

加速度感覚提示機用パラレルマニピュレータの 並列実時間動力学計算と H_∞ 加速度フィードバック制御

中村 仁彦*1 山根 克*1 岡田 昌史*1 小嶺 徳晃*2

Parallel Real-Time Dynamics Computation and H_∞ Acceleration Feedback Control of Parallel Manipulators for Acceleration Display

Yoshihiko Nakamura*1, Katsu Yamane*1, Masafumi Okada*1 and Noriaki Komine*2

In this paper, we propose a control scheme of parallel manipulators focusing on the accuracy of acceleration on the endplate, which is an important factor when parallel manipulators are used as acceleration displays. We use two controllers — dynamic controller to achieve accuracy of position and to stabilize the system, and H_∞ controller to feedback the acceleration measured on the endplate. The main problem of dynamic control is computational complexity. In order to reduce computation time for inverse dynamics, parallel processing method called multi-thread programming is applied. H_∞ controller is added outside the closed loop of dynamic control to remove the vibration due to the elasticity of the links and the influence of modeling errors in the dynamic controller.

Key Words: Parallel Manipulator, Dynamics Computation, Parallel Computation, H_∞ Control, Acceleration Display

1. はじめに

パラレルメカニズムとは、エンドプレートとベースプレートが並列に配置された複数の脚リンク機構によって結合された構造を持つ閉リンク機構の一種であり、数多くの優れた特徴を持っているため、さまざまな分野で今後の利用が期待されている機構である [1]。パラレルメカニズムの応用分野の一つとして、本研究ではドライビングシミュレータなどの加速度感覚提示機を取り上げ、精密な加速度感覚を発生させるための制御方法について考える。

一般のマニピュレータでは位置や速度の精度が要求されるのに対し、加速度感覚提示機では加速度の精度が重要である。また人間を乗せるため可搬重量が大きいことも求められる。そのため、油圧駆動で出力の大きいスチュワートプラットホーム型のパラレルマニピュレータ [2] が広く用いられてきた。しかし、この機構は油圧駆動系の非線形性のため精密な加速度・力制御を行うことが難しい。この欠点を改良するため、Fig. 1 のように回転関節または球面関節で構成されたパラレルメカニズムがいくつか提案されている [3] [4]。この機構ではベースプレートに固定された回転関節を駆動するため、アクチュエータとして

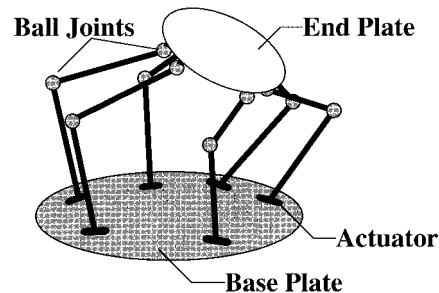


Fig. 1 Parallel mechanism with rotational/spherical joints

制御の容易なサーボモータなどを用いることができる。また、可動部分にアクチュエータがないのも大きな特徴である。本研究では、各脚リンク機構が六つの回転関節で構成されたパラレルマニピュレータを製作し、加速度感覚提示機への応用を目指す。

加速度感覚提示機は構造体自身が大きいので、動力学を考慮した制御を行うことが望ましい。動的制御は、機構の動特性を考慮して駆動トルクを計算する制御方法で、慣性力が無視できない高速運動などに有効であり、シリアルマニピュレータに対してはすでに多くの成果が報告されている [5]。しかし、動的制御では測定された関節値からエンドリンクの位置・姿勢を計算する順運動学計算と、目標加速度を実現するための関節トルクを計算する逆運動学計算が必要であり、サンプリングタイムが長くなることによる問題が起こりやすい。パラレルメカニズム

原稿受付 1999年5月26日

*1 東京大学大学院工学系研究科機械情報工学専攻

*2 島津製作所

*1 Department of Mechano-Informatics, University of Tokyo

*2 Shimadzu Corporation

では、受動関節の関節値が測定できないこと、および閉リンク機構であることからいずれも膨大な計算を必要とする。

パラレルメカニズムの順運動学問題は数学の分野でも研究が行われている [6] [7] が、制御を目的とした計算法としてはヤコビアンを用いた近似計算法 [8] や数学的手法を応用した計算法 [9] が提案されている。また逆動力学では、一般の閉リンク機構 [10] ~ [12] やパラレルメカニズム [13] の逆動力学・順動力学計算の効率化を目的とした研究が行われている。動的制御の研究としては、内山ら [14] が脚リンクの質量がエンドプレートに集中しているとする簡易動力学モデルを用いて計算量を減らし、実時間で動的制御を実現したものがある。この方法は高速性を重視した軽量のマニピュレータに対しては有効だが、本研究で対象としているような大規模な構造では脚リンクの慣性力なども無視できないため、厳密な動力学モデルが必要である。

動力学計算に要する時間を短縮する方法として、パラレルメカニズムの機構の並列性を利用した並列計算が考えられる [13] [15]。小嶺ら [16] はこれを利用し、並列計算により計算時間を短縮したが、実時間で厳密な動的制御には至っていない。並列計算は一般のリンク系の動力学計算でも注目されるようになってきている [17] が、その実装にはプロセス間通信や同期に伴うオーバーヘッド、利用できるソフトウェアやハードウェアの問題がある。並列計算の実装方法を確立することはパラレルメカニズムの制御だけでなく、様々な目的の運動計算の計算時間短縮に貢献すると考えられる。

本論文では、リアルタイムでの厳密な動的制御を実現するために、並列性が高く、並列計算による計算時間短縮の効果が大きい動的制御アルゴリズムを提案する。次に、マルチスレッドプログラミングにより並列計算をインプリメントすることを提案し、これにより実時間で動的制御を実現する。動的制御実験の結果、位置追従性に関しては性能の向上が見られたものの、高速運動においてはリンクの弾性変形による大きな振動と摩擦による立ち上がりの遅れが観察された。これらを除去するため、エンドプレート上で発生する加速度を測定しフィードバックするコントローラを動的制御系の外に追加することを考え、 H_∞ 制御によりコントローラを設計する。本研究で採用した設計法は、システム同定とそれに基づく補償器設計を繰り返し系統的に行うことができるもので、従来の補償器設計に伴う試行錯誤をなくし、効率的な設計が可能である。最後に、動的制御系と加速度フィードバック制御系を組み合わせた制御系により、高速運動でもエンドプレート上に滑らかで正確な加速度が実現されることを実験結果により示す。

2. 並列性を考慮した動的制御アルゴリズム

2.1 パラレルメカニズムの動力学

パラレルメカニズムの逆動力学計算は次の二つの手順で行う [13]。

- (1) パラレルメカニズムの閉ループを仮想的に切断した仮想開リンク機構が、もとのパラレルメカニズムと同じ運動を行うのに必要な駆動トルク τ_O の計算
ここでは開リンク機構と同様に Newton-Euler Formulation [18] などを適用する。

- (2) τ_O を実際のパラレルメカニズムの駆動トルク τ_A に変換する計算

仮想開リンク機構の関節値を θ_O 、実際の駆動関節の関節値を θ_A 、パラレルメカニズムの状態を表す一般化座標を θ_G として、二つのヤコビアン W, S を次のように定義する。

$$W = \frac{\partial \theta_O}{\partial \theta_G} \quad (1)$$

$$S = \frac{\partial \theta_A}{\partial \theta_G} \quad (2)$$

仮想仕事の原理から、仮想開リンク機構のトルク τ_O 、実際の駆動トルク τ_A 、一般化力 τ_G の間に以下の関係がある。

$$\tau_G = W^T \tau_O \quad (3)$$

$$\tau_G = S^T \tau_A \quad (4)$$

この2式を用いて (1) で計算された τ_O から τ_G を経て τ_A を計算する。

6自由度のパラレルメカニズムでは、一般化座標 θ_G としてエンドプレートの位置・姿勢を用いることができる。このとき二つのヤコビアン W, S は各脚リンク機構のヤコビアンから容易に構成することができる [13]。

2.2 動的制御に必要な計算

本論文では、中村 [13] が提案したパラレルメカニズムの逆動力学計算法と、Luhら [5] の分解加速度制御則を適用し、パラレルメカニズムの動的制御を行う。この制御系のブロック線図は Fig. 2 のようになる。具体的な計算手順を以下に示す [13] [16]。

- (1) 駆動関節の現在の関節値 θ_A よりエンドプレートの現在位置・姿勢 x_E を計算する (順運動学)
パラレルメカニズムでは順運動学計算が極めて困難なため、ヤコビアンを用いた逐次計算法 [8] により数値的に求める。
- (2) x_E より全関節角 θ_{all} を計算する (逆運動学)
エンドプレートの位置・姿勢から受動関節角を求める。シリアルマニピュレータの逆運動学問題と同様一般的な解法はなく、機構に応じた解析解 [18] またはヤコビアンを用いた逐次計算法を用いる。
- (3) Resolved acceleration control [5] に基づく実現すべきエンドプレート加速度の計算
次式によりエンドプレートの実現すべき加速度を計算する。

$$\ddot{x}_E = \ddot{x}_{Ed} + K_V(\dot{x}_{Ed} - \dot{x}_E) + K_P(x_{Ed} - x_E) \quad (5)$$

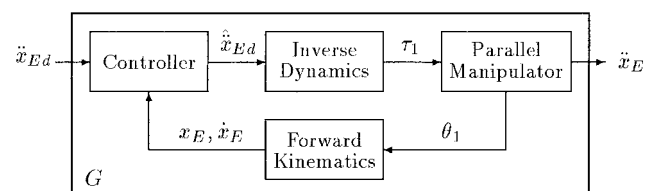


Fig. 2 Block diagram of dynamic control

ここで x_{Ed} はエンドプレートの目標位置・姿勢, K_V , K_P は定数のゲイン行列である.

- (4) トルク変換ヤコビアン計算
トルク変換に用いるヤコビアン W , S を計算する.
- (5) トルク計算 (逆力学)
仮想開リンク機構の駆動トルク τ_O を Newton-Euler Formulation [18] などにより計算する.
- (6) 閉リンクでのトルクへの変換
式(3)(4)を用いて τ_O を実際の駆動トルク τ_A に変換する.

回転入力型パラレルメカニズムでは受動関節の関節角が計測できないため(2)の計算が必要となる. また(6)は(5)で得られた仮想開リンク機構におけるトルクを実際の閉リンク機構におけるトルクに変換する計算である.

これらのうち(2)と(5)を並列に計算することにより計算時間を短縮できる可能性がある. しかし, 並列計算可能な部分が並列計算不可能な部分(3)と(4)によって二つに分かれ, 並列計算を行ったときにオーバーヘッドが大きくなると考えられる. また切替の回数が多いとインプリメントも困難になる. したがって, 並列計算部分はなるべく一つにまとまっていることが望ましい.

そのために, 回転入力型パラレルメカニズムの6本の脚を, エンドプレートをエンドリンクとする6関節の独立したシリアルメカニズムとみなし, 各脚の運動学, 動力学的計算に統合化計算法[19]をそのまま適用することにより, アルゴリズムの並列性を高めることを考える.

2.3 並列性を考慮したアルゴリズム

(3)を各脚で独立に計算できるようにするため, 以下のようにして各脚の独立性を高める.

- リンク座標系は通常のマニピュレータと同様に配置するが, エンドリンク座標系の姿勢はエンドプレート座標系の姿勢と一致させる.
- 各シリアルメカニズムの手先をエンドプレート座標系の原点に置く. これにより, エンドリンクの目標速度と目標加速度はエンドプレートと一致する. また, 各脚のエンドリンクの目標位置を得るにはエンドプレートの目標位置から脚リンク機構のベースの位置を差し引けばよい.
- 仮想開リンク機構の逆力学を計算する際には, 任意の1本の脚を除きエンドリンクの質量, 慣性行列としてそれぞれ 0 , O を与える. これは, エンドプレートとの間の関節を仮想切断される脚にも仮想的なエンドプレートを与えることにより, すべての脚で式(5)を計算することができるようにするためである. 質量, 慣性行列を零にするのは閉リンク機構から仮想開リンク機構への変換の際に脚先の関節が仮想的に切断されることに対応する.

これにより, 次のような新たな動的制御アルゴリズムが構築される.

- (1) 順運動学計算を行いエンドプレートの現在位置を求める.
- (2) 以下, 各脚で並列に計算する.
 - (a) エンドプレートの目標位置を脚のエンドリンクの目標位置に換算する.

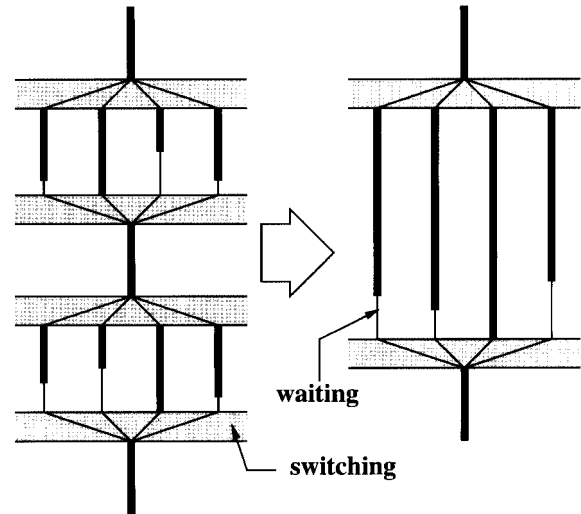


Fig. 3 Unifying parallel-computation part

- (b) 逆運動学計算を行い, すべての関節角を求める. これで関節角がすべて計測できる開リンク機構とまったく同じ情報が得られたことになる.
 - (c) 統合化計算法[19]に従って順運動学, 逆力学の計算を行い, 仮想トルクを求める.
 - (d) トルク変換に用いるヤコビアンを作る.
- (3) 得られたトルクを実際のパラレルメカニズムのトルクに変換する.

すべての脚で運動学, 動力学的計算を行うため, 計算には若干の重複が生じる. 例えば, エンドリンクの現在の速度は実際にはエンドプレートの速度であり, すべての脚で同じ値を計算している. しかし, 重複する計算のほとんどは統合化計算法の後のステップで必要になるものであり, 重複による総計算量の増大は小さい. このため, Fig. 3 左のように直列計算部分を間に入れることで並列計算部分を二つに分割するよりも, Fig. 3 右のように多少の重複を含むものの並列計算部分を大きな一つのかたまりにする方が, 並列計算させた際の計算時間に対しては効率的である.

3. 実時間動的制御の実現

3.1 マルチスレッド・プログラミングの特徴

複数のCPUを用いて並列計算を行う場合には様々なシステム構成が考えられる. 小嶺ら[16]は, 並列計算を実現するシステムを1台のPCとベクトル演算・割り込み処理に優れた特殊な計算機で構成することを提案している. PCなど一般の目的で使用される計算機は, 割り込みや入出力などマニピュレータの制御に必要な処理が不得手であることが多く, 専用の計算機を用いることでそれらの処理に伴うオーバーヘッドを減らすことができる. しかし, このような方法の欠点は汎用性がないことである. あるシステムで採用した計算アーキテクチャは, 他のシステムでは動作しない. 場合によってはその計算機専用の言語でプログラミングしなければならないことがある.

そこで, 計算アーキテクチャや制御プログラムの汎用性を高

Table 1 Computation time for each step

step	time	remarks
(1) forward kinematics	0.92 [msec]	
(2) inverse dynamics	2.58 [msec]	0.43 [msec] × 6
(3) torque transform	0.19 [msec]	
subtotal (1)	3.69 [msec]	for dynamic control
(4) read counter	0.60 [msec]	6 channels
(5) output torque	0.30 [msec]	6 channels
subtotal (2)	0.90 [msec]	for I/O
total	4.59 [msec]	

めるために、一般に広く用いられている計算機と並列処理アーキテクチャを利用することを考える。

Windows95, Windows NT, Solaris など、最近の新しい OS のほとんどは“マルチスレッド” [20] と呼ばれる並列処理法をサポートしている。マルチスレッドとは、一つのプロセスの中に実行経路(スレッド)を複数生成してそれらを並行して実行する並列処理法で、もともとはユーザインタフェースと計算を同時に処理するアプリケーションの開発を容易にするために生み出されたものである。

並列計算法としてマルチスレッドを利用することの利点としては次のようなものが挙げられる。

- 並列処理に必要な様々な手続き(同期, 共有メモリ, 変数のロックなど)が簡単に利用できる。
 - 開発したプログラムは、同じ OS であれば CPU 数によらず(計算速度は変化するが)どの計算機でも実行できる。
- 以上のことから、本研究ではマルチスレッド・プログラミングにより並列計算を実現し、計算時間の短縮を図ることとする。

3.2 アルゴリズムの時間解析

2章の動的制御アルゴリズムを単一のスレッドでインプリメントし、計算時間を計測した結果を Table 1 に示す。なお、使用した CPU は Pentium Pro 200 [MHz], OS は Windows NT である。また Windows NT 上でデバイスの入出力を行うため、デバイスドライバ開発ツール WinRT を利用している。入出力に関連する(4.5)については、デバイスのオープン・クローズに時間がかかるため全チャンネルの読み書きをまとめてを行うこととし、1チャンネルあたりの処理時間は評価しなかった。

すべての処理を単一のスレッドで実行すると1ステップあたりの計算時間は 4.59 [msec], サンプル周波数は約 220 [Hz] となる。計算機自体の性能が高いため [16] と比べると高速に計算されているが、動的制御を有効にするためにはさらなる高速化が求められる。

3.3 マルチスレッド・プログラミングによるインプリメント

マルチスレッドプログラムでは、実行と同時にプライマリスレッド(primary thread)と呼ばれるスレッドが生成され、他のスレッドはプライマリスレッドから生成される。そこでまずプライマリスレッドで他のスレッドの管理とメインの計算を行うこととし、その下に逆動力学計算のスレッドを各脚1個ずつ計6個および順運動学計算のスレッド1個、合わせて7個のスレッドを生成する。

これらのスケジューリングは次のように行う (Fig. 4 参照)。

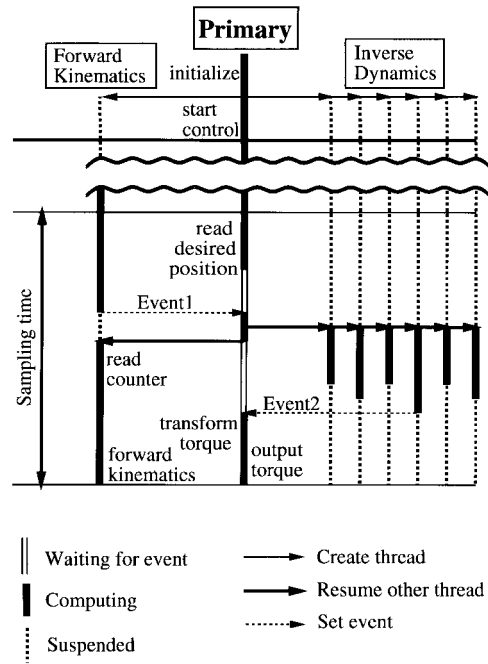


Fig. 4 Scheduling of the threads

ただしマルチスレッドでは CPU 時間の割当を明示的に指定することができないため、同期変数 Event とスレッドの中断 (Suspend) /再開 (Resume) を利用してスレッド間の同期をとる。同期変数とは、あるスレッドでそれをシグナル状態にする (Set Event) ことにより他のスレッドにイベントの発生を通知することのできる変数をいい、計算終了の通知などに利用する。

- (1) プライマリスレッド—初期化後、制御を開始し、動作中は目標位置の読み込みと開リンクから閉リンクへのトルク変換, D/A ボードへの出力を行う。
 - (a) 目標位置を読み、順運動学計算の終了を待つ。
 - (b) 順運動学計算が終了したら逆動力学計算スレッドと順運動学計算スレッドを再開させ、逆動力学計算終了を待つ。
 - (c) 全脚の逆動力学計算が終了するとトルク変換を行い、結果を D/A ボードに出力する。
- (2) 順運動学計算スレッド—カウンタボードから回転角を読み込み、順運動学計算を行う。
 - (a) 順運動学計算の終了後 Event1 をシグナル状態にし、自らを中断する。
 - (b) すぐプライマリスレッドにより再開されるので、次のステップに入る。
- (3) 逆動力学計算スレッド (6 個)—逆動力学計算を行い、開リンクにおける仮想トルクを計算する。
 - (a) 目標位置の読み込みと順運動学計算が終了して逆動力学が計算できる状態になったらプライマリスレッドにより再開される。
 - (b) 1~5 番目に計算を終了したスレッドは自らを中断

Table 2 Comparison of computation time between parallel and serial computations

	with I/O	without I/O
serial	4.59 [msec]	3.69 [msec]
parallel	1.92 [msec]	1.02 [msec]
gain	0.42	0.28

し、最後に計算を終了したスレッドは Event2 をシグナル状態にして自らを中断する。

3.4 並列化後の計算時間

上の方法で並列計算をインプリメントし、Pentium Pro 200 [MHz] CPU を四つ搭載した PC で動的制御計算を行ったところ、計算時間は 1.92 [msec] と大幅に短縮された。これでサンプリングタイム 2 [msec] (サンプリング周波数は約 500 [Hz]) でのリアルタイムの動的制御が可能である。

Table 2 に並列化による計算時間短縮の効果をまとめた。Windows NT ではデバイスを介した入出力に時間がかかるので、入出力にかかわる計算を除いた時間での比較も行っている。四つの CPU を使っているが、実際には常時四つ以上のスレッドが動作しているわけではなく、またスレッドの切替や同期に時間がかかるため、計算時間が必ずしも 4 分の 1 になるわけではない。

4. 動的制御実験

動的制御の有効性を確認するためにいくつかの実験を行った。なお比較のために、エンドプレートの位置の PID 制御による結果を合わせて示す。また 4.4 節以外の実験では摩擦などは考慮していない。

4.1 実験システム

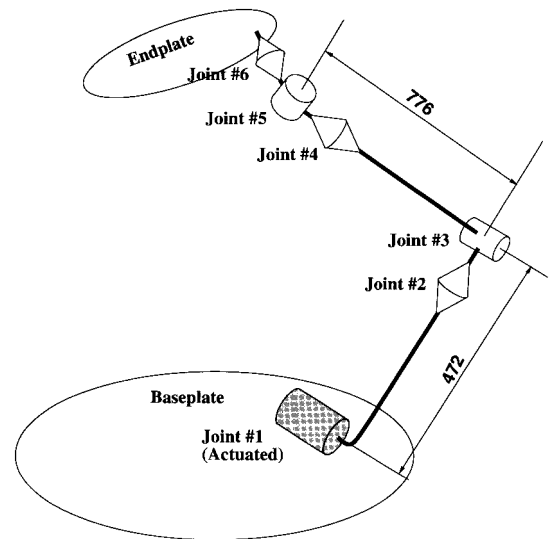
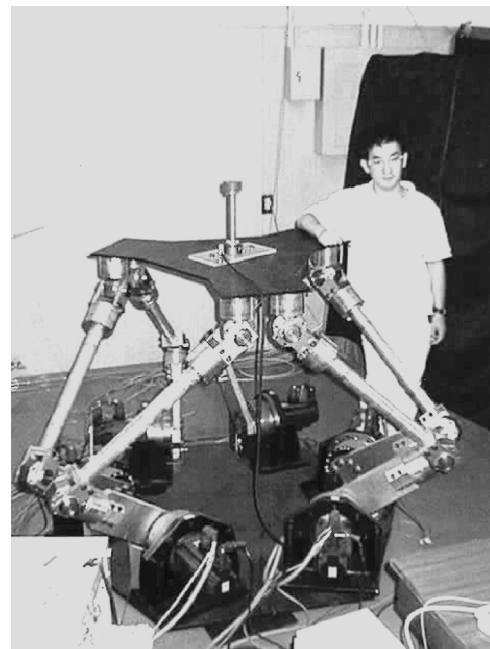
実験に使用した平行マニピュレータの写真を Fig. 6 に示す。この平行マニピュレータは、武田ら [3] が運動伝達指数が最適化されるように設計した機構を、ドライビングシミュレータのプラットフォームとして応用するために仕様変更したものであり、70 [kg] の物体を載せて 2 [G] の加速度を発生する能力を持つ。平行マニピュレータの各脚の関節構成を Fig. 5 に、各リンクの動力学パラメータを Table 3 にそれぞれ示す。

制御用計算機としては Pentium Pro 200 [MHz] CPU を並列計算のために四つ搭載した PC を使用し、制御プログラムは Microsoft Visual C++ で書かれている。PC により計算された制御トルクは、D/A ボードを通してモータのサーボバックに入力される。モータの回転角は、モータに付属したエンコーダのパルスをカウンタボードでカウントすることにより得る。また、リアルタイム性を確保するため、カウンタボードに付属するタイマからの入力で割り込みをかける。以上の入出力を Windows NT 上で処理するため、デバイスドライバ開発用ツール WinRT を使用している。入出力のシステムを Fig. 7 に示す。

4.2 制御則とゲイン

動的制御

分解加速度制御により、先に述べた計算で得られたトルクを指令値としてサーボに入力する。なお、以下の実験では分解加速度制御におけるゲイン行列を対角行列とし、値は各軸共通で $K_P = 2.00 \times 10^3$, $K_D = 2.85 \times 10^2$ とした。

**Fig. 5** Leg of the Parallel Manipulator**Fig. 6** Parallel Manipulator**Table 3** Dynamic parameters of the leg

Link	Mass [kg]	Inertia (kg · m ²)		
		I_{xx}	I_{yy}	I_{zz}
1	6.90	0.153	0.146	0.0123
2	3.79	0.00938	0.00398	0.0114
3	8.35	0.526	0.525	0.0101
4	5.10	0.0197	0.00480	0.0171
5	3.79	0.0152	0.00788	0.0133
Endplate	30.0	1.14	1.14	2.28

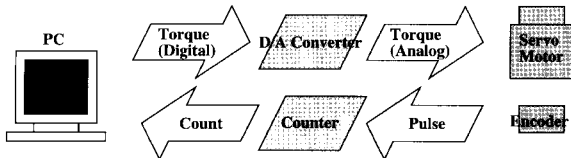


Fig. 7 I/O system of the parallel manipulator

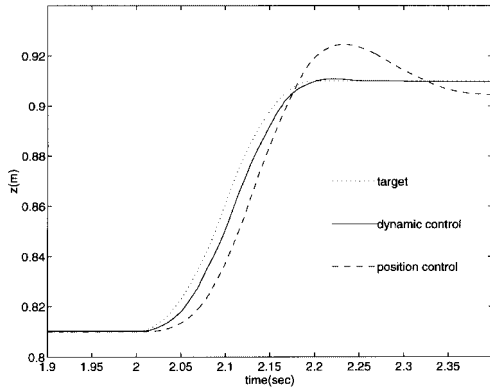


Fig. 8 Endplate position in upward motion

位置制御

まずエンドプレートに加える一般化力を次のように定める .

$$f = K_P(x_{Ed} - x_E) + K_D \frac{d}{dt}(x_{Ed} - x_E) + K_I \int (x_{Ed} - x_E) dt \quad (6)$$

ここで x_E, x_{Ed} はそれぞれエンドプレートの現在位置・姿勢, および目標位置・姿勢である . 次に, エンドプレートの位置に対するモータ角のヤコビアン J を用いて, 次式のように f をモータの出力に変換する .

$$\tau = J^T f \quad (7)$$

エンドプレートの現在位置・姿勢 x_E を求めるために順運動学計算が必要になるほか, このシステムでは入出力に時間がかかるため, サンプリングタイムは動的制御と同様 2 [msec] . となった . また, 以下の実験ではゲインを各軸共通で $K_P = 3.80 \times 10^5$, $K_D = 1.75 \times 10^4$, $K_I = 1.85 \times 10^4$ とした .

4.3 軌道追従性

直線運動

上方へ $10 [m/s^2]$ で 0.1 秒間加速・減速する目標軌道を与えた場合の結果が Fig. 8 である . 動的制御により時間遅れ, オーバーシュート, 重力による定常偏差が改善されていることが分かる . 動的制御における偏差は摩擦の影響と考えられる .

曲線運動

水平面内の曲線運動の結果を Fig. 9 に示す . 始点 (右下) から終点 (左上) の所要時間は 0.4 秒である . 動的制御により遠心力が補償され, 軌道追従性が向上している . また終点付近のオーバーシュートも動的制御では見られない .

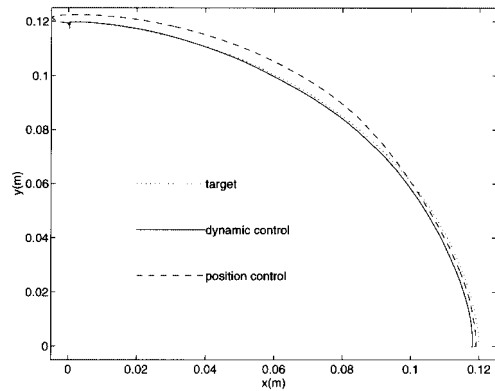


Fig. 9 Endplate position in curve in xy plane

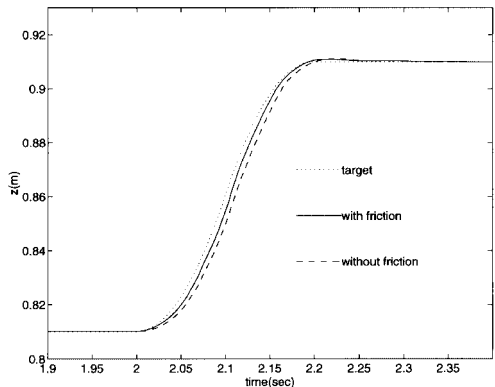


Fig. 10 Endplate position in upward motion when friction is considered

4.4 摩擦補償

逆動力学で計算されたトルクに入力関節の静止摩擦 60 [Nm], 粘性摩擦 100 [Nm/(rad/s)] を加え, 4.3 節の直線運動を行わせた結果を Fig. 10 に示す . 軌道追従性は摩擦を考慮しないときに比べてさらに向上している . しかし, ここでは考慮しなかった受動関節の摩擦などにより依然として誤差が残っている .

5. 加速度フィードバック

5.1 エンドプレート上の加速度

本マニピュレータは, 加速度感覚提示機としてシミュレータへの応用を考えて製作されている . したがって一般の作業用マニピュレータとは異なり, 位置決め精度や軌道追従性よりもエンドプレートで発生する加速度の精度が重要である .

z 方向に $5 [m/s^2]$, $-5 [m/s^2]$ の目標加速度 (点線) を与えたときに, エンドプレート上で発生した加速度を測定した結果を Fig. 11 に示す . 動的制御により時間遅れや精度が改善されているが, 依然として時間遅れが見られるほか, 構造の弾性によると考えられる振動が発生していることが分かる .

時間遅れは 4.4 節の結果から摩擦が主な原因と考えられる . しかし, すでにマニピュレータが組み上がっている状態では, 現実問題として摩擦を正確に測定することは困難である . また, 振動は現在検出しているモータ角には反映されないため, 現在

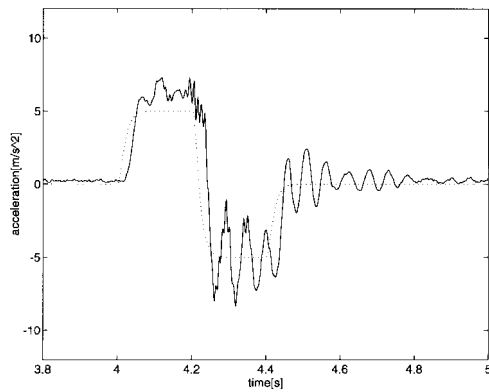


Fig. 11 Acceleration measured on the endplate in vertical motion

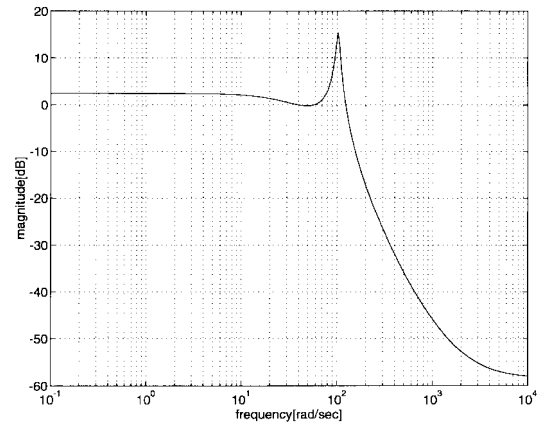


Fig. 12 Bode plot of vertical direction system

の制御法では除去することはできない。

そこで、エンドプレート上の加速度をフィードバックし、発生する加速度の精度を高めることを考える。

5.2 同定と補償器の統合化設計

統合化設計とは、同定実験の結果を用いて補償器の再設計を行う操作を繰り返し、希望の閉ループ系の実現を目指す設計法である。この設計法の重要な点は、実験結果をいかに補償器再設計に活用するかという点にあり、今回は目標加速度から実際の値までの閉ループ系全体を一つのシステムとみなして同定し、これを用いて H_∞ 制御則によって補償器の設計を行うという方法 [21] をとる。利点としては次のようなものが挙げられる。

- 解析的には解けない振動の除去が可能
- 開ループ同定なのでシステム同定が比較的容易
- ロバスト安定化、低感度化のために適用される H_∞ 制御則は周波数領域での評価を行うため振動除去に向く

5.3 設計手順

制御系の設計は以下の手順で行われる。

(1) 同定

補償器設計のための準備として、動的制御系により安定化された \ddot{x}_d から \ddot{x} までの実際のシステムを G_m として同定する。同定のための加速度入力として M 系列信号を用い、Box-Jenkins モデル [22] により同定する。ここで、動的制御系により 6 軸相互間の干渉は取り除かれていると考え、各軸をおのおの 1 入力 1 出力系として個別に同定する。同定の結果得られた鉛直方向のボード線図を Fig. 12 に示す。

(2) 補償器の設計

次式の評価関数 J に対し H_∞ 制御則を適用して補償器 K を設計する。

$$J = \left\| \begin{array}{c} W_T K (I + G_m K)^{-1} G_m \\ W_S G_m (I + K G_m)^{-1} \end{array} \right\|_\infty \quad (8)$$

ここでモデル化誤差としては、 $G = G_m(1 + \Delta)$ のような乗法的誤差 Δ を考え、 W_T によってこれに対するロバスト安定化、 W_S によって感度低減化をめざしている。

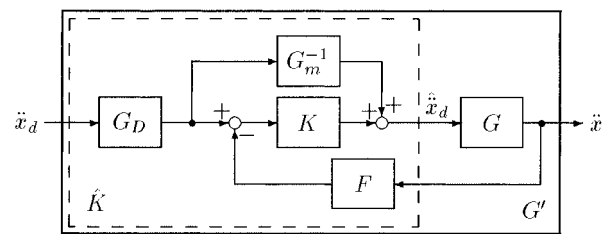


Fig. 13 Acceleration feedback system

ただし、 W_T, W_S はシステム同定時の実験結果よりスペクトル解析によって得る。

(3) 補償器の組み込み

補償器 K を Fig. 13 の 2 自由度制御系 [23] に組み込む。ここで G_D は希望する応答、 F はセンサのノイズを除去するためのローパスフィルタであり、それぞれ次のように設定した。

$$G_D = \frac{100^2}{(s+100)^2}, \quad F = \frac{1}{s+300} \quad (9)$$

最終的に得られるコントローラは、目標加速度 \ddot{x}_d とセンサの出力 \ddot{x} を入力とし、動的制御系への参照加速度入力 $\hat{\ddot{x}}_d$ を出力とする \hat{K} (Fig. 13 の点線部) となる。

(4) 繰り返し

補償器が更新されることで閉ループ特性が変化することが予想される。そこで (3) の制御系を新たな制御対象 G' (Fig. 13 の太線) とし (1) に戻って同定・設計・組み込みの手順を繰り返す。

5.4 実験結果

動的制御系に加速度フィードバックコントローラを追加した制御系に対し、Fig. 11 と同じ目標加速度を与えたときの応答を Fig. 14 に示す。加速度フィードバックを行うことにより、立ち上がりの遅れは約 4 分の 1、振動の振幅は約 3 分の 1 に減少した。

動的制御系の固有振動数に近い 16 [Hz] の正弦波状の目標加速度入力を与えた場合の応答を Fig. 15 (加速度フィードバックなし)、Fig. 16 (加速度フィードバックあり) に示す。コン

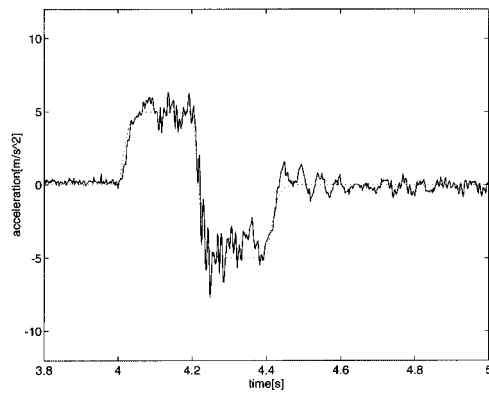


Fig. 14 Measured acceleration in vertical motion with feedback

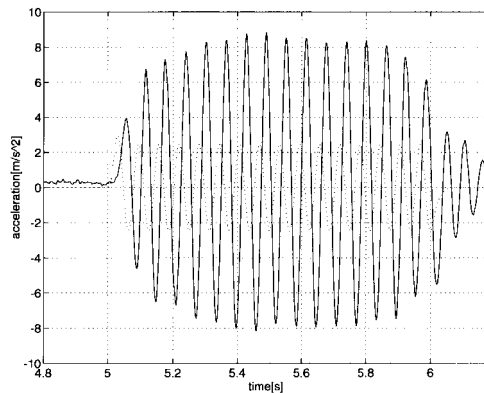


Fig. 15 Response to sine-wave input without acceleration feedback

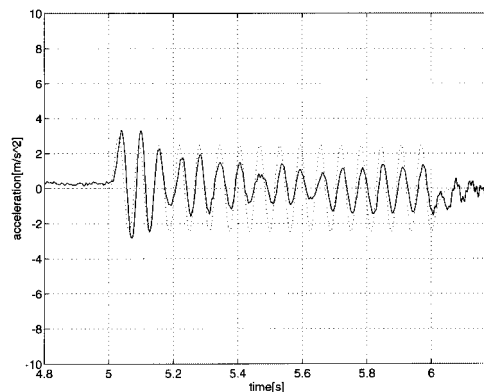


Fig. 16 Response to sine-wave input with acceleration feedback

トローラを追加したことにより周波数特性が変化し、極が取り除かれたことが分かる。

6. ま と め

本研究ではパラレルメカニズムの動的制御と加速度フィードバック制御に関して以下の結論を得た。

- 並列性が高く、並列計算による計算時間短縮の効果の大きい動的制御アルゴリズムを構築した。

- マルチスレッド・プログラミングにより汎用性のある動的制御プログラムを開発し、2 [msec] のサンプリングタイムで実時間での厳密な動的制御を実現した。
- 実験により動的制御が位置追従性の向上に効果があることを示した。しかし、高速運動においては入力角では観測できない大きな振動が見られた。
- エンドプレート上の加速度をフィードバックするコントローラを同定実験結果をもとに設計し、動的制御系の外に追加した。この結果、高速運動においてもエンドプレート上に滑らかで正確な加速度を実現することができることを確認した。

なお、本研究は文部省科学研究費 (B) (2) 06555065 (財) 国際ロボット FA 技術開発センター (現・製造科学技術センター) の高速パラレルメカニズム開発委員会 (委員長: 新井健生, 1995–1997), および中山隼雄科学技術文化財団の援助を受けて行われたものである。

参 考 文 献

- [1] 新井健生ほか編: 特集パラレルメカニズム, 日本ロボット学会誌, vol.10, no.6, pp.695–769, 1992.
- [2] D. Stewart: “A Platform with Six Degrees of Freedom,” Proceedings of the Institution of Mechanical Engineers, vol.180–1, no.15, pp.371–386, 1965–1966.
- [3] 武田, 舟橋, 市丸: “運動伝達性を考慮した 6 自由度パラレルマニピュレータ機構の開発”, 第 11 回日本ロボット学会学術講演会予稿集, pp.853–856, 1993.
- [4] 内山, 飯村, 多羅尾, ビエロ, 外山: “6 自由度高速パラレルロボット HEXA の開発”, 日本ロボット学会誌, vol.12, no.3, pp.451–458, 1994.
- [5] J.Y.S. Luh, M.W. Walker and R.P.C. Paul: “Resolved Acceleration Control of Mechanical Manipulators,” IEEE Transactions on Automatic Control, vol.25, no.3, pp.468–474, 1980.
- [6] M. Raghavan: “The Stewart Platform of General Geometry Has 40 Configurations,” Journal of Mechanical Design, Transactions of the ASME, vol.115, pp.277–282, 1993.
- [7] B. Mourrain: “The 40 “Generic” Positions of a Parallel Robot,” Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation, pp.173–182, 1993.
- [8] 多羅尾, 内山, 益川, 佐渡友: “6 自由度高速パラレルロボット HEXA”, 日本ロボット学会パラレルメカニズム研究専門委員会報告書, pp.74–84, 1993.
- [9] 中村, 山根: “高精度高速パラレルメカニズムの開発—第 3 報: 動的制御系の設計”, 第 14 回日本ロボット学会学術講演会予稿集, pp.785–786, 1996.
- [10] J.F. Kleinfinger and W. Khalil: “Dynamic Modeling of Closed-Loop Robots,” Proceedings of the 16th International Symposium on Industrial Robots, pp.401–412, 1986.
- [11] Y. Nakamura and M. Ghodoussi: “Dynamics Computation of Closed-Link Robot Mechanisms with Nonredundant and Redundant Actuators,” IEEE Transactions on Robotics and Automation, vol.5, no.3, pp.294–302, 1989.
- [12] M. Ait-Ahmed and M. Renaud: “Dynamic Modeling of Closed-Chain Mechanisms and Its Application for a 6 D.O.F. Actuated Manipulator,” Proceedings of the 1st International Conference in Electronics and Automatic Control, pp.203–209, 1992.
- [13] 中村仁彦: “パラレルメカニズムの動力学”, 日本ロボット学会誌, vol.10, no.6, pp.709–714, 1992.
- [14] 内山, 佐渡友, 益川: “パラレルロボット HEXA の動的制御実験”, 日本ロボット学会誌, vol.14, no.2, pp.297–304, 1996.
- [15] J.-P. Merlet: Les robots parallèles. Hermès, Paris, 1997.

- [16] 小嶺, 名嶋, 中村, 吉本: “動力学補償を用いたパラレルメカニズムの制御”, 日本機械学会ロボティクス・メカトロニクス講演会'96 予稿集, pp.1059-1062, 1996.
- [17] A. Fijany, I. Sharf and G.M.T. D'Eleuterio: “Parallel $O(\log N)$ Algorithms for Computation of Manipulator Forward Dynamics,” IEEE Transactions on Robotics and Automation, vol.11, no.3, pp.389-400, 1995.
- [18] J.J. Craig: Introduction to Robotics: Mechanics and Control. Addison-Wesley, 1986.
- [19] 中村, 横小路, 花房, 吉川: “ロボットマニピュレータの運動学と動力学の統合化計算”, 計測自動制御学会論文集, vol.23, no.5, pp.71-78, 1987.
- [20] B. Lewis and D.J. Berg: Threads Primer. Sun Soft Inc., 1996.
- [21] 杉江, 岡田: “閉ループ同定を利用した補償器の繰り返し設計法”, 計測自動制御学会論文集, vol.32, no.3, pp.313-319, 1996.
- [22] 足立修一: ユーザのためのシステム同定理論. 計測自動制御学会, 1993.
- [23] 前田, 杉江: アドバンスト制御のためのシステム制御理論. 朝倉書店, 1990.



中村仁彦 (Yoshihiko Nakamura)

1954年9月22日生。1977年京都大学工学部精密工学科卒業。1982年同大学大学院博士課程退学。同年より1987年まで同大学助手。1987年より1991年3月までカリフォルニア大学サンタバーバラ校助教授, 準教授。同年4月より東京大学工学部機械情報工学科に勤務, 現在教授。工学博士。運動学的冗長性, 閉リンク機構の駆動冗長性, 複数のロボット機構の協調, 宇宙ロボットの制御, 非ホロノミックな拘束を受ける機械の運動制御, 外科手術用ロボットなどに興味をもつ。計測自動制御学会, システム制御情報学会, 日本機械学会, 日本コンピュータ外科学会, IEEE, ASMEなどの会員。(日本ロボット学会正会員)



岡田昌史 (Masafumi Okada)

1969年3月21日生。1992年3月京都大学工学部精密工学科卒業。1994年3月同大学大学院修士課程(応用システム科学専攻)修了。1996年9月同大学大学院博士課程修了, 博士(工学)。1996年10月日本学術振興会特別研究員(PD)。1997年2月東京大学大学院工学系研究科リサーチ・アソシエイト(日本学術振興会未来開拓学術研究推進事業研究員)となり現在に至る。ヒューマノイドロボット, ロバスト制御の研究に従事。システム制御情報学会, 計測自動制御学会, IEEEの会員。(日本ロボット学会正会員)



山根 克 (Katsu Yamane)

1974年8月1日生。1997年東京大学工学部機械情報工学科卒業, 1999年同大学大学院工学系研究科機械情報工学専攻修士課程修了, 同博士後期課程進学。マルチボディシステムの運動計算, ヒューマンフィギュアの運動計算, 人間の運動データやキーフレームに基づくヒューマノイド・CGキャラクターの運動生成, 力学的整合性を考慮した多様な運動の合成などに興味をもつ。(日本ロボット学会学生会員)



小嶺徳晃 (Noriaki Komine)

1971年9月23日生。1994年東京大学工学部機械情報工学科卒業, 1996年同大学大学院工学系研究科機械情報工学専攻修士課程修了。同年, 島津製作所入社, 材料試験機的设计に従事。