

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4743700号  
(P4743700)

(45) 発行日 平成23年8月10日(2011.8.10)

(24) 登録日 平成23年5月20日(2011.5.20)

(51) Int.Cl.

F I

<b>B 2 5 J</b>	<b>13/02</b>	<b>(2006.01)</b>	B 2 5 J	13/02	
<b>B 2 5 J</b>	<b>3/00</b>	<b>(2006.01)</b>	B 2 5 J	3/00	C
<b>B 2 5 J</b>	<b>19/06</b>	<b>(2006.01)</b>	B 2 5 J	19/06	
<b>B 2 5 J</b>	<b>9/16</b>	<b>(2006.01)</b>	B 2 5 J	9/16	
<b>G 0 5 B</b>	<b>19/045</b>	<b>(2006.01)</b>	G 0 5 B	19/045	

請求項の数 9 (全 28 頁)

(21) 出願番号 特願2005-325278 (P2005-325278)  
 (22) 出願日 平成17年11月9日(2005.11.9)  
 (65) 公開番号 特開2007-130714 (P2007-130714A)  
 (43) 公開日 平成19年5月31日(2007.5.31)  
 審査請求日 平成20年3月27日(2008.3.27)

(出願人による申告)平成17年度独立行政法人情報通信研究機構、研究テーマ「超高速知能ネットワーク社会に向けた新しいインタラクション・メディアの研究開発」に関する委託研究、産業活力再生特別措置法第30条の適用を受ける特許出願

(73) 特許権者 393031586  
 株式会社国際電気通信基礎技術研究所  
 京都府相楽郡精華町光台二丁目2番地2  
 (74) 代理人 100067828  
 弁理士 小谷 悦司  
 (74) 代理人 100096150  
 弁理士 伊藤 孝夫  
 (74) 代理人 100109438  
 弁理士 大月 伸介  
 (72) 発明者 荒川 佳樹  
 京都府相楽郡精華町光台二丁目2番地2  
 株式会社国際電気通信基礎技術研究所内

審査官 松浦 陽

最終頁に続く

(54) 【発明の名称】 制御装置、制御方法及び制御プログラム

(57) 【特許請求の範囲】

【請求項1】

制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換する変換手段と、

前記変換手段により変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行する無誤差演算手段とを備え、

前記無誤差演算手段は、前記データ演算処理における除算を可変長分数データ形式で表現し、

前記制御対象は、関節を介して複数のリンクを連結したリンク機構であり、

前記変換手段は、前記リンク機構の所定部位の目標3次元位置データ及び目標方向データからなる目標データをP、前記リンク機構の各関節角度を とし、目標データPから関節角度 を求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換することを特徴とする制御装置。

【請求項2】

前記無誤差演算手段は、前記データ演算処理において除算の結果に余りが発生しない場合は当該除算を実行することを特徴とする請求項1記載の制御装置。

【請求項3】

前記無誤差演算手段は、前記可変長分数データ形式で表現された分母データ及び分子デ

10

20

ータを約分することを特徴とする請求項 1 又は 2 記載の制御装置。

【請求項 4】

前記変換手段は、前記データ演算処理に使用されるデータを固定長数値データ形式から可変長分数データ形式へ変換し、

前記無誤差演算手段は、前記変換手段により変換された可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行することを特徴とする請求項 1 ~ 3 のいずれかに記載の制御装置。

【請求項 5】

前記無誤差演算手段は、 $\theta_i = \theta'_i + \Delta\theta_i$  ( $i = 1, \dots, n$ ) (ここで、 $\theta_i$  は関節  $i$  の求める関節角度、 $\theta'_i$  は微小時間  $t$  前の関節  $i$  の関節角度、 $\Delta\theta_i$  は微小時間  $t$  経過後の関節  $i$  の微小関節角度変化量、 $n$  は求める関節角度 の数) とし、 $n$  元連立一次方程式  $A \mathbf{x} = B$  (ここで、 $A$  はその要素  $a_{ij}$  ( $i = 1, \dots, n, j = 1, \dots, n$ ) がリンク機構のリンク長及び微小時間  $t$  前の関節  $i$  の関節角度  $\theta'_i$  により決定される係数行列、 $B$  はその要素  $b_i$  ( $i = 1, \dots, n$ ) が目標データ  $P$  及び微小時間  $t$  前の位置方向データ  $P'$  により決定されるベクトル) を、可変長分数データ形式のデータを用いて、無誤差演算により解くことを特徴とする請求項 1 ~ 4 のいずれかに記載の制御装置。

10

【請求項 6】

前記変換手段は、前記無誤差演算手段により求められた可変長分数データ形式の関節角度 を固定長数値データ形式へ変換することを特徴とする請求項 1 ~ 5 のいずれかに記載の制御装置。

20

【請求項 7】

操作者の所定部位の 3 次元位置及び方向を検出する検出手段をさらに備え、

前記検出手段により検出された 3 次元位置及び方向を前記リンク機構の所定部位の目標 3 次元位置データ及び目標方向データとして用いて、前記リンク機構の所定部位を目標 3 次元位置及び目標方向に制御することを特徴とする請求項 1 ~ 6 のいずれかに記載の制御装置。

【請求項 8】

制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換する第 1 のステップと、

30

変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行する第 2 のステップとを含み、

前記第 2 のステップは、前記データ演算処理における除算を可変長分数データ形式で表現するステップを含み、

前記制御対象は、関節を介して複数のリンクを連結したリンク機構であり、

前記第 1 のステップは、前記リンク機構の所定部位の目標 3 次元位置データ及び目標方向データからなる目標データを  $P$ 、前記リンク機構の各関節角度を  $\theta$  とし、目標データ  $P$  から関節角度  $\theta$  を求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換するステップを含むことを特徴とする制御方法。

40

【請求項 9】

制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換する変換手段と、

前記変換手段により変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行する無誤差演算手段としてコンピュータを機能させ、

前記無誤差演算手段は、前記データ演算処理における除算を可変長分数データ形式で表

50

現し、

前記制御対象は、関節を介して複数のリンクを連結したリンク機構であり、

前記変換手段は、前記リンク機構の所定部位の目標3次元位置データ及び目標方向データからなる目標データをP、前記リンク機構の各関節角度をとし、目標データPから関節角度を求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換することを特徴とする制御プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、制御対象を制御する制御装置、制御方法及び制御プログラムに関し、特に、関節を介して複数のリンクを連結したリンク機構の所定部位、例えば、先端部を目標3次元位置及び目標方向に制御する制御装置、制御方法及び制御プログラムに関する。

【背景技術】

【0002】

従来のロボットアームの制御方式としては、スレーブアームと同じ軸構成等を有するマスターアームを操作者が操作してスレーブアームを制御するマスターアーム方式、操作者の腕と手の主要部（各関節）にセンサを装着して操作者の腕と手の関節の動きを検出してロボットアームを制御するモーションキャプチャー方式、操作者の手首・手の甲等に付けた3次元位置方向センサにより操作者の手首・手の甲の3次元位置及び方向を検出し、このデータを用いてアーム（リンク）の各関節角度を求める逆変換を行ってロボットアームを制御する逆変換方式等の種々の方式がある。

【0003】

上記のマスターアーム方式では、操作者側には、制御されるスレーブアームだけでなく、マスターアームが必要となり、システムが重装備になる。また、モーションキャプチャー方式では、操作者の腕と手の複数箇所にセンサを装着する必要があるため、操作者にとって煩わしいものとなる。

【0004】

一方、逆変換方式では、操作者は手の甲等に1個のセンサのみを装着するだけで済み、操作者にとって煩わしさが無いが、操作者の手首の位置と方向データから各関節角度を求める逆変換が必要となる。この逆変換では、人の腕のような高い自由度（6自由度以上）を持つ軸構成における各関節角度を求める必要があるため、逆変換が非線形連立方程式となり、複雑な計算となる。

【0005】

このため、本願発明者は、逆変換が非線形連立方程式とならず、単純且つ少ない演算量で厳密解を求めることができるロボットアーム・ハンドの操作方法を提案した（特許文献1参照）。この操作方法では、ロボットアームの操作者の手首に付けたセンサにより検出された3次元位置及び方向からアームの各関節角度を求めるために、浮動小数点演算を用いた誤差有り演算を用いている。この浮動小数点フォーマットを用いた実数表現は、非常に小さな数から大きな数まで表現でき、計算効率がよいという優れた特徴があるため、逆変換のような実数の数値計算に適している。

【特許文献1】特開2005-46931号公報

【発明の開示】

【発明が解決しようとする課題】

【0006】

しかしながら、上記の浮動小数点演算を用いた数値制御では、以下のような問題がある。

（1）浮動小数点演算では、数値演算誤差が発生し、且つその蓄積が起るため、演算結果に大きな誤差が含まれる場合があり、ロボットアームの先端位置が大幅にずれる場合がある。

10

20

30

40

50

(2) 上記の数値演算誤差とその蓄積のために、当初想定していなかった状態が発生し、結果として制御が破綻してロボットアーム動作が暴走する状況が発生する。

(3) 上記のロボットアーム動作の暴走を防止するためには、暴走対策を施す必要があり、制御が複雑化する。

【0007】

また、数値演算処理は、加算、減算、乗算及び除算の四則演算を用いて行われるが、除算は、割り切れない場合に必ず誤差が発生し、除算を含む四則演算すべてを無誤差で実行することは困難である。この場合、除算を使用しないことも考えられるが、通常、数値制御処理には加減乗除算の四則演算が用いられるため、除算が使えないということは、適用できる範囲が狭くなり、実用性が低下する。

10

【0008】

本発明の目的は、制御動作を安定化及び高信頼化することができるとともに、制御対象を高精度に制御することができ、さらに、汎用性の高い且つ適用範囲の広い制御演算を実現することができる制御装置、制御方法及び制御プログラムを提供することである。

【課題を解決するための手段】

【0009】

本発明に係る制御装置は、制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換する変換手段と、前記変換手段により変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行する無誤差演算手段とを備え、前記無誤差演算手段は、前記データ演算処理における除算を可変長分数データ形式で表現し、前記制御対象は、関節を介して複数のリンクを連結したリンク機構であり、前記変換手段は、前記リンク機構の所定部位の目標3次元位置データ及び目標方向データからなる目標データをP、前記リンク機構の各関節角度を  $\theta$  とし、目標データPから関節角度  $\theta$  を求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換するものである。

20

【0010】

本発明に係る制御装置では、制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータが固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換され、変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理が無誤差で実行される。ここで、データ演算処理において除算が発生した場合、当該除算が保留されて可変長分数データ形式で表現されるので、可変長分数データ形式で表現された数値に対するその後の演算を継続することができ、データ演算処理を最後まで実行することができる。また、誤差が発生する除算を実際には実行していないので、数値演算誤差が発生することがなく、その蓄積も起こらない。この結果、除算を含む四則演算すべてを無誤差演算の枠組みで処理することができるので、制御動作を安定化及び高信頼化することができるとともに、制御対象を高精度に制御することができ、さらに、汎用性の高い且つ適用範囲の広い制御演算を実現することができる。

30

40

また、目標データPから関節角度  $\theta$  を求めるデータ演算処理において、除算を含む四則演算すべてを無誤差演算の枠組みで処理することができるので、リンク機構の制御動作を安定化及び高信頼化することができるとともに、リンク機構を高精度に制御することができる。さらに、汎用性の高い且つ適用範囲の広い制御演算を実現することができる。

【0011】

前記無誤差演算手段は、前記データ演算処理において除算の結果に余りが発生しない場合は当該除算を実行することが好ましい。

【0012】

この場合、データ演算処理において除算の結果に余りが発生しないときは当該除算が実

50

行されるので、数値演算誤差の発生及びその蓄積を防止しながら、誤差が発生しない除算を実行して除算の結果を表すデータ量を削減することができ、データ演算処理におけるデータ量の削減及び計算効率の向上を図ることができる。

【0013】

前記無誤差演算手段は、前記可変長分数データ形式で表現された分母データ及び分子データを約分することが好ましい。

【0014】

この場合、可変長分数データ形式で表現された分母データ及び分子データが約分されるので、数値データの桁数及びデータ量を削減することができ、データ演算処理におけるデータ量の削減及び計算効率の向上を図ることができる。

10

【0015】

前記変換手段は、前記データ演算処理に使用されるデータを固定長数値データ形式から可変長分数データ形式へ変換し、前記無誤差演算手段は、前記変換手段により変換された可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行することが好ましい。

【0016】

この場合、可変長分数データ形式のデータを用いてデータ演算処理が無誤差で実行されるので、使用するデータ形式が固定長数値データ形式と可変長分数データ形式との2種類のみ限定され、データの管理が容易となり、また、無誤差演算手段では可変長分数データ形式のデータのみを用いることとなり、無誤差演算手段の構成も簡略化することができる。

20

【0019】

前記無誤差演算手段は、 $\theta_i = \theta'_i + \Delta\theta_i$  ( $i = 1, \dots, n$ ) (ここで、 $\theta_i$  は関節  $i$  の求める関節角度、 $\theta'_i$  は微小時間  $t$  前の関節  $i$  の関節角度、 $\Delta\theta_i$  は微小時間  $t$  経過後の関節  $i$  の微小関節角度変化量、 $n$  は求める関節角度の数) とし、 $n$  元連立一次方程式  $A \cdot \theta = B$  (ここで、 $A$  はその要素  $a_{ij}$  ( $i = 1, \dots, n, j = 1, \dots, n$ ) がリンク機構のリンク長及び微小時間  $t$  前の関節  $i$  の関節角度  $\theta'_i$  により決定される係数行列、 $B$  はその要素  $b_i$  ( $i = 1, \dots, n$ ) が目標データ  $P$  及び微小時間  $t$  前の位置方向データ  $P'$  により決定されるベクトル) を、可変長分数データ形式のデータを用いて、無誤差演算により解くことが好ましい。

30

【0020】

この場合、目標データ  $P$  から関節角度  $\theta$  を求めるデータ演算処理を、可変長分数データ形式の係数データを用いて無誤差で解くことができるので、データ演算処理が除算を含む場合でも、データ演算処理における数値演算誤差の発生及びその蓄積を完全に排除することができる。

【0021】

前記変換手段は、前記無誤差演算手段により求められた可変長分数データ形式の関節角度  $\theta$  を固定長数値データ形式へ変換することが好ましい。

【0022】

この場合、データ演算処理により求められた関節角度  $\theta$  が可変長分数データ形式から固定長数値データ形式へ変換されるので、固定長数値データ形式の関節角度  $\theta$  を用いて通常の制御を行うことができ、種々の制御対象を容易に制御することができる。

40

【0023】

上記制御装置は、操作者の所定部位の3次元位置及び方向を検出する検出手段をさらに備え、前記検出手段により検出された3次元位置及び方向を前記リンク機構の所定部位の目標3次元位置データ及び目標方向データとして用いて、前記リンク機構の所定部位を目標3次元位置及び目標方向に制御することが好ましい。

【0024】

この場合、リンク機構の所定部位を操作者の所定部位の動きに応じた目標3次元位置及び目標方向に高精度に制御することができる。

50

## 【 0 0 2 5 】

本発明に係る制御方法は、制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換する第1のステップと、変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行する第2のステップとを含み、前記第2のステップは、前記データ演算処理における除算を可変長分数データ形式で表現するステップを含み、前記制御対象は、関節を介して複数のリンクを連結したリンク機構であり、前記第1のステップは、前記リンク機構の所定部位の目標3次元位置データ及び目標方向データからなる目標データをP、前記リンク機構の各関節角度をとし、目標データPから関節角度をを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換するステップを含むものである。

10

## 【 0 0 2 6 】

本発明に係る制御プログラムは、制御対象を制御するために用いられる制御データを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換する変換手段と、前記変換手段により変換された可変長整数データ形式又は可変長分数データ形式のデータを用いて、前記データ演算処理を無誤差で実行する無誤差演算手段としてコンピュータを機能させ、前記無誤差演算手段は、前記データ演算処理における除算を可変長分数データ形式で表現し、前記制御対象は、関節を介して複数のリンクを連結したリンク機構であり、前記変換手段は、前記リンク機構の所定部位の目標3次元位置データ及び目標方向データからなる目標データをP、前記リンク機構の各関節角度をとし、目標データPから関節角度をを求めるデータ演算処理に使用されるデータを固定長数値データ形式から可変長整数データ形式、又は可変長整数データ形式の分母データ及び分子データから構成される可変長分数データ形式へ変換するものである。

20

## 【発明の効果】

## 【 0 0 2 7 】

本発明によれば、制御対象を制御するために用いられる制御データを求めるデータ演算処理において、除算が発生した場合、当該除算が保留されて可変長分数データ形式で表現されるので、データ演算処理を最後まで実行することができるとともに、数値演算誤差が発生することがなく、その蓄積も起こらないため、除算を含む四則演算すべてを無誤差演算の枠組みで処理することができ、制御動作を安定化及び高信頼化することができるとともに、制御対象を高精度に制御することができ、さらに、汎用性の高い且つ適用範囲の広い制御演算を実現することができる。

30

## 【発明を実施するための最良の形態】

## 【 0 0 2 8 】

以下、本発明の一実施の形態による制御装置の一例であるロボットアーム制御装置を用いたロボットアーム制御システムについて図面を参照しながら説明する。

## 【 0 0 2 9 】

図1は、本発明の一実施の形態によるロボットアーム制御装置を用いたロボットアーム制御システムの構成を示すブロック図である。図1に示すロボットアーム制御システムは、ロボットアーム制御装置1、ロボットアーム部2、3次元操作グローブ3、3次元位置方向センサ4及び3次元ディスプレイ5を備える。

40

## 【 0 0 3 0 】

ロボットアーム部2は、一对のロボットアーム21及びハンド22、胴体部23並びにカメラ部24を備え、人の上半身(頭部を除く)と同程度の大きさと同程度の自由度とを有し、両腕と両手の機能を実現している。

## 【 0 0 3 1 】

ロボットアーム21は、肩、肘及び手首に関節を有し、これらの関節を介してリンクと

50

なる上腕 2 1 a 及び前腕 2 1 b が連結されたリンク機構（マニプレータ）であり、人の腕と同程度の大きさ及び自由度を有する。ハンド 2 2 は、人の手と同程度の大きさ及び自由度を有する触覚付きの 5 本指ロボットハンドであり、前腕 2 1 b の先端に取り付けられる。ハンド 2 2 には触覚センサ（図示省略）が設けられており、物をつかんだ時の触覚（圧力）がロボットアーム制御装置 1 を介して 3 次元操作グローブ 3 へ伝達される。

【 0 0 3 2 】

なお、図示を省略しているが、ロボットアーム 2 1 及びハンド 2 2 は、その内部にモータ等の駆動源を有し、この駆動源により任意の角度に駆動される。また、本実施の形態では、ロボットアーム 2 1 を人の腕に模倣させ、ハンド 2 2 を人の手指に模倣させているが、ロボットアーム及びハンドの構成は、この例に特に限定されず、ハンド 2 2 の指の数を増減させる等の種々の変更が可能である。また、リンク機構の関節数及びリンク数も、上記の例に特に限定されず、種々の変更が可能である。

10

【 0 0 3 3 】

カメラ部 2 4 は、3 次元カメラ（ステレオカメラ）等から構成され、ロボットアーム制御装置 1 を介して、撮影した画像を 3 次元ディスプレイ 5 へ出力する。3 次元ディスプレイ 5 は、カメラ部 2 4 により撮影された画像を表示し、操作者 M に提示する。なお、撮影された画像の提示方法は、この例に特に限定されず、3 次元ディスプレイに替えて、操作者 M に装着させたヘッドマウントディスプレイに表示させる等の種々の変更が可能である。

【 0 0 3 4 】

3 次元操作グローブ 3 は、公知の 3 次元操作用データグローブから構成され、操作者 M の両手にそれぞれ装着される。3 次元操作グローブ 3 は、操作者 M の手の甲及び指の動き（関節角度）を検出し、検出した関節角度データをロボットアーム制御装置 1 へ出力する。3 次元操作グローブ 3 の手首の位置には、3 次元位置方向センサ 4 が取り付けられ、3 次元位置方向センサ 4 は、前腕 2 1 b の先端部の目標 3 次元位置及び目標方向として、操作者 M の手首の 3 次元位置及び方向を検出し、検出した 3 次元位置データ及び方向データをロボットアーム制御装置 1 へ出力する。

20

【 0 0 3 5 】

なお、目標 3 次元位置及び目標方向を検出又は入力する方法は、上記の例に特に限定されず、データグローブ又はデータスーツ等から操作者の所定部位の 3 次元位置及び方向を検出したり、ハプティクス・デバイス等を用いて入力したりしてもよい。

30

【 0 0 3 6 】

ロボットアーム制御装置 1 は、操作者 M の両手に装着された 3 次元操作グローブ 3 及び 3 次元位置方向センサ 4 からの検出結果を基に、ロボットアーム 2 1 及びハンド 2 2 の各関節角度を算出する。ハンド 2 2 に関しては、3 次元操作グローブ 3 が手の甲の曲がり（手首の）角度及び指の関節角度を検出し、ロボットアーム制御装置 1 は、これらの角度データに基づいてハンド 2 2 を制御する。

【 0 0 3 7 】

一方、ロボットアーム 2 1 に関しては、操作者 M の手首に装着した 3 次元位置方向センサ 4 から操作者 M の手首の 3 次元位置及びその方向、すなわち、ロボットアーム 2 1 の先端部の目標 3 次元位置及び目標方向が得られるだけである。このため、ロボットアーム 2 1 の先端部の目標 3 次元位置データ及び目標方向データから、ロボットアーム 2 1 の各関節角度を計算する必要がある。このように、ロボットアーム 2 1 の先端部の 3 次元位置及び方向からロボットアーム 2 1 の各関節角度を計算することを逆変換と呼び、これとは逆に、ロボットアーム 2 1 の各関節角度からロボットアーム 2 1 の先端部の 3 次元位置及び方向を計算することを順変換と呼ぶことにする。

40

【 0 0 3 8 】

図 2 は、図 1 に示すロボットアーム制御装置 1 の構成を示すブロック図である。図 2 に示すように、ロボットアーム制御装置 1 は、外部インターフェース部 1 1、固定長数値データ記憶部 1 2、データ変換処理部 1 3、可変長分数データ記憶部 1 4、誤差有制御処理

50

部 1 5 及び無誤差制御処理部 1 6 を備える。誤差有制御処理部 1 5 は、固定長数値データ演算処理部 1 5 a 及び浮動小数点演算部 1 5 b を備え、無誤差制御処理部 1 6 は、可変長分数データ演算処理部 1 6 a 及び可変長分数演算部 1 6 b を備える。

【 0 0 3 9 】

ロボットアーム制御装置 1 は、CPU (中央演算処理装置)、ROM (リードオンリメモリ)、RAM (ランダムアクセスメモリ)、外部記憶装置、記録媒体駆動装置、入力装置、表示装置、外部機器インターフェース等を備えるコンピュータから構成することができる。この場合、後述するロボットアーム制御処理 (位置制御処理) を実行するためのロボットアーム制御プログラム (位置制御プログラム) を、CPU 等を用いて実行することにより、外部インターフェース部 1 1、固定長数値データ記憶部 1 2、データ変換処理部 1 3、可変長分数データ記憶部 1 4、誤差有制御処理部 1 5 及び無誤差制御処理部 1 6 としてコンピュータを機能させることができる。なお、ロボットアーム制御装置 1 の構成は、この例に特に限定されず、上記の各機能の一部又は全部を専用のハードウェアから構成する等の種々の変更が可能である。

10

【 0 0 4 0 】

外部インターフェース部 1 1 は、3次元位置方向センサ 4 から3次元位置データ及び方向データを取得して固定長数値データ記憶部 1 2 に格納する。また、外部インターフェース部 1 1 は、ロボットアーム制御処理により求められた関節角度を固定長数値データ記憶部 1 2 から読み出してロボットアーム部 2 へ出力し、ロボットアーム 2 1 の各関節角度がロボットアーム制御処理により求められた関節角度に制御される。

20

【 0 0 4 1 】

固定長数値データ記憶部 1 2 は、誤差有制御処理部 1 5 において処理される種々のデータを固定長数値データ形式で記憶し、可変長分数データ記憶部 1 4 は、無誤差制御処理部 1 6 において処理される種々のデータを可変長分数データ形式で記憶する。

【 0 0 4 2 】

データ変換処理部 1 3 は、固定長数値データ記憶部 1 2 から浮動小数点データを読み出し、読み出した浮動小数点データを可変長分数データへ変換し、変換した可変長分数データを可変長分数データ記憶部 1 4 に格納する。また、データ変換処理部 1 3 は、可変長分数データ記憶部 1 4 から可変長分数データを読み出し、読み出した可変長分数データを浮動小数点データへ変換し、変換した浮動小数点データを固定長数値データ記憶部 1 2 に格納する。

30

【 0 0 4 3 】

なお、本発明に用いられる固定長数値データは、上記の浮動小数点データに特に限定されず、固定長整数データを用いてもよく、この場合、浮動小数点演算部 1 5 b を固定長整数演算部に変更して固定長整数演算部により固定長整数演算を実行するようにしてもよい。例えば、3次元位置方向センサ 4 が出力する値 (角度) が、 $-255 \sim +255$  等の整数値であり、ロボットアーム 2 1 の関節を動作させるモータに対する出力 (回転角度) も、 $-1000 \sim +1000$  パルス等の整数値である場合に固定長整数データを用いることができる。

【 0 0 4 4 】

誤差有制御処理部 1 5 は、ロボットアーム部 2 のロボットアーム 2 1 及びハンド 2 2 の制御処理のうち、固定長数値演算の一例である浮動小数点演算を用いた誤差有制御処理を実行する。固定長数値データ演算処理部 1 5 a は、固定長数値データ記憶部 1 2 から浮動小数点データを読み出し、浮動小数点演算部 1 5 b を用いて座標変換処理等 (sin、cos 演算) を浮動小数点演算で実行し、演算結果を固定長数値データ記憶部 1 2 に格納する。浮動小数点演算部 1 5 b は、加算器、減算器、乗算器及び除算器としての機能を有し、浮動小数点演算を用いて、加算、減算、乗算及び除算を実行する。

40

【 0 0 4 5 】

無誤差制御処理部 1 6 は、ロボットアーム部 2 のロボットアーム 2 1 及びハンド 2 2 の制御処理のうち、可変長分数演算を用いた無誤差制御処理を実行する。可変長分数データ

50



演算処理部 16 a は、可変長分数データ記憶部 14 から可変長分数データを読み出し、制御方程式の求解等を可変長分数演算部 16 b を用いて可変長分数演算で実行し、演算結果を可変長分数データ記憶部 14 に格納する。

【0046】

可変長分数演算部 16 b は、加算器、減算器及び乗算器としての機能を有し、可変長分数演算を用いて、加算、減算及び乗算を無誤差で実行する。また、可変長分数演算部 16 b は、除算器としての機能をも有し、割り切れる場合は除算を行い、その結果を分母を 1 とする可変長分数データ形式で保持し、割り切れない場合は除算を保留し、そのまま可変長分数データ形式で保持し、割り切れないが、約分できる場合は、約分を実行し、その結果を可変長分数データ形式で保持し、以降の演算を進める。

10

【0047】

図 3 は、数値データの種々のフォーマットの例を示す図である。数値データは、固定長数値データと可変長数値データとに大別される。前者（固定長数値データ）の代表例には、計算機においてよく用いられる、図 3 の (a) に示す固定長整数データ（例えば、32 ビット）と、図 3 の (b) に示す浮動小数点データ（例えば、64 ビット）とが該当する。図 3 の (a) に示すように、固定長整数データは、符号部及び数値（整数）部から構成される。図 3 の (b) に示すように、浮動小数点データは、符号部、指数部及び仮数部から構成され、このようなデータ形式で固定長数値データ記憶部 12 に格納される。これらの固定長数値データは、固定長であるため、計算効率が良い反面、数値表現精度が限定され、演算誤差の発生とその蓄積とが起る。

20

【0048】

一方、可変長数値データには、図 3 の (c) に示す可変長整数データと、図 3 の (d) に示す可変長分数データとが該当する。図 3 の (c) ように、可変長整数データは、語長部、符号部及び数値（整数）部から構成される。図 3 の (d) ように、可変長分数データは、語長部、符号部及び数値（整数）部から構成される分子データと、語長部、符号部及び数値（整数）部から構成される分母データとが一对にされたデータであり、このようなデータ形式で可変長分数データ記憶部 14 に格納される。これらの可変長数値データでは、数値表現精度に限界がないため、加減乗算を無誤差で演算することができ、演算誤差の発生とその蓄積とが起らないという利点がある。

【0049】

しかしながら、可変長数値データでは、データ長が固定でないため、計算効率が劣る。また、除算の場合は、割り切れずに循環小数となる場合がある。この場合、無限桁数の数値となるので、可変長数値データ形式でも対応することができず、無誤差演算を行うことはできない。

30

【0050】

そこで、本実施の形態では、可変長分数データ記憶部 14 及び無誤差制御処理部 16 を用いて、可変長整数データを分子と分母とに用いた分数形式で表現して記憶及び演算を行っている。すなわち、被除数  $a/b$ （ここで、 $a$  は可変長整数データ形式の分子データ、 $b$  は可変長整数データ形式の分母データ）を除数  $c/d$ （ここで、 $c$  は可変長整数データ形式の分子データ、 $d$  は可変長整数データ形式の分母データ）で割る除算を実行する場合、除数  $c/d$  の逆数  $d/c$  を乗算して  $a \times d (= e)$  及び  $b \times c (= f)$  の分子部及び分母部の乗算を実行する。次に、乗算後の被除数  $e$  を除数  $f$  で割る除算において、割り切れる場合は除算を行い、その結果を、分母を 1 とする可変長分数データ形式で表現して記憶及び演算を行い、一方、割り切れない場合は除算を保留し、そのまま可変長分数データ形式で記憶し、以降の演算を進める。また、除算において、割り切れないが、約分できる場合は、約分を実行し、約分後の可変長分数データを記憶してその後の演算に用いている。

40

【0051】

上記の方法の利点としては、以下が挙げられる。

(1) 加減乗算に加えて、除算も無誤差演算の枠組みで行うことができるため、非常に汎用性の高い且つ適用性の広い無誤差演算を実現できる。

50

(2) 除算を分数形式データで表現するので、割り切れる場合は除算を無誤差で実行することができ、一方、割り切れない場合は、除算を分数形式データとして保留したまま無誤差演算を進めることができる。

(3) 除算を分数形式データで表現するので、割り切れない場合でも、約分することができ、数値桁数を削減できる場合が多いため、データ量の削減及び計算効率の向上に大きく寄与することができる。

【0052】

なお、除算において割り切れる場合、上記の可変長分数データ形式で表現することなく、除算結果を可変長整数データ形式で表現するようにしてもよく、可変長整数データ記憶部、可変長整数データ演算処理部及び可変長整数演算部を付加して、可変長整数データ演算処理部及び可変長整数演算部を用いて可変長整数演算を実行するようにしてもよい。この場合、分母データを可変長整数データ記憶部に記憶する必要がなくなり、データ量をより削減することができる。

10

【0053】

次に、本実施の形態の制御対象となるロボットアーム21について詳細に説明する。図4は、図1に示すロボットアーム部2における右側のロボットアーム21の構成を示す模式図であり、図5は、図4に示す右側のロボットアーム21の座標系を説明するための図である。なお、左側のロボットアーム21も、図4及び図5に示す右側のロボットアーム21と同様に構成されているので、詳細な説明は省略する。

【0054】

20

図4に示すように、グローバル座標系の原点Oを上腕21aの右肩(第1関節)に、グローバル座標系のX座標軸を胴体部23(図1参照)の正面方向に沿って床面と平行方向に、グローバル座標系のY座標軸を右肩から左肩方向に、グローバル座標系のZ座標軸を床面から天井への垂直方向にそれぞれ設定する。また、ロボットアーム21の上腕21aの長さを $e_1$ 、前腕21bの長さを $e_2$ とし、上腕21aの肩関節のうち第1関節(Y軸回り)の関節角度 $\theta_1$ 、第2関節(X軸回り)の関節角度 $\theta_2$ 、第3関節(Z軸回り)の関節角度 $\theta_3$ とし、上腕21aと前腕21bとの間の第4関節(Y軸回り)の関節角度 $\theta_4$ 、第5関節(Z軸回り)の関節角度 $\theta_5$ とし、前腕21bとハンド22との間の第6関節(Y軸回り)の関節角度 $\theta_6$ とする。

【0055】

30

このとき、3次元位置方向センサ4が装着されている前腕の手首のローカル座標系からグローバル座標への変換行列を $H_s$ とすると、以下の式(1)のように表現することができる。

【0056】

【数1】

$$H_s = \begin{pmatrix} u_{xs} & u_{ys} & u_{zs} & x_s \\ v_{xs} & v_{ys} & v_{zs} & y_s \\ w_{xs} & w_{ys} & w_{zs} & z_s \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

40

【0057】

ここで、図5に示すように、手首に装着された3次元位置方向センサ4の3次元位置は $P_s(x_s, y_s, z_s)$ 、X軸方向の単位ベクトルは $d_{xs}(u_{xs}, v_{xs}, w_{xs})$ 、Y軸方向の単位ベクトルは $d_{ys}(u_{ys}, v_{ys}, w_{ys})$ 、Z軸方向の単位ベクトルは $d_{zs}(u_{zs}, v_{zs}, w_{zs})$ となる。これらの値はすべて、3次元位置方向センサ4から取得されるので、既知となる。一方、 $H_s$ の各要素は、各アーム(上腕21a及び前腕21b)の軸の長さ $e_1, e_2$ と各関節角度 $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ を用いて、以下のように表現することができる。

50

【 0 0 5 8 】

【 数 2 】

$$\left. \begin{aligned} u_{xs} &= C_1 C_3 C_4 C_5 + S_1 S_2 S_3 C_4 C_5 - S_1 C_2 S_4 C_5 - C_1 S_3 S_5 + S_1 S_2 C_3 S_5 \\ v_{xs} &= C_2 S_3 C_4 C_5 + S_2 S_4 C_5 + C_2 C_3 S_5 \\ w_{xs} &= -S_1 C_3 C_4 C_5 + C_1 S_2 S_3 C_4 C_5 - C_1 C_2 S_4 C_5 + S_1 S_3 S_5 + C_1 S_2 C_3 S_5 \end{aligned} \right\} (2)$$

$$\left. \begin{aligned} u_{ys} &= -C_1 C_3 C_4 S_5 - S_1 S_2 S_3 C_4 S_5 + S_1 C_2 S_4 S_5 - C_1 S_3 C_5 + S_1 S_2 C_3 C_5 \\ v_{ys} &= -C_2 S_3 C_4 S_5 - S_2 S_4 S_5 + C_2 C_3 C_5 \\ w_{ys} &= S_1 C_3 C_4 S_5 - C_1 S_2 S_3 C_4 S_5 + C_1 C_2 S_4 S_5 + S_1 S_3 C_5 + C_1 S_2 C_3 C_5 \end{aligned} \right\} (3) \quad 10$$

$$\left. \begin{aligned} u_{zs} &= C_1 C_3 S_4 + S_1 S_2 S_3 S_4 + S_1 C_2 C_4 \\ v_{zs} &= C_2 S_3 S_4 - S_2 C_4 \\ w_{zs} &= -S_1 C_3 S_4 + C_1 S_2 S_3 S_4 + C_1 C_2 C_4 \end{aligned} \right\} (4) \quad 20$$

$$\left. \begin{aligned} x_s &= -C_1 C_3 S_4 e_2 - S_1 S_2 S_3 S_4 e_2 - S_1 C_2 C_4 e_2 - S_1 C_2 e_1 \\ y_s &= -C_2 S_3 S_4 e_2 + S_2 C_4 e_2 + S_2 e_1 \\ z_s &= S_1 C_3 S_4 e_2 - C_1 S_2 S_3 S_4 e_2 - C_1 C_2 C_4 e_2 - C_1 C_2 e_1 \end{aligned} \right\} (5)$$

【 0 0 5 9 】

ここで、 $C_n = \cos(\theta_n)$ 、 $S_n = \sin(\theta_n)$  ( $n = 1, \dots, 5$ ) であり、関節角度  $\theta_n$  は、アーム動作により刻々と変化するので、時間  $t$  の関数となり、 $\theta_n'$  を微小時間  $\Delta t$  前の関節角度値 (直前の関節角度) とし、微小時間  $\Delta t$  経過後における微小関節角度変化量を  $\Delta \theta_n$  とすると、以下のように表現することができる。 30

$$\theta_n = \theta_n' + \Delta \theta_n \quad (6)$$

ここで、微小時間  $\Delta t$  前の関節角度値  $\theta_n'$  は既知であり、微小時間  $\Delta t$  は制御時間間隔である。

【 0 0 6 0 】

上記の前提を基に、以下のように近似することができる。なお、下記式では、 $(\Delta \theta_n)^2$  は微小であるので、 $(\Delta \theta_n)^2$  以上の高次項は無視できるとした。

$$\sin(\theta_n) = \sin(\theta_n' + \Delta \theta_n) = \sin(\theta_n') + (\sin(\theta_n'))' \Delta \theta_n = \sin(\theta_n') + \cos(\theta_n') \Delta \theta_n \quad (7a)$$

$$\cos(\theta_n) = \cos(\theta_n' + \Delta \theta_n) = \cos(\theta_n') + (\cos(\theta_n'))' \Delta \theta_n = \cos(\theta_n') - \sin(\theta_n') \Delta \theta_n \quad (7b) \quad 40$$

上記の式 (7a)、(7b) を式 (2) ~ (5) へ代入すると、以下となる。なお、ここでも、 $\Delta \theta_n$  等の高次の  $\Delta \theta_n$  は微小であり、無視できるとした。

【 0 0 6 1 】

【数 3】

$$\begin{aligned}
u_{xs} &= C_1 C_3 C_4 C_5 + S_1 S_2 S_3 C_4 C_5 - S_1 C_2 S_4 C_5 - C_1 S_3 S_5 + S_1 S_2 C_3 S_5 \\
&= (C_1' C_3' C_4' C_5' + S_1' S_2' S_3' C_4' C_5' - S_1' C_2' S_4' C_5' - C_1' S_3' S_5' + S_1' S_2' C_3' S_5') \\
&\quad + (-S_1' C_3' C_4' C_5' + C_1' S_2' S_3' C_4' C_5' - C_1' C_2' S_4' C_5' + S_1' S_3' S_5' + C_1' S_2' C_3' S_5') \Delta\theta_1 \\
&\quad + (S_1' C_2' S_3' C_4' C_5' + S_1' S_2' S_4' C_5' + S_1' C_2' C_3' S_5') \Delta\theta_2 \\
&\quad + (-C_1' S_3' C_4' C_5' + S_1' S_2' C_3' C_4' C_5' - C_1' C_3' S_5' - S_1' S_2' S_3' S_5') \Delta\theta_3 \\
&\quad + (-C_1' C_3' S_4' C_5' - S_1' S_2' S_3' S_4' C_5' - S_1' C_2' C_4' C_5') \Delta\theta_4 \\
&\quad + (-C_1' C_3' C_4' S_5' - S_1' S_2' S_3' C_4' S_5' + S_1' C_2' S_4' S_5' - C_1' S_3' C_5' + S_1' S_2' C_3' C_5') \Delta\theta_5 \\
v_{xs} &= C_2 S_3 C_4 C_5 + S_2 S_4 C_5 + C_2 C_3 S_5 \\
&= C_2' S_3' C_4' C_5' + S_2' S_4' C_5' + C_2' C_3' S_5' \\
&\quad + (-S_2' S_3' C_4' C_5' + C_2' S_4' C_5' - S_2' C_3' S_5') \Delta\theta_2 \\
&\quad + (C_2' C_3' C_4' C_5' - C_2' S_3' S_5') \Delta\theta_3 \\
&\quad + (-C_2' S_3' S_4' C_5' + S_2' C_4' C_5') \Delta\theta_4 \\
&\quad + (-C_2' S_3' C_4' S_5' - S_2' S_4' S_5' + C_2' C_3' C_5') \Delta\theta_5 \\
w_{xs} &= -S_1 C_3 C_4 C_5 + C_1 S_2 S_3 C_4 C_5 - C_1 C_2 S_4 C_5 + S_1 S_3 S_5 + C_1 S_2 C_3 S_5 \\
&= -S_1' C_3' C_4' C_5' + C_1' S_2' S_3' C_4' C_5' - C_1' C_2' S_4' C_5' + S_1' S_3' S_5' + C_1' S_2' C_3' S_5' \\
&\quad + (-C_1' C_3' C_4' C_5' - S_1' S_2' S_3' C_4' C_5' + S_1' C_2' S_4' C_5' + C_1' S_3' S_5' - S_1' S_2' C_3' S_5') \Delta\theta_1 \\
&\quad + (C_1' C_2' S_3' C_4' C_5' + C_1' S_2' S_4' C_5' + C_1' C_2' C_3' S_5') \Delta\theta_2 \\
&\quad + (S_1' S_3' C_4' C_5' + C_1' S_2' C_3' C_4' C_5' + S_1' C_3' S_5' - C_1' S_2' S_3' S_5') \Delta\theta_3 \\
&\quad + (S_1' C_3' S_4' C_5' - C_1' S_2' S_3' S_4' C_5' - C_1' C_2' C_4' C_5') \Delta\theta_4 \\
&\quad + (S_1' C_3' C_4' S_5' - C_1' S_2' S_3' C_4' S_5' + C_1' C_2' S_4' S_5' + S_1' S_3' C_5' + C_1' S_2' C_3' C_5') \Delta\theta_5
\end{aligned}
\tag{8}$$

【 0 0 6 2 】

【数4】

$$\begin{aligned}
u_{ys} &= -C_1 C_3 C_4 S_5 - S_1 S_2 S_3 C_4 S_5 + S_1 C_2 S_4 S_5 - C_1 S_3 C_5 + S_1 S_2 C_3 C_5 \\
&= -C'_1 C'_3 C'_4 S'_5 - S'_1 S'_2 S'_3 C'_4 S'_5 + S'_1 C'_2 S'_4 S'_5 - C'_1 S'_3 C'_5 + S'_1 S'_2 C'_3 C'_5 \\
&\quad + (S'_1 C'_3 C'_4 S'_5 - C'_1 S'_2 S'_3 C'_4 S'_5 + C'_1 C'_2 S'_4 S'_5 + S'_1 S'_3 C'_5 + C'_1 S'_2 C'_3 C'_5) \Delta\theta_1 \\
&\quad + (-S'_1 C'_2 S'_3 C'_4 S'_5 - S'_1 S'_2 S'_4 S'_5 + S'_1 C'_2 C'_3 C'_5) \Delta\theta_2 \\
&\quad + (C'_1 S'_3 C'_4 S'_5 - S'_1 S'_2 C'_3 C'_4 S'_5 - C'_1 C'_3 C'_5 - S'_1 S'_2 S'_3 C'_5) \Delta\theta_3 \\
&\quad + (C'_1 C'_3 S'_4 S'_5 + S'_1 S'_2 S'_3 S'_4 S'_5 + S'_1 C'_2 C'_4 S'_5) \Delta\theta_4 \\
&\quad + (-C'_1 C'_3 C'_4 C'_5 - S'_1 S'_2 S'_3 C'_4 C'_5 + S'_1 C'_2 S'_4 C'_5 + C'_1 S'_3 S'_5 - S'_1 S'_2 C'_3 S'_5) \Delta\theta_5 \\
v_{ys} &= -C_2 S_3 C_4 S_5 - S_2 S_4 S_5 + C_2 C_3 C_5 \\
&= -C'_2 S'_3 C'_4 S'_5 - S'_2 S'_4 S'_5 + C'_2 C'_3 C'_5 \\
&\quad + (S'_2 S'_3 C'_4 S'_5 - C'_2 S'_4 S'_5 - S'_2 C'_3 C'_5) \Delta\theta_2 \\
&\quad + (-C'_2 C'_3 C'_4 S'_5 - C'_2 S'_3 C'_5) \Delta\theta_3 \\
&\quad + (C'_2 S'_3 S'_4 S'_5 - S'_2 C'_4 S'_5) \Delta\theta_4 \\
&\quad + (-C'_2 S'_3 C'_4 C'_5 - S'_2 S'_4 C'_5 - C'_2 C'_3 S'_5) \Delta\theta_5 \\
w_{ys} &= S_1 C_3 C_4 S_5 - C_1 S_2 S_3 C_4 S_5 + C_1 C_2 S_4 S_5 + S_1 S_3 C_5 + C_1 S_2 C_3 C_5 \\
&= S'_1 C'_3 C'_4 S'_5 - C'_1 S'_2 S'_3 C'_4 S'_5 + C'_1 C'_2 S'_4 S'_5 + S'_1 S'_3 C'_5 + C'_1 S'_2 C'_3 C'_5 \\
&\quad + (C'_1 C'_3 C'_4 S'_5 + S'_1 S'_2 S'_3 C'_4 S'_5 - S'_1 C'_2 S'_4 S'_5 + C'_1 S'_3 C'_5 - S'_1 S'_2 C'_3 C'_5) \Delta\theta_1 \\
&\quad + (-C'_1 C'_2 S'_3 C'_4 S'_5 - C'_1 S'_2 S'_4 S'_5 + C'_1 C'_2 C'_3 C'_5) \Delta\theta_2 \\
&\quad + (-S'_1 S'_3 C'_4 S'_5 - C'_1 S'_2 C'_3 C'_4 S'_5 + S'_1 C'_3 C'_5 - C'_1 S'_2 S'_3 C'_5) \Delta\theta_3 \\
&\quad + (-S'_1 C'_3 S'_4 S'_5 + C'_1 S'_2 S'_3 S'_4 S'_5 + C'_1 C'_2 C'_4 S'_5) \Delta\theta_4 \\
&\quad + (S'_1 C'_3 C'_4 C'_5 - C'_1 S'_2 S'_3 C'_4 C'_5 + C'_1 C'_2 S'_4 C'_5 - S'_1 S'_3 S'_5 - C'_1 S'_2 C'_3 S'_5) \Delta\theta_5
\end{aligned}
\tag{9}$$

【0063】

【数 5】

$$\begin{aligned}
 u_{zs} &= C_1 C_3 S_4 + S_1 S_2 S_3 S_4 + S_1 C_2 C_4 \\
 &= C_1' C_3' S_4' + S_1' S_2' S_3' S_4' + S_1' C_2' C_4' \\
 &\quad + (-S_1' C_3' S_4' + C_1' S_2' S_3' S_4' + C_1' C_2' C_4') \Delta\theta_1 \\
 &\quad + (S_1' C_2' S_3' S_4' - S_1' S_2' C_4') \Delta\theta_2 \\
 &\quad + (-C_1' S_3' S_4' + S_1' S_2' C_3' S_4') \Delta\theta_3 \\
 &\quad + (C_1' C_3' C_4' + S_1' S_2' S_3' C_4' - S_1' C_2' S_4') \Delta\theta_4 \\
 v_{zs} &= C_2 S_3 S_4 - S_2 C_4 \\
 &= C_2' S_3' S_4' - S_2' C_4' \\
 &\quad + (-S_2' S_3' S_4' - C_2' C_4') \Delta\theta_2 \\
 &\quad + C_2' C_3' S_4' \Delta\theta_3 \\
 &\quad + (C_2' S_3' C_4' + S_2' S_4') \Delta\theta_4 \\
 w_{zs} &= -S_1 C_3 S_4 + C_1 S_2 S_3 S_4 + C_1 C_2 C_4 \\
 &= -S_1' C_3' S_4' + C_1' S_2' S_3' S_4' + C_1' C_2' C_4' \\
 &\quad + (-C_1' C_3' S_4' - S_1' S_2' S_3' S_4' - S_1' C_2' C_4') \Delta\theta_1 \\
 &\quad + (C_1' C_2' S_3' S_4' - C_1' S_2' C_4') \Delta\theta_2 \\
 &\quad + (S_1' S_3' S_4' + C_1' S_2' C_3' S_4') \Delta\theta_3 \\
 &\quad + (-S_1' C_3' C_4' + C_1' S_2' S_3' C_4' - C_1' C_2' S_4') \Delta\theta_4
 \end{aligned}
 \tag{10}$$

【0064】

【数 6】

$$\begin{aligned}
 x_s &= -C_1 C_3 S_4 e_2 - S_1 S_2 S_3 S_4 e_2 - S_1 C_2 C_4 e_2 - S_1 C_2 e_1 \\
 &= -C'_1 C'_3 S'_4 e_2 - S'_1 S'_2 S'_3 S'_4 e_2 - S'_1 C'_2 C'_4 e_2 - S'_1 C'_2 e_1 \\
 &\quad + (S'_1 C'_3 S'_4 e_2 - C'_1 S'_2 S'_3 S'_4 e_2 - C'_1 C'_2 C'_4 e_2 - C'_1 C'_2 e_1) \Delta \theta_1 \\
 &\quad + (-S'_1 C'_2 S'_3 S'_4 e_2 + S'_1 S'_2 C'_4 e_2 + S'_1 S'_2 e_1) \Delta \theta_2 \\
 &\quad + (C'_1 S'_3 S'_4 e_2 - S'_1 S'_2 C'_3 S'_4 e_2) \Delta \theta_3 \\
 &\quad + (-C'_1 C'_3 C'_4 e_2 - S'_1 S'_2 S'_3 C'_4 e_2 + S'_1 C'_2 S'_4 e_2) \Delta \theta_4 \\
 y_s &= -C_2 S_3 S_4 e_2 + S_2 C_4 e_2 + S_2 e_1 \\
 &= -C'_2 S'_3 S'_4 e_2 + S'_2 C'_4 e_2 + S'_2 e_1 \\
 &\quad + (S'_2 S'_3 S'_4 e_2 + C'_2 C'_4 e_2 + C'_2 e_1) \Delta \theta_2 \\
 &\quad - C'_2 C'_3 S'_4 e_2 \Delta \theta_3 \\
 &\quad + (-C'_2 S'_3 C'_4 e_2 - S'_2 S'_4 e_2) \Delta \theta_4 \\
 z_s &= S_1 C_3 S_4 e_2 - C_1 S_2 S_3 S_4 e_2 - C_1 C_2 C_4 e_2 - C_1 C_2 e_1 \\
 &= S'_1 C'_3 S'_4 e_2 - C'_1 S'_2 S'_3 S'_4 e_2 - C'_1 C'_2 C'_4 e_2 - C'_1 C'_2 e_1 \\
 &\quad + (C'_1 C'_3 S'_4 e_2 + S'_1 S'_2 S'_3 S'_4 e_2 + S'_1 C'_2 C'_4 e_2 + S'_1 C'_2 e_1) \Delta \theta_1 \\
 &\quad + (-C'_1 C'_2 S'_3 S'_4 e_2 + C'_1 S'_2 C'_4 e_2 + C'_1 S'_2 e_1) \Delta \theta_2 \\
 &\quad + (-S'_1 S'_3 S'_4 e_2 - C'_1 S'_2 C'_3 S'_4 e_2) \Delta \theta_3 \\
 &\quad + (S'_1 C'_3 C'_4 e_2 - C'_1 S'_2 S'_3 C'_4 e_2 + C'_1 C'_2 S'_4 e_2) \Delta \theta_4
 \end{aligned}
 \tag{11}$$

【0065】

上記各式より、式(1)は、以下のように表現することができる。

【0066】

10

20

30

【数 7】

$$H_s = H'_s + \Delta H_s \quad (12)$$

$$\begin{pmatrix} u_{xs} \\ v_{xs} \\ w_{xs} \\ u_{ys} \\ v_{ys} \\ w_{ys} \\ u_{zs} \\ v_{zs} \\ w_{zs} \\ x_s \\ y_s \\ z_s \end{pmatrix} = \begin{pmatrix} u'_{xs} \\ v'_{xs} \\ w'_{xs} \\ u'_{ys} \\ v'_{ys} \\ w'_{ys} \\ u'_{zs} \\ v'_{zs} \\ w'_{zs} \\ x'_s \\ y'_s \\ z'_s \end{pmatrix} + \begin{pmatrix} h_{011} & h_{012} & h_{013} & h_{014} & h_{015} \\ h_{021} & h_{022} & h_{023} & h_{024} & h_{025} \\ h_{031} & h_{032} & h_{033} & h_{034} & h_{035} \\ h_{041} & h_{042} & h_{043} & h_{044} & h_{045} \\ h_{051} & h_{052} & h_{053} & h_{054} & h_{055} \\ h_{061} & h_{062} & h_{063} & h_{064} & h_{065} \\ h_{071} & h_{072} & h_{073} & h_{074} & h_{075} \\ h_{081} & h_{082} & h_{083} & h_{084} & h_{085} \\ h_{091} & h_{092} & h_{093} & h_{094} & h_{095} \\ h_{101} & h_{102} & h_{103} & h_{104} & h_{105} \\ h_{111} & h_{112} & h_{113} & h_{114} & h_{115} \\ h_{121} & h_{122} & h_{123} & h_{124} & h_{125} \end{pmatrix} \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \\ \Delta\theta_5 \end{pmatrix} \quad (13)$$

【0067】

そこで、 $h_{101}, h_{111}, h_{121}, h_{021}, h_{081}$  は、未知数が 5 個あるので、式 (13) の中から適切な 5 つの式を選択し、例えば、以下の 5 元連立一次方程式の解として求めることができる。

【0068】

【数 8】

$$\begin{pmatrix} h_{101} & h_{102} & h_{103} & h_{104} & h_{105} \\ h_{111} & h_{112} & h_{113} & h_{114} & h_{115} \\ h_{121} & h_{122} & h_{123} & h_{124} & h_{125} \\ h_{021} & h_{022} & h_{023} & h_{024} & h_{025} \\ h_{081} & h_{082} & h_{083} & h_{084} & h_{085} \end{pmatrix} \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \\ \Delta\theta_5 \end{pmatrix} = \begin{pmatrix} x_s - x'_s \\ y_s - y'_s \\ z_s - z'_s \\ v_{xs} - v'_{xs} \\ v_{zs} - v'_{zs} \end{pmatrix} \quad (14)$$

【0069】

また、上式の左辺の行列の各要素を係数  $a_{ij}$  ( $i = 1, \dots, 5, j = 1, \dots, 5$ )、右辺のベクトルの各要素を係数  $b_i$  ( $i = 1, \dots, 5$ ) で表すと、以下のように表現することができる。

【0070】

10

20

30

40



【数 9】

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \\ \Delta\theta_5 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} \quad (15)$$

10

【0071】

上記の5元連立一次方程式の解法では、加減乗除算が繰り返されるために、浮動小数点演算（固定長数値演算）を用いた数値計算では、数値演算誤差とその蓄積のために解の誤差が大きくなる場合が発生する。また、ゼロで除算が行われる場合（以下、この場合を「ゼロ割」という）も発生する。結果として、これらの演算誤差、その蓄積及びゼロ割のために、ロボットアーム動作が不安定になり、暴走及び破綻が発生する。

【0072】

このような数値演算の誤差が大きくなること、ロボットアームの動作が暴走及び破綻することは、ロボットアーム制御処理においては許されない。ロボットアームの動作精度が低下したり、又は動作が暴走及び破綻すると、人に危害を加えたり、周辺の物を壊してしまふこととなる。また、除算があると、除算は数値演算に計算精度の低下等様々な悪影響を及ぼすことはもちろんのこと、除算では割り切れない場合が発生し（無限桁数が必要な循環小数の発生）、無誤差演算を行うことができない。

20

【0073】

このため、本願発明者は、特願2005-164567号明細書において、上記連立一次方程式の解法処理に関して、可変長数値（整数）データを用い且つ除算を一切使用せず、誤差を発生させない無誤差演算を行う無誤差連立一次方程式解法処理方法を考案した。これにより、安定且つ高信頼性なロボット制御方法を実現した。

【0074】

しかしながら、除算を用いない無誤差演算方法は、適用できる範囲が非常に限定され、且つ計算を進めていくと、際限なく数値桁数が増大し、計算効率が著しく低下していく。結果として、実用的ではない場合も多々ある。このため、本願発明者は、上記の特願2005-164567号明細書に記載の方法をベースにして、この方法の欠点をなくす方法を考案した。

30

【0075】

すなわち、本実施の形態では、式(15)の連立一次方程式の解法演算において、上記した可変長整数を用いた分数形式のデータフォーマット（可変長分数データ形式）で除算を表現して無誤差演算を行う。例えば、ガウス・ジョルダンの消去法を用いて、連立一次方程式の解を求める処理を可変長分数データ処理演算により行い、各関節角度  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$  を以下のようにして求めている。

40

【0076】

まず、式(15)の係数  $a_{ij}$  ( $i = 1, \dots, 5, j = 1, \dots, 5$ ) 及び係数  $b_i$  ( $i = 1, \dots, 5$ ) を用いて、 $k = 1, \dots, 5$  として、下記の式を繰り返し演算することにより係数  $a^k_{ij}$  及び係数  $b^k_i$  を求める。

$$a^{k+1}_{ij} = a^k_{ij} \quad (i = k) \quad (16a)$$

$$a^{k+1}_{ij} = a^k_{ij} - a^k_{kj} a^k_{ik} / a^k_{kk} \quad (i \neq k) \quad (16b)$$

$$b^{k+1}_i = b^k_i \quad (i = k) \quad (16c)$$

$$b^{k+1}_i = b^k_i - b^k_k a^k_{ik} / a^k_{kk} \quad (i \neq k) \quad (16d)$$

上記の係数  $a^k_{ij}$  及び係数  $b^k_i$  により、式(15)は以下の形式となる。

【0077】

50

【数 1 0】

$$\begin{pmatrix} a_{11}^6 & 0 & 0 & 0 & 0 \\ 0 & a_{22}^6 & 0 & 0 & 0 \\ 0 & 0 & a_{33}^6 & 0 & 0 \\ 0 & 0 & 0 & a_{44}^6 & 0 \\ 0 & 0 & 0 & 0 & a_{55}^6 \end{pmatrix} \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \\ \Delta\theta_5 \end{pmatrix} = \begin{pmatrix} b_1^6 \\ b_2^6 \\ b_3^6 \\ b_4^6 \\ b_5^6 \end{pmatrix} \quad (17)$$

10

【0 0 7 8】

上記のように、式(17)の左辺の係数行列は、対角要素以外が0となる対角行列となる。式(6)及び式(17)から、各関節角度  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$  は、以下の式により求めることができる。なお、手首の角度  $\theta_6$  は、3次元操作グローブ3により直接取得することができるので、上記の無誤差連立一次方程式解法処理からは求めていない。

【0 0 7 9】

【数 1 1】

$$\begin{cases} \theta_1 = \theta_1' + \Delta\theta_1 = \theta_1' + b_1^6 / a_{11}^6 \\ \theta_2 = \theta_2' + \Delta\theta_2 = \theta_2' + b_2^6 / a_{22}^6 \\ \theta_3 = \theta_3' + \Delta\theta_3 = \theta_3' + b_3^6 / a_{33}^6 \\ \theta_4 = \theta_4' + \Delta\theta_4 = \theta_4' + b_4^6 / a_{44}^6 \\ \theta_5 = \theta_5' + \Delta\theta_5 = \theta_5' + b_5^6 / a_{55}^6 \end{cases} \quad (18)$$

20

【0 0 8 0】

上記の解法を基に、ロボットアーム制御装置1は、上記の無誤差連立一次方程式解法処理(目標データPから関節角度  $\theta_i$  を求めるデータ演算処理)を以下のようにして実行している。

30

【0 0 8 1】

外部インターフェース部11は、ロボットアームの21の先端部の目標3次元位置(3次元位置方向センサ4の3次元位置  $P_s$ )及び目標方向(X、Y、Z軸方向の単位ベクトル  $d_{xs}, d_{ys}, d_{zs}$ )からなる目標データPを3次元位置方向センサ4から取得し、固定長数値データ記憶部12に浮動小数点データ形式で記憶させる。このとき、固定長数値データ記憶部12には、微小時間  $t$  前の位置方向データ  $P'$  及び微小時間  $t$  前の関節角度データ  $\theta'_i$  が記憶されており、また、上腕21aの長さ  $e_1$ 、前腕21bの長さ  $e_2$  等の初期値は予め記憶されているものとする。

【0 0 8 2】

ここで、上記の微小時間  $t$  前の位置方向データ  $P'$  は、微小時間  $t$  前の目標データではなく、微小時間  $t$  前の実際の3次元位置データ及び方向データであり、微小時間  $t$  前の関節角度データ  $\theta'_i$  もアームの実際の関節角度である。これは、後述するように、 $P'$ 、 $\theta'_i$  及びPから  $\theta_i$  を求め(線形近似後、連立一次方程式を無誤差で解く)、そして、 $\theta_i = \theta'_i + \Delta\theta_i$  から  $\Delta\theta_i$  を求めており、ここで(線形)近似しているため、このから求めた実際の位置方向データは目標データと一致しない。そこで、次の時刻の計算では、微小時間  $t$  前の目標データ等ではなく、微小時間  $t$  前の実際の3次元位置データ及び方向データ、並びに微小時間  $t$  前の実際の関節角度データを用いている。

40

【0 0 8 3】

固定長数値データ演算処理部15aは、固定長数値データ記憶部12から微小時間  $t$

50

前の関節角度  $\theta_i$ 、上腕 2 1 a の長さ  $e_1$ 、前腕 2 1 b の長さ  $e_2$  を読み出し、浮動小数点演算部 1 5 b を用いて浮動小数点演算を実行することにより式 ( 1 5 ) の係数データ  $a_{ij}$  を算出し、固定長数値データ記憶部 1 2 に記憶させる。また、固定長数値データ演算処理部 1 5 a は、固定長数値データ記憶部 1 2 から目標データ P 及び微小時間  $t$  前の位置方向データ P' を読み出し、浮動小数点演算部 1 5 b を用いて浮動小数点演算を実行することにより式 ( 1 5 ) の係数データ  $b_i$  を算出し、固定長数値データ記憶部 1 2 に記憶させる。

【 0 0 8 4 】

データ変換処理部 1 3 は、固定長数値データ記憶部 1 2 から係数  $a_{ij}$  及び係数  $b_i$  を順次読み出し、浮動小数点データ形式から可変長分数データ形式に変換した係数  $a_{ij}$  及び係数  $b_i$  を、係数  $a^1_{ij}$  及び係数  $b^1_i$  として可変長分数データ記憶部 1 4 に格納する。

10

【 0 0 8 5 】

可変長分数データ演算処理部 1 6 a は、可変長分数データ記憶部 1 4 から係数  $a^1_{ij}$  及び係数  $b^1_i$  を読み出し、可変長分数演算部 1 6 b を用いて可変長分数演算 ( 無誤差演算 ) を実行することにより、式 ( 1 6 a ) 乃至 ( 1 6 d ) を用いて係数  $a^{k+1}_{ij}$  及び  $b^{k+1}_i$  を順次算出し、最終的に係数  $a^6_{ij}$  及び  $b^6_i$  を可変長分数データ記憶部 1 4 に格納する。

【 0 0 8 6 】

また、可変長分数データ演算処理部 1 6 a は、可変長分数データ記憶部 1 4 から係数  $a^6_{ii}$  及び係数  $b^6_i$  を読み出し、可変長分数演算部 1 6 b を用いて可変長分数演算 ( 無誤差演算 ) を実行することにより、式 ( 1 7 ) を基に微小関節角度変化量  $\Delta \theta_i = b^6_i / a^6_{ii}$  を算出し、さらに、式 ( 1 8 ) に従って関節角度  $\theta_i = \theta'_i + \Delta \theta_i$  を算出し、可変長分数データ記憶部 1 4 に格納する。

20

【 0 0 8 7 】

これらの可変長分数演算において、可変長分数データを用いて、加算、減算及び乗算はすべて実行され、除算は割り切れる場合に実行され、割り切れない場合には保留され、さらに、約分ができる場合は約分が実行される。

【 0 0 8 8 】

データ変換処理部 1 3 は、可変長分数データ記憶部 1 4 から関節角度  $\theta_i$  を順次読み出して可変長分数データ形式から浮動小数点データ形式に変換し、浮動小数点データ形式の関節角度  $\theta_i$  を固定長数値データ記憶部 1 2 に格納する。

30

【 0 0 8 9 】

外部インターフェース部 1 1 は、固定長数値データ記憶部 1 2 から浮動小数点データ形式の関節角度  $\theta_i$  を読み出してロボットアーム部 2 へ出力する。ロボットアーム部 2 は、ロボットアーム 2 1 の各関節角度が関節角度  $\theta_i$  となるようにロボットアーム 2 1 を動作させる。

【 0 0 9 0 】

上記の制御動作により、操作者 M は、3次元ディスプレイ 5 を用いて、カメラ部 2 4 により撮影された画像を見ながら、両手に装着した3次元操作グローブ 3 を用いて、ロボットアーム 2 1 及びハンド 2 2 を操作し、操作者 M の腕と手の動きとがロボットアーム 2 1 及びハンド 2 2 により忠実に再現される。

40

【 0 0 9 1 】

本実施の形態では、データ変換処理部 1 3 が変換手段の一例に相当し、無誤差制御処理部 1 6 ( 可変長分数データ演算処理部 1 6 a 及び可変長分数演算部 1 6 b ) が無誤差演算手段の一例に相当し、3次元位置方向センサ 4 が検出手段の一例に相当する。

【 0 0 9 2 】

次に、上記のように構成されたロボットアーム制御装置 1 によるロボットアーム制御処理について説明する。図 6 は、図 1 に示すロボットアーム制御装置 1 によるロボットアーム制御処理を説明するためのフローチャートである。

50

## 【 0 0 9 3 】

図 6 に示すように、まず、ステップ S 1 1 において、外部インターフェース部 1 1 は、3 次元位置方向センサ 4 から目標となる 3 次元位置方向データ  $H_s$  を取得して固定長数値データ記憶部 1 2 に格納する。

## 【 0 0 9 4 】

次に、ステップ S 1 2 において、誤差有制御処理部 1 5 の固定長数値データ演算処理部 1 5 a は、固定長数値データ記憶部 1 2 から 3 次元位置方向データ  $H_s$  等を読み出し、浮動小数点演算部 1 5 b を用いて、式 ( 1 5 ) の係数  $a_{i j}$  ( $i = 1, \dots, 5, j = 1, \dots, 5$ ) 及び係数  $b_i$  ( $i = 1, \dots, 5$ ) を求め、固定長数値データ記憶部 1 2 に格納する。

10

## 【 0 0 9 5 】

次に、ステップ S 1 3 において、データ変換処理部 1 3 は、固定長数値データ記憶部 1 2 から係数  $a_{i j}$  及び係数  $b_i$  を順次読み出して浮動小数点データ形式から可変長分数データ形式へ変換し、可変長分数データ形式の係数  $a^1_{i j}$  及び係数  $b^1_i$  を可変長分数データ記憶部 1 4 に格納する。

## 【 0 0 9 6 】

なお、上記のステップ S 1 2 及び S 1 3 では、浮動小数点演算を用いて 3 次元位置方向データ  $H_s$  等から係数  $a_{i j}$  及び係数  $b_i$  を求めた後に、可変長分数データへフォーマット変換したが、この例に特に限定されず、データ変換処理部 1 3 により 3 次元位置方向データ  $H_s$  を浮動小数点データから可変長分数データに変換し、その後、無誤差制御処理部 1 6 により可変長分数データの 3 次元位置方向データ  $H_s$  等から可変長分数データの係数  $a^1_{i j}$  及び係数  $b^1_i$  を求めるようにしてもよい。

20

## 【 0 0 9 7 】

ここで、図 6 に示すステップ S 1 3 における固定長数値データ ( 浮動小数点数値データ ) から可変長分数データへの変換処理を詳細に説明する。図 7 は、図 6 に示すステップ S 1 3 における浮動小数点数値データから可変長分数データへの変換処理を説明するためのフローチャートである。

## 【 0 0 9 8 】

図 7 に示すように、まず、ステップ S 3 1 において、データ変換処理部 1 3 は、固定長数値データ記憶部 1 2 から浮動小数点数値データ  $d$  を読み出し、浮動小数点数値データ  $d$  を上位から  $(n + 1)$  桁目で四捨五入をする ( ステップ S 5 1 ) 。ここで、桁数  $n$  は要求される数値精度に応じて任意に設定することができる。例えば、 $d = 1 2 3 4 5 . 6 7 8 9$ 、 $n = 7$  とすると、四捨五入した結果は  $d = 1 2 3 4 5 . 6 8$  となる。

30

## 【 0 0 9 9 】

次に、ステップ S 3 2 において、データ変換処理部 1 3 は、四捨五入後の数値  $d$  が整数か小数点以下の数値を持つかどうかを判定し、整数である場合は、ステップ S 3 3 に処理を移行して可変長分数データの分子  $a = d$ 、分母  $b = 1$  に設定し、この可変長分数データ  $a / b$  を、可変長分数データ記憶部 1 4 に格納した後、図 6 に示すステップ S 1 4 へ処理を戻す。

## 【 0 1 0 0 】

一方、数値  $d$  が小数点以下の数値を持つ場合は、データ変換処理部 1 3 は、ステップ S 3 4 において、数値  $d$  の小数点以下の桁数  $n'$  を求め、ステップ S 3 5 において、分子  $a = d \times 10^{n'}$ 、分母  $b = 10^{n'}$  とし、この可変長分数データ  $a / b$  を可変長分数データ記憶部 1 4 に格納した後、図 6 に示すステップ S 1 4 へ処理を戻す。上記の例では、小数点以下の数値を持ち、 $n' = 2$  となるので、分子  $a = 1 2 3 4 5 . 6 8 \times 10^2 = 1 2 3 4 5 6 8$ 、分母  $b = 10^2 = 1 0 0$  となり、これらのデータが一对のデータとして可変長分数データ記憶部 1 4 に記憶される。

40

## 【 0 1 0 1 】

再び、図 6 を参照して、上記のステップ S 1 3 の処理が終了した後、ステップ S 1 4、S 1 5 及び S 1 6 において、可変長分数データ演算処理部 1 6 a は、初期設定として、 $k$

50

= 1、 $i = 1$ 、 $j = 1$ に設定する。次に、ステップS 17において、可変長分数データ演算処理部16aは、可変長分数データ記憶部14から係数 $a^k_{ij}$ 、 $a^k_{kk}$ 、 $a^k_{kj}$ 、 $a^k_{ik}$ を読み出し、可変長分数演算部16bを用いて、 $i = k$ の場合は $a^{k+1}_{ij} = a^k_{ij}$ に設定し、 $i \neq k$ の場合は $a^{k+1}_{ij} = a^k_{ij} - a^k_{kj} a^k_{ik} / a^k_{kk}$ を算出し、係数 $a^{k+1}_{ij}$ を可変長分数データ記憶部14に格納する。

【0102】

次に、ステップS 18において、可変長分数データ演算処理部16aは、 $j < 5$ であるか否かを判断し、 $j < 5$ の場合、ステップS 27において $j$ を1だけインクリメントしてステップS 17以降の処理を継続し、 $j < 5$ でない( $j = 5$ )場合、ステップS 19へ処理を移行する。

10

【0103】

$j < 5$ でない場合、ステップS 19において、可変長分数データ演算処理部16aは、可変長分数データ記憶部14から係数 $b^k_i$ 、 $b^k_k$ 、 $a^k_{ik}$ 、 $a^k_{kk}$ を読み出し、可変長分数演算部16bを用いて、 $i = k$ の場合は $b^{k+1}_i = b^k_i$ に設定し、 $i \neq k$ の場合は $b^{k+1}_i = b^k_i - b^k_k a^k_{ik} / a^k_{kk}$ を算出し、係数 $b^{k+1}_i$ を可変長分数データ記憶部14に格納する。

【0104】

次に、ステップS 20において、可変長分数データ演算処理部16aは、 $i < 5$ であるか否かを判断し、 $i < 5$ の場合、ステップS 26において $i$ を1だけインクリメントしてステップS 16以降の処理を継続し、 $i < 5$ でない( $i = 5$ )場合、ステップS 21へ処理を移行する。

20

【0105】

$i < 5$ でない場合、ステップS 21において、可変長分数データ演算処理部16aは、 $k < 5$ であるか否かを判断し、 $k < 5$ の場合、ステップS 25において $k$ を1だけインクリメントしてステップS 15以降の処理を継続し、 $k < 5$ でない( $k = 5$ )場合、ステップS 22へ処理を移行する。

【0106】

上記の処理により、可変長分数データ記憶部14から $a^k_{ij}$ 及び $b^k_i$ が読み出され、無誤差制御処理部16により、可変長分数演算部16bを用いて式(16a)乃至(16d)の演算処理が行われ、演算結果である $a^{k+1}_{ij}$ 及び $b^{k+1}_i$ が、再び可変長分数データ記憶部14に記憶される。ここでの演算の除算において、割り切れる場合は除算が実行され(分母が1となる)、割り切れない場合は、除算は保留される(分数形式のままとなる)。また、割り切れない場合で、約分ができる場合は約分が行われ、分子データ及び分母データの数値桁数の削減が行われる。これらの点に関して以下のステップS 22における除算も同様である。

30

【0107】

次に、ステップS 22において、可変長分数データ演算処理部16aは、可変長分数データ記憶部14から係数 $a^6_{ii}$ 及び係数 $b^6_i$ ( $i = 1, \dots, 5$ )を順次読み出し、可変長分数演算部16bを用いて、微小関節角度変化量 $\theta_i = b^6_i / a^6_{ii}$ ( $i = 1, \dots, 5$ )を算出し、可変長分数データ記憶部14に格納する。

40

【0108】

ここで、図6に示すステップS 17、S 19及びS 22の除算における約分処理を詳細に説明する。本実施の形態では、ロボットアーム制御処理における除算において、例えば、可変長整数データ $a$ 及び $b$ に対する除算 $a \div b$ が発生すると、可変長分数データ $a/b$ として表現され、分子データ $a$ と分母データ $b$ とが1つのペアデータとして可変長分数データ記憶部14に記憶され、割り切れる場合は除算が実行され、さらに、約分ができる場合は約分が行われる。

【0109】

図8は、図6に示すステップS 17、S 19及びS 22の除算における約分処理を説明するためのフローチャートである。図8に示すように、約分処理では、まず、ステップS

50

41において、可変長分数データ演算処理部16aは、可変長分数データa/bを可変長分数データ記憶部14から読み込む。

【0110】

次に、ステップS42において、可変長分数データ演算処理部16aは、可変長分数演算部16bを用いて、分子データa及び分母データbの下位ビットにおいて「0」が連続するビット数 $n_a$ 及び $n_b$ を求める。例えば、 $a = 12 = 1100$ （2進数）、 $b = 6 = 110$ （2進数）であるとする、 $n_a = 2$ 、 $n_b = 1$ となる。

【0111】

次に、ステップS43において、可変長分数データ演算処理部16aは、可変長分数演算部16bを用いて、ビット数 $n_a$ と $n_b$ とを比較し、大きくない方をnとする。上記の例では、 $n = n_b = 1$ となる。

10

【0112】

次に、ステップS44において、可変長分数データ演算処理部16aは、可変長分数演算部16bを用いて、分子データa及び分母データbのビットを下位方向にnビットシフトさせ、それぞれa'及びb'とする。上記の例では、 $n = 1$ であるため、分子データa及び分母データbのビットを下位方向に1ビットシフトし、それぞれ $a' = 6 = 110$ （2進数）及び $b' = 3 = 11$ （2進数）となる。

【0113】

次に、可変長分数データ演算処理部16aは、可変長分数演算部16bを用いて、ステップS45において、a'とb'とを比較し、小さくない方をA、大きくない方をBとし、ステップS46において、両者の差 $C = A - B$ を求める。上記の例では、 $A = 6$ 、 $B = 3$ となり、 $C = 6 - 3 = 3$ となる。

20

【0114】

次に、可変長分数データ演算処理部16aは、可変長分数演算部16bを用いて、ステップS47において、 $C = 0$ であるか否かを判断し、 $C = 0$ の場合は、ステップS51において、Aが最大公約数となり、分子データa'及び分母データb'をAで除算し、約分された分数データa/bを再び可変長分数データ記憶部14に記憶させた後、図6に示すステップS17、S19及びS22へ処理を戻す。

【0115】

一方、 $C = 0$ でない場合、可変長分数データ演算処理部16aは、可変長分数演算部16bを用いて、ステップS48において、 $A = B$ 、 $B = C$ に設定し、ステップS49において、 $A < B$ であるか否かを判断し、 $A < B$ でない場合は、ステップS46に戻って以降の処理を継続し、 $A < B$ である場合は、ステップS50において、AとBとを入れ替えた後、ステップ46に戻って以降の処理を継続する。この処理では、BとCとを比較し、小さくない方をA、大きくない方をBとし、 $C = 0$ となるまで、処理を繰り返す。上記の例では、 $C = 0$ となるのは、 $A = 3$ 、 $B = 3$ となるときであり、最大公約数は3となり、 $a/b$ を約分すると、 $a/b = 12/6 = a'/b' = 6/3 = (6 \div 3) / (3 \div 3) = 2/1$ となる。

30

【0116】

再び、図6を参照して、上記のステップS22の処理が終了した後、ステップS23において、可変長分数データ演算処理部16aは、可変長分数データ記憶部14から直前の関節角度値 $\theta'_i$ 及び微小関節角度変化量 $\Delta\theta_i$  ( $i = 1, \dots, 5$ )を順次読み出し、可変長分数演算部16bを用いて、式(18)に従って関節角度 $\theta_i = \theta'_i + \Delta\theta_i$  ( $i = 1, \dots, 5$ )を算出し、可変長分数データ記憶部14に格納する。このように、ステップS14~S24及びS25~S27までの処理が無誤差演算を用いて実行される。

40

【0117】

次に、ステップS24において、データ変換処理部13は、可変長分数データ記憶部14から関節角度 $\theta_i$  ( $i = 1, \dots, 5$ )を順次読み出して可変長分数データ形式から浮動小数点データ形式に変換し、浮動小数点データ形式の関節角度 $\theta_i$ を固定長数値データ記憶部12に格納する。この処理では、分数データを浮動小数点数値に変換するために除算

50

すなわち分子を分母で割る処理が行われる。この演算により誤差が発生する可能性があるが、この処理は1回の除算を行うだけであるので、除算に十分な精度が確保されていれば、誤差が問題になることはなく、誤差の管理が可能であり、誤差の蓄積も起こらない。

【0118】

また、ステップS24において、外部インターフェース部11は、固定長数値データ記憶部12から関節角度 $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ を読み出してロボットアーム部2へ出力する。ロボットアーム部2は、ロボットアーム21の各関節角度が関節角度 $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ となるようにロボットアーム21を動作させる。

【0119】

なお、手首の関節角度 $\theta_6$ は、3次元操作グループ3から直接取得することができるので、3次元操作グループ3により検出された関節角度 $\theta_6$ がロボットアーム制御装置1を介してロボットアーム部2へ出力され、ロボットアーム部2は、ロボットアーム21の手首の関節角度が関節角度 $\theta_6$ となるようにロボットアーム21を動作させる。上記の処理が制御時間間隔ごとに繰り返され、ロボットアーム21の先端位置が操作者Mの手首の位置に応じて正確に制御される。

【0120】

ここで、図6に示すステップS24における可変長分数データから浮動小数点数値データへの変換処理を詳細に説明する。図9は、図6に示すステップS24における可変長分数データから浮動小数点数値データへの変換処理を説明するためのフローチャートである。

【0121】

図9に示すように、データ変換処理部13は、まず、ステップS61において、可変長分数データ記憶部14から可変長分数データ $a/b$ を読み出して、分子データ $a$ の上位から $(n+1)$ 桁目で四捨五入をし、これを仮数部 $a'$ とし、ステップS62において、分子データ $a$ の指数部 $i$ を求める。ここで、桁数 $n$ は要求される数値精度に応じて任意に設定することができる。例えば、 $a/b = 123456/789$ 、 $n = 3$ とすると、四捨五入した結果は、 $a' = 1.23$ となり、 $i = 5$ となる。

【0122】

次に、データ変換処理部13は、ステップS63において、分母データ $b$ の上位から $(n+1)$ 桁目で四捨五入をし、これを仮数部 $b'$ とし、ステップS64において、分母データの指数部 $j$ を求める。上記の例では、 $b' = 7.89$ 、 $j = 2$ となる。

【0123】

次に、データ変換処理部13は、ステップS65において、 $a'$ を $b'$ で割った仮数部 $c$ を求め、ステップS66において、その指数部 $k = i - j$ を求める。上記の例では、 $c = a'/b' = 1.23/7.89 = 0.156$ 、 $k = i - j = 5 - 2 = 3$ となる。

【0124】

最後に、ステップS67において、データ変換処理部13は、求めた仮数部と指数部とで構成される浮動小数点数値 $d$ を正規化して、固定長数値データ記憶部12に格納して処理を終了する。上記の例では、正規化は、 $d = 0.156 \times 10^3 = 1.56 \times 10^2$ となる。なお、この例では、説明を分かりやすくするために、10進数を用いて説明してきたが、実際の実装は2進数を用いて行われる。

【0125】

上記の処理により、本実施の形態では、可変長分数データ演算を用いて無誤差演算制御を実現しているため、以下の効果を奏することができる。

(1) 除算を含む四則演算すべてを用いて、無誤差演算で制御処理を行うことができ、汎用性、実用性が高い。

(2) 可変長分数データ処理では、約分をすることができる。これにより、無誤差演算の欠点であった数値桁数の際限のない増大を抑えることができ、データ量を削減し、演算処理を効率化及び高速化することができる。

(3) 制御演算誤差及びその蓄積をなくすことにより、制御処理の破綻をなくし、高精度

10

20

30

40

50

な位置制御を実現することができる。

(4) リンク機構(ロボットアーム)動作の暴走等をなくすことができ、ロボット動作を安定化及び高信頼化することができる。

(5) 上記との関連で、ロボットアームの暴走防止対策等が不要となり、制御プログラム及び制御システムを単純化することができる。

【0126】

なお、本実施の形態では、浮動小数点データを可変長分数データへ変換して無誤差演算を実行したが、この例に特に限定されず、浮動小数点データを可変長整数データへ変換し、除算が発生した場合に可変長整数データを可変長分数データへ変換して表現する等の種々の変更が可能である。

10

【0127】

また、操作者の手首の位置及び方向を検出し、これらを目標位置及び目標方向として制御したが、本発明による制御例は、この例に特に限定されず、3次元操作グローブ3、3次元位置方向センサ4及び3次元ディスプレイ5を省略し、テーチングを行うことにより固定長数値データ記憶部12に記憶した目標位置及び目標方向となるようにロボットアーム21を制御する等の他の制御例にも適用可能である。また、ロボットアーム21を制御対象としたが、この例に特に限定されず、ハンド22を制御対象にする等の種々の変更が可能である。

【図面の簡単な説明】

【0128】

20

【図1】本発明の一実施の形態によるロボットアーム制御装置を用いたロボットアーム制御システムの構成を示すブロック図である。

【図2】図1に示すロボットアーム制御装置の構成を示すブロック図である。

【図3】数値データの種々のフォーマットの例を示す図である。

【図4】図1に示すロボットアーム部における右側のロボットアームの構成を示す模式図である。

【図5】図4に示す右側のロボットアームの座標系を説明するための図である。

【図6】図1に示すロボットアーム制御装置によるロボットアーム制御処理を説明するためのフローチャートである。

【図7】図6に示すステップS13における浮動小数点数値データから可変長分数データへの変換処理を説明するためのフローチャートである。

30

【図8】図6に示すステップS17、S19及びS22の除算における約分処理を説明するためのフローチャートである。

【図9】図6に示すステップS24における可変長分数データから浮動小数点数値データへの変換処理を説明するためのフローチャートである。

【符号の説明】

【0129】

- 1 ロボットアーム制御装置
- 2 ロボットアーム部
- 3 3次元操作グローブ
- 4 3次元位置方向センサ
- 5 3次元ディスプレイ
- 11 外部インターフェース部
- 12 固定長数値データ記憶部
- 13 データ変換処理部
- 14 可変長分数データ記憶部
- 15 誤差有制御処理部
- 15a 固定長数値データ演算処理部
- 15b 浮動小数点演算部
- 16 無誤差制御処理部

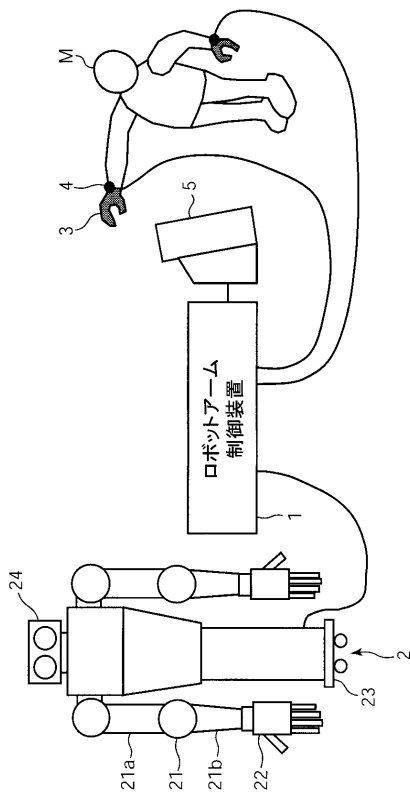
40

50

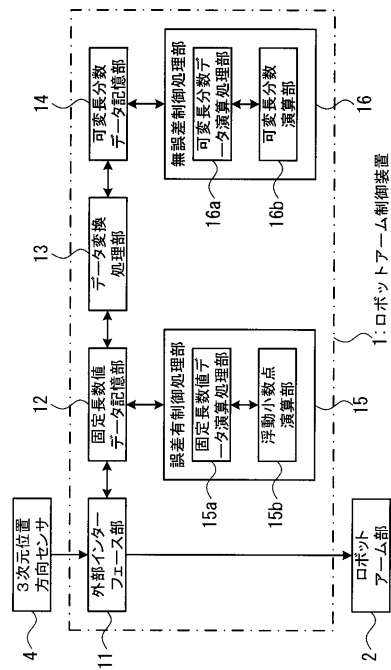


- 16 a 可変長分数データ演算処理部
- 16 b 可変長分数演算部
- 21 ロボットアーム
- 21 a 上腕
- 21 b 前腕
- 22 ハンド
- 23 胴体部
- 24 カメラ部

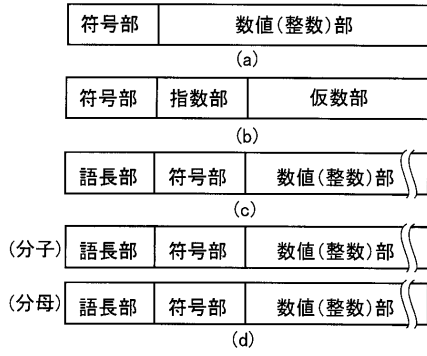
【図1】



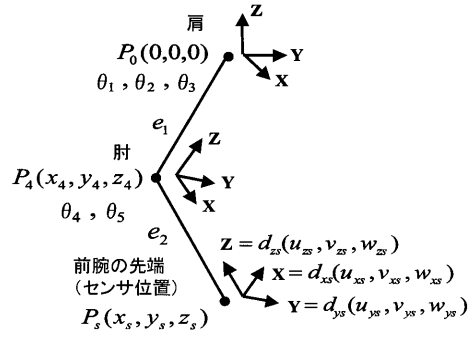
【図2】



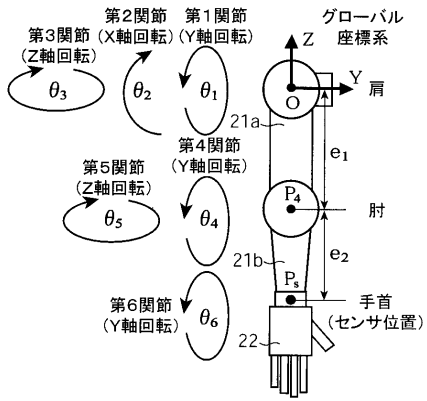
【図3】



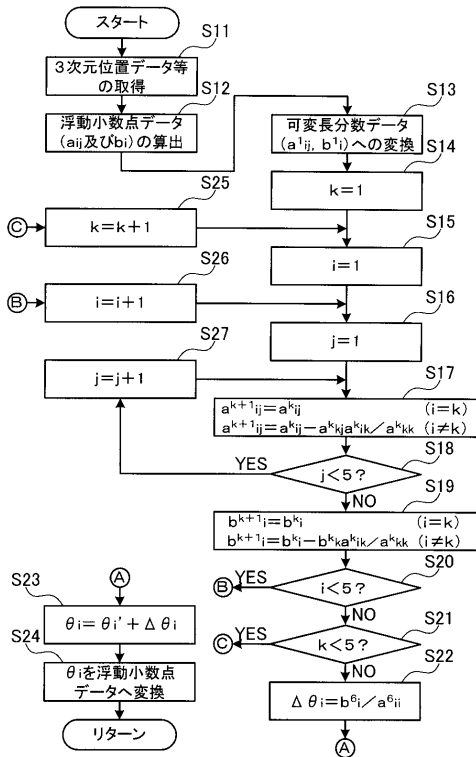
【図5】



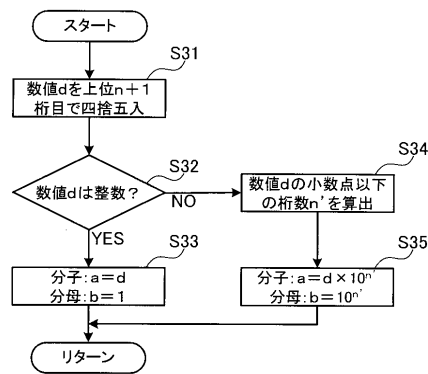
【図4】



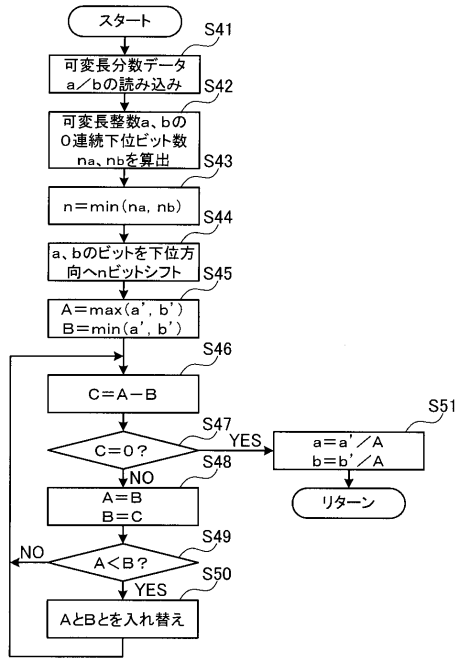
【図6】



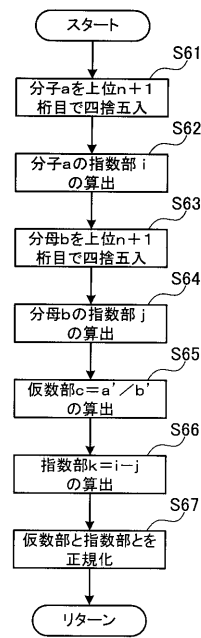
【図7】



【図8】



【図9】



---

フロントページの続き

- (56)参考文献 再公表特許第2002/058894(JP, A1)  
国際公開第2002/058894(WO, A1)  
特開2005-046931(JP, A)  
特開2007-007837(JP, A)

(58)調査した分野(Int.Cl., DB名)

B25J 1/00 - 21/02  
G05B 19/18 - 19/46  
G05B 19/00 - 19/16  
G05B 24/00 - 24/04  
G05B 15/00 - 15/02  
G05B 99/00