並列処理システムによる 作業座標系非線形フィードバック制御

橋 本 浩 一* 木 元 努* 大 橋 謙 次*
赤 井 健 二** 鹿 児 島 昌之* 木 村 英 紀

Cartesian-Space Nonlinear Feedback Control with a Parallel Processing System

Koichi HASHIMOTO*, Tsutomu KIMOTO*, Kenji OHASHI*,
Kenji AKAI**, Masayuki KAGOSHIMA*, Hidenori KIMURA*

The dynamic equations of a manipulator are modeled by a system of second-order nonlinear differential equations with its kinematic and dynamic parameters. On the basis of this model, the manipulator can be controlled in a strict way by compensating all dynamic torques/forces consist of inertial, centrifugal, Coriolis, and frictional effects. However, since these dynamic torques/forces are highly complex functions of joint positions and velocities, the computational burden for evaluating these torques is significant. Further, the Cartesian-space control, in which positions and velocities are controlled directly in Cartesian work space, requires considerable amount of additional computations for inverse kinematics. Hence it has been difficult to implement the realtime model-based control with conventional commercially available microprocessors.

This paper presents a real-time implementation scheme of Cartesian-space nonlinear feedback control based on a new parallel computation scheme called *Resolved Newton-Euler algorithm*. Some experiments on the basic three joints of the PUMA 560 manipulator are also reported by using a parallel processing system with multiple microprocessors. A sampling period of 0.79msec is achieved and fairly good control performances are obtained.

Key Words : Cartesian-space Control, Model-based Control, Parallel Processing, Transputer, Occam.

1. は じ め に

マニピュレータの機構学的および動力学的パラメータを用いると、マニピュレータの運動方程式は非線形2階微分方程式系でかなり正確にモデル化される。つまり、パラメータが精度よく求まれば逆動力学問題を解くことにより、慣性力、遠心力、コリオリ力、重力、および、摩擦力をすべて補償することができ、厳密な制御が行えるはずである¹⁾。しかし現状では、実用化されているマニピュレータに対してこのような非線形フィードバック

制御は行われていない。その主な理由としては、

(1) 逆動力学問題を解くのに要する計算量が非常に多い。さらに、作業座標系で直接制御するためには機構学の計算も必要である。

(2) マニピュレータの各種パラメータを精度よく求めることが困難である。

(3) ギアによる減速機構のため摩擦が大きく、そのモデル化が困難である。

などがあげられる。(2)の問題に対しては、最近いくつかの同定アルゴリズムが開発され^{2,3)}、統一的評価基準も提案されている⁴⁾。(3)の問題は、ダイレクトドライブマニピュレータの開発により本質的に解決されつつあ

原稿受付 1989 年 7 月 27 日

* 大阪大学工学部 ** 山武ハネウェル株式会社

る。一方、(1)の問題については、多くの研究者により逆動力学計算の手数を減らしたアルゴリズムが提案された⁵⁻¹¹⁾。それらのうちマニピュレータ特有の構造を利用しない場合に最も効率的なアルゴリズムは Newton-Euler 法⁵⁾であろう¹²⁾。しかし NE アルゴリズムを用いても、ハードウェアを改良しなければ実時間処理は困難である。ハードウェアの工夫を行うとき、工夫とはなんらかの意味で並列処理を取り入れることと解釈できる。

まず、最も粒度の低いレベルの並列処理として特徴づけることのできるのは、マニピュレータ制御専用のハードウェアを設計、開発するアプローチである^{13,14)}。このアプローチでは非常に高速な処理装置が期待できるが、現状の LSI 技術にそぐわないアーキテクチャが要求されたり、開発コストが莫大なものになるなどの問題があり実用化はされていない。

つぎに粒度の低いレベルでは、ビットスライスアーキテクチャに基づくもの¹⁵⁾や、ワードレベルでの並列処理に基づくものである。ほとんどの DSP やアレイプロセッサは乗算器と加算器がパイプラインで使えるなどのワードレベルの並列処理を用いており、16)-22) のアプローチはこのレベルに属する。DSP やアレイプロセッサを用いて実用的に十分高速な処理が実現されているが、いずれも、ソフトウェアには低レベルの言語を用いている。その結果ソフトウェアの開発に時間がかかり仕様の変更に対応できないという欠点を持つ。

実際のマニピュレータに対して適用された例として、例えば、末広と高瀬は作業座標系動的制御をサンプリング周期 2.5 msec で実現した¹⁶⁾。サーボルーチンはアセンブラで書かれている。16 ビット固定小数点演算に基づき、速度の2乗項を省略し、マニピュレータの機構的特徴を利用して計算量を減らし、三角関数はテーブルルックアップ方式で処理するなどの簡略化がされており、これにより、高速な処理が可能となったが精度が落ちた。また、マニピュレータおよび制御法が変更されると、ほとんどすべてのソフトウェアをアセンブラで書き直さなければならないので、汎用性を持っているとはいえない。一方、Khosla らは関節座標系での制御をサンプリング周期 2 msec で実現している^{17,18)}が、マニピュレータの機構的特徴を利用して計算量を減らし⁸⁾、ソフトウェアにはやはりアセンブラを用いているので、汎用的であるとはいえない。An らは汎用マイクロプロセッサで C 言語を用いて 3 軸制御している^{19,20)}が、サンプリング周期は 7.5 msec で十分な処理速度ではない。しかし今後は、高速高機能高級言語対応の DSP も市販されつつあり、それにとまって、上記のアプローチの実用性が増すと思われる。

最後に、最も汎用的なアプローチとして複数のマイクロプロセッサに基づく方法がある。そのうち、スケジューリングアルゴリズムを用いる方法^{23,24)}ではプロセッサをバスで結合する必要があるが、バス結合で最も重要な通信の方法について述べられておらず、バスの溢れによる性能低下が懸念される。また、Nigam と Lee の提案したアーキテクチャ²⁵⁾では、バスの制御が非常に複雑になり、バスの制御に要する時間のために計算時間が増加すると思われる。一方、プロセッサ間をローカルな通信線で結合する方法は PTP (Point-To-Point) 通信と呼ばれ、ネットワーク形状が固定されるがバスのアービトレーションの問題がない。したがって通信と同期が非常に簡単にプロセッサの性能をおとすことなく実現できるという特徴を持つ²⁶⁾。著者らは複数のマイクロプロセッサの PTP 通信による実現に適した逆動力学計算の並列処理アルゴリズムとして Resolved Newton-Euler (RNE) 法を提案した²⁷⁾。このアルゴリズムは、逆動力学の計算を並列に実行可能な、ほぼ同じ計算量を持つ複数のサブプロセスに分解したものである。これらのサブプロセスを複数のプロセッサに割り当てることにより並列処理が行える。全てのサブプロセスはたがいに局所的かつ規則的に通信を行うので、通信の方法は近傍のプロセッサとの PTP 通信でよい。逆機構学の計算(ヤコビアンおよびその逆行列の計算)は RNE 法の一部のサブプロセッサと並行して計算可能であり、RNE アルゴリズムを用いることにより作業座標系での制御のための計算を高速に行うことができる。

本報告では、作業座標系での動的な制御 (Resolved Acceleration Control²⁸⁾) を行うための並列処理アルゴリズムと並列処理システムについて述べ、PUMA 560 を用いて実験した例を示す。並列処理システムはホストとなる 16 ビットパーソナルコンピュータ (NEC PC9801) と 32 ビットのマイクロプロセッサ (Transputer²⁹⁾) のネットワークで構成される。全てのプログラムは並列処理記述用高級言語 (Occam^{30,31)}) で書かれた。動力学および機構学パラメータは CAD を用いて設計図から求めた。また、摩擦に関しては粘性摩擦のみを考慮し、実験により同定した。PUMA 560 の基本 3 軸による軌道制御がサンプリング周期 0.79 msec で実行され、良好な制御性能が得られた。

本報告は、以下のように構成される。2 章ではマニピュレータのモデルと作業座標系での 3 種類の制御法、3 章で並列処理アルゴリズム、そして 4 章で並列処理アーキテクチャについて述べる。作業座標系での実時間制御の実験結果および計算時間についての報告を 5 章で行う。

2. 作業座標系における制御法

2.1 マニピュレータのモデル

n -自由度のマニピュレータに対し、作業座標系におけるハンドの位置を x 、速度を $\dot{x}$ 、加速度を $\ddot{x}$ と書き、関節座標系における関節位置を q 、速度を $\dot{q}$ 、加速度を $\ddot{q}$ と書くとき、マニピュレータの機構学的モデル ψ は、

$$x = \psi(q) \quad (1)$$

で与えられる。ヤコビアン J を用いて速度、加速度の関係式は、

$$\dot{x} = J\dot{q} \quad (2)$$

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (3)$$

となる。作業座標系で表されたハンドで発生する力 f と各関節で発生するトルク τ の間には仮想仕事の原理より

$$\tau = J^T f \quad (4)$$

が成り立つ。

一方、マニピュレータの動力学モデル R はマニピュレータへの入力をアクチュエータトルク τ 、出力を関節位置 q 、速度 $\dot{q}$ 、加速度 $\ddot{q}$ とみなし、

$$[q, \dot{q}, \ddot{q}] = R(\tau) \quad (5)$$

で与えられる。逆に、関節データ $q, \dot{q}, \ddot{q}$ を達成するためのアクチュエータトルク τ を求めるための微分方程式

$$\tau = R^{-1}(q, \dot{q}, \ddot{q}) \quad (6)$$

$$= H(q)\ddot{q} + C(q, \dot{q}) + g(q) + \tau_c \operatorname{sgn}(\dot{q}) + \tau_v \dot{q} \quad (7)$$

は逆動力学モデルと呼ばれる。 H は慣性行列、 C は遠心力、コリオリ力の項、 g は重力項、 τ_c はクローン摩擦、 τ_v は粘性摩擦を表す。ふつうこれらのパラメータの正確な値はわからないので、それらの推定値 $\hat{H}, \hat{C}, \hat{g}, \hat{\tau}_c, \hat{\tau}_v$ を用いて制御することになる。

2.2 Stiffness Control¹⁹⁾

作業座標系において発生すべき力 f をスカラーの位置のゲイン K_p 、速度のゲイン K_v を用いて

$$f = K_p(x_d - x) + K_v(\dot{x}_d - \dot{x}) \quad (8)$$

と考える。ここで x_d は作業座標系での望ましい軌道、 $\dot{x}_d$ は望ましい速度を意味する。いま、(4) 式にもとづき、アクチュエータトルクを

$$\tau = J^T(K_p(x_d - x) + K_v(\dot{x}_d - \dot{x})) \quad (9)$$

と決める。この制御則をここでは Stiffness Control と呼ぶことにする。ブロック図を Fig.1 に示す。

2.3 Resolved Motion Rate Control³²⁾

関節座標系での微小な変位 δq に対し、その方向にアクチュエータトルク τ を加えるという制御法が考えられる。作業座標系での微小変位 δx を $K_1(x_d - x) + K_2(\dot{x}_d - \dot{x})$ とすると、制御則は (2) 式にもとづいて、ス

Fig.1 Stiffness Control.

カラー k を用いて

$$\tau = k\delta q = J^{-1}k\delta x \quad (10)$$

$$= J^{-1}(K_p(x_d - x) + K_v(\dot{x}_d - \dot{x})) \quad (11)$$

と決定できる。ここで、 $K_p = kK_1$ 、 $K_v = kK_2$ である。この制御法は Resolved Motion Rate Control と呼ばれる。ブロック図を Fig.2 示す。

2.4 Resolved Acceleration Control²⁸⁾

いま、操作加速度 $\ddot{x}$ を

$$\ddot{x} = \ddot{x}_d + K_p(x_d - x) + K_v(\dot{x}_d - \dot{x}) \quad (12)$$

のように決めると、(3) 式より操作関節加速度 $\ddot{q}$ は

$$\ddot{q} = J^{-1}(\ddot{x} - \dot{J}\dot{q}) \quad (13)$$

となる。逆動力学モデル ((6) 式、(7) 式) にもとづき、アクチュエータトルク τ を

$$\tau = \hat{R}^{-1}(q, \dot{q}, \ddot{q}) \quad (14)$$

$$= \hat{H}(q)\ddot{q} + \hat{C}(q, \dot{q}) + \hat{g}(q) + \hat{\tau}_c \operatorname{sgn}(\dot{q}) + \hat{\tau}_v \dot{q} \quad (15)$$

と決める制御法は Resolved Acceleration Control と呼ばれる。ブロック図を Fig.3 に示す。ここで $\hat{\cdot}$ はそのパラメータが推定値であることを示す。

3. 並列処理アルゴリズム

2.4 で述べた作業座標系での動的制御 (Resolved Acceleration Control) を 5 つのステップに分け、3 軸制御の場合に対し、浮動小数点演算ユニット付きの 32 ビットマイクロプロセッサ T800²⁹⁾ を用いて逐次処理を行い、各々のステップに要する計算時間を調べた。結果を

Fig.2 Resolved Motion Rate Control.

Fig.3 Resolved Acceleration Control.

Table 1 に示す。全体を5ステップに分割したが、第5ステップの逆力学計算に要する時間が半分以上をしめる。

そこで著者は、複数のマイクロプロセッサでの実現に適した逆力学計算の並列処理のアルゴリズム (Resolved Newton-Euler Algorithm) を開発した²⁷⁾。これは、Luh らによって提案された Newton-Euler 法⁵⁾を複数のサブプロセスに分解し並列性を抽出したものである。並行して処理可能なサブプロセスを明示的にプロセッサに割り当てることにより並列処理を記述する。この RNE アルゴリズムを用いると、NE アルゴリズムを直感的に分かりやすい手法でタスクアサインメントする方法³³⁾ (ここではこの方法を Parallel Newton-Euler 法と呼ぶ) の 60% 以下の計算時間で済む。3 軸の場合に対し、このことをデータ依存関係図を用いて以下に示す。

NE 法は、正順アルゴリズムと、逆順アルゴリズムか

Table 1. Tasks for resolved acceleration control and computational time for each tasks.

STEP	Computations of...	Time(μ s) [†]
1	$q, \dot{q}$ from encoder signal, $\sin q, \cos q$	291 ^{††}
2	x in Eq.(1)	183
3	$J, \dot{J}\dot{q}$ in Eqs.(2) and (3)	203
4	$\ddot{q}$ in Eq.(13)	125
5	τ in Eq.(15) -- ID ^{†††}	886

By T800 (20 MHz), sequential program.

[†] Times are measured by using transputer's internal clock.

^{††} The computations required by interpolation of desired trajectory $x_d, \dot{x}_d$ and $\ddot{x}_d$ are included in STEP 1.

^{†††} ID means Inverse Dynamics.

らなる。正順部では、マニピュレータのベースからバンドに向かって、各リンクの角速度 ω 、角加速度 $\dot{\omega}$ 、各リンク座標系の原点の加速度 $\dot{v}$ 、各リンクの重心の加速度 $\dot{v}^*$ 、各ジョイントに作用する力 f とモーメント n の順に計算する。逆順部では、バンドからベースに向かって、各リンクに対して前のリンクから作用する力 f とモーメント n 、そして、アクチュエータトルク τ の順に計算する。データ依存関係を Fig.4

に示す。各列にならべられたプロセスを、各リンク用のプロセッサに割り当てれば、直感的に分かりやすい並列処理方法 (PNE 法) が得られる。Fig. 4 は、たとえば、 $\dot{\omega}^2$ は $\dot{\omega}^1$ と $\dot{\omega}^2$ に依存していることを表す。つまり、“リンク 2 プロセッサは $\dot{\omega}^2$ の計算を終了し、リンク 1 プロセッサから $\dot{\omega}^1$ を受け取ると、 $\dot{\omega}^2$ の計算を開始することができる” ということを表している。Fig. 4 より、リンク 3 プロセッサは $\dot{\omega}^3$ の計算を開始する前、そしてリンク 1 プロセッサは N_1 の計算を終了した後、アイド

Fig.4 Data dependence of NE algorithm.

ルの時間があり効率がよくない。

そこで RNE 法では, ω, v , および v^* は, 3次元ベクトル係数を持つジョイントの速度 $\dot{q}$ の一次結合として

$$\omega^k = \sum_{i=1}^k \omega_i^k \dot{q}_i \quad (16)$$

$$v^k = \sum_{i=1}^k v_i^k \dot{q}_i \quad (17)$$

$$v^{k*} = \sum_{i=1}^k v_i^{k*} \dot{q}_i \quad (18)$$

と表せることに注目する。ここで, 角速度と速度の表現に含まれる係数ベクトルはそれぞれ, angular partial velocity, linear partial velocity と呼ばれる³⁴⁾。 $i \neq j$ なる i, j に対し, $\dot{q}_i$ の係数は $\dot{q}_j$ の係数に依存しないので, すべてのジョイント変数の係数は並行して計算可能である。またリンク i のアクチュエータトルクは, これらの partial velocities を用いて

$$\tau_i = \sum_{k=i}^n (N_k \cdot \omega_i^k + F_k \cdot v_i^{k*}) \quad (19)$$

のように求めることができる。これより, RNE 法のデータ依存関係は Fig.5 のように表すことができる。ここで, ξ は 3×3 行列

$$\xi_i^k = (\omega_i^k, v_i^k, v_i^{k*}) \quad (20)$$

を意味する。Fig. 4, Fig. 5 において, 各々のステップはほぼ同量の計算量を有し計算時間も同等である。したがって, ステップ数を比較することにより, RNE 法にもとづくと PNE 法を用いる場合の約 60% の計算時間で済むことが示される。

複数のプロセッサで並列処理を行う場合, とくに通信

Fig.5 Data dependence of RNE algorithm.

の速度があまり速くない場合, 計算量と通信量のトレードオフに注意してアルゴリズムの開発を行わなければならない。RNE 法においても, 通信のオーバーヘッドを考慮して通信のタイミングを決定することにより PNE 法の 60% 以下に計算時間を短縮することができる。

逆動力学計算において, ξ および ω は $\dot{q}$ に依存しないので, ξ および ω の計算は Table 1 のステップ 3, 4 と並行して実行可能である。また, 目標軌道 $x_d, \dot{x}_d, \ddot{x}_d$ は与えられた目標ポイント間の軌道を 5 次多項式で加速度が連続となるように内挿することによりオンラインで求める。この計算は 1 サンプル周期先の目標軌道を前もって求めておくことにすれば逆動力学計算と並行して実行可能である。これらのことに注意して並列処理のプログラムを書くことにより, プロセッサの使用効率を上げることができる。

逆機構学計算において, 3 自由度マニピュレータの場合, $J = [J_1, J_2, J_3]$ と書くとき,

$$\begin{aligned} j\dot{q} = & (z_0 \times J_1) \dot{q}_1^2 + (z_1 \times J_2) \dot{q}_2^2 + (z_2 \times J_3) \dot{q}_3^2 \\ & + 2(z_0 \times J_3) \dot{q}_1 \dot{q}_3 + 2(z_1 \times J_3) \dot{q}_2 \dot{q}_3 \\ & + 2(z_0 \times J_2) \dot{q}_1 \dot{q}_2 \end{aligned} \quad (21)$$

が成り立つ。ここで z_i は基本座標系における第 $i+1$ ジョイントの回転軸方向ベクトルである。3 自由度の場合は (21) 式および J^{-1} を直接計算しても実行速度にはあまり影響しないが, 6 自由度マニピュレータに拡張する場合は spherical wrist の特徴を利用し, $\ddot{x} \rightarrow \ddot{q}$ の変換を簡単化する必要があると思われる^{35,36)}。

4. 並列処理アーキテクチャ

並列処理システムの構成を Fig.6 に示す。コントローラは, 16 ビットのパーソナルコンピュータ (NEC PC9801 VX41) をホストとし, 拡張スロットに Transputer T414 がアドオンされている。T414 は 1 チップ上に 10MIPS の 32 ビットマイクロプロセッサ, 4 本の 20Mbps の通信用シリアルリンク, 2K バイトの RAM とメモリーインターフェースをもっている²⁹⁾。最大の特徴はシリアルリンクで, これにより他のプロセッサと PTP 通信することが可能になる。

ネットワークは浮動小数点演算ユニット付きの Transputer T800 で構成されている。T800 は T414 のオンチップメモリを 4K バイトに拡張し 1.5Mflops の FPU をつけたものである²⁹⁾。C011 リンクアダプタはシリアルデータと 8 ビットのパラレルデータの変換を行う²⁹⁾。Transputer と C011 はシリアルリンクで通信する。

関節角度はエンコーダの出力をアップダウンカウンタを用いて計測する。関節角速度は関節角度の差分により

Fig. 6 Configuration of the controller.

得る。パルスカウンタは 16 ビットのものを用いた。16 ビットデータと 2 バイトのデータの間の変換を行うための 16/2*8 インターフェースを製作した。モータへの入力には PUMA 付属の VAL コントローラのサーボアンプへの電流指令で行う。D/A コンバータは 12 ビットのものを用いた。

プログラムはすべて TDS (Transputer Development System³⁷⁾) の環境下で、並列処理記述用言語 Occam で書かれた。Occam のチャンネルと Transputer のリンクにより PTP 通信が非常に簡単に実現される³⁹⁾。

5. 実験結果

PUMA560 の基本 3 軸を用いて軌道制御を行った。手首 3 軸は第 3 リンクに固定した。PUMA560 の機構学的および動力学的パラメータは設計データにもとづき CAD で求めた。摩擦は粘性摩擦のみを考慮し、往復することで重力の影響が相殺されるような軌道を等速で動かし、それに必要な駆動トルクを求め同定した。

第 1 リンクの回転軸と床面との交点を原点として、この回転軸に沿って上向きに z 軸を、第 2 リンクの床面への投影方向へ y 軸をとった。そしてこれらと右手系をな

Table 2 (a) Desired points in Cartesian space

Point	p1	p2	p3	p4	p5	p6
p_x	50	-15	50	-15	50	50
p_y	15	50	15	50	15	15
p_z [cm]	120	100	80	60	40	120
t [sec]	1.7	1.7	1.7	1.7	2.5	

(b) Desired points in Cartesian space

Point	p1	p2	p3	p4	p5
p_x	50	24	-2	-15	50
p_y	15	29	43	50	15
p_z [cm]	120	40	120	40	120
t [sec]	2.0	3.0	2.0	3.0	

すように x 軸をとって基本座標系とした。Table 2(a), (b) に示される 2 通りの軌道に対して実験を行った。 p_x, p_y, p_z はハンドの目標ポイント、 t は各ポイント間の移動時間を示す。目標軌道は加速度が連続関数となるように 5 次の多項式で各ポイント間を内挿した。

各種制御方式、Stiffness Control (SC), Resolved Motion-Rate Control (MR), Resolved Acceleration Control (RA) を用いた場合の基本座標系での軌道とトラッキングエラーを Fig.7~12 に示す。制御性能を最大誤差で評価したものを Table 3 に示す。各々の制御方式に対しゲイン K_p, K_v は独立に実験により求めた。実験的に K_p, K_v の上限を求め、 K_p は上限の 80%, K_v は 40% に設定した。すべての制御方式においてサンプリング周期は 0.79 msec で実験した。これらの結果より、モデルにもとづく動的制御 (RA) を用いることで制御性能が改善されることが示された。

並列処理アルゴリズムにおけるタスクとその所用時間を Table 4 に示す。第 1 ステップでは processor #1-#3 は、エンコーダからデータを読み、 q を求め、その $\sin, \cos$ を計算している。プロセッサ間の通信は 20 Mbps で行われ、32 bit データの通信には $2.4 \mu\text{sec}$ かかる。エンコーダから角度情報を取り入れるのに要する時間は D/A コンバータにデータを転送するのに要する時間とほぼおなじで $4.3 \mu\text{sec}$ である。 $65 \mu\text{sec}$ のうちの残りのほとんどは三角関数の計算に要した時間である。第 2-第 4 ステップでは processor #0 が逆機構学計算を行っておりクリティカルパスとなっている。第 5 ステップでは processor #1-#3 は RNE 法にもとづき逆動力学計算を行っている。 ξ, ω の計算は processor #0 の第 3,

第4ステップと並行して実行される。したがって、第5ステップに要する時間は Table 1 に示されている第5ステップの所要時間の 1/3 以下になる。マニピュレータの機構的特徴に依存しないアルゴリズムに対し、逆動力学的計算に要した時間の総計は、3個のプロセッサで並列処理した場合は $323 \mu\text{sec}$ 、1個のプロセッサによる逐次処理では $886 \mu\text{sec}$ である。RNE アルゴリズムを用いると、3個のプロセッサにより、36.5%の時間で処理

可能となりかなり並列処理効率が高いことが示された。このことから、6軸の制御用にミシステムを拡張すれば（プロセッサを8個用いて spherical wrist の特徴を利用すれば）6軸の作業座標系の動的制御も実時間で実現可能であると思われる。

6. 結 論

作業座標系における非線形フィードバック制御

Fig.7 Trajectory (a) and tracking errors of x-axis.

Fig.8 Trajectory (a) and tracking errors of y-axis.

Fig.9 Trajectory (a) and tracking errors of z-axis.

Fig.10 Trajectory (d) and tracking errors of x-axis.

Fig.11 Trajectory (d) and tracking errors of y-axis.

Fig.12 Trajectory (d) and tracking errors of z-axis.

Table 3. Tracking error

TRJ	LAW	x [mm]	y [mm]	z [mm]
(a)	SC	1.65	1.39	1.88
	MR	1.83	0.96	1.83
	RA	1.46	0.72	1.41
(b)	SC	1.38	0.94	1.89
	MR	1.07	0.63	1.61
	RA	0.84	0.31	1.19

Table 4. Tasks for resolved acceleration control with resolved Newton-Euler algorithm.

STEP	Computations of...		Time(μ s)
	Processor #1-#3	Processor #0	
1	q $\sin q, \cos q$	Timer interruption	65
2	$\dot{q}$	x	165
3	ID	$J, J\dot{q}$	206
4		$\ddot{x}, \ddot{q}$	128
5	323	Interporation of $x_d, \dot{x}_d$ and $\ddot{x}_d$	231

をPUMA560に実装した。RNE アルゴリズムにもとづく並列処理システムにより基本3軸の制御がサンプリング周期1msec以下で実現された。6軸の制御は8個のプロセッサを用いれば、やはり1msec以下で実現できることが予想される。コントローラは高級言語でプログラミングでき十分な汎用性がある。軌道制御における良好な制御性能が示されたが、力制御などのいろいろなモデルにもとづく制御がこの並列処理システムにより実時間で実現できると思われる。さらに、より複雑なタスクを実行するためにはモデルの精度をあげることが必要であり、各種のセンサ（力センサ、視覚等）との融合も望まれる。これらは今後の課題であり、ここで述べたシステムはこれらの課題の解決に向けて中心的な役割をはたすことが期待される。

謝 辞

PUMA560を提供していただきました久保田鉄工株式会社に感謝致します。PUMA560のパラメータ計算にCADを使わせていただき、さらに、多くの助言を下さいました川崎重工業株式会社の黒崎 康充氏、西 義和氏、五百井 清氏、狩屋 芳樹氏に感謝致します。

参 考 文 献

- 1) K. S. Fu, R. C. Gonzalez and C. S. G. Lee, Robotics : control, sensing, vision, and intelligence, New York, McGraw-Hill, 1987.
- 2) 川崎, "ロボットアームのパラメータ同定", 計測と制御, Vol.28, No.4, pp.344-350, 1989.
- 3) 前田, "ロボットアームの動的モデルと同定", 日本ロボット学会誌, Vol.7, No.2, pp.203-208, 1989.
- 4) 吉川, "マニピュレータに対するパラメータ同定試験の評価", 計測自動制御学会論文集, Vol.23, No.11, pp.111-113, 1987.
- 5) J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "On-line computational scheme for mechanical manipulators", ASME J. Dynamic Systems, Measurements and Control, Vol.102, pp.69-76, 1980.
- 6) J. M. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity", IEEE Trans. System, Man and Cybernetics, Vol.SMC-10, No.11, pp.730-736, 1980.
- 7) T. R. Kane and D. A. Levinson, "The use of Kane's dynamical equations in robotics", Int. J. Robotics Research, Vol.2, No.3, pp.3-21, 1983.
- 8) P. K. Khosla and C. P. Newman, "Computational requirements of customized Newton-Euler algorithms", J. Robotic Systems, Vol.2, No.3, pp.309-327, 1985.
- 9) M. Khalil, J. F. Kleinfinger and M. Gautier, "Reducing the computational burden of the dynamic models of robots", Proc. IEEE Int. Conf. Robotics and Automation, San Francisco, California, pp.525-531, April 1986.
- 10) 大須賀, 前田, 吉田, 小野, "仮想パラメータを用いた逆動力学問題の高速解法", 第31回システムと制御研究発表講演会, pp.149-150, 1987.
- 11) C. A. Balafoutis, P. Mislis and R. V. Patel, "A Cartesian tensor approach for fast computation of manipulator dynamics", Proc. IEEE Int. Conf. Robotics and Automation, Philadelphia, Pennsylvania, pp.1348-1357, April 1988.
- 12) 吉川, "計算トルク法によるディジタル制御", 日本ロボット学会誌, Vol.7, No.3, pp.237-242, 1989.
- 13) R. H. Lathrop, "Parallelism in manipulator dynamics", Int. J. Robotics Research, Vol.4, No.2, pp.80-102, 1985.
- 14) C. S. G. Lee and P. R. Chang, "Efficient parallel algorithm for robot inverse dynamics computation", IEEE Trans. System, Man and Cybernetics, Vol.SMC-16, No.4, pp.532-542, 1986.
- 15) M. Rahman and D. G. Meyer, "A cost-effective high-performance bit-serial architecture for robot inverse dynamics computation", IEEE Trans. System, Man and Cybernetics, Vol.SMC-17, No.6, pp.1050-1058, 1987.
- 16) 末広, 高瀬, "直接計算方式による作業座標サーボに基づくマニピュレーションシステム", 日本ロボット学会誌, Vol.3, No.2, pp.95-105, 1985.
- 17) P. K. Khosla and T. Kanade, "Experimental evaluations of nonlinear feedback and feedforward control schemes for manipulators", Int. J. Robotics Research, Vol.7, No.1, pp.18-28, 1988.

- 18) P. K. Khosla and T. Kanade, "Real-time implementation and evaluation of computed torque scheme", IEEE Trans. Robotics and Automation, Vol RA-5, No.2, pp. 245-253, 1989.
- 19) C. H. An, C. G. Atkeson and J. M. Hollerbach, Model-based control of a robot manipulator, Cambridge, Massachusetts, MIT Press, 1988.
- 20) C. H. An, C. G. Atkeson, J. S. Griffiths and J. M. Hollerbach, "Experimental evaluation of feedforward and computed torque control", IEEE Trans. Robotics and Automation, Vol RA-5, No.3, pp. 368-373, 1989.
- 21) 吉川, 中村, 横小路, "DSP による多関節マニピュレータの動的制御のための実時間計算", 日本ロボット学会誌, Vol.6, No.3, pp.175-183, 1988.
- 22) H. Mayeda, K. Kusamoto and K. Ohashi, "Robot controller with digital signal processor for the resolved acceleration control", Proc. U. S. A.-JAPAN Symposium on Flexible Automation, Minneapolis, pp.147-152, July 1988.
- 23) J. Y. S. Luh and C. S. Lin, "Scheduling of parallel computation for a computer-controlled mechanical manipulator", IEEE Trans. System, Man and Cybernetics, Vol:SMC-12, No.2, pp. 214-234, 1984.
- 24) H. Kasahara and S. Narita, "Parallel-processing of robot arm control on a multimicroprocessor system", IEEE J. Robotics and Automation, Vol RA-1, No. 2, pp. 104-113, 1985.
- 25) R. Nigam and C. S. G. Lee, "A multiprocessor-based controller for the control of mechanical manipulators", IEEE J. Robotics and Automation, Vol RA-1, No.4, pp. 173-182, 1985.
- 26) M. Milencvic, Operating systems: concepts and design, Chapter 5; Interprocess communication and synchronization, New York, McGraw-Hill, pp. 175-227, 1987.
- 27) K. Hashimoto and H. Kimura, "A new parallel algorithm for inverse dynamics", Int. J. Robotics Research, Vol.8, No.1, pp. 63-76, 1989.
- 28) J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "Resolved acceleration control of mechanical manipulators", IEEE Trans. Automatic Control, Vol AC-25, No.3, pp. 468-474, 1980.
- 29) Inmos Ltd., Transputer reference manual, Herts, UK. Prentice Hall, 1988.
- 30) Inmos Ltd., Occam 2 reference manual, Herts, UK. Prentice Hall, 1988.
- 31) R. H. Perrott, Parallel programming, Chapter 9; Occam: a distributed computing language, Workingham, UK. Addison-Wesley, pp. 113-128, 1987.
- 32) D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses", IEEE Trans. Man-Machine Systems, Vol MMS-10, No.2, pp. 47-53, 1969.
- 33) 吉川, ロボット制御基礎論, pp. 149-155, コロナ社, 1988.
- 34) T. R. Kane and D. A. Levinson, Dynamics: theory and applications, New York, McGraw-Hill, 1985.
- 35) R. Featherstone, "Position and velocity transformations between robot end-effector coordinates and joint angles", Int. J. Robotics Research, Vol.2, No.2, pp. 35-45, 1983.
- 36) J. M. Hollerbach and G. Sahar, "Wrist-partitioned, inverse kinematic acceleration and manipulator dynamics", Int. J. Robotics Research, Vol.2, No.4, pp. 61-76, 1983.
- 37) Inmos Ltd., Transputer development system, Herts, UK. Prentice Hall, 1988.
- 38) Inmos Ltd., Communicating process architecture, Herts, UK. Prentice Hall, 1988.