 グローバル座標系 Σ_2 第2関節に固定した座標系 Σ_e ロボットの先端に固定した座標系、	
iT_j	第 i 座標系 Σ_i から、第 j 座標系 Σ_j への座標変換行列。 例 1T_2 第1関節の座標系から第2関節の座標系への座標変換行列。 0T_e グローバル座標系からロボット先端の座標系への座標変換行列。	20
Tr_j	第 j 関節の固定端の位置姿勢を表す座標変換行列。 関節の取付位置を表す。	
ΔTr_j	上記 Tr_j の誤差を表す座標変換行列。	
r_j	上記 ΔTr_j を平行移動成分とオイラー角成分で表現したベクトル。 すなわち $r_j = \text{CoordTrans}^{-1}(\Delta Tr_j)$	
$Tv_j(q)$	第 j 関節の移動端の位置姿勢を表す座標変換行列。 回転関節の場合は角度 q_i の回転行列である。	30
ΔTv_j	上記 $Tv_j(q)$ の誤差を表す座標変換行列。	
v_j	上記 ΔTv_j を平行移動成分とオイラー角成分で表現したベクトル すなわち $v_j = \text{CoordTrans}^{-1}(\Delta Tv_j)$	
ΔTu_j	上記2種類の誤差 ΔTr_j と ΔTv_j を統合した座標変換行列。	
w_i	関節にかかる力とモーメント。 力とモーメントの6要素を持つベクトルでレンチとも呼ばれる	
J	ロボット先端の位置姿勢に対するヤコビ行列	
D	上記ヤコビ行列の行列式の値	

【0020】

40

[第1実施形態]

図1は、本発明の第1実施形態に係るロボット装置の概略構成を示す斜視図である。図1に示すように、ロボット装置100は、多関節のロボット200と、ロボット200の動作を制御する制御部であるロボット制御装置300と、を備えている。また、ロボット装置100は、ロボット制御装置300に複数の教示点のデータを送信する教示装置としてのティーチングペンダント400を備えている。ティーチングペンダント400は、人が操作するものであり、ロボット200やロボット制御装置300の動作を指定するのに用いる。

【0021】

ロボット200は、垂直多関節のロボットである。具体的に説明すると、ロボット200 50

0 は、垂直多関節のロボットアーム 201 と、ロボットアーム 201 の先端に取り付けられたエンドエフェクタとしてのロボットハンド 202 と、を備えている。以下、エンドエフェクタがロボットハンド 202 である場合について説明するが、これに限定するものではなく、ツール等であってもよい。

【0022】

ロボットアーム 201 の基端は、台座 B に固定された固定端である。ロボットアーム 201 の先端は、自由端である。第 1 実施形態において、ロボット 200 の先端とは、ロボットアーム 201 の先端、即ちロボットハンド 202 のことである。

【0023】

ロボット 200、即ちロボットアーム 201 は、複数の関節、例えば 6 つ関節（6 軸） $J_1 \sim J_6$ を有している。ロボットアーム 201 は、各関節 $J_1 \sim J_6$ を各関節軸 $A_1 \sim A_6$ まわりにそれぞれ回転駆動する複数（6 つ）のアクチュエータ 10 を有している。

【0024】

ロボットアーム 201 は、複数のリンク（フレーム） $210_0 \sim 210_6$ が各関節 $J_1 \sim J_6$ で回転可能に連結されている。ここで、基端側から先端側に向かって、リンク $210_0 \sim 210_6$ が順に直列に連結されている。ロボットアーム 201 は、可動範囲の中であれば、任意の 3 次元位置で任意の 3 方向の姿勢に、ロボット 200 の先端（ロボット 200 の手先）、即ちロボットハンド 202 を向けることができる。

【0025】

ロボットアーム 201 の位置及び姿勢は、座標系で表現することができる。座標系 0 は、ロボットアーム 201 の基端、即ち台座 B に固定した座標系を表し、座標系 e はロボットアーム 201 の先端に固定した座標系を表す。

【0026】

各関節 $J_1 \sim J_6$ に設けたアクチュエータ 10 は、各関節 $J_1 \sim J_6$ を駆動する。即ち、各アクチュエータ 10 は、関節 J_i で連結された一対のリンク 210_{i-1} 、 210_i のうち、一方のリンク 210_{i-1} に対して他方のリンク 210_i を相対的に駆動する。ここで、6 軸のロボット 200 の場合、 $i = 1 \sim 6$ である。関節 J_i （固定端に対して相対的に移動する移動端）には、座標系 i が固定されている。ここで、関節 J_i の固定端とは、リンク 210_{i-1} のことであり、関節 J_i の移動端とは、リンク 210_i のことである。

【0027】

つまり、リンク 210_i の一端が、関節 J_{i-1} の移動端であり、リンク 210_i の他端が、関節 J_i の固定端である（ただし、 $i = 2 \sim 5$ ）。リンク 210_0 については、リンク 210_0 の一端が台座 B に固定されており、リンク 210_0 の他端が関節 J_1 の固定端である。リンク 210_6 については、リンク 210_6 の一端が移動端である。リンク 210_6 の他端には、ロボットハンド 202 が取り付けられている。グローバル座標系 0 は、リンク 210_0 の一端に設けられている。

【0028】

各アクチュエータ 10 は、電動モータ 11 と、電動モータ 11 に接続された減速機 12 とからなる。各減速機 12 は、直接、又は不図示のベルト、ベアリング又はギア等の伝達部材を介してそれぞれの関節 $J_1 \sim J_6$ で相対的に駆動されるリンクに接続されている。

【0029】

各減速機 12 には、エンコーダ等の関節角度検出部 13 が配置されており、各関節 $J_1 \sim J_6$ の角度を検出できる。各関節 $J_1 \sim J_6$ の角度を検出することで、ロボット制御装置 300 がロボット 200 の先端の位置姿勢を計算でき、ロボットアーム 201 の実際の軌道を求めることができる。

【0030】

なお、各アクチュエータ 10 の構成は、これに限定するものではなく、例えば人工筋肉等であってもよい。

【0031】

10

20

30

40

50

ロボットアーム 201 の内部には、各アクチュエータ 10 の電動モータ 11 を駆動制御する駆動制御部としてのサーボ制御装置 230 が配置されている。

【0032】

サーボ制御装置 230 は、各関節 $J_1 \sim J_6$ の関節位置が指令軌道に追従するよう各電動モータ 11 に電流を出力し、各電動モータ 11 の動作を制御する。ここで、第 1 実施形態では、各関節 $J_1 \sim J_6$ が回転関節であるので、関節位置は関節角度である。指令軌道は、各関節 $J_1 \sim J_6$ の関節指令値、第 1 実施形態では各関節 $J_1 \sim J_6$ が回転関節であるので角度指令値である。

【0033】

なお、サーボ制御装置 230 が 1 つの制御装置で構成されている、即ち、複数の電動モータ 11 を統括して駆動制御するものとして説明するが、各電動モータ 11 にそれぞれ対応した関節制御部で各電動モータ 11 を駆動制御するようにしてもよい。

10

【0034】

図 1 には 4 つの教示点 p_1, p_2, p_3, p_4 を図示している。これらの教示点は、例えば人がティーチングペンダント 400 を用いてロボット制御装置 300 に与える。ここで、教示点 p_1, p_2, p_3, p_4 はいわゆる 3 次元空間内の点ではない。

【0035】

ロボットアーム 201 (ロボット 200) の自由度を表すパラメータを関節角度として、ロボットアーム 201 の 6 関節 $J_1 \sim J_6$ の関節角度をそれぞれ $\theta_1 \sim \theta_6$ とする。ロボットアーム 201 のコンフィグレーションは $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ で表され、関節空間上では、1 つの点とみなすことができる。このように、ロボットアーム 201 の自由度を表すパラメータを座標軸の値とした場合、ロボットアーム 201 のコンフィグレーションは関節空間上の点として表現することができる。つまり、関節空間は、ロボットアーム 201 (ロボット 200) の関節角度を座標軸とする空間である。

20

【0036】

また、ロボット 200 の先端、即ちロボットハンド 202 には、ツールセンターポイント (TCP) が設定されている。TCP は、位置を表す 3 つのパラメータ (x, y, z) と、姿勢 (回転) を表す 3 つのパラメータ (α, β, γ) 、即ち 6 つのパラメータ $(x, y, z, \alpha, \beta, \gamma)$ で表され、タスク空間上では、1 つの点としてみなすことができる。つまり、タスク空間は、これら 6 つの座標軸で規定された空間である。

30

【0037】

ロボット 200 を関節空間で移動させる場合では、教示点は関節角度である。タスク空間移動の場合では、教示点は 3 次元空間の位置および姿勢であり、これは図 1 のように、 p_1, p_2, p_3, p_4 を座標系として表現される。つまり教示点はいわゆる 3 次元空間内の点ではなく、一つのベクトルとして理解できる。

【0038】

ロボット制御装置 300 は、コンピュータで構成されている。ロボット制御装置 300 は、教示点を結ぶ軌道、すなわち教示点を補間した軌道を生成する。

【0039】

具体的に説明すると、ロボット制御装置 300 は、教示点間を補間するロボット 200 の経路 (補間経路) を生成する。ロボット 200 の経路とは、関節空間におけるロボット 200 のコンフィグレーションの軌跡、もしくはタスク空間におけるロボット 200 の TCP の軌跡である。換言すれば、ロボット 200 の経路とは、関節空間又はタスク空間の点の順序集合である。ロボット 200 の軌道とは、時間をパラメータとして経路を表したものであり、第 1 実施形態では、時刻毎のロボット 200 の各関節 $J_1 \sim J_6$ の関節指令値 (角度指令値) の集合である。ロボット 200 の関節は複数あるため、各関節 $J_1 \sim J_6$ で関節指令値の同期がとられる。教示点間を補間する補間方法としては、Spline 補間、B-Spline 補間、ベジェ曲線、直線補間、円弧補間、関節補間などがある。

40

【0040】

第 1 実施形態では、ティーチングペンダント 400 等により教示された教示点を用いて

50

初期に設定した軌道为目标とする。そのため、タスク空間における軌道为目标軌道 x (図 1 には点線で示す)、サーボ制御装置 230 に指令する関節空間における軌道を指令軌道 u という。

【0041】

これらの軌道は、動作開始から停止までの時間を T_{total} とし、一定の時間間隔 (例えば、2 [ms]) を t とすると、 T_{total} / t 個の要素を持つ時系列データである。特に、指令軌道 u は、各関節 $J_1 \sim J_6$ について、 T_{total} / t 個の要素を持つ時系列データ (関節指令値) である。さらに、教示点はベクトルであるので、目標軌道 x 及び指令軌道 u は、ベクトルの時系列データ、すなわち 2 次元の配列データである。

10

【0042】

ロボット 200、即ちロボットアーム 201 の関節の角度は、ロボット制御装置 300 から指令軌道 u を指令されたサーボ制御装置 230 で駆動制御する。サーボ制御装置 230 は指令軌道 u を入力とし、ロボットアーム 201 を動作させ、動作結果である実際の軌道 y を出力する。実際の軌道 y は目標軌道 x と同じベクトルの時系列データ、すなわち 2 次元の配列データである。なお、目標軌道 x は、各関節に対し、関節角度で与えてもよいし、直交座標系の座標値で与えてもよい。以下、指令軌道 u の関節 $J_1 \sim J_6$ に出力する関節角度の指令値 (関節指令値) を $q_1 \sim q_6$ と記述する。

【0043】

次に、ロボット制御装置 300 について説明する。図 2 は、本発明の第 1 実施形態に係るロボット装置のロボット制御装置を示すブロック図である。ロボット制御装置 300 は、コンピュータで構成されており、制御部としての CPU (Central Processing Unit) 301 を備えている。また、ロボット制御装置 300 は、記憶部として、ROM (Read Only Memory) 302、RAM (Random Access Memory) 303、HDD (Hard Disk Drive) 304 を備えている。また、ロボット制御装置 300 は、記録ディスクドライブ 305、各種のインタフェース 311 ~ 314 を備えている。

20

【0044】

CPU 301 には、ROM 302、RAM 303、HDD 304、記録ディスクドライブ 305、各種のインタフェース 311 ~ 314 が、バス 310 を介して接続されている。ROM 302 には、BIOS 等の基本プログラムが格納されている。RAM 303 は、CPU 301 の演算処理結果等、各種データを一時的に記憶する記憶装置である。

30

【0045】

HDD 304 は、CPU 301 の演算処理結果や外部から取得した各種データ (最良指令軌道や最良評価値を含む) 等を記憶する記憶装置であると共に、CPU 301 に、後述する演算処理を実行させるためのプログラム 330 を記録するものである。プログラム 330 は、ロボット制御プログラムを含み、CPU 301 に各種演算処理を実行させるためのプログラムである。CPU 301 は、HDD 304 に記録 (格納) されたプログラム 330 に基づいてロボット制御方法の各工程を実行する。

【0046】

記録ディスクドライブ 305 は、記録ディスク 331 に記録された各種データやプログラム等を読み出すことができる。

40

【0047】

ティーチングペンダント 400 はインタフェース 311 に接続されており、CPU 301 はインタフェース 311 及びバス 310 を介してティーチングペンダント 400 からの教示点のデータの入力を受ける。

【0048】

サーボ制御装置 230 は、インタフェース 314 に接続されており、CPU 301 は、指令軌道 (関節指令値) のデータを所定時間間隔でバス 310 及びインタフェース 314 を介してサーボ制御装置 230 に出力する。

【0049】

50

インタフェース 312 には、モニタ 321 が接続されており、モニタ 321 には、CPU 301 の制御の下、各種画像が表示される。インタフェース 313 は、書き換え可能な不揮発性メモリや外付け HDD 等の外部記憶装置 322 が接続可能に構成されている。

【0050】

なお、第 1 実施形態では、コンピュータ読み取り可能な記録媒体が HDD 304 であり、HDD 304 にプログラム 330 が格納される場合について説明するが、これに限定するものではない。プログラム 330 は、コンピュータ読み取り可能な記録媒体であれば、いかなる記録媒体に記録されていてもよい。例えば、プログラム 330 を供給するための記録媒体としては、図 2 に示す記録ディスク 331、外部記憶装置 322 等を用いてもよい。具体例を挙げて説明すると、記録媒体として、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性メモリ、ROM 等を用いることができる。

10

【0051】

図 3 は、本発明の第 1 実施形態に係るロボット制御方法を示すフローチャートである。この図 3 に示すフローチャートには、プログラム 330 に従ったロボット制御装置 300、具体的には CPU 301 の制御を示す。

【0052】

CPU 301 は、ロボット 200 の各関節の目標角度となる関節指令値 q_i を、教示点のデータに基づいて生成し、誤差を補正するための補正量（補正值）を用いて関節指令値 q_i を補正し、補正後の新たな関節指令値 q_i を出力する。ここで、 q の添え字 i は、本実施形態の 6 軸のロボット 200 の場合、 $i = 1 \sim 6$ である。

20

【0053】

補正後の新たな関節指令値 q_i は、一定の時間間隔、例えば 2 [ms] といった時間間隔で各関節 $J_1 \sim J_6$ の電動モータ 11 を駆動制御するサーボ制御装置 230 に指令される。したがって、図 3 に示すフローチャートの処理も、その時間間隔ごとに実行される。前述したように、この時間間隔を t で表す。また、本実施形態では、関節 $J_1 \sim J_6$ は回転関節として説明するが、関節が直動関節であっても、数式を含めて同じ議論である。

【0054】

CPU 301 は、目標関節角度である関節指令値 q_i から、ロボット 200 の幾何パラメータを用い、ロボット 200 の先端の位置姿勢を計算する。この計算方法は、周知の順機構学計算（順運動学計算）により行う。

30

【0055】

第 i 関節 J_i から第 j 関節 J_j までの座標変換行列を ${}^i T_j$ と記述すると、CPU 301 は、次式（4）でグローバル座標系 ${}_0$ から見たロボット 200 の先端の位置姿勢 ${}^0 T_e$ を計算する。

【0056】

【数 5】

$${}^0 T_e = {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6 {}^6 T_e = \left(\prod_{i=1,6} {}^{i-1} T_i \right) {}^6 T_e \quad (4)$$

40

【0057】

さらに、各関節の座標変換行列 ${}^i T_j$ を次式（5）のように 2 段に分けて考える。

【0058】

【数 6】

$${}^{i-1} T_i = Tr_i Tv_i(q_i) \quad (5)$$

【0059】

ここで、 Tr_i は、第 $(i-1)$ 関節 J_{i-1} の移動端（座標系 ${}_{i-1}$ の原点）に対

50

する第 i 関節 J_i の固定端の位置姿勢である。関節 J_i はアクチュエータ 10 により動作するが、この座標変換行列で、関節 J_i の固定端、即ちアクチュエータ 10 の固定端の位置姿勢を表す。例えば Tr_2 は、第 1 関節 J_1 の移動端（座標系 Σ_1 の原点）に対する第 2 関節 J_2 の固定端の位置姿勢である。この座標変換行列はロボットアーム 201 を設計する時に決定される固定値である。

【 0 0 6 0 】

Tv_i は、第 i 関節 J_i の固定端に対する移動端の位置姿勢である。この座標変換行列でアクチュエータ 10 の移動端、即ち関節の移動端の位置姿勢を表す。関節指令値 q_i を変数に持つ座標変換行列で、関節の動作を表す。6 軸のロボット 200 の場合は回転座標変換となる。図 1 に示す 6 軸のロボット 200 の場合、 Tv_1 は、関節 J_1 の移動端が、
10
関節 J_1 の固定端に対して A_1 軸回りに相対的に q_1 回転する回転座標変換行列になる。

【 0 0 6 1 】

また、CPU 301 は、ステップ S1 において、ロボット 200 の関節角度に対するロボット 200 の先端の次式 (6) のヤコビ行列 J を計算する。

【 0 0 6 2 】

【数 7】

$$J = \frac{\partial \text{CoodTrans}^{-1}({}^0T_e)}{\partial q_i} \quad (6)$$

20

【 0 0 6 3 】

さらに、CPU 301 は、このヤコビ行列の行列式の絶対値 D を計算する。ヤコビ行列や行列式の具体的な計算手順については周知なので、詳細な説明は省略する。

【 0 0 6 4 】

次に、CPU 301 は、関節指令値 q_i を時間微分し、関節 J_i の角度の角速度と角加速度を計算する (S2)。具体的には、CPU 301 は、一定時間間隔 t ごとに与えられる関節指令値 q_i を次式 (7)、(8) のように数値微分する。

【 0 0 6 5 】

【数 8】

$$\dot{q}_i = \frac{q_i - q_i^{(1)}}{\Delta t} \quad (7)$$

30

$$\ddot{q}_i = \frac{\dot{q}_i - \dot{q}_i^{(1)}}{\Delta t} \quad (8)$$

【 0 0 6 6 】

式 (7)、(8) 中の右肩の (1) は、前回の値を表す。CPU 301 は、この計算が終わった後、式 (9)、(10) のように前回の値を更新する。

【 0 0 6 7 】

【数 9】

$$q_i^{(1)} \leftarrow q_i \quad (9)$$

40

$$\dot{q}_i^{(1)} \leftarrow \dot{q}_i \quad (10)$$

【 0 0 6 8 】

矢印は代入の意味である。次に、CPU 301 は、関節指令値 q_i 、角速度及び角加速度から、ロボット 200 の動力学パラメータを用い、関節 J_i にかかる力及びトルク w_i を計算する。この計算方法は逆動力学計算として知られている。動力学パラメータとは、各関節 J_i を構成する部品の重心位置や質量、慣性テンソル等である。また、 w_i は、力とトルクの 6 要素からなるベクトルで、レンチと呼ばれる。

50

【 0 0 6 9 】

ここで、第1実施形態では、関節 J_i は、アクチュエータ 10 と、一対のリンク 210_{i-1} 、リンク 210_i と、で構成されている。アクチュエータ 10 の入力軸には、一対のリンク 210_{i-1} 、リンク 210_i のうち的一方が固定され、アクチュエータ 10 の出力軸には、一対のリンク 210_{i-1} 、リンク 210_i のうち他方が固定されている。したがって、第1実施形態では、後述する関節 J_i の位置姿勢誤差 u_i として、関節 J_i の固定端で発生する位置姿勢誤差 r_i と、関節 J_i の移動端で発生する位置姿勢誤差 v_i との2つに分類することができる。また、これらの誤差が発生する原因も異なる。

【 0 0 7 0 】

関節 J_i の固定端で発生する位置姿勢誤差 r_i の主な原因は、製作誤差である。これは、関節 J_i のリンクを設置する組立て誤差や、部品の製作誤差等である。

10

【 0 0 7 1 】

関節 J_i の移動端で発生する位置姿勢誤差 v_i の主な原因は、アクチュエータの外力による変形である。これはアクチュエータ 10 を構成する軸受けや減速機 12 の変形である。

【 0 0 7 2 】

CPU301は、関節 J_i の変形による関節 J_i の位置姿勢誤差（第1位置姿勢誤差） v_i を求める（S4：関節誤差算出工程）。位置姿勢誤差 v_i は、関節 J_i の変形に起因するリンク 210_{i-1} に対するリンク 210_i の相対的な位置姿勢誤差、即ち、関節 J_i の固定端に対する移動端の相対的な位置姿勢誤差である。ステップS4では、CPU301は、関節 J_i にかかる力及びトルク w_i と関節変形のパラメータから、関節 J_i の移動端の位置姿勢誤差 v_i を計算する。関節変形のパラメータとは、剛性行列や粘性行列などであり、記憶部（例えばHDD304）に予め記憶されている。ここでは剛性を考慮する。剛性行列を K_i とすると以下の式（11）となる。

20

【 0 0 7 3 】

【数10】

$$v_i = K_i^{-1} w_i \quad (11)$$

【 0 0 7 4 】

また、この関節 J_i の移動端の位置姿勢誤差 v_i を座標変換行列で表現すると次のようになる。

30

【 0 0 7 5 】

【数11】

$$\Delta T v_i = \text{CoordTrans}(v_i) \quad (12)$$

【 0 0 7 6 】

次に、CPU301は、ロボット200の製作誤差による関節 J_i の位置姿勢誤差（第2位置姿勢誤差） r_i を求める（S5：位置姿勢誤差算出工程）。位置姿勢誤差 r_i は、リンク 210_{i-1} の製作誤差に起因する、第 $(i-1)$ 関節 J_{i-1} の移動端（座標系 $i-1$ の原点）に対する第 i 関節 J_i の固定端の相対的な位置姿勢誤差である。ステップS5では、CPU301は、ロボット200の製作誤差のパラメータから、関節 J_i の固定端の位置姿勢誤差 r_i を計算する。

40

【 0 0 7 7 】

ここで、ロボット200の製作誤差のパラメータ、即ち関節 J_i の固定端の位置姿勢誤差 r_i は、並進3方向と回転3方向の6要素からなるベクトルである。この製作誤差は、ロボット200の製作時に3次元座標測定装置などを用いて、直接計測したり、ロボット200を製作した後に同定したりすることにより、予め求めて記憶部（例えばHDD304）に記憶させておく。CPU301は、この製作誤差のパラメータを記憶部（例えばH

50

D D 3 0 4) から読み出して、次式 (1 3) により、関節 J_i の固定端の位置姿勢誤差 r_i から座標変換行列を求める。

【 0 0 7 8 】

【 数 1 2 】

$$\Delta Tr_i = CoordTrans(r_i) \quad (13)$$

【 0 0 7 9 】

以上で 2 種類の誤差、即ち関節移動端の位置姿勢誤差 Tv_i と関節固定端の位置姿勢誤差 Tr_i が計算できる。つまり、関節 J_i の固定端と移動端で発生する 2 種類の誤差を、後のステップで補正することができる。

10

【 0 0 8 0 】

次に、C P U 3 0 1 は、2 種類の誤差 Tv_i , Tr_i から、関節 J_i の位置姿勢誤差 Tu_i を計算する (S 6 : 関節誤差算出工程) 。

【 0 0 8 1 】

図 4 は、各関節の位置姿勢誤差を求めるための説明図である。第 ($i - 1$) 関節 J_{i-1} から第 i 関節 J_i への座標変換行列は次の式 (1 4) のように書ける。

【 0 0 8 2 】

【 数 1 3 】

$${}^{i-1}T_i = Tr_i \Delta Tr_i Tv_i(q_i) \Delta Tv_i \quad (14)$$

20

【 0 0 8 3 】

ステップ S 1 で示した座標変換行列と比較すると、誤差の項が追加されているところが異なる。C P U 3 0 1 は、統合した関節の位置姿勢誤差 Tu_i を、次式 (1 5) で計算する。

【 0 0 8 4 】

【 数 1 4 】

$$\Delta Tu_i = Tv_i(q_i)^{-1} \Delta Tr_i Tv_i(q_i) \Delta Tv_i \quad (15)$$

30

【 0 0 8 5 】

すると、第 ($i - 1$) 関節 J_{i-1} から第 i 関節 J_i への座標変換行列は次のように簡単化される。

【 0 0 8 6 】

【 数 1 5 】

$${}^{i-1}T_i = Tr_i Tv_i(q_i) \Delta Tu_i \quad (16)$$

【 0 0 8 7 】

次工程の準備のため、C P U 3 0 1 は、関節 J_i の位置姿勢誤差 Tu_i をベクトル表現 u_i に変換する。

40

【 0 0 8 8 】

【 数 1 6 】

$$u_i = CoordTrans^{-1}(\Delta Tu_i) \quad (17)$$

【 0 0 8 9 】

関節 J_i の位置姿勢誤差 u_i は、座標系 $i-1$ の原点 (特に、 $i = 2 \sim 6$ の場合は関節 J_{i-1} の移動端) に対する関節 J_i の移動端の相対的な誤差である。

【 0 0 9 0 】

50

CPU301は、以上のステップS4～S6（関節誤差算出工程）により、ロボット200の関節 J_i の位置姿勢誤差 u_i を、各関節 $J_1 \sim J_6$ についてそれぞれ求める。

【0091】

CPU301はステップS4～S6で求めた関節 J_i の位置姿勢誤差 u_i に含まれるアクチュエータ10の駆動方向の誤差成分を用いて関節指令値 q_i を補正するための第1補正量を、各関節 $J_1 \sim J_6$ についてそれぞれ求める（S7：第1補正量算出工程）。

【0092】

第1実施形態では、CPU301は、座標系 $i-1$ における関節 J_i の位置姿勢誤差 u_i から、関節 J_i のアクチュエータ10の駆動方向の誤差成分を分離して第1補正量とする。

10

【0093】

具体例を挙げて説明すると、まず、関節 J_i の駆動方向を表す成分だけ1で、他がゼロのベクトル、 s_i を定義する。例えば、関節 J_i が座標系 $i-1$ で定義されたZ軸回りの回転関節の場合は次のようになる。

【0094】

【数17】

$$s_i = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^t \quad (18)$$

【0095】

20

もしX方向の直動関節の場合は、 $s_i = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^t$ である。

【0096】

このベクトル s_i を用いると、誤差 u_i の中の関節の駆動方向成分は次式（19）のように書ける。

【0097】

【数18】

$$s_i^t u_i \quad (19)$$

【0098】

30

これを第1補正量とする。つまり、式（19）は、関節 J_i の位置姿勢誤差 u_i のうち、関節 J_i の駆動方向の誤差成分 $s_i^t u_i$ のみを抽出する演算である。CPU301は、式（19）により抽出した誤差成分 $s_i^t u_i$ を、第1補正量とする。

【0099】

また、CPU301は、ステップS4～S6で求めた関節 J_i の位置姿勢誤差 u_i から関節 J_i のアクチュエータ10の駆動方向の誤差成分 $s_i^t u_i$ を除いた残差 u_i' を、各関節 $J_1 \sim J_6$ についてそれぞれ求める（S8：残差算出工程）。

【0100】

誤差 u_i から関節 J_i の駆動方向の誤差成分 $s_i^t u_i$ を除いた残差 u_i' は次式（20）で計算できる。

40

【0101】

【数19】

$$u_i' = (1 - s_i^t s_i) u_i \quad (20)$$

【0102】

ここで、式（20）中、右辺かっこの中は、分離した駆動方向の誤差成分をゼロにして除くために使う 6×6 の行列である。

【0103】

次に、CPU301は、補正すべき残差 u_i' の大きさを調節する（S9）。誤差のう

50

ち、関節 J_i のアクチュエータ 10 の駆動方向の誤差成分は関節空間で補正するのに対し、残差 u_i' はタスク空間で補正計算する。後述するように、タスク空間での補正は、残差 u_i' から、グローバル座標系 ${}_0$ におけるロボット 200 の先端の位置姿勢誤差を計算し、これを逆機構計算（逆運動学計算）により各関節 $J_1 \sim J_6$ の補正量を計算する。したがって、逆機構計算における特異点の問題が発生する。特異点の近傍では補正量の値が非常に大きくなり、ロボット 200 の関節が大きく動いてしまう可能性がある。

【0104】

そこで、特異点に対する距離に応じて補正すべき残差の大きさを次式（21）で調節する。

【0105】

10

【数20】

$$u_i' \leftarrow f(D) u_i' \quad (21)$$

【0106】

ここで、矢印は代入の意味である。また、 $f(D)$ は、特異点に対する距離に応じて 0 ~ 1 の値をとる重み係数を表す関数である。この関数 $f(D)$ が特異点上で 0 になるようにすれば、補正すべき誤差がゼロになり、その結果、補正量もゼロになる。つまり、特異点（又は特異点近傍）でロボット 200 の関節が大きく動くことを防止できる。

【0107】

20

具体的にはステップ S1 で求めたヤコビ行列 J の行列式が特異点に近づくに連れてゼロになることを利用して次のように定義する。関数 $f(D)$ は、ヤコビ行列 J の行列式の値を D とし、その絶対値を $|D|$ とし、予め定めた特異点の判定値 D_0 を用いて次式（22）となる。

【0108】

【数21】

$$f(D) = \begin{cases} -2\left(\frac{|D|}{D_0}\right)^3 + 3\left(\frac{|D|}{D_0}\right)^2 & |D| < D_0 \\ 1 & D_0 \leq |D| \end{cases} \quad (22)$$

30

【0109】

図5（a）は、式（22）の重み係数を示す関数を示すグラフである。図5（a）に示すように、絶対値 $|D|$ が小さくなるほど、重み係数 $f(D)$ が小さくなるようにするのが好ましい。

【0110】

以上、CPU301 は、ロボット 200 の先端におけるヤコビ行列の行列式の絶対値 $|D|$ に応じて、0 から 1 までの間の重み係数を、残差 u_i' に掛け算して残差 u_i' を補正する。

【0111】

40

次に、CPU301 は、残差 u_i' による各関節 J_i の位置姿勢誤差 Tu_i' を式（23）により計算する（S10）。

【0112】

【数22】

$$\Delta Tu_i' = \text{CoordTrans}(u_i') \quad (23)$$

【0113】

次に、CPU301 は、各関節 $J_1 \sim J_6$ の残差 u_i' に基づくロボット 200 の先端の位置姿勢誤差を求める（S11：先端誤差算出工程）。

50

【 0 1 1 4 】

具体的には、誤差を含まないロボット 2 0 0 の先端の位置姿勢から、関節の位置姿勢誤差 $T u_i'$ を含むロボット 2 0 0 の先端の位置までの座標変換行列 T_e を式 (2 4) により計算する。

【 0 1 1 5 】

【 数 2 3 】

$$\Delta T_e = \left(\prod_i \left(Tr_i Tv_i(q_i - s_i' u_i) \right) \right)^{-1} \prod_i \left(Tr_i Tv_i(q_i - s_i' u_i) \Delta T u_i' \right) \quad (24)$$

10

【 0 1 1 6 】

右辺の関節動作を表す座標変換 $T v_i$ の引数 $q_i - s_i' u_i$ は、関節 J_i の駆動方向の誤差成分を目標値から差し引く工夫であり、本実施形態の特徴の一つである。この工夫によりロボット 2 0 0 の先端の位置姿勢誤差 T_e の大きさを小さくすることができる。

【 0 1 1 7 】

また、上式 (2 4) を使う代わりに、関節の位置姿勢誤差 $T u_i'$ を順次ロボット 2 0 0 の先端の座標系に座標変換していく、次式 (2 5) の簡便な式で計算してもよい。

【 0 1 1 8 】

【 数 2 4 】

$$\Delta T_e = \prod_i \left({}^e T_i \Delta T u_i' {}^e T_i^{-1} \right) \quad (25)$$

20

【 0 1 1 9 】

次に、CPU 3 0 1 は、S 1 1 にて求めたロボット 2 0 0 の先端の位置姿勢誤差 T_e に基づく関節 J_i の位置姿勢誤差を用いて関節指令値 q_i を補正するための第 2 補正量 q_i を、各関節 $J_1 \sim J_6$ についてそれぞれ求める (S 1 2 : 第 2 補正量算出工程) 。このとき、CPU 3 0 1 は、ロボット 2 0 0 の先端の誤差 T_e から逆機構計算 (逆運動学計算) により、各関節 $J_1 \sim J_6$ の第 2 補正量 q_i を計算する。

【 0 1 2 0 】

ここで、誤差を含む場合と、含まない場合のロボット 2 0 0 の先端の位置から、それぞれ関節角度を計算し、その差を補正量とする方法が考えられる。しかしこの方法は逆機構計算を 2 回実行する必要がある。

30

【 0 1 2 1 】

そこで、第 1 実施形態では、CPU 3 0 1 は、より簡便で実用的な計算としてステップ S 1 で求めたヤコビ行列 J の逆行列を用い、次式 (2 6) のように第 2 補正量 q 、即ち ($q_1, q_2, q_3, q_4, q_5, q_6$) を計算する。

【 0 1 2 2 】

【 数 2 5 】

$$\Delta q = J^{-1} CoordTrans^{-1} \left({}^0 T_e \Delta T_e {}^0 T_e^{-1} \right) \quad (26)$$

40

【 0 1 2 3 】

ここで、右辺のかっこ内の計算は、ロボット 2 0 0 の先端の誤差 T_e をグローバル座標系 $_0$ での表現に変更するものである。

【 0 1 2 4 】

ところで、ロボット 2 0 0 の特異点では、ヤコビ行列 J の逆行列を計算することができないので、一般逆行列で代用する。また、特異点付近ではすでにステップ S 9 で補正すべき残差が調整されて小さくなっているため、第 2 補正量 q_i が大きくなりすぎることはない。

【 0 1 2 5 】

50

CPU301は、ステップS7で分離した駆動方向成分の第1補正量 $s_i^t u_i$ と、第2補正量 q_i とを足し合わせ、その結果を新たな補正量（総補正量） q_i とする（S13）。

【0126】

具体的には次式（27）となる。矢印は代入の意味である。

【0127】

【数26】

$$\Delta q_i \leftarrow \Delta q_i + s_i^t u_i \quad (27)$$

【0128】

10

次に、第1実施形態では、CPU301は、補正強度を調節する（S14）。第1実施形態では、ロボット制御装置300、即ちCPU301は、関節指令値 q_i を補正する補正モードと、関節指令値 q_i を補正しない非補正モードと、を有している。また、CPU301は、非補正モードから補正モードに切り替わる際の遷移モード（第1遷移モード）と、補正モードから非補正モードに切り替わる際の遷移モード（第2遷移モード）と、を有している。

【0129】

補正モードと非補正モードとは、作業者の指示により（例えばティーチングペンダント400からの信号により）切り替えられるが、CPU301は、直ちに補正モード又は非補正モードに切り替えずに、遷移モードを経てから切り替える。図3に示すフローチャートは、CPU301が遷移モードのときのフローチャートであり、CPU301が補正モードのときは、図3においてステップS14が省略される。

20

【0130】

関節指令値 q_i の補正の開始時において、補正の有無を調節する必要がある。そこで第1実施形態では、CPU301は、補正強度を時間とともに0から1に変化する係数を表す関数 $g(t)$ を用いて調節する。

【0131】

【数27】

$$\Delta q_i \leftarrow g(t) \Delta q_i \quad (28)$$

30

【0132】

即ち、CPU301は、遷移モード（第1遷移モード）においては、第1補正量と第2補正量とを足し合わせた総補正量 q_i に、時間が経過するに連れて0から1に増加する係数 $g(t)$ を掛け算して総補正量 q_i を補正する。

【0133】

時間の経過とともに0から1に増加する関数は次式（29）のようなものである。 t_0 は切替時間で、例えば0.5秒といった値である。

【0134】

【数28】

$$g(t) = \begin{cases} 0 & t < 0 \\ t/t_0 & 0 \leq t < t_0 \\ 1 & t_0 \leq t \end{cases} \quad (29)$$

40

【0135】

図5（b）は、式（29）の係数を示す関数を示すグラフである。また、より滑らかに切り替える次式（30）の関数も有効である。

【0136】

【数 2 9】

$$g(t) = \begin{cases} 0 & t < 0 \\ -2(t/t_0)^3 + 3(t/t_0)^2 & 0 \leq t < t_0 \\ 1 & t_0 \leq t \end{cases} \quad (30)$$

【0 1 3 7】

図 5 (c) は、式 (3 0) の係数を示す関数を示すグラフである。

【0 1 3 8】

また、関節指令値 q_i の補正の終了時において、補正の有無を調節する必要がある。そこで第 1 実施形態では、C P U 3 0 1 は、補正強度を時間とともに 0 から 1 に変化する係数を表す関数を用いて調節する。この関数は、図 5 (b) 又は図 5 (c) に示す関数を左右反転し、1 から 0 に減少するようにすればよい。

10

【0 1 3 9】

即ち、C P U 3 0 1 は、遷移モード (第 2 遷移モード) においては、第 1 補正量と第 2 補正量とを足し合わせた総補正量 q_i に、時間が経過するに連れて 1 から 0 に減少する係数を掛け算して総補正量 q_i を補正する。

【0 1 4 0】

次に、C P U 3 0 1 は、各関節 $J_1 \sim J_6$ について、第 1 補正量及び第 2 補正量を用いて関節指令値 q_i を補正する (S 1 5 : 補正工程) 。つまり、C P U 3 0 1 は、最後に求めた補正量 q_i を、式 (3 1) に示すように関節指令値 q_i から減算して、新たな関節指令値 q_i として出力する。

20

【0 1 4 1】

【数 3 0】

$$q_i \leftarrow q_i - \Delta q_i \quad (31)$$

【0 1 4 2】

なお、ステップ S 1 4 では、ステップ S 1 3 にて求めた総補正量 q_i を補正し、補正された総補正量 q_i を関節指令値 q_i から減算する補正を行う場合について説明したがこれに限定するものではない。

30

【0 1 4 3】

即ち、ステップ S 1 4 では、第 1 補正量 $s_i^t u_i$ 及び第 2 補正量 q_i のうち少なくとも一方 (好ましくは、第 1 補正量 $s_i^t u_i$) を補正してもよい。その際、C P U 3 0 1 は、遷移モード (第 1 遷移モード) において、第 1 補正量及び第 2 補正量のうち少なくとも一方に、時間が経過するに連れて 0 から 1 に増加する係数を掛け算して、第 1 補正量及び第 2 補正量のうち少なくとも一方を補正すればよい。そして、C P U 3 0 1 は、ステップ S 1 5 では、少なくとも一方が補正された第 1 補正量及び第 2 補正量を、関節指令値 q_i から減算する補正を行えばよい。

【0 1 4 4】

また、C P U 3 0 1 は、遷移モード (第 2 遷移モード) において、第 1 補正量及び第 2 補正量のうち少なくとも一方に、時間が経過するに連れて 1 から 0 に減少する係数を掛け算して、第 1 補正量及び第 2 補正量のうち少なくとも一方を補正すればよい。そして、C P U 3 0 1 は、ステップ S 1 5 では、少なくとも一方が補正された第 1 補正量及び第 2 補正量を、関節指令値 q_i から減算する補正を行えばよい。

40

【0 1 4 5】

また、C P U 3 0 1 は、ステップ S 1 5 では、補正モードのときには、補正していない第 1 補正量と第 2 補正量とを、関節指令値 q_i から減算する補正を行えばよい。

【0 1 4 6】

以上、関節 J_i の関節指令値 q_i を補正するための第 1 補正量 $s_i^t u_i$ は、関節 J_i のアクチュエータ 1 0 の駆動方向の誤差成分を用いて直接求められるので、第 1 補正量 s

50

${}_i^t u_i$ に含まれる補正誤差を小さくすることができる。

【0147】

また、関節 J_i の位置姿勢誤差 u_i に含まれるアクチュエータの駆動方向の誤差成分を除いた残差 u_i' は、アクチュエータの駆動方向の誤差成分よりも小さい。したがって、各関節の残差 u_i' に基づくロボット 200 の先端の位置姿勢誤差から求まる第 2 補正量 q_i は、第 1 補正量 $s_i^t u_i$ よりも小さい。

【0148】

したがって、ロボットの先端の位置姿勢誤差から各関節の第 2 補正量 q_i を求めた際に第 2 補正量 q_i に補正誤差が含まれていたとしても、第 2 補正量 q_i は小さいので第 2 補正量 q_i に含まれる補正誤差も小さくなる。よって、第 1 実施形態によれば、第 1 補正量及び第 2 補正量を求めるのに、収束計算など、複雑なループ計算を行う必要がないので、総補正量 q_i の算出に要する計算量、即ち計算時間も低減することができる。また、補正誤差の小さい第 1 補正量及び第 2 補正量を用いて関節指令値 q_i が補正されるので、ロボット 200 の先端の位置姿勢誤差が低減され、高精度にロボット 200 を動作させることができる。

【0149】

また、補正しない非補正モードから補正する補正モードに遷移する第 1 遷移モードの時、時間経過とともに 0 から 1 に増加する係数を総補正量（又は第 1 補正量と第 2 補正量のうち少なくとも一方）に掛け算することにより、各関節の急峻な動きが防止される。

【0150】

反対に、補正モードから非補正モードに遷移する第 2 遷移モード時には、時間経過とともに 1 から 0 に減少する係数を総補正量（又は第 1 補正量と第 2 補正量のうち少なくとも一方）に掛け算することにより、各関節の急峻な動きが防止される。このように、各遷移モードにおいてスムーズな切り換えを可能とすることで、各遷移モードにおける各関節の急峻な動きが防止される。

【0151】

また、即ちヤコビ行列の行列式の絶対値 $|D|$ 、即ち特異点までの距離に応じて、求めた残差を補正するので、特異点近傍でも安定してロボット 200 を動作させることができる。

【0152】

次に、シミュレーションした結果について説明する。第 1 実施形態の特徴の一つは、関節 J_i の位置姿勢誤差から、関節 J_i のアクチュエータ 10 の駆動方向成分を分離することである。そこで、この分離計算を行って求めた補正量の補正誤差と、分離計算を行わず求めた補正量の補正誤差とを検討した。

【0153】

図 6 は、シミュレーションに用いたロボット 200 の軸構成を示す模式図である。図 7 は、シミュレーションに用いたロボット 200 を示す斜視図である。座標系 $_0$ はロボット 200 の基端（ベース）即ちリンク 210₀ に固定した座標系でグローバル座標系である。座標系 $_e$ はロボット 200 の先端に固定した座標系である。

【0154】

6 つの関節 $J_1 \sim J_6$ はすべて回転関節で、それぞれの関節 $J_1 \sim J_6$ に固定した座標系を $_1 \sim _6$ 、それぞれの関節 $J_1 \sim J_6$ の回転角度を $q_1 \sim q_6$ と表す。第 5 関節 J_5 と第 6 関節 J_6 の座標系は重なっているが、回転の方向が異なる。

【0155】

図 7 に示したように、座標系 $_0$ の原点（ベース）から第 2 関節 J_2 （座標系 $_2$ の原点）までの距離 L_a を 200 [mm] とする。第 2 関節 J_2 （座標系 $_2$ の原点）から第 3 関節 J_3 （座標系 $_3$ の原点）までの距離 L_b を 260 [mm] とする。第 3 関節 J_3 （座標系 $_3$ の原点）から第 5 関節 J_5 （座標系 $_5$ の原点）までの距離 L_c を 260 [mm] とする。第 5 関節 J_5 （座標系 $_5$ の原点）からロボット 200 の先端（ロボットアーム 201 の先端、座標系 $_e$ の原点）までの距離 L_d を 40 [mm] とする。

【 0 1 5 6 】

図 8 は、評価したロボット 2 0 0 の姿勢を示す斜視図である。ロボット 2 0 0 のベースからロボット 2 0 0 の先端までの座標変換行列⁰T_eの位置を(4 0 0, 0, 4 0 0)とし、ロボット 2 0 0 の先端の姿勢は、図 8 のように水平方向とした。

【 0 1 5 7 】

関節の位置姿勢誤差は、前述したように、関節固定端の位置姿勢誤差 r_j と関節移動端の位置姿勢誤差 v_j の 2 種類ある。

【 0 1 5 8 】

位置誤差については影響が小さいのでゼロとした。姿勢誤差は関節移動端の誤差のうち、関節 J_i のアクチュエータ 1 0 の回転方向を 0 . 5 度、それ以外を 0 . 0 5 度とした。これは関節が減速機 1 2 に接続されており、他の回転方向に対して剛性が低いからである。

【 0 1 5 9 】

以上をまとめると、次のように誤差を設定した。

【 0 1 6 0 】

【 数 3 1 】

$$\begin{aligned}\Delta Tr_i &= \text{CoordTrans}(0 \quad 0 \quad 0 \quad 0.05 \text{deg} \quad 0.05 \text{deg} \quad 0.05 \text{deg}) \\ \Delta Tv_i &= \text{CoordTrans}(0 \quad 0 \quad 0 \quad 0.05 \text{deg} \quad 0.05 \text{deg} \quad 0.5 \text{deg})\end{aligned}\quad (32)$$

【 0 1 6 1 】

以上の計算条件において、まず第 1 実施形態による分離計算有りの補正は、上述した図 3 のフローチャートに従って行った。この計算条件は、特異点近傍ではないので、ステップ S 9 の残差の調整は行わなかった。すなわち $f(D) = 1$ とした。また、補正強度の調整も行わなかった。すなわち $g(t) = 1$ とした。

【 0 1 6 2 】

一方、比較する計算は、関節の駆動方向成分を分離せず、上式(32)に示した関節の誤差からロボット 2 0 0 の先端の位置姿勢誤差を計算し、これと目標値との差を計算して関節の補正量を計算した。

【 0 1 6 3 】

関節の誤差の駆動方向成分を分離しない場合と、分離した場合について、計算結果を次表に示す。

【 0 1 6 4 】

【 表 1 】

ロボット先端の目標位置(4 0 0、0、4 0 0)における補正誤差。単位はmm

分離計算	X	Y	Z
無し	-0.178	-0.071	-0.057
有り(第 1 実施形態)	-0.014	-0.040	-0.003

【 0 1 6 5 】

上記誤差ベクトルの大きさを比較すると、分離計算無しの場合は、0 . 2 0 0 [mm] 分離計算有り(第 1 実施形態)の場合は、0 . 0 4 3 [mm] (約 1 / 5)であった。これにより、補正誤差を約 1 / 5 に改善できる。

【 0 1 6 6 】

以上、第 1 実施形態によれば、関節誤差を精度よく補正できるので、ロボット 2 0 0 を高精度に動作させることができる。

【 0 1 6 7 】

[第 2 実施形態]

次に、本発明の第 2 実施形態に係るロボット装置におけるロボット制御方法について説

明する。図 9 は、本発明の第 2 実施形態に係るロボット制御方法を示すフローチャートである。なお、第 2 実施形態におけるロボット装置の構成は、第 1 実施形態で説明した図 1 及び図 2 に示すロボット装置の構成と同様であるため、説明を省略する。

【 0 1 6 8 】

図 9 に示すフローチャートには、プログラム 3 3 0 に従ったロボット制御装置 3 0 0、具体的には CPU 3 0 1 の制御を示す。

【 0 1 6 9 】

第 2 実施形態では、第 1 実施形態に対し、全体の誤差から関節駆動方向成分を分離する順番が異なる。ここでは、主に異なる点について説明する。図 1 で説明したように、ロボット 2 0 0 の関節 $J_1 \sim J_6$ は、モータ 1 1 と減速機 1 2 とを組合せたアクチュエータ 1 0 により駆動される。減速機 1 2 の回転方向の剛性は、軸受で支持される他の方向に比べて小さい。そのため、関節 $J_1 \sim J_6$ の剛性は、回転方向が最も弱い。したがって、減速機 1 2 の回転方向（アクチュエータ 1 0 の駆動方向）のねじれ変形だけに注目しても、大きな効果を期待できる。

【 0 1 7 0 】

関節 J_i の位置姿勢誤差 u_i は、関節移動端の位置姿勢誤差 v_i と、関節固定端の位置姿勢誤差 r_i の 2 つから構成されている。したがって、減速機 1 2 のねじれ変形を受け持つ関節移動端の位置姿勢誤差 v_i を計算した後に、第 1 補正量を分離しても、大きな効果を期待できる。

【 0 1 7 1 】

図 9 に示すステップ S 2 1 ~ S 2 3、S 2 9 ~ S 3 5 は、それぞれ図 3 に示すステップ S 1 ~ S 3、S 9 ~ S 1 4 と同じであるため、説明を省略する。

【 0 1 7 2 】

ステップ S 2 4 では、CPU 3 0 1 は、ステップ S 4 と同様、関節 J_i にかかる力及びトルク w_i と、関節変形のパラメータから、関節 J_i の移動端の位置姿勢誤差 v_i を計算する。ただし、座標変換行列 $T v_i$ は計算する必要がないので、第 1 実施形態に対して計算が少ない。

【 0 1 7 3 】

ステップ S 2 5 では、CPU 3 0 1 は、第 1 位置姿勢誤差である位置姿勢誤差 v_i から、関節 J_i のアクチュエータ 1 0 の駆動方向の誤差成分を分離して第 1 補正量 $s_i^t v_i$ とする（第 1 補正量算出工程）。即ち、このステップ S 2 5 では、CPU 3 0 1 は、関節 J_i の位置姿勢誤差 v_i に含まれるアクチュエータ 1 0 の駆動方向の誤差成分を用いて第 1 補正量 $s_i^t v_i$ を求める。

【 0 1 7 4 】

そして、ステップ S 2 6 では、CPU 3 0 1 は、以下の式 (33) を用いて残差 v_i' を求める。

【 0 1 7 5 】

【数 3 2】

$$v_i' = (1 - s_i s_i^t) v_i \quad (33)$$

【 0 1 7 6 】

ステップ S 2 7 では、CPU 3 0 1 は、ロボット 2 0 0 の製作誤差パラメータから、関節 J_i の固定端の位置姿勢誤差 r_i と $T r_i$ を計算する。このステップ S 2 7 は図 3 のステップ S 5 と同じである。

【 0 1 7 7 】

ステップ S 2 8 では、CPU 3 0 1 は、残差 v_i' と関節固定端の位置姿勢誤差 $T r_i$ を統合し、関節 J_i の位置姿勢誤差 $T u_i'$ を計算する。ここで、ダッシュ (') が付いているのは、すでに関節駆動方向の成分を分離した後であることを表す。

第 1 実施形態と同様、次の式 (34) 及び (35) で計算する。

10

20

30

40

50

【 0 1 7 8 】

【 数 3 3 】

$$\Delta T v_i' = \text{CoordTrans}(v_i') \quad (34)$$

$$\Delta T u_i = T v_i(q_i)^{-1} \Delta T r_i T v_i(q_i) \Delta T v_i' \quad (35)$$

【 0 1 7 9 】

以上、第 2 実施形態によれば、第 1 実施形態と同様、関節誤差を精度よく補正できるので、ロボット 200 を高精度に動作させることができる。

10

【 0 1 8 0 】

[第 3 実施形態]

次に、本発明の第 3 実施形態に係るロボット装置におけるロボット制御方法について説明する。図 10 は、本発明の第 3 実施形態に係るロボット装置のロボットの斜視図である。図 11 は、本発明の第 3 実施形態に係るロボット装置のロボット制御装置を示すブロック図である。なお、第 3 実施形態において、第 1、第 2 実施形態と同様の構成については同一符号を付して説明を省略する。

【 0 1 8 1 】

第 1 実施形態では関節 J_i の固定端の位置姿勢誤差を、製作誤差のパラメータとして予め与えられるとして説明した。第 3 実施形態では、ロボット 200 の各位置に設けた複数の温度センサの検知値から製作誤差のパラメータを補正する点が第 1 実施形態と異なる。

20

【 0 1 8 2 】

図 10 に示すロボット 200 において、リンク 210_{i-1} ($i = 1 \sim 6$) には、一対の温度センサ $500A_i$, $500B_i$ が固定して設けられている。なお、第 3 実施形態では各リンク 210 に 2 つの温度センサ $500A$, $500B$ を固定する場合について説明するが、温度センサの数は、いくつあってもかまわない。

【 0 1 8 3 】

これらの温度センサ $500A$ ($500A_1 \sim 500A_6$), $500B$ ($500B_1 \sim 500B_6$) は、図 11 に示すように、ロボット制御装置 300B のインタフェース 315 を介してバス 310 に接続されている。CPU 301 は、バス 310 を介して、各温度センサ $500A$, $500B$ の検知結果を取得する。CPU 301 は、プログラム 330B に従って、各関節 $J_1 \sim J_6$ の関節指令値 $q_1 \sim q_6$ を補正する。

30

【 0 1 8 4 】

図 12 は、本発明の第 3 実施形態に係るロボット制御方法を示すフローチャートである。この図 12 に示すフローチャートには、プログラム 330B に従ったロボット制御装置 300、具体的には CPU 301 の制御を示す。

【 0 1 8 5 】

図 12 に示すステップ $S41 \sim S44$ 、 $S47 \sim S56$ は、それぞれ図 3 に示すステップ $S1 \sim S4$ 、 $S6 \sim S15$ と同じであるため、説明を省略する。

【 0 1 8 6 】

ステップ $S45$ において、CPU 301 は、温度センサ $500A$ ($500A_1 \sim 500A_6$), $500B$ ($500B_1 \sim 500B_6$) の値を読み込む。温度センサは、数が多いほど正確に温度の分布がわかるので好ましいが、数が多いほどコストが上がるので、1 つの関節 J_i につき、2 ~ 3 個程度が現実的である。第 3 実施形態では、1 つの関節につき 2 個のセンサを設置した場合について説明する。

40

【 0 1 8 7 】

図 13 (a) は、本発明の第 3 実施形態におけるリンク 210_{i-1} の模式図である。図 13 (b) は、温度によって変形したリンク 210_{i-1} を示す模式図である。図 13 (a) 及び図 13 (b) には、関節 J_{i-1} に固定した座標系 $i-1$ から関節 J_i に固定した座標系 i までの間のリンク 210_{i-1} を図示している。リンク 210_{i-1} に

50

は2個の温度センサ500A_i, 500B_iが取り付けられている。常温(例えば25[
])に対し、温度センサ500A_i, 500B_iにて検知された温度(検知値)の温度
上昇の値をt_{a_i}, t_{b_i}とする。2つの温度センサ500A_i, 500B_iの間の距離
をL₁とし、リンク210_{i-1}の長さ、すなわち座標系_{i-1}の原点と座標系_iの
原点との間の距離をL₂とする。

【0188】

リンク210_{i-1}の線熱膨張係数をとし、座標系_{i-1}から座標系_iへの座
標変換誤差を考える。

【0189】

まず、温度上昇に伴う上下方向、すなわちX方向の伸びは、2つの温度センサ500A_i, 500B_iの
10
の平均値から次のように計算できる。

【0190】

【数34】

$$\frac{\alpha L_2 (ta_i + tb_i)}{2} \quad (36)$$

【0191】

また、Y軸まわりの回転誤差は次のように計算できる。

【0192】

【数35】

20

$$\frac{\alpha L_2 (ta_i - tb_i)}{L_1} \quad (37)$$

【0193】

また、Z軸方向の平行移動成分は次のように計算できる。

【0194】

【数36】

$$-\frac{\alpha L_2^2 (ta_i - tb_i)}{2L_1} \quad (38)$$

30

【0195】

次に、ステップS46において、CPU301は、ロボット200の製作誤差のパラメ
ータに、ステップS45にて計算した温度上昇の影響を足し算し、関節J_iの固定端の位
置姿勢誤差r_iを計算する。

【0196】

具体的には、2つの温度センサ500A_i, 500B_iによる温度t_{a_i}, t_{b_i}を用
い、6自由度ある位置姿勢誤差を次のように計算する。

【0197】

40

【数37】

$$r_i \leftarrow \begin{bmatrix} r_{i,1} + \alpha L_2 (ta_i + tb_i) / 2 \\ r_{i,2} \\ r_{i,3} - \alpha L_2^2 (ta_i - tb_i) / 2L_1 \\ r_{i,4} \\ r_{i,5} + \alpha L_2 (ta_i - tb_i) / L_1 \\ r_{i,6} \end{bmatrix} \quad (39)$$

50

【0198】

ここで矢印は代入を、 $r_{i,j}$ は第 i 関節 J_i のロボット 200 の製作誤差パラメータの第 j 成分を表す。例えば、 $j = 1$ の場合は、 X 方向の平行移動成分である。以上のように、関節の固定側の位置姿勢誤差 r_i を計算し、その座標変換行列 $T r_i$ を計算する。

【0199】

このように、ステップ S45、S46 において、CPU301 は、ロボット 200 の温度に応じて製作誤差のパラメータを補正して、関節 J_i の固定側の位置姿勢誤差 r_i 及びその座標変換行列 $T r_i$ を計算する。

【0200】

このように、第3実施形態では、リンク 210_{i-1} の温度（温度センサ $500A_i$ 、 $500B_i$ の温度検出結果）から、リンク 210_{i-1} の温度変形を計算することができる。具体的には、リンク 210_{i-1} の線熱膨張係数、長さ及び温度を掛け合わせれば、リンク 210_{i-1} の温度による伸びを計算することができる。また、少なくとも2つの温度センサ $500A_i$ 、 $500B_i$ によりリンク 210_{i-1} の温度を検出しているため、リンク 210_{i-1} の曲がりを計算することができる。この温度変形を関節の位置姿勢誤差 r_i に加えることにより、位置姿勢誤差 r_i を補正することができる。

10

【0201】

第3実施形態によれば、ロボット 200 に固定した温度センサ $500A_i$ 、 $500B_i$ の値により、関節 J_i の固定端に生じる位置姿勢誤差 r_i を推定し補正することができる。したがって、温度変化があっても、高精度の動作をするロボット 200 を実現することができる。

20

【0202】

また、第3実施形態における温度による誤差補正は、必ずしも全関節について必要わけではない。例えば、根元に近い第2関節 J_2 は、熱変形による角度変化の影響が大きい。この第2関節 J_2 だけ補正しても効果が期待できる。

【0203】

〔第4実施形態〕

次に、本発明の第4実施形態に係るロボット装置におけるロボット制御方法について説明する。図14は、本発明の第4実施形態に係るロボット装置におけるロボット及びその周辺部材を示す斜視図である。なお、第4実施形態において、第1～第3実施形態と同様の構成については同一符号を付して説明を省略する。第1実施形態では関節の位置姿勢誤差は予め与えられていることを想定していたが、第4実施形態では、実際にロボット 200 を動作させて、関節の位置姿勢誤差 r_j を求める（補正する）部分が異なる。 $j = 1 \sim 6$ である。以下、異なる部分を中心に説明する。

30

【0204】

図14に示す6軸のロボット 200 において、ロボット 200 の先端は、ロボットハンド 202 である。ロボットハンド 202 には、第1嵌合部としての爪 202A が形成されている。ロボット 200 は台座 B に固定されている。また、第2嵌合部としての嵌合穴 270A が形成された嵌合部材である位置マーク部材 270 が、台座 B に固定されている。位置マーク部材 270 は、ロボット 200 の動作領域に配置されている。

40

【0205】

図3のステップ S5、図9のステップ S27 又は図12のステップ S46 において、CPU301 は、製作誤差パラメータを用いて位置姿勢誤差 r_j を求めたときに、位置姿勢誤差 r_j を補正する。具体的には、CPU301 は、ロボットハンド 202 の爪 202A を、位置マーク部材 270 の嵌合穴 270A に嵌合させて、ロボット 200 の先端位置のずれから、関節の補正值を求めて記憶部（例えば HDD304）に記憶させておく。そして、図3のステップ S5、図9のステップ S27 又は図12のステップ S46 では、CPU301 は、記憶部（例えば HDD304）から読み出した補正值で位置姿勢誤差 r_j を補正する。

【0206】

50

なお、ロボット 200 の先端の形状（第 1 実施形態では、ロボットハンド 202 の爪 202A の外形）及び嵌合穴 270A は、四角錐の一部分の形状とするが、これに限定するものではない。例えば円錐の一部分の形状であってもよく、位置決め可能であれば形状は任意でよい。

【0207】

図 15 (a) は、ロボットハンド 202 を位置マーク部材 270 に嵌合する前の状態を示す模式図である。図 15 (b) は、ロボットハンド 202 を位置マーク部材 270 に嵌合した状態を示す模式図である。

【0208】

図 15 (a) 及び図 15 (b) に示すように、ロボット 200 の各関節を動作させて、ロボットハンド 202 の爪 202A を位置マーク部材 270 の嵌合穴 270A に嵌合させる。この嵌合作業は、ロボット 200 における組立作業で通常行われる動作である。例えば、ロボットハンド 202 に取り付けられた力覚センサ（不図示）を使い、ロボット 200 の手先が外力に反応して動くインピーダンス制御を行うことによって、爪 202A を嵌合穴 270A に押し付け、倣わせることができる。

【0209】

CPU 301 は、こうして嵌合した時、各関節のアクチュエータ 10 の位置からロボット 200 の先端の位置を計算する。ロボット 200 の先端の位置は様々な誤差を反映して、当初の目標の位置とは異なる。このロボット 200 の先端の位置の誤差を T_p とする。次に、どの関節が誤差を生じているか仮定する。例えば、第 1 関節を選択すると、誤差 T_p により第 1 関節の取付誤差、つまりロボット 200 の取付位置姿勢を推定する。具体的には、第 j 関節を選択したとし、第 j 関節の取付誤差 T_{ppj} を次の式から推定する。

【0210】

【数 38】

$$\Delta T_{ppj} = \left({}^0T_j^{-1} {}^0T_e \right) \Delta T_p \left({}^0T_j^{-1} {}^0T_e \right)^{-1} \quad (40)$$

【0211】

これは、ロボット 200 の先端の座標系で表現された、ロボット 200 の先端位置の誤差 T_p を選択した関節の座標系に座標変換する式である。これを第 j 関節の位置姿勢誤差 r_j に加える。

【0212】

【数 39】

$$r_j \leftarrow r_j + \text{CoordTrans}^{-1}(\Delta T_{ppj}) \quad (41)$$

【0213】

ここで、矢印は代入を表す。

【0214】

第 4 実施形態によれば、関節の取付位置姿勢が不明な場合でも、これを推定し、補正することで、より高精度にロボットを動作させることができる。

【0215】

なお、本発明は、以上説明した実施形態に限定されるものではなく、本発明の技術的思想内で多くの変形が可能である。また、本発明の実施形態に記載された効果は、本発明から生じる最も好適な効果を列挙したに過ぎず、本発明による効果は、本発明の実施形態に記載されたものに限定されない。

【0216】

〔その他の実施形態〕

本発明は、上述の実施形態の 1 以上の機能を実現するプログラムを、ネットワーク又は

10

20

30

40

50

記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサがプログラムを読み出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

【０２１７】

また、第１～第４実施形態は、ロボットにおけるすべての関節が回転関節として説明したが、これに限定するものではなく、複数の関節のうち一部又は全部が、直動関節であってもよい。

【０２１８】

また、第１～第４実施形態では、垂直多関節のロボットの場合について説明したが、これに限定するものではない。例えば、水平多関節のロボット、パラレルリンクのロボット、直交ロボット等、種々のロボットについて本発明は適用可能である。

10

【０２１９】

また、ロボット装置が、撮像部としてのカメラを更に備えていてもよい。この場合、カメラによりロボットを撮像し、制御装置（ＣＰＵ）が、関節変形のパラメータや製作誤差のパラメータを用いて各関節の位置姿勢誤差を求める代わりに、カメラから得られた撮像画像に基づいて各関節の位置姿勢誤差を求めてもよい。

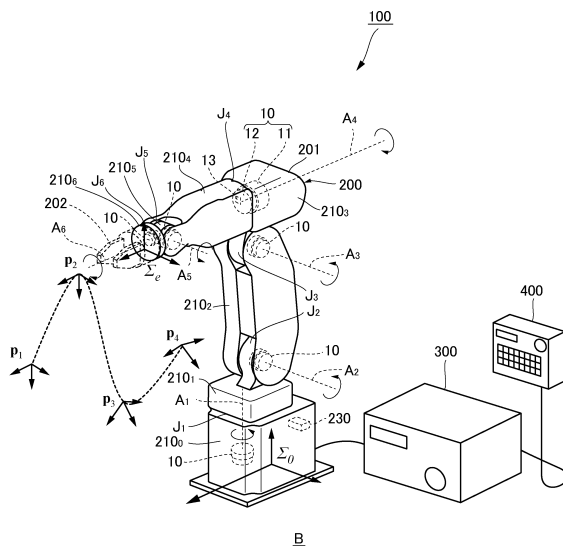
【符号の説明】

【０２２０】

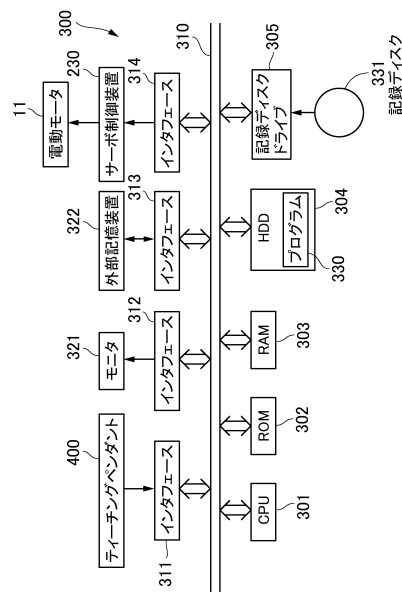
１０…アクチュエータ、１００…ロボット装置、２００…ロボット、３００…ロボット制御装置、３０１…ＣＰＵ（制御部）

20

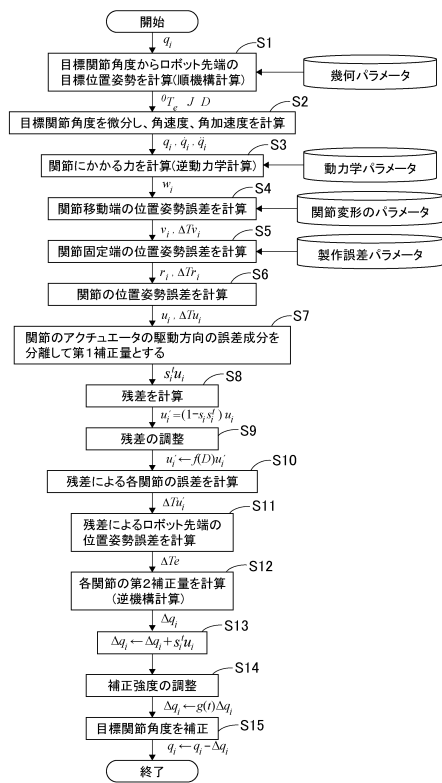
【図１】



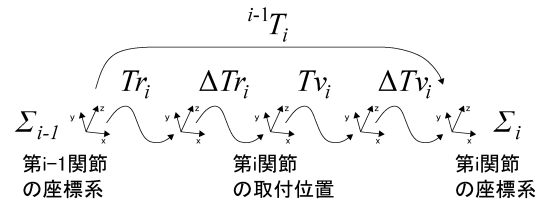
【図２】



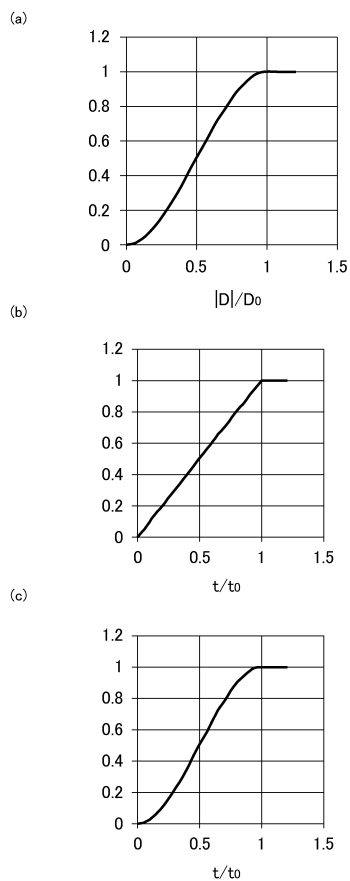
【図 3】



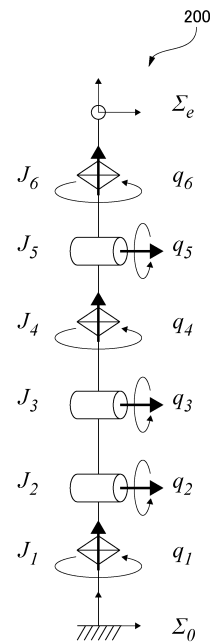
【図 4】



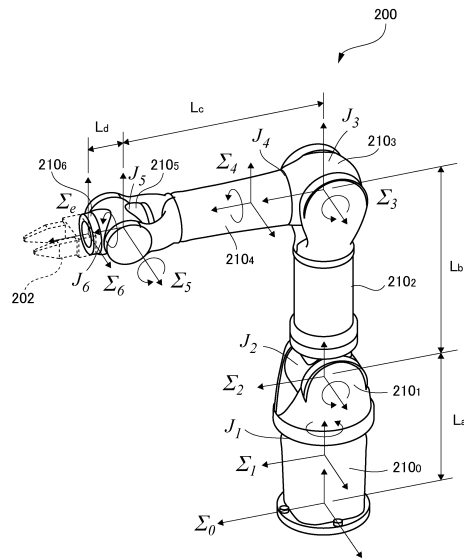
【図 5】



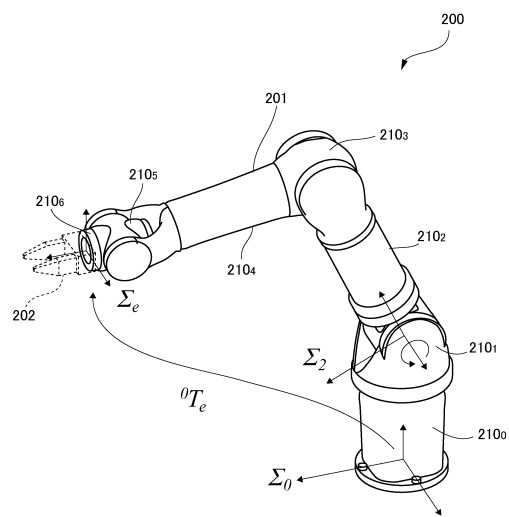
【図 6】



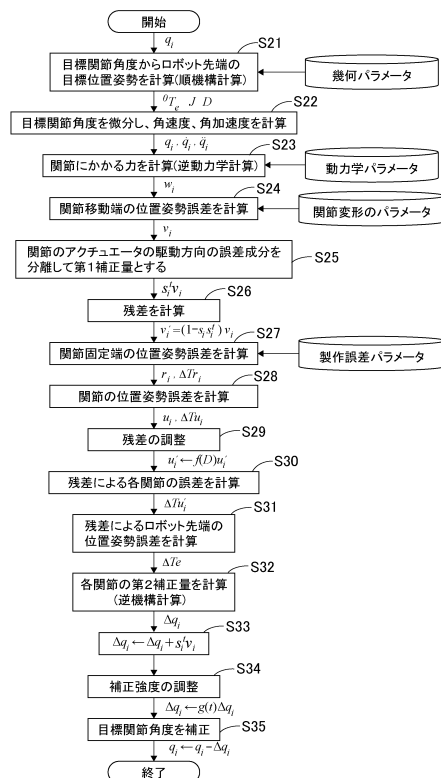
【図 7】



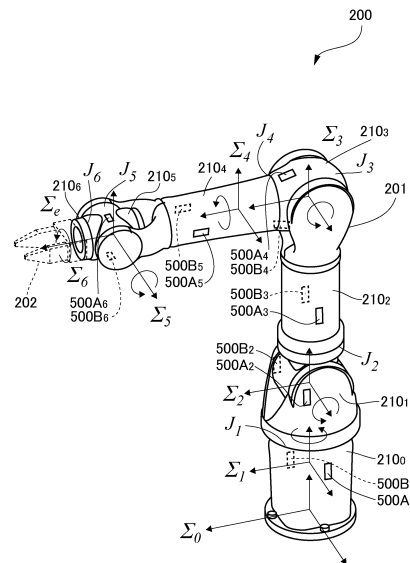
【図 8】



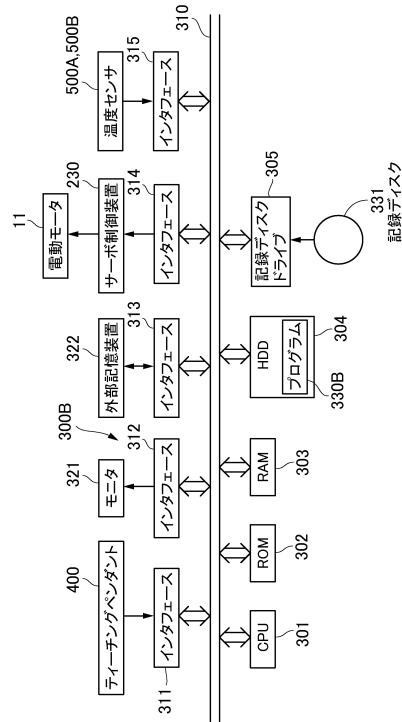
【図 9】



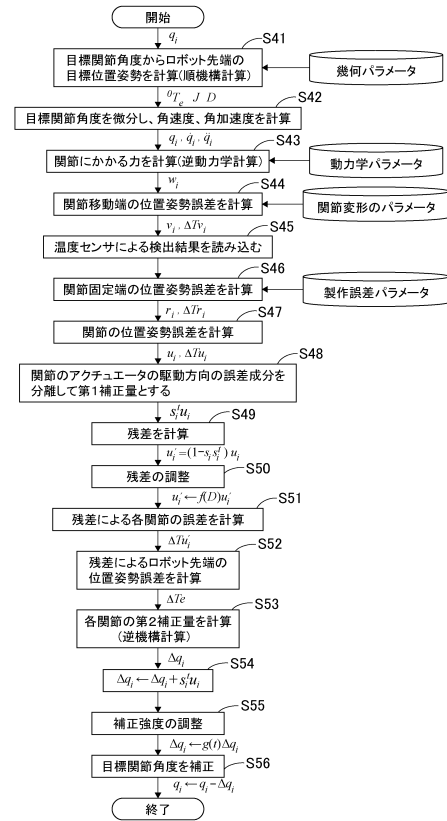
【図 10】



【図 1 1】

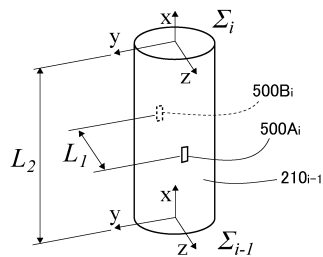


【図 1 2】

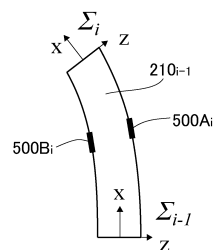


【図 1 3】

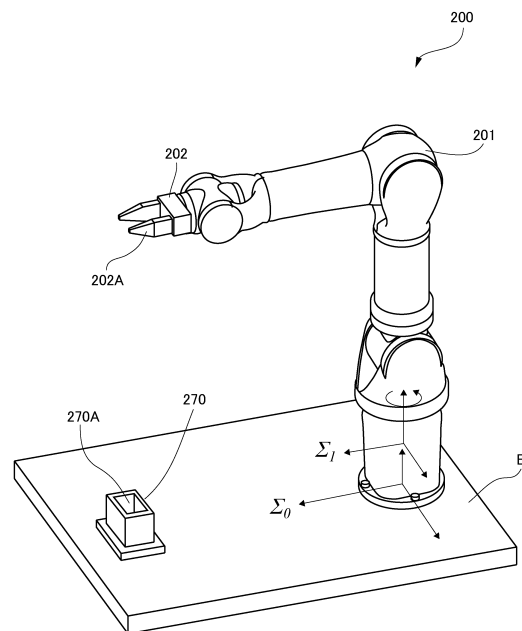
(a)



(b)

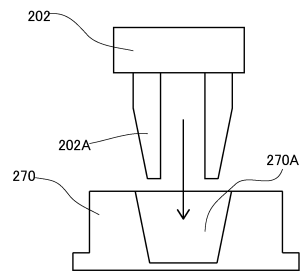


【図 1 4】

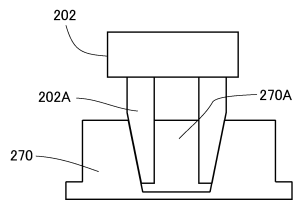


【図 15】

(a)



(b)



フロントページの続き

(56)参考文献 特開2009-142904(JP,A)
特開昭58-137585(JP,A)
特開昭59-107882(JP,A)
特開昭61-281304(JP,A)
特開平04-233602(JP,A)
特開平06-099372(JP,A)
特開平08-108382(JP,A)
特開平10-000582(JP,A)
特開2013-166224(JP,A)
特開2014-104529(JP,A)
特開2014-151427(JP,A)
国際公開第2013/175777(WO,A1)
米国特許第04763276(US,A)

(58)調査した分野(Int.Cl., DB名)

B25J 9/10-13/08
G05B 19/18