

—修士学位論文—

機能性に基づいた構造可変型ロボットの
位置と力の協調制御システムに関する研究

A Study on a Function-based Position and Force
Cooperative Control System
for Reconfigurable Robots



平成21年度

三重大学大学院工学研究科
博士前期課程 電気電子工学専攻
上村 章仁

目次

第1章 緒言	1
1.1 研究背景と目的	1
1.2 論文の構成	3
第2章 構造可変型ロボットの構想	4
2.1 構造可変型ロボットの基礎構造	4
2.2 構造可変型ロボットに適した制御システムの概要	5
2.3 非接触伝送	6
第3章 中央制御部と分散知能の制御システム	9
3.1 多自由度ロボットアームの動力学 [15]	9
3.1.1 動力学	9
3.1.2 逆動力学	10
3.2 中央制御部	12
3.3 分散知能	12
3.3.1 外乱オブザーバ [9]	13
3.3.2 係数図法 [10, 11]	15
3.4 ロボットの運動学	17
3.4.1 一般的な運動学 [16]	17
3.4.2 構造可変型ロボットに適した逆運動学 [3, 4, 5, 12]	21
3.4.3 作業空間オブザーバ [9]	22
第4章 機能性に基づく分散制御 [6, 7]	24
4.1 機能性の概念	24
4.2 機能空間への座標変換	27
4.3 構造可変型ロボットに適した機能性に基づく分散制御	29

第5章 機能性に基づく構造可変型ロボットの制御システム	30
5.1 提案する制御システムにおける座標変換	30
5.2 提案する機能性に基づく制御システム	31
5.2.1 インピーダンスモデル	33
5.3 中央制御部の制御器設計	34
5.4 構造可変型ロボットに適した姿勢の制御方法	36
5.4.1 ヤコビ行列を用いた冗長指令の生成方法 [15]	36
5.4.2 簡便な冗長指令生成方法 [4]	37
5.4.3 勾配法の比例ゲインの一決定法	38
5.4.4 機能性に基づく制御システムの拡張	39
第6章 カセンサレス制御システムの構築	41
6.1 トルク反力の推定	41
6.1.1 トルク反力推定オブザーバ [8]	41
6.1.2 重力トルク, 摩擦トルク, 減速機の効率	42
6.1.3 分散知能の拡張	47
6.1.4 トルク反力推定の実験	47
6.2 座標変換の問題	51
6.3 計算の省力化の提案	52
6.4 中央制御部の拡張	53
6.5 機能性に基づく制御システムのセンサレス化	54
第7章 シミュレーションと実験	55
7.1 実験1 (構造の組み替えとタスク変化の伴う協調作業)	56
7.2 実験2 (冗長指令を付加した協調作業)	62
7.3 実験3 (カセンサレス制御システムのシミュレーション)	65
7.4 実験4 (カセンサレス制御システムの実機実験)	68
7.5 実験5 (協調把持のカセンサレス制御)	70
第8章 まとめと今後の課題	72
8.1 まとめ	72
8.2 今後の課題について	73
参考文献	75

謝辞	77
論文目録	78

第1章 緒言

1.1 研究背景と目的

従来、工場では低コスト化を目的とし、同じ製品を多量に生産する「少品種多量生産」が行われてきた。そのため、工場では作業効率の向上のために作業を分担し、それぞれの作業工程で専用のロボットが決められた作業のみを行っている現状がある。しかし、近年社会ではあらゆる面での多様化が進んでいる。それに伴い、低コスト化を目的とした「少品種多量生産」では多様化するニーズに柔軟に対応していくことが困難になりつつあり、「多品種少量生産」または「変種変量生産」への移行が余儀なくされている。したがって、これからの工場で利用されるロボットに対して、多様な作業が実現できる柔軟性を求められることになるが、決められた作業のみを行っている従来のロボットではそれらに対応できなくなりつつある。

このような問題の一解決方法として、構造の組み替えが容易なロボットシステム[1, 2]が注目されている。容易な構造の組み替えにより、様々な要求に応えることができ、かつ状況変化に対しても柔軟な対応が可能となる。さらに、故障に対しても故障箇所のみを取り替える事で迅速な対処ができるという利点も有する。

これらを背景に、我々は構造の組み替えを前提とした構造可変型ロボットシステムを提案している。我々の提案する構造可変型ロボットシステムは、ロボットを各関節ごとに分割したモジュールをあらかじめ構成しておき、それらを組み替えることで作業の目的や環境に対応する。また、モジュールはそれぞれが、機構、機能が異なるものとし、作業や環境に応じて必要なモジュールを選択し、多様なロボットを構成可能にしていくことを目指している。他研究でも、構造の組み替えを前提とするロボットシステムとして、複数のモジュールから構成されるモジュール型ロボットシステム[1]や、セル構造化ロボット[2]などが提案されてはいるが、これらのシステムはすべてが均一なパーツから構成されている点が我々の提案と異なる。そのため、提案する構造可変型ロボットシステムは、より多様な組み合わせのロボットが構成可能であることに期待できるが、その分、構造の変更による制御システムの対応が複雑化する。したがって、このようなシステムを実現するためには、構造の組み替えの対応を単純化し、

柔軟に対応可能なシステムを構成する必要がある。現状では、ロボットの構造を組み替えるときに生じる以下の問題点に対応する制御システムを構成することが望ましいと考えている。

問題 1 … 応答特性の変化

問題 2 … 運動学の変更

問題 3 … 多様な作業の実現のための位置、力の制御の変更

これらを踏まえて、先行研究では、中央制御部と分散知能を配置する分散制御システムが提案されている [3, 4]。この分散制御システムは、運動学を解き、各関節への指令を生成する中央制御部とその指令にしたがって各関節を制御する複数の分散知能を構成することで計算の集中を回避し、構造の組み替えに対する対応を単純化している。分散知能では、各関節の応答特性が構造に依存しないように制御器パラメータを自律的に更新することで問題 1 を解決している。そして、中央制御部で使用する逆運動学では、一般的な逆運動学として必要となるヤコビ行列の逆行列を使用しない逆運動学の解法である仮想速度伝播アルゴリズム (VVTA) や仮想位置伝播アルゴリズム (VPTA) が考案されている [3, 4, 5]。これらの運動学は、構造の組み替えによる計算式の変更が容易なため、問題 2 を解決することができる。先行研究では、これらを用いた分散制御システムによって、構造可変型ロボット単体の手先位置の制御システム、そして、力センサの使用を前提とした力制御システムが考案されており、シミュレーションにおいて有効性が確認されている。

本研究では先行研究で提案された制御システムをさらに応用し、構造可変型ロボットを用いて、多様な作業を達成することを目指す。そこで、複数のロボットが一つの作業を共有する協調作業に着目し、それを可能とする制御アルゴリズムを提案する。ロボットを複数用いることで単体では不可能な作業を実現でき、構造可変型ロボットを用いる協調作業となれば、さらに多様かつ汎用性の高い作業の実現が期待できる。

そこで、この目的を達成するために、作業を構成する要素の最小単位を考え、その組み合わせで作業を表現する「機能性」という概念に基づく分散制御システム [6, 7] を導入する。そのシステムでは、制御システムにおける制御器に対しても最小単位となるものを設計し、それらの組み合わせにより、要求される作業を達成する指令を生成する。すなわち、仕様の変化に応じた位置や力制御器の組み替えが、制御システム内の制御器でも可能となるため、状況に応じて機構 (モジュール) が組み替えられるロボッ

トに適した様々な指令を生成できる。したがって、このような制御システムを構成することにより、問題 3 の解決が期待できる。

さらに、本研究では、力制御システムのセンサレス化を検討する。この制御システムでは、トルク反力推定オブザーバ [8] を利用し、各分散知能でアクチュエータにかかるトルク反力を推定する。そして、構造可変型ロボットに適した組み替えに対する対応が容易な運動学による計算方法を提案することで、各関節のトルク反力推定値から容易に手先の反力を推定し、力の座標変換で生じる運動学の問題も回避する。

最後に、提案するシステムが上記の問題 1, 2, 3 を解決する制御システムであることを、構造可変型ロボットを模擬した実機システムを用いた実験によって確認する。また、それらの制御が力センサを用いずに可能なことを実機実験により確認する。

1.2 論文の構成

本論文の構成を以下に述べる。まず第 2 章において本研究で対象とする構造可変型ロボットシステムの概要について詳しく説明する。この章では、構造の組み替えを行うために重要な要素となる、分散知能、中央制御部を搭載した構造可変型ロボットを紹介し、それぞれの役割を簡単に述べる。そして、第 3 章において先行研究で提案されている分散知能と中央制御部の制御システムについて、構造の変更に応じて分散知能が自律的に制御器パラメータを更新する方法や、構造可変型ロボットに適した逆運動学である仮想速度伝播アルゴリズムについて述べる。第 4 章において本研究で用いる「機能性に基づく分散制御システム」について説明した後、そのシステムを構造可変型ロボットに適した形で導入した制御システムを第 5 章で提案する。第 6 章では提案する制御システムの力センサレス化を検討し、簡単な計算で可能となる力の座標変換方法についても述べる。第 7 章で提案する制御システムの有効性を、構造可変型ロボットを模擬した実機システムによる実験によって確認する。最後に第 8 章で本研究におけるまとめと今後の課題を述べる。

第2章 構造可変型ロボットの構想

構造の組み替えに柔軟に対応可能なロボットの機構や制御システムを構成するため、我々は、電力・情報の非接触伝送システム、分散知能、中央制御部を搭載することで組み替えに柔軟に対応するロボットシステムの実現を目指している。この章では、我々が提案する構造の組み替えに対する柔軟な対応方法を簡単に説明する。

2.1 構造可変型ロボットの基礎構造

提案する構造可変型ロボットの構想図を図 2.1 に示す。

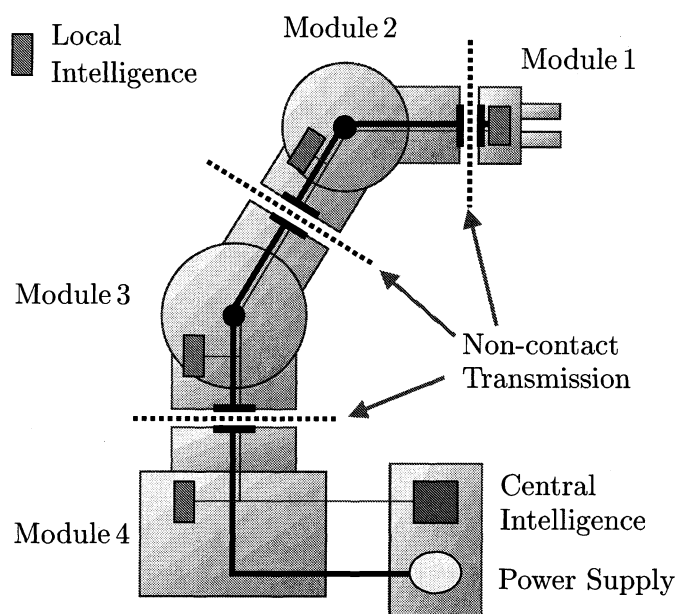


図 2.1: 構造可変型ロボットの基礎構造

図 2.1 に示されるように、構造可変型ロボットは中央制御部 (Central Intelligence) と組み替え可能な複数のモジュールから成り立ち、各モジュールには分散知能 (Local Intelligence) が配置されている。

各モジュールは電磁誘導による非接触伝送によって電力と情報が伝送される。すなわち、各モジュールが物理的な配線でつながれていない機構となるため、構造の組み替えが容易となる。そのため、構造の組み替えにより、様々な作業や環境への対応が可能となり、ロボットの故障時に対しても、故障箇所を組み替えることで迅速な対処が期待できる。また、非接触伝送は、配線の接続による磨耗や動作範囲の制限といった問題も解決する。

2.2 構造可変型ロボットに適した制御システムの概要

組み替えに対して柔軟に対応するため、構造可変型ロボットシステムでは、演算能力のある中央制御部と複数の分散知能を配置している。その関係を図2.2に示し、以下にそれぞれの役割をまとめる。

分散知能

- 1… モジュールの物理パラメータを保持
- 2… モジュールのアクチュエータの速度制御
- 3… 各関節のノミナル慣性パラメータに応じた制御器パラメータの自律更新

中央制御部

- 1… 構成されたロボットの各関節のノミナル慣性パラメータの計算
- 2… 順運動学、逆運動学を解き、各分散知能への指令を生成
- 3… 作業に応じた位置、力制御器の選択

新たにロボットが組み替えられたとき、分散知能と中央制御部がどのように対応するかを説明する。まず、各モジュールに配置された分散知能が物理パラメータを中央制御部へ送信する。中央制御部はそのパラメータから、どのようなロボットが構成されているのかを把握し、そのロボットの運動学を計算する。また、ロボットの各関節のノミナル慣性パラメータを計算し、各分散知能へ送信する。各分散知能はそのパラメータを用いて、あらかじめ設計されていたアクチュエータの速度制御システムの制御器パラメータを更新する。以上がロボットが組み替えられた場合における前準備で必要

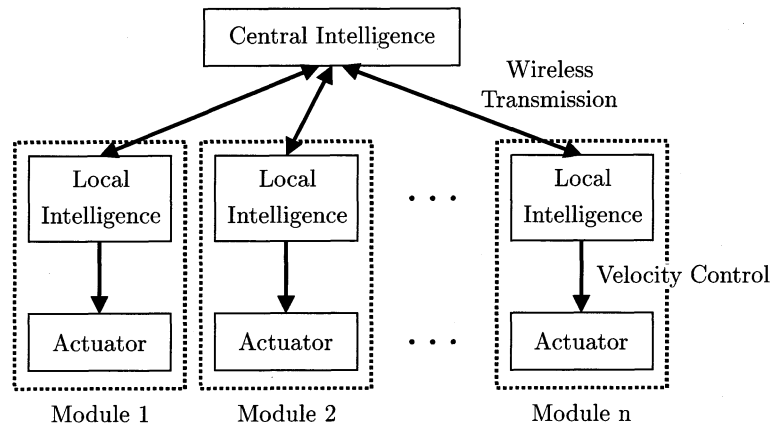


図 2.2: 構造可変型ロボットのシステム図

な対応である。このように、各モジュールごとに知能を配置することにより、人が行わなければならない再設定などの操作を減らすことができ、構造の変更への対応をより簡単化できる。

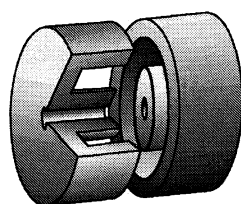
なお、分散知能と中央制御部の制御システムの詳細については、第3章で述べる。

2.3 非接触伝送

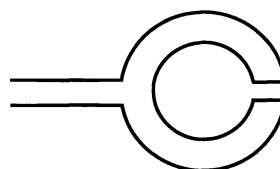
構造可変型ロボットにおける情報と電力を非接触伝送について簡単に説明する。電力は図 2.3 (a) に示される非接触トランスを用いて、電磁結合させることで伝送する。電磁誘導を行う時に必要な交流を作り出すため、図 2.4 に示されるようなトランスとインバータとコンバータを用いる。結合部は非接触トランスにより物理的な接続はなく、電氣的に接続されているため、電力を1次側(中央制御部)から2次側(各分散知能)に伝送することが可能となる。また、電力の非接触伝送はモジュール間の微小空間を介して電力を供給するため、漏れが生じ、通常の配線より効率が悪化することが考えられるが、インバータのスイッチング周波数における共振回路の適用などにより伝送利得の改善が可能であることがわかっている。

情報伝送では、図 2.3 (b) のように銅線を配置したアンテナを用い、アンテナに電流を流したときに作り出される磁界によって伝送を行う。この時、電力伝送からの電磁ノイズは、このアンテナの形状より、内側と外側の銅線での電流の向きが逆になるため、全体でキャンセルすることができる。

これらの接続は図 2.5 のように行う。このように、各モジュールは物理的に接続せずに、電力と情報が伝送されることにより、構造の組み替えが容易なものとなる。



(a) 非接触トランス



(b) 情報伝送用アンテナ

図 2.3: 非接触伝送

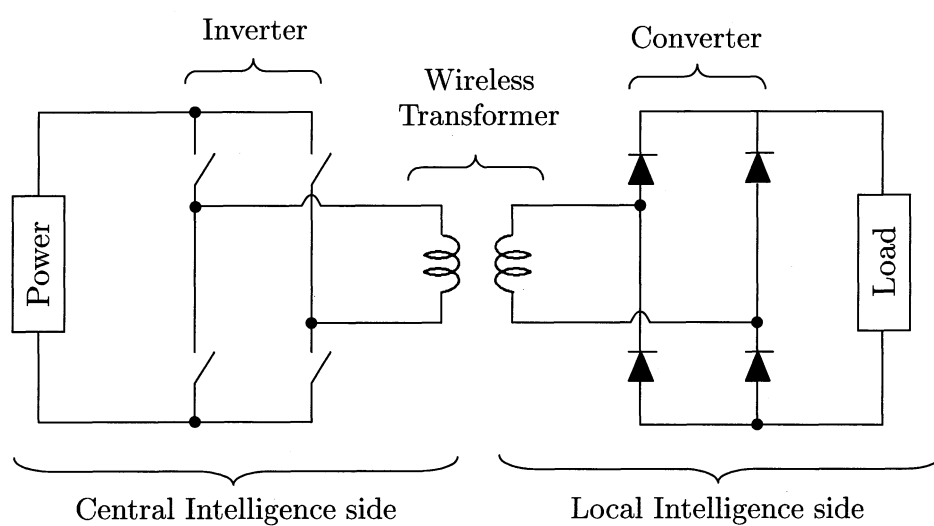


図 2.4: 電力の非接触伝送

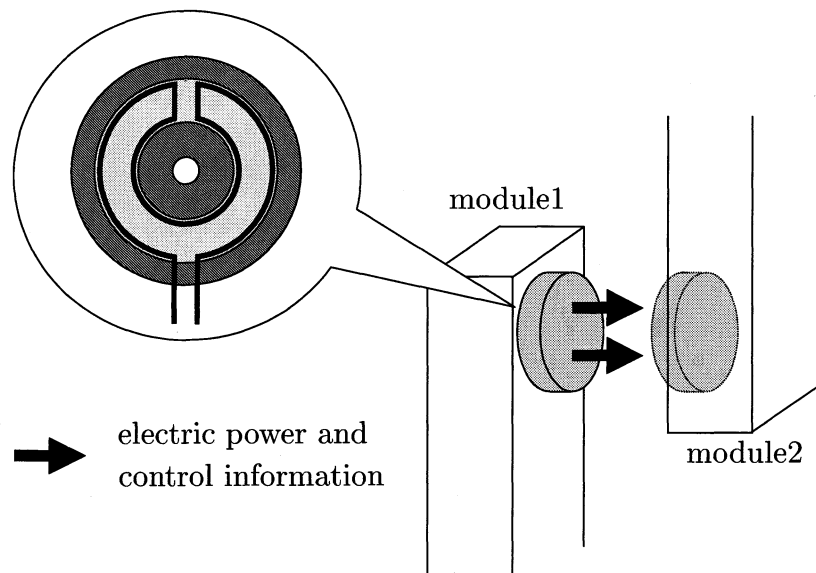


図 2.5: モジュール接続イメージ図

第3章 中央制御部と分散知能の制御システム

前章では分散知能と中央制御部を配置する構造可変型ロボットシステムの概要について述べた。本章ではそれらに対して提案されている制御システムに焦点をあて、その詳細を説明する。

なお、本章以降、手先の位置 \boldsymbol{x} ，速度 $\dot{\boldsymbol{x}}$ ，加速度 $\ddot{\boldsymbol{x}}$ ，力 \boldsymbol{f} を直交座標系で定義する空間を作業空間，各アクチュエータの角度 $\boldsymbol{\theta}$ ，角速度 $\dot{\boldsymbol{\theta}}$ ，角加速度 $\ddot{\boldsymbol{\theta}}$ ，トルク $\boldsymbol{\tau}$ を回転座標系で定義する空間を関節空間と呼ぶものとする。

パラメータの上添え字 \bullet^{cmd} は指令値， \bullet^{ref} は指令値と応答を制御器に通して計算した参照入力， \bullet^{res} は応答， \bullet^{dis} は外乱を表すものとする。

3.1 多自由度ロボットアームの動力学 [15]

ロボットを物理的に解釈するために、「動力学」について説明する。ここでロボットの「動力学」とは、力の次元までの関係を考慮したものであり、関節トルク $\boldsymbol{\tau}$ から物体の運動を求めるものである。それに対して、3.4節で後述するロボットの「運動学」は、ロボットの幾何学構造から計算される関節空間と手先空間の関係を表したものであり、力を考慮しないものである。

3.1.1 動力学

まず、一般的なロボットアームの動力学モデルについて説明する。一般に、ロボットアームはラグランジュ法やニュートンオイラー法の誘導から、(3.1) 式のような運動方程式で表すことができる。

$$\boldsymbol{J}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{D}\dot{\boldsymbol{\theta}} + \boldsymbol{f}(\dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{\tau} \quad (3.1)$$

ここで $\boldsymbol{J}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}$ は慣性力項， $\boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ は遠心力およびコリオリ力項， $\boldsymbol{D}\dot{\boldsymbol{\theta}}$ は粘性摩擦力項， $\boldsymbol{f}(\dot{\boldsymbol{\theta}})$ はクーロン摩擦力項， $\boldsymbol{g}(\boldsymbol{\theta})$ は重力項，そして $\boldsymbol{\tau}$ は関節トルクである。このように、

駆動入力である関節トルク τ から運動方程式を解いて物体の運動を求めることを動力学、または、次節で述べる逆動力学と区別して、順動力学という。

3.1.2 逆動力学

ロボットの制御に関して、ある関節空間の指令 θ^{cmd} が与えられたとき、それを実現する関節トルク τ を求めたいことがある。これは前節で説明した動力学とは逆の立場になるため、一般に逆動力学という。逆動力学モデルを利用したロボットアームの制御方法の代表例として、計算トルク法 [15] を利用した加速度制御システムについて説明する。

計算トルク法では、物理パラメータに基づいて、与えられた運動を実現するのに必要なトルク参照入力 τ^{ref} を計算する。すなわち、(3.1) 式を (3.2) 式のように書きかえ、必要なトルク参照入力 τ^{ref} を計算する。

$$\tau^{\text{ref}} = \hat{J}(\theta^{\text{res}}) \ddot{\theta}^{\text{ref}} + \hat{h}(\theta^{\text{res}}, \dot{\theta}^{\text{res}}) + \hat{D} \dot{\theta}^{\text{res}} + \hat{f}(\dot{\theta}^{\text{res}}) + \hat{g}(\theta^{\text{res}}) \quad (3.2)$$

ここで、 $\hat{\bullet}$ は物理パラメータの推定値を示し、角加速度参照入力 $\ddot{\theta}^{\text{ref}}$ は、フィードバック制御器の出力から得られるものである。例えば、フィードバック制御器を PD 制御器とするならば、角加速度参照入力 $\ddot{\theta}^{\text{ref}}$ は (3.3) 式のように計算される。

$$\ddot{\theta}^{\text{ref}} = k_p(\theta^{\text{cmd}} - \theta^{\text{res}}) + k_d(\dot{\theta}^{\text{cmd}} - \dot{\theta}^{\text{res}}) \quad (3.3)$$

ここで、 k_p は比例ゲイン、 k_d は微分ゲインである。このように、角度応答 θ^{res} 、角速度応答 $\dot{\theta}^{\text{res}}$ 、角加速度参照入力 $\ddot{\theta}^{\text{ref}}$ から物理パラメータを考慮してトルク参照入力 τ^{ref} を計算するものが加速度ベースの計算トルク法である。

計算トルク法は、制御過程で逆動力学モデルを利用し、モデルや物理パラメータが適切であれば誤差なく制御できるという利点がある。ただし、必要な物理パラメータを高精度に導出する必要があることから、複雑な運動方程式を解かなければならないという欠点も含む。従来の構造が固定化されているロボットであるならば、一度必要な物理パラメータが推定できればよく、この欠点が大きく影響することはないため、計算トルク法は有効な制御方法の一つとして考えることができる。

しかしながら、構造が頻繁に組み替えられるロボットには、複雑な運動方程式を組み替えるたびに導出しなければならないという観点から、計算トルク法のようなすべての物理パラメータを導出する必要がある制御方法は、その欠点が大きく影響してしまう。さらに、制御するのに必要な参照入力を計算する最適な制御器も、当然ながら

組み替えごとに異なることも問題となる。そのため、本研究では動力学までは立ち入らず、簡単な運動学のみで計算でロボットアームを制御し、制御器パラメータに関しても単純な代数計算のみで自律的に更新可能な制御方法を提案する。次節以降で、提案されている中央制御部、分散知能の制御システムを説明する。

3.2 中央制御部

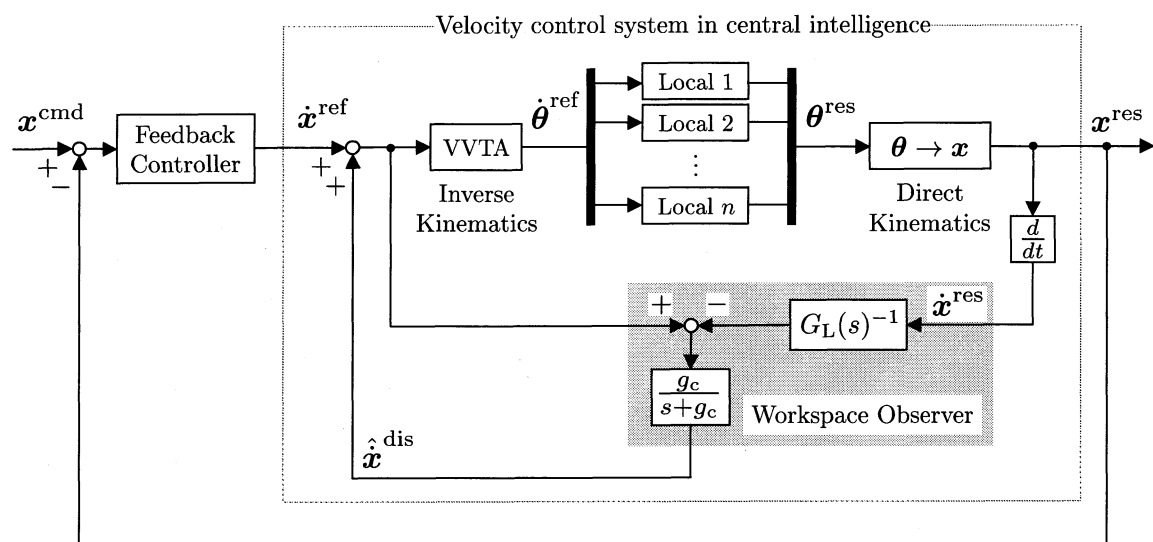


図 3.1: 中央制御部 (手先の速度制御システム)

中央制御部はロボット全体を統括する役割を持つ。すなわち、各分散知能への指令を生成する部分となる。例として、ロボット単体の手先の位置を制御する中央制御部のブロック図を図 3.1 に示す。

中央制御部は、作業空間における手先の位置指令 x^{cmd} と手先の位置応答 x^{res} の偏差をフィードバック制御に通し、速度参照入力 \dot{x}^{ref} を計算する。その後、その参照入力 \dot{x}^{ref} を仮想速度伝播アルゴリズム (VVTA) を用いて関節空間における速度参照入力 $\dot{\theta}^{\text{ref}}$ へ変換し、その参照入力を分散知能へ送信する。そして、分散知能から各関節の角度応答 θ^{res} を受信し、手先の位置応答 x^{res} を計算する。また、この手先の位置応答 x^{res} をフィードバックし、速度参照入力 \dot{x}^{ref} を計算するフィードバック制御器を構成したものが先行研究で提案されていた構造可変型ロボット単体のための手先位置制御システムである。

以下の小節では、図 3.1 中の分散知能、中央制御部の運動学、作業空間オブザーバについて説明する。

3.3 分散知能

作業変化や環境変化に対応するためにロボットの構造を組み替えるとき、そのロボットが有する物理パラメータは変化する。その都度、新たにその情報を入力し、制御シ

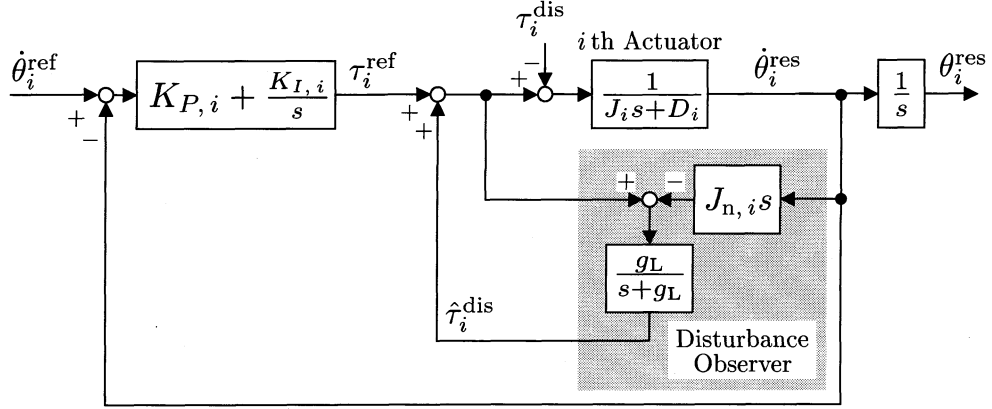


図 3.2: 分散知能 (速度制御システム)

システムを再設計・再調整することは非常に非効率的である。そこで、構造の組み替えに適した制御システムを構築するために、各モジュールに分散知能を配置し、そのモジュールのパラメータを保持し、必要に応じて中央制御部へ送信し、自律的な制御システムの再設計・再調整を可能とすることで、柔軟な構造の組み替えが期待できる。そのような分散知能の制御システムとして、図 3.2 に示される外乱オブザーバを付加した速度ベースの制御システムが考案されている。次節にて、その制御システムを説明する。

3.3.1 外乱オブザーバ [9]

分散知能における速度制御システムでは、外乱オブザーバを適用している。外乱オブザーバについて説明するために、まず図 3.3 (a) について考える。ここで下添え字 \bullet_i は、 i 番目のモジュールに付加された分散知能 i の変数もしくはパラメータであることを表す。図 3.3 中の τ_i^{dis} は動力学モデルとして示した (3.1) 式における相互慣性力項 $\{\mathbf{I} - \text{diag} \hat{\mathbf{J}}(\boldsymbol{\theta}^{\text{res}})\} \ddot{\boldsymbol{\theta}}^{\text{res}}$ 、遠心コリオリ力項 $\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ 、粘性摩擦力項 $\mathbf{D} \dot{\boldsymbol{\theta}}$ 、クーロン摩擦力項 $\mathbf{f}(\dot{\boldsymbol{\theta}})$ 、重力項 $\mathbf{g}(\boldsymbol{\theta})$ のすべての情報を含んだ外乱であり、 τ_i^{ext} はロボットに外力が加わったときの外力トルクである。このブロック図では、アクチュエータのトルク参照入力 τ_i^{ref} とアクチュエータの速度応答 $\dot{\theta}_i^{\text{res}}$ を微分しノミナル慣性 $J_{n,i}$ を乗じたものとの差をとり、推定外乱 $\hat{\tau}_i^{\text{dis}}$ を算出する。このとき、推定外乱 $\hat{\tau}_i^{\text{dis}}$ は (3.4) 式のように推定される。

$$\hat{\tau}_i^{\text{dis}} = \frac{g_L}{s + g_L} \{ (J_i - J_{n,i}) s \dot{\theta}_i^{\text{res}} + \tau_i^{\text{dis}} + \tau_i^{\text{ext}} \} \quad (3.4)$$

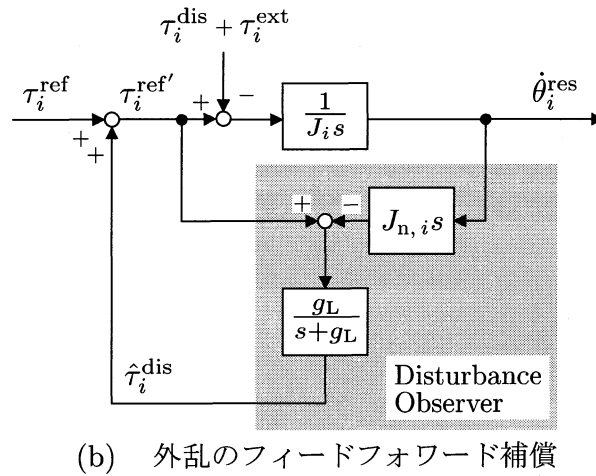
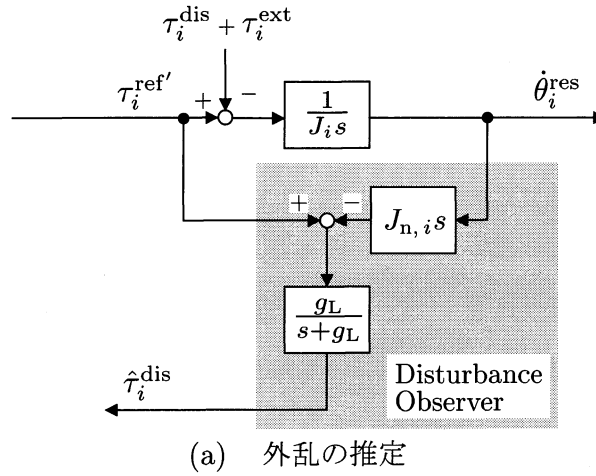


図 3.3: 外乱オブザーバ

(3.4) 式において, g_L は外乱オブザーバのローパスフィルタのカットオフ周波数であり, 右辺の第一項はロボットの姿勢の変動等に起因する慣性変動による項, 第二項は外力とその他の外乱による項である。これらを一つの外乱として一括して推定・補償するため, 図 3.3 (b) のように外乱のフィードフォワード補償を行うと (3.5) 式が算出できる。

$$\dot{\theta}_i^{\text{res}} = \frac{1}{J_{n,i}s} \left\{ \tau_i^{\text{ref}} - \frac{s}{s + g_L} (\tau_i^{\text{dis}} + \tau_i^{\text{ext}}) \right\} \quad (3.5)$$

(3.5) 式より, $\tau_i^{\text{dis}} + \tau_i^{\text{ext}}$ にはハイパスフィルタが付加されていると考えることができるため, 外乱オブザーバのカットオフ周波数 g_L 以下においてその項はほぼ無視できる。すなわち, 逆動力学を計算せずに相互慣性力, 遠心コリオリ力などの干渉項を補償することができ, 制御対象はノミナル値 $J_{n,i}$ に固定化される。

外乱オブザーバの補償が十分に行われ, τ_i^{ref} から $\dot{\theta}_i^{\text{res}}$ までの伝達関数が $1/(J_{n,i}s)$ に

なっていると仮定すると、図 3.2 の伝達関数は (3.6) 式のように表される。

$$\dot{\theta}_i^{\text{res}} = \frac{K_{P,i}s + K_{I,i}}{J_{n,i}s^2 + K_{P,i}s + K_{I,i}} \dot{\theta}_i^{\text{ref}} \quad (3.6)$$

(3.6) 式にはリンク間の干渉や慣性変動による項が含まれないため、速度制御システムの制御器設計を簡単に行うことができる。

このように、外乱オブザーバを各モジュールの分散知能に備えることによって、組み替え時の速度制御システムの設計問題を簡単に扱うことができる。

ロボットの構造の組み替え時、中央制御部は各モジュールからリンク長、質量、重心位置などの物理パラメータを収集し、適切なノミナル慣性 $J_{n,i}$ を算出し、各分散知能へ送信する。各分散知能はそのノミナル慣性を用いて、次節で述べる係数図法により適切な制御器パラメータを更新する。

3.3.2 係数図法 [10, 11]

図 3.2 に示される制御器 ($K_{P,i}, K_{I,i}$) の設計については、真鍋氏より考案されている簡単な代数計算より設計を行うことができる係数図法を用いる。係数図法について簡単に説明する。

ある制御系の特性方程式が (3.7) 式で与えられたとする。

$$M(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0 \quad (3.7)$$

ここで、安定性に関わる値である安定度指標 γ_i 、整定時間に関わる値である等価時定数 τ_v を (3.8), (3.9) 式で与える。

$$\gamma_i = \frac{a_i^2}{a_{i+1}a_{i-1}} \quad (3.8)$$

$$\tau_v = \frac{a_1}{a_0} \quad (3.9)$$

安定度指標 γ_i の与え方の指針として、(3.10) 式で表される標準形がある。

$$\gamma_1 = 2.5, \quad \gamma_{n-1} = \gamma_{n-2} = \cdots = \gamma_2 = 2 \quad (3.10)$$

標準形では安定性や応答性が非常に優れた特性となることが知られており、本研究ではこの標準形を用いて制御器を設計する。

係数図法に従って図 3.2 の制御器パラメータ $K_{P,i}, K_{I,i}$ を求める。図 3.2 の特性方程式は (3.11) 式となる。

$$M(s) = J_{n,i}s^2 + K_{P,i}s + K_{I,i} \quad (3.11)$$

そのため、制御器パラメータ $K_{P,i}, K_{I,i}$ と安定度指標 γ_1 、等価時定数 τ_v の関係式は (3.12), (3.13) 式となる。

$$\gamma_1 = \frac{K_{P,i}^2}{J_{n,i} K_{I,i}} \quad (3.12)$$

$$\tau_v = \frac{K_{P,i}}{K_{I,i}} \quad (3.13)$$

(3.12), (3.13) 式を $K_{P,i}, K_{I,i}$ について整理すると (3.14), (3.15) 式のように表せる。

$$K_{P,i} = \frac{J_{n,i} \gamma_1}{\tau_v} \quad (3.14)$$

$$K_{I,i} = \frac{J_{n,i} \gamma_1}{\tau_v^2} \quad (3.15)$$

この (3.14), (3.15) 式を用い、 γ_1 を標準形とし ($\gamma_1 = 2.5$)、 τ_v を仕様により決定することで簡単に制御器パラメータを決定できる。

このような手順で設計された速度制御システムは (3.14), (3.15) 式を (3.6) 式に代入することにより以下のような応答として表せる。

$$\dot{\theta}_i^{\text{res}} = G_L(s) \dot{\theta}_i^{\text{ref}} \quad (3.16)$$

$$G_L(s) = \frac{\gamma_1 \tau_v s + \gamma_1}{\tau_v^2 s^2 + \gamma_1 \tau_v s + \gamma_1} \quad (3.17)$$

また、すべての分散知能で同様の速度制御システムが構成されるため、パラメータ変動や外乱の影響が十分に抑制されていれば、各モジュールに対して (3.18) 式が成り立ち、すべてのモジュールの応答特性を (3.19) 式のように $G_L(s)$ にそろえることができる。

$$\begin{bmatrix} \dot{\theta}_1^{\text{res}} \\ \dot{\theta}_2^{\text{res}} \\ \vdots \\ \dot{\theta}_n^{\text{res}} \end{bmatrix} = \begin{bmatrix} G_L(s) & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & G_L(s) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^{\text{ref}} \\ \dot{\theta}_2^{\text{ref}} \\ \vdots \\ \dot{\theta}_n^{\text{ref}} \end{bmatrix} \quad (3.18)$$

$$\dot{\boldsymbol{\theta}}^{\text{res}} = G_L(s) \dot{\boldsymbol{\theta}}^{\text{ref}} \quad (3.19)$$

外乱オブザーバの補償が十分に行われる範囲においては、どのようなモジュールに対しても (3.19) 式が成り立ち、ロボットの構造を意識する必要がない。この点が、構造可変型ロボットに対して速度ベースの制御システムを持つ分散知能を構成する利点となる。

3.4 ロボットの運動学

ロボットを制御するには、ロボットの運動学を解かなければならない。関節空間における各関節の角度(直動軸なら移動量)や角速度の情報から、作業空間における手先の位置や速度を求めることを順運動学といい、手先の位置から各関節の角度、角速度を求めることを逆運動学という。以下の小節において、一般的な運動学を説明した後、構造可変型ロボットに適した運動学について述べる。

3.4.1 一般的な運動学 [16]

順運動学

ロボットを制御するとき、一般的に得られる情報は各関節のエンコーダからの角度情報であり、そこから手先の位置や速度を計算し、タスクを行う手先の情報をフィードバックして制御する。そのため、関節空間における各関節の角度(直動軸なら移動量)や角速度の情報から、作業空間における手先の位置や速度を求める順運動学を計算する必要がある。この順運動学を計算する手法の代表例である Denavit-Hartenberg の表記法(DH法)について説明する。

DH法の準備として、図3.4のように座標系を定義する。ここで使用するリンク座標系は以下のように定義される。

1. リンク i の関節軸 i (回転軸) を z_i 軸とする座標系 Σ_i を設定する。
2. x_i 軸は z_i 軸と z_{i+1} 軸との共通垂線とし、その方向は z_i 軸から z_{i+1} 軸へ向かう方向とする。共通垂線と z_i 軸との交点が Σ_i の座標原点 O_i である。例外として、 z_i 軸と z_{i+1} 軸が平行な場合は、共通法線が無数に存在するため、 x_{i-1} 軸と z_i 軸との交点に Σ_i の座標原点 O_i を定める。
3. y_i 軸は右手座標系をもとに設定する。

また、DH法で用いるパラメータは以下のとおりである。

1. θ_i (リンク間角度) $\cdots z_i$ 軸回りに右ネジ方向に測った x_{i-1} 軸から x_i 軸への角度
2. d_i (リンク間距離) $\cdots z_i$ 軸の正方向に沿って測った x_{i-1} 軸から x_i 軸への距離
3. α_i (リンクねじれ角) $\cdots x_{i-1}$ 軸回りに右ネジ方向に沿って測った z_{i-1} 軸から z_i 軸への角度

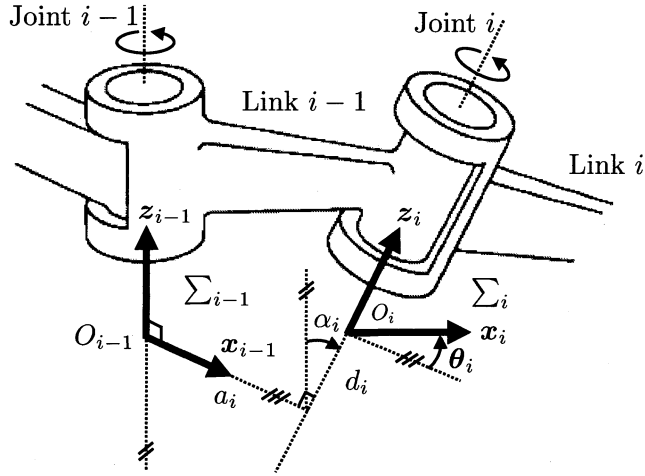


図 3.4: DH 法で用いる座標系

4. a_i (リンク長さ) $\cdots x_{i-1}$ 軸の正方向に沿って測った z_{i-1} 軸から z_i 軸への距離

これらの4つのパラメータのうち、 α_i と a_i はリンクの幾何学的構造により決定される定数であり、回転関節の場合は θ_i が変数で d_i は定数、直動関節の場合は d_i が変数で θ_i は定数となる。

これらの4つのパラメータを用いた座標変換の手順を以下に示す。

1. x_{i-1} 軸に沿って a_i の並進
2. x_{i-1} 軸まわりに α_i の回転
3. 回転後の z_{i-1} 軸に沿って d_i の並進
4. 並進後の z_{i-1} 軸のまわりに θ_i の回転 (Σ_{i-1} が Σ_i に一致)

これらの4つの変換を順次行うことにより、(3.20) 式の同次変換行列が求められる。

$$\begin{aligned}
 {}^{i-1}\mathbf{T}_i &= \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & a_i \\ C_{\alpha_i} S_{\theta_i} & C_{\alpha_i} C_{\theta_i} & -S_{\alpha_i} & -S_{\alpha_i} d_i \\ -S_{\alpha_i} S_{\theta_i} & S_{\alpha_i} C_{\theta_i} & C_{\alpha_i} & C_{\alpha_i} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}
 \end{aligned}$$

ここで, $S_{\theta_i} = \sin \theta_i$, $C_{\theta_i} = \cos \theta_i$, $S_{\alpha_i} = \sin \alpha_i$, $C_{\alpha_i} = \cos \alpha_i$ とする。この同次変換行列を用いることで, n 関節ロボットのリンク座標系 Σ_n と基準座標系 Σ_0 を関係づける変換式は (3.21) 式のように表現できる。

$${}^0T_n = {}^0T_1 {}^1T_2 \cdots {}^{n-1}T_n \quad (3.21)$$

さらに, 座標系 Σ_n からみた手先の位置 \mathbf{x}_{end} は, 関節 n から手先までのリンク長を L_n とすると (3.22) 式のような関節角度 $\boldsymbol{\theta}$ の関数として求められる。

$$\mathbf{x}_{\text{end}} = \mathbf{T}(\boldsymbol{\theta}) = {}^0T_n [L_n \ 0 \ 0 \ 1]^T \quad (3.22)$$

このように, DH 法で順運動学を解くことにより, 簡単な同次変換行列の乗算の組み合わせで, 関節角度 $\boldsymbol{\theta}$ から手先の位置 \mathbf{x}_{end} の関係式を求めることができる。

構造可変型ロボットのための制御システムにこの DH 法を組み込む場合, まず, 各分散知能が, 各モジュールがもつ同次変換行列を保持しておき, 組み替え時に中央制御部へその情報を送信する。そして, 中央制御部が (3.22) 式を解くことで系統的に順運動学を計算できる。すなわち, 組み替えに応じた対応は容易に行うことができると考えられる。

逆運動学

一般的に, タスクを行うのはロボットの手先である。つまり, ロボットの制御は, 直交座標系である作業空間で定義される手先の位置や速度に指令を与え, 制御入力进行計算する。そして, 得られた作業空間における制御入力を, 関節空間における各関節のための制御入力へ変換し, 各関節のアクチュエータへ入力する。このようにロボットを制御するには, 手先の位置や速度の情報を各関節の位置や速度の情報に座標変換する逆運動学を計算する必要がある。

この逆運動学を計算する手法の一般的な方法として, ヤコビ行列の逆行列を用いる速度次元の逆運動学を説明する。まず, (3.22) 式の両辺を時間 t について微分すると (3.23) 式を得る。

$$\dot{\mathbf{x}}_{\text{end}} = \frac{\partial \mathbf{T}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{d\boldsymbol{\theta}}{dt} = \mathbf{J}_{\text{aco}}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (3.23)$$

ここで $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})$ は作業空間の自由度と関節空間の自由度が等しいとき正方行列となり, ある手先速度を実現する関節角速度が (3.24) 式より一意に求まる。

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_{\text{aco}}^{-1}(\boldsymbol{\theta}) \dot{\mathbf{x}}_{\text{end}} \quad (3.24)$$

また、作業空間の自由度に比べて、関節空間の自由度が多いとき、 $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})$ は横長の行列となり、ある手先速度を実現する関節角速度が一意に決まらなくなる。この場合、擬似逆行列 $\mathbf{J}_{\text{aco}}^{\dagger}(\boldsymbol{\theta})$ を用いた方法があり、これを用いる場合、内積 $\boldsymbol{\theta}^T \cdot \boldsymbol{\theta}$ を最小化する最小 2 乗解が (3.25) 式から求まる。

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_{\text{aco}}^{\dagger}(\boldsymbol{\theta}) \dot{\mathbf{x}}_{\text{end}} \quad (3.25)$$

$$\mathbf{J}_{\text{aco}}^{\dagger}(\boldsymbol{\theta}) = \mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta}) \{ \mathbf{J}_{\text{aco}}(\boldsymbol{\theta}) \mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta}) \}^{-1} \quad (3.26)$$

以上の計算を行うことで作業空間の手先速度を関節空間の角速度へ座標変換することが可能となる。

しかし、本研究では構造可変型ロボットを対象としており、当然ながら、組み替えごとにヤコビ行列 $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})$ の内容は変化していく。ヤコビ行列 $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})$ の列数はそのロボットの関節自由度の数であるため、構造に依存してこの行列の導出は困難になりうる。さらに、擬似逆行列を計算することは構造に依存して計算式が大幅に変わっていく、計算量が急激に増加していく問題も存在する。したがって、構造が変化していくロボットを制御するとき、ヤコビ行列を求めて逆行列 (擬似逆行列) を構成して逆運動学を解くシステムは、組み替え時の対応が複雑になると考えられる。他の論文においても、組み替えを前提とするモジュール型ロボットのような構造変化を前提とするロボットを制御するにはヤコビ行列を用いることは適さないことが言及されている [12]。このような観点から、構造可変型ロボットに適した逆運動学としてヤコビ行列を用いないものが望ましい。次節でこの逆運動学を説明する。

3.4.2 構造可変型ロボットに適した逆運動学 [3, 4, 5, 12]

前節で示した問題に対して、矢向氏はヤコビ行列の逆行列を用いない逆運動学の解法である仮想力伝播アルゴリズム (VFTA) を提案した [12]。そして、その VFTA を構造可変型ロボットに適したアルゴリズムへと応用するため、仮想的な力を速度に置き換える仮想速度伝播アルゴリズム (VVTA) が三輪氏によって考案され、仮想的な力を位置に置き換える仮想位置伝播アルゴリズム (VPTA) が片岡氏によって考案されている [3, 4, 5]。本研究で提案する制御システムでは VVTA を用いて逆運動学を解くため、以下、VVTA について詳しく説明する。

まず図 3.5 のように座標を定義する。座標系 Σ_{i+1} に (3.27) 式に示される速度参照入力 $\dot{\mathbf{x}}_{i+1}^{\text{ref}}$ が与えられたとする。

$$\dot{\mathbf{x}}_{i+1}^{\text{ref}} = \begin{bmatrix} \dot{x}_{r_{i+1}}^{\text{ref}} & \dot{x}_{\theta_{i+1}}^{\text{ref}} & \dot{x}_{n_{i+1}}^{\text{ref}} \end{bmatrix}^T \quad (3.27)$$

ここで下添え字の \bullet_r は軸方向、 \bullet_θ は隣接する関節の回転方向、 \bullet_n はこれらの軸に直交する方向として定義する。 θ 成分の速度参照入力 $\dot{x}_{\theta_{i+1}}^{\text{ref}}$ は (3.28) 式を用いて関節 i の参照値となり、残りの成分は (3.29) 式を用いて次の関節の参照入力として伝播される。

$$\dot{\theta}_i^{\text{ref}} = \dot{x}_{\theta_{i+1}}^{\text{ref}} / L_{i+1} \quad (3.28)$$

$$\dot{\mathbf{x}}_i^{\text{ref}} = {}^i\mathbf{R}_{i+1}^{-1} \begin{bmatrix} \dot{x}_{r_{i+1}}^{\text{ref}} & 0 & \dot{x}_{n_{i+1}}^{\text{ref}} \end{bmatrix}^T \quad (3.29)$$

ここで ${}^i\mathbf{R}_{i+1}$ は 3×3 の座標系 Σ_i から座標系 Σ_{i+1} への変換行列であり、DH 法の同次変換行列の並進を表す 4 行目と 4 列目のベクトルを省略したものに相当する。隣接する関節の回転軸は図 3.5 (a) に示されるように平行、または図 3.5 (b) に示されるように垂直に接続されていることが多く、関節 i と関節 $i-1$ の回転軸が平行であるとき (3.29) 式は (3.30) 式となり、垂直であるとき (3.29) 式は (3.31) 式となる。

$$\dot{\mathbf{x}}_i^{\text{ref}} = \begin{bmatrix} \dot{x}_{r_{i+1}}^{\text{ref}} \cos q_i & \dot{x}_{r_{i+1}}^{\text{ref}} \sin q_i & \dot{x}_{n_{i+1}}^{\text{ref}} \end{bmatrix}^T \quad (3.30)$$

$$\dot{\mathbf{x}}_i^{\text{ref}} = \begin{bmatrix} \dot{x}_{r_{i+1}}^{\text{ref}} \cos q_i & \dot{x}_{n_{i+1}}^{\text{ref}} & \dot{x}_{r_{i+1}}^{\text{ref}} \sin q_i \end{bmatrix}^T \quad (3.31)$$

また、ロボットに直動リンクが存在する場合も、その進行方向を θ 、それに直交する方向を r 、 n と定義して (3.29) 式を用いて伝播を行い、そのリンクへの指令は (3.28) 式を用いずに θ 方向の指令を直接加えることで適用可能となる。このような関節から関節への参照入力の伝播を繰り返すことで、作業空間で与えられた参照入力を関節空間の参照入力に変換する。

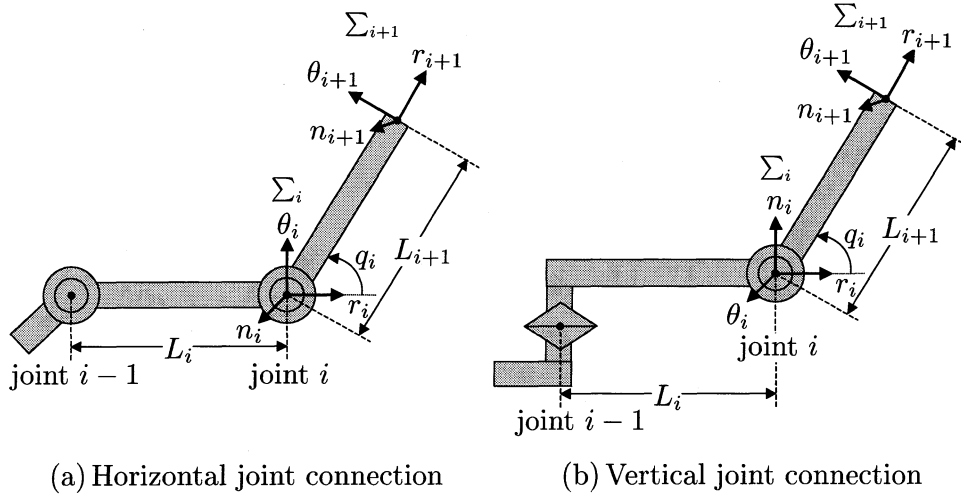


図 3.5: 関節空間における座標系

一般的な逆運動学の解法としてヤコビ行列の逆行列を用いた場合であれば、構造変更時に一からヤコビ行列を再構成する必要があるのに対し、このアルゴリズムは変更した軸分の伝播計算を変更するのみであり、大幅なシステム変更を必要としない。また、関節数の増加により、計算量が急激に増加することもなく、アルゴリズムも単純である。そのため、構造可変型ロボットに適した逆運動学であるといえる。

3.4.3 作業空間オブザーバ [9]

前節で説明した VVTA は完全な逆運動学の解を与えているとは限らず、手先に与えられた速度参照入力を各リンクに分配できずに余りが生じる可能性がある。さらに、分散知能における (3.19) 式の応答も、すべての関節で達成できているとは限らないことや、作業空間において動力的な干渉も生じる。そこで、それらが原因で生じてしまう手先の速度偏差を一括して作業空間における外乱 $\dot{\mathbf{x}}^{\text{dis}}$ によるものとみなし、これを推定・補償するため、作業空間オブザーバを導入する。作業空間オブザーバは、図 3.1 における速度参照入力 $\dot{\mathbf{x}}^{\text{ref}}$ と、速度応答 $\dot{\mathbf{x}}^{\text{res}}$ から推定外乱 $\hat{\mathbf{x}}^{\text{dis}}$ を (3.32), (3.33) 式を用いて算出し、図 3.1 に示すようにフィードフォワード補償することで実際に混入する外乱 $\dot{\mathbf{x}}^{\text{dis}}$ を相殺する。

$$\dot{\mathbf{x}}^{\text{dis}} = \dot{\mathbf{x}}^{\text{ref}} - G_L^{-1}(s)\dot{\mathbf{x}}^{\text{res}} \quad (3.32)$$

$$\hat{\mathbf{x}}^{\text{dis}} = \frac{g_c}{s + g_c} \dot{\mathbf{x}}^{\text{dis}} \quad (3.33)$$

ここで g_c は作業空間オブザーバのローパスフィルタのカットオフ周波数であり、この周波数以下の外乱が補償される。関節空間において (3.19) 式が成立するのであれば、(3.32) 式により作業空間のノミナルな応答特性を、関節空間におけるノミナルな応答特性 $G_L(s)$ に設定することで、推定外乱 $\hat{\mathbf{x}}^{\text{dis}}$ は VVTA の余りと作業空間の動力学的な干渉のみに依存したものとなり、作業空間オブザーバで過大な補償が必要となる可能性が低くなる。作業空間オブザーバでの補償が十分に行われていると仮定すると、作業空間では次式が成立する。

$$\dot{\mathbf{x}}^{\text{res}} = G_L(s)\dot{\mathbf{x}}^{\text{ref}} = \frac{\gamma_1\tau_v s + \gamma_1}{\tau_v^2 s^2 + \gamma_1\tau_v s + \gamma_1}\dot{\mathbf{x}}^{\text{ref}} \quad (3.34)$$

(3.34) 式はロボットの手先の応答が速度次元において構造に依存しないことを意味している。

これは次章以下で述べる指令を生成するアルゴリズムに対して、非常に有益な効果をもたらす。詳しくは第 4, 5 章で述べる。

第4章 機能性に基づく分散制御[6, 7]

本論文では、提案する制御システムで多様な協調作業を実現することを目的として、先行研究で提案されていた制御システムを応用し、機能性に基づく分散制御システムを構築する。その準備として本章ではまず、「機能性に基づいた分散制御」について簡単な例を用いて説明する。

なお、本章以降、パラメータの下添え字 \bullet_r は作業空間におけるパラメータ、下添え字 \bullet_f は機能性を用いるために新たに導入する機能空間を示すものとする。

4.1 機能性の概念

「機能」という制御システムの持つ役割を明確化するための概念について説明するため、以下のように定義する。

「タスク (task)」… ロボットの制御システムに対して使用者が要求する仕様。

「機能 (function)」… タスクを最小要素に分割したもの。タスクは機能の組み合わせで表される。

これらの定義について「ある物体を把持しながら別の場所へ移動させる」というタスクを例に説明する。このタスクは「物体の把持」と「物体の移動」の2つの機能に分解することができ、この2つの機能が同時に行われているものと考えられる。「物体の把持」は物体からの反力 f を制御することで実現でき、「物体の移動」はロボットの位置 x を制御すれば実現できる。このとき、自由度が1のロボットを2体を用いるシステムを例にとると、このタスクと機能の関係を図4.1のように解釈することができる。

では、この機能性を利用した制御システムの設計について述べる。通常、ロボット2体のシステムのための制御器は、図4.2に示されるシステムのように、個々のロボットをそれぞれ制御するように設計される。

しかしこの場合、各ロボットに対しての指令しか与えることができないため、物体をどの程度の力で把持するかや、物体をどの場所に移動させるかという要求を直接指令として設定することはできない。つまり、これらの要求を満足する個々のロボットの指令を設計者が前もって計算し、間接的に要求を満たすように個々のロボットを制

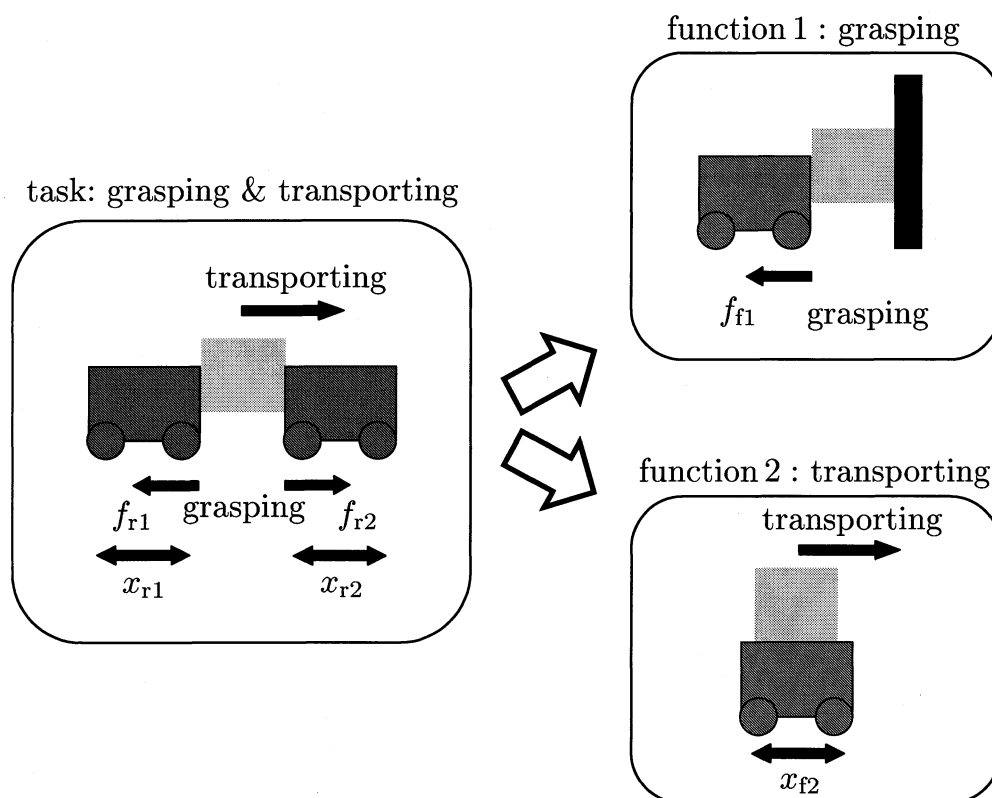


図 4.1: タスクから機能への概念図

御するしかない。このような制御システムは、制御する対象が多自由度化し、要求が複雑化するほど、設計者の負担を増加させてしまう。

それに対して、機能性に基づく分散制御システムでは、制御器の設計をこれらの機能に対して行う。すなわち、上記の例では、個々のロボットを制御する制御器ではなく、物体の把持力を制御する制御器と物体の移動場所を制御する制御器が設計され、図 4.2 の制御システムが図 4.3 のような機能性に基づく制御システムに置き換わる。機能に対して制御器設計を行うことにより、その制御器がもつ役割が明白になり、動作指令の設定や制御器設計が簡単化できる。また、それらの機能に基づく制御器の組み合わせを変更することで、仕様の変化に柔軟に対応できる制御システムを設計することが可能となる。

しかし、このような機能性に基づく制御器設計法は、複数の制御器が複数のアクチュエータに入力を加えるため、多対多の複雑な関係となってしまう。このような関係を整理するため、ロボットが定義される空間 (作業空間) から、機能に基づく空間 (機能空間) への変換が必要となる。

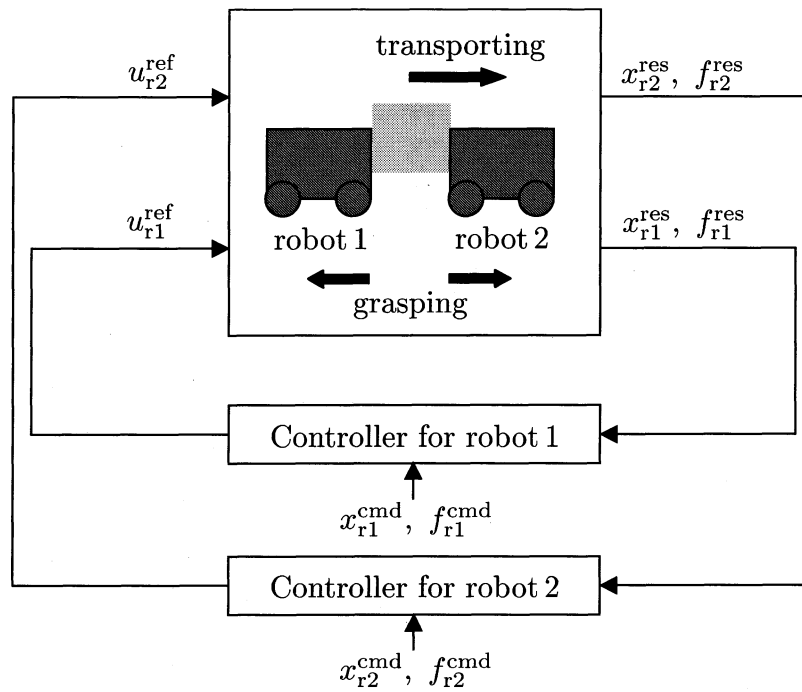


図 4.2: 一般的な制御システム

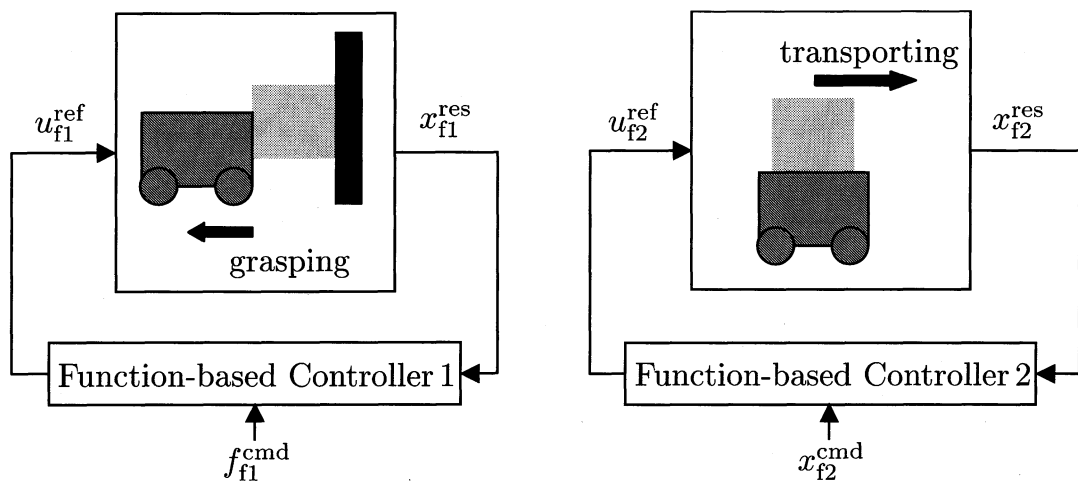


図 4.3: 機能性に基づく制御システム

4.2 機能空間への座標変換

機能に基づく制御器とロボットとの複雑な関係を整理するため、座標変換を行う。それぞれが独立な機能座標系を定義し、その空間上で機能に基づく制御器を構成することで、それぞれの制御器の単純な重ねあわせにより、各ロボットへの制御入力を生成することが可能となる。

ここで、作業空間 \mathbf{x}_r から新たに、各作業ごとに (4.1) 式で導かれる複数の機能空間 \mathbf{x}_f を以下のように定義する。

$$\mathbf{x}_f = \mathbf{T} \mathbf{x}_r \quad (4.1)$$

$$\mathbf{x}_f = \begin{bmatrix} x_{f1} & x_{f2} & \cdots & x_{fN} \end{bmatrix}^T$$

$$\mathbf{x}_r = \begin{bmatrix} x_{r1} & x_{r2} & \cdots & x_{rM} \end{bmatrix}^T$$

x は位置情報、下添え字 \bullet_r は作業空間、下添え字 \bullet_f は機能空間を示す。 N は作業空間におけるシステムの総自由度を示し、付加できる機能の数の上限となる。正則行列 \mathbf{T} は変換行列と呼ばれ、作業座標から機能座標を抽出する行列である。また、速度 $\dot{\mathbf{x}}_r$ 、加速度 $\ddot{\mathbf{x}}_r$ 、力次元の制御入力 \mathbf{u}_r 、反力 \mathbf{f}_r の変換も (4.1) 式と同様に (4.2), (4.3), (4.4), (4.5) 式のように定義する。

$$\dot{\mathbf{x}}_f = \mathbf{T} \dot{\mathbf{x}}_r \quad (4.2)$$

$$\ddot{\mathbf{x}}_f = \mathbf{T} \ddot{\mathbf{x}}_r \quad (4.3)$$

$$\mathbf{u}_f = \mathbf{T} \mathbf{u}_r \quad (4.4)$$

$$\mathbf{f}_f = \mathbf{T} \mathbf{f}_r \quad (4.5)$$

次に、定義された機能空間上での動力学について考える。各ロボットの運動方程式を (4.6) 式で表す。

$$\mathbf{m}_r \ddot{\mathbf{x}}_r + \boldsymbol{\mu} \dot{\mathbf{x}}_r = \mathbf{u}_r + \mathbf{f}_r \quad (4.6)$$

$$\mathbf{m}_r = \begin{bmatrix} m_{r1} & 0 & \cdots & 0 \\ 0 & m_{r2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & m_{rM} \end{bmatrix}$$

$$\boldsymbol{\mu}_r = \begin{bmatrix} \mu_{r1} & 0 & \cdots & 0 \\ 0 & \mu_{r2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mu_{rM} \end{bmatrix}$$

(4.6) 式において, m_r はロボットの質量であり, μ_r は摩擦係数である。ここで, すべてのロボットのパラメータが等しい, すなわち, $m = m_{r1} = m_{r2} = \cdots = m_{rM}$, $\mu = \mu_{r1} = \mu_{r2} = \cdots = \mu_{rM}$ であるとき, (4.7) 式への変形が成立する。

$$\begin{aligned} Tm\ddot{\mathbf{x}}_r + T\mu\dot{\mathbf{x}}_r &= T\mathbf{u}_r + T\mathbf{f}_r \\ mT\ddot{\mathbf{x}}_r + \mu T\dot{\mathbf{x}}_r &= T\mathbf{u}_r + T\mathbf{f}_r \\ m\ddot{\mathbf{x}}_f + \mu\dot{\mathbf{x}}_f &= \mathbf{u}_f + \mathbf{f}_f \end{aligned} \quad (4.7)$$

(4.7) 式では, 新たに定義された機能空間における実在しない仮想のロボットについての運動方程式であると解釈でき, 各仮想のロボットはそれぞれが独立な座標系に存在する。すなわち, 機能座標上で働く制御器はそれぞれが仮想のロボットを制御しているため, 互いに独立に動作可能である。

次に, 機能から変換行列 T を設定する方法について述べる。ある 1 つの機能に対応する機能座標 x_{fj} は (4.8), (4.9) 式のように求められる。

$$x_{fj} = \mathbf{t}_{\text{function},j}^T \cdot \mathbf{x}_r \quad (4.8)$$

$$\mathbf{t}_{\text{function},j} = \begin{bmatrix} t_1 & t_2 & \cdots & t_M \end{bmatrix}^T \quad (4.9)$$

ここで $\mathbf{t}_{\text{function},j}$ ($j = 1, 2, \dots, N$) は変換ベクトルと呼ばれる。各機能はそれぞれ (4.8) 式で表される変換ベクトル $\mathbf{t}_{\text{function},j}$ を持つ。それらは一つの機能における作業空間と機能空間の関係を表したものであり, 変換行列の行ベクトルとなる。すべての機能はそれぞれに適切な変換ベクトルが設定され, 変換行列は (4.10) 式から決定される。

$$T = \begin{bmatrix} \mathbf{t}_{\text{function},1}^T \\ \mathbf{t}_{\text{function},2}^T \\ \vdots \\ \mathbf{t}_{\text{function},N}^T \end{bmatrix} \quad (4.10)$$

ここで, 図 4.1 の座標系を用いて, 変換行列 T の設定方法の具体例について説明する。「物体の把持」の機能では, 各ロボットの反力の差を制御するため, $\bullet_{f1} = \bullet_{r1} - \bullet_{r2}$ という座標系で制御することが可能となる。「物体の移動」の機能では, 2 つのロボットの中心, つまり和の座標系を制御すればよいため, $\bullet_{f2} = \bullet_{r1} + \bullet_{r2}$ という座標系で制御する。したがって, このときの変換行列 T は (4.11) 式のように求められる。

$$T = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (4.11)$$

このように、この例ではロボットの総自由度が2であったため、2つの機能を同時に動作させることが可能となる。これが多自由度の大規模システムとなれば、さらに多くの機能を付加でき、より高度なタスクが可能となる。以上が機能性に基づく分散制御システムの一例である。

4.3 構造可変型ロボットに適した機能性に基づく分散制御

本研究では、機能に基づくこの手法の「着脱可能な構成要素としての制御器設計」という考え方が構造可変型ロボットに非常に有効であるということを主張する。この考え方はパソコンの周辺機器の設計と類似する。この手法では、まず想定される機能すべてをあらかじめ作っておく。そして、周辺機器をパソコンに接続するように、タスクや環境の変化に応じて必要となった機能呼び出す。機能の組み合わせ数は非常に多く、多様なパターンの挙動が想定されるが、その一方で機能に基づく個々の制御器設計は単純かつ明確である。

構造可変型ロボットでは構造の組み替えを前提としているため、タスクや環境の変化が多く行われることが想定される。そのため、この手法を用い、制御器に対しても同様に組み替えを容易に行うことが可能な制御構造をとることは非常に有効であると考えられる。

しかし課題として、各ロボットのパラメータが同等なものでなければ、独立な機能座標を定義できないという制約が存在することが文献[6]より指摘されている。つまり、同じ構造であるロボットの協調作業でなければ、この機能性を用いたシステムを構成できないため、これは構造の変更に大きな制限を加えることになる。

一方で、我々の提案している構造可変型ロボットに適した制御システムは、各モジュールの分散知能における外乱オブザーバによる制御対象のノミナル化、係数図法による速度制御器の自律調整、そして中央制御部における作業空間オブザーバの補償により、ロボットの手先の応答特性を(3.34)式に示されるようなリンク長、重さ、慣性モーメントなどのパラメータに依存しないように正規化している。すなわち、各ロボットの手先の応答特性は、それぞれ同等のパラメータを持つものとして評価できる状況にある。本研究ではこの点を利用し、手先の応答特性のみを考慮して機能性に基づく分散制御システムを適用することで、実際には各モジュールのパラメータや関節数の異なるロボットに対してもこの手法を適用可能にする。詳しくは第5章で述べる。

第5章 機能性に基づく構造可変型ロボットの制御システム

本章では、構造可変型ロボットに多様なタスクを行わせること目的とした制御システムを構築する。そこで、第3章で説明した制御システムを応用し、構造の変更、そしてタスクの変更に柔軟に対応可能な制御システムへと拡張することを目指す。また、多様な作業を可能とするため、複数のロボット用いた協調作業に着目する。ロボットを複数用いることで単体では不可能な作業が可能となり、構造可変型ロボットの協調作業となればさらに汎用性のある多様な作業の実現が期待できる。

また、一般に協調作業を行うロボット群は、システムが大規模化するため、タスク変化や状況変化における対応の高難度化、そして故障による大きな損害という欠点が挙げられるが、構造可変型ロボットによる協調作業となれば、容易な組み替えが可能という利点を活かすことでそれらの欠点を補うことができる。すなわち、構造可変型ロボットの利点と協調作業の利点の双方をうまく活かすシステムが期待できる。

以上のような理由から、本研究では前節で説明した様々な協調作業を可能とする機能性という概念に基づく制御システムを中央制御部へ組み込んだものを提案する。

5.1 提案する制御システムにおける座標変換

まず、4.3節で言及したように、(3.34)式が成り立ち、中央制御部は構造変化に依存せず、各構造可変型ロボットを同等のパラメータを持った仮想のロボットとして扱うことができると想定して座標変換を行う。すなわち、(3.34), (4.1), (4.2)式より、(5.1)式のように変形できる。

$$\begin{aligned}\dot{\mathbf{x}}_r^{\text{res}} &= \frac{\gamma_1 \tau_v s + \gamma_1}{\tau_v^2 s^2 + \gamma_1 \tau_v s + \gamma_1} \dot{\mathbf{x}}_r^{\text{ref}} \\ \mathbf{T} \dot{\mathbf{x}}_r^{\text{res}} &= \frac{\gamma_1 \tau_v s + \gamma_1}{\tau_v^2 s^2 + \gamma_1 \tau_v s + \gamma_1} \mathbf{T} \dot{\mathbf{x}}_r^{\text{ref}} \\ \dot{\mathbf{x}}_f^{\text{res}} &= \frac{\gamma_1 \tau_v s + \gamma_1}{\tau_v^2 s^2 + \gamma_1 \tau_v s + \gamma_1} \dot{\mathbf{x}}_f^{\text{ref}}\end{aligned}\tag{5.1}$$

(5.1) 式は、機能空間における各要素 \dot{x}_{fj}^{es} ($j = 1, 2, \dots, N$) が、他の要素の影響を受けず、 $\dot{x}_{fj}^{\text{ref}}$ のみに影響し、それぞれが独立であることを示す。そのため、それぞれの機能空間で各制御器を動作すれば、複数の機能を同時かつ独立に制御することができる。このような複数の機能を独立に制御できる利点は、文献 [7] で提案されているモータの角度情報から機能空間を定義する場合は、協調作業を行う各ロボットがノミナル慣性など同等のパラメータを持つことが条件であった。しかし、本研究では (3.34) 式に示されるロボットの構造に依存しない手先の応答特性のみを座標変換するため、各モジュールのパラメータや関節数の異なるロボットでもこの利点が保持される。

5.2 提案する機能性に基づく制御システム

第 4 章で説明した「機能性に基づいた分散制御」を中央制御部へ組み込んだ制御システムの概略図を図 5.1 に示す。ここで、 \mathbf{I}_2 は 2 行 2 列の単位行列である。

本論文では簡単のため、2 次元平面における 2 体のロボット用いた協調制御を実現する制御システムを考えるものとする。また、各ロボットは n 関節、 m 関節のロボットとする。

図 5.1 における各ロボットの分散知能は、制御するアクチュエータの数 (関節数) だけ必要となるため、ロボット 1 には n 個、ロボット 2 には m 個が配置される。また、ロボット 2 体を制御するため、図 3.1 の点線で囲われた構造可変型ロボット単体の速度制御システムを 2 つ用いてそれぞれのロボットの手先速度を制御する。そして、手先速度参照入力 $\dot{\mathbf{x}}_r^{\text{ref}}$ の計算方法が、一般的なフィードバック制御器ではなく、機能性に基づく制御器の組み合わせに置き換わっている。

2 体のロボットの手先位置 $\mathbf{x}_{r1} = [x_{r1}, y_{r1}]^T$ 、 $\mathbf{x}_{r2} = [x_{r2}, y_{r2}]^T$ はそれぞれ自由度が 2 であるため、合計で 4 つの自由度が存在する。そのため、4 つの機能を独立に制御することが可能となり、図 5.1 中の変換行列 \mathbf{T} の構造は 4×4 の行列となり、以下の式で座標変換が行われる。

$$\begin{aligned} \mathbf{x}_f &= \mathbf{T} \mathbf{x}_r \\ \mathbf{x}_r &= \begin{bmatrix} x_{r1} & x_{r2} & y_{r1} & y_{r2} \end{bmatrix}^T \\ \mathbf{x}_f &= \begin{bmatrix} x_{f1} & x_{f2} & x_{f3} & x_{f4} \end{bmatrix}^T \end{aligned} \quad (5.2)$$

(5.2) 式に従い、構造可変型ロボット 1, 2 の 2 次元平面の手先の位置情報は、4 つの機能空間 $(x_{f1}, x_{f2}, x_{f3}, x_{f4})$ における情報へ変換され、制御器へ入力される。そして、そ

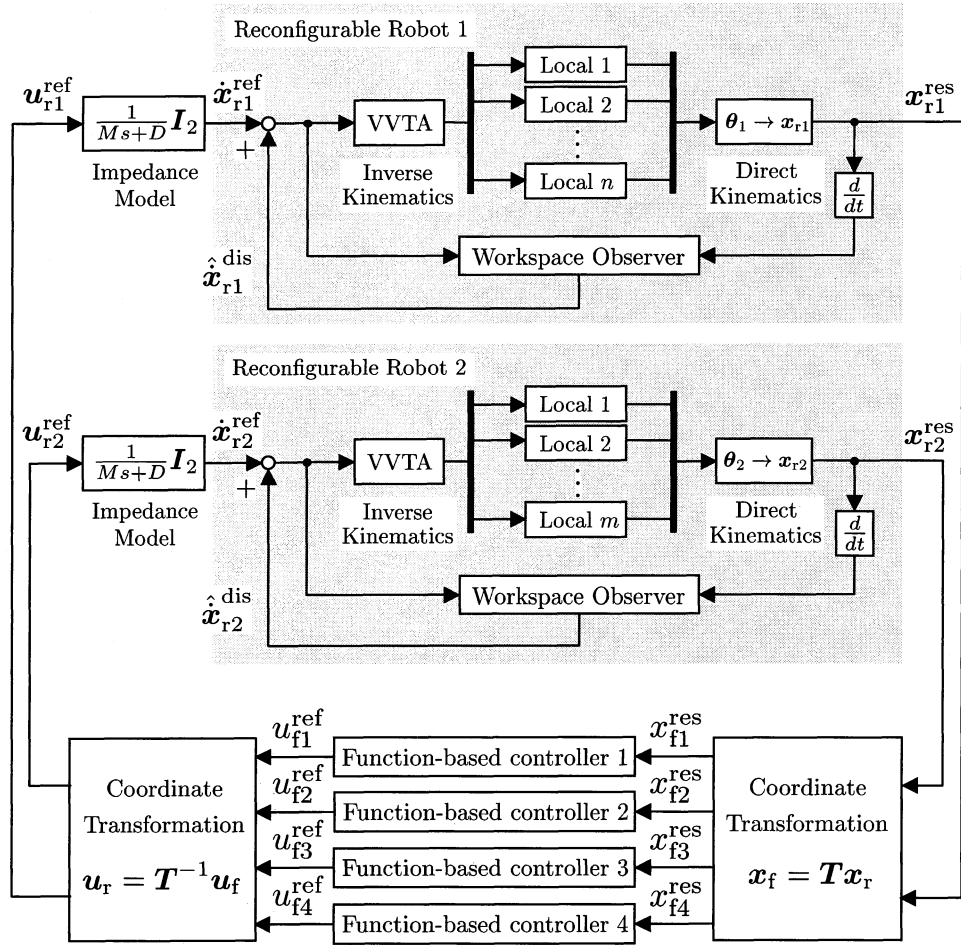


図 5.1: 提案する制御システムの概略図

れらの制御器を経て機能空間上での制御入力 $\mathbf{u}_f^{\text{ref}}$ が生成され、再び作業空間における各ロボットの制御入力 $\mathbf{u}_r^{\text{ref}}$ へと逆変換される。なお、この概略図では位置と制御入力のみ座標変換を行っているが、速度 $\dot{\mathbf{x}}_r$ 、加速度 $\ddot{\mathbf{x}}_r$ 、反力 \mathbf{f}_r も同様に (4.2), (4.3), (4.5) 式を用いて座標変換が行われる。その後、インピーダンスモデルを通じて速度参照入力 $\dot{\mathbf{x}}_r^{\text{ref}}$ が生成され、その参照入力に対して VVTA を適用する。最後に、関節空間における速度参照入力 $\dot{\boldsymbol{\theta}}^{\text{ref}}$ を各分散知能へ送信する。

また、分散知能から受信した各関節の角度応答を $\boldsymbol{\theta}^{\text{res}}$ を、ロボットを組み替えた段階で DH 法より計算した順運動学モデルにより手先位置 $\mathbf{x}_r^{\text{res}}$ を計算する。そして、再び機能空間の変数 $\mathbf{x}_f^{\text{res}}$ へ変換し、上述した方法により、分散知能への参照入力を計算する。以上が今回提案する制御系での中央制御部の役割となる。

5.2.1 インピーダンスモデル

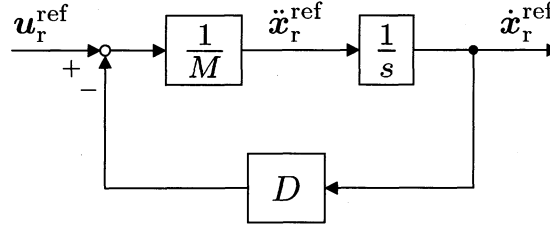


図 5.2: インピーダンスモデル

図 5.1 に示されたインピーダンスモデルについてもう少し詳しく説明する。提案する制御システムでは、各分散知能でアクチュエータの速度制御を行うことを前提としている。つまり中央制御部では速度次元の指令を分散知能へ送る必要がある。そこで、ロボットに力次元の参照入力 $\mathbf{u}_r^{\text{ref}}$ が与えられたとき、その参照入力による速度変化を、設計者が定める質量・粘性系の応答になるように入力を与える方法を用いて、力参照値 $\mathbf{u}_r^{\text{ref}}$ を速度参照値 $\dot{\mathbf{x}}_r^{\text{ref}}$ へ変換する。設計者が定める質量を M 、粘性を D とすると、その応答は (5.3) 式で表され、図 5.2 のブロック図のようにシステム上で実現される。

$$M\ddot{\mathbf{x}}_r^{\text{ref}} + D\dot{\mathbf{x}}_r^{\text{ref}} = \mathbf{u}_r^{\text{ref}} \quad (5.3)$$

また、作業変更時の制御器の組み替え時に $\mathbf{u}_{r1}^{\text{ref}}$, $\mathbf{u}_{r2}^{\text{ref}}$ は不連続に変化するおそれがあるが、このインピーダンスモデルは 1 次のローパスフィルタであるため、 $\mathbf{u}_{r1}^{\text{ref}}$, $\mathbf{u}_{r2}^{\text{ref}}$ が不連続に変化したとしてもなめらかな速度参照入力 $\dot{\mathbf{x}}_r^{\text{ref}}$ を生成することができる。

5.3 中央制御部の制御器設計

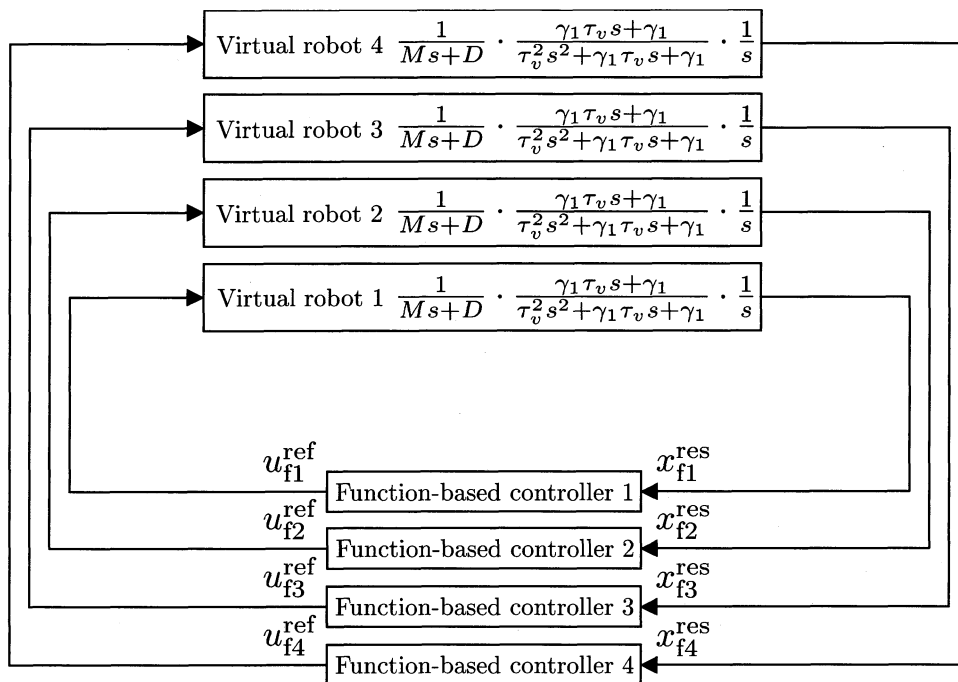


図 5.3: 機能空間における制御システム

中央制御部の制御器について説明する。中央制御部では (5.1) 式で表されるような機能空間上の仮想的なロボットを制御していると考えられる。つまり、図 5.1 の制御システムは、図 5.3 の制御システムのように解釈することができる。これは図 5.1 中の中央制御部の制御器 1, 2, 3, 4 の設計は、ロボットのリンク数 (分散知能の数) や各リンク長、慣性モーメントなどのパラメータに依存せずに行うことができることを意味する。つまり、一度これらの制御器パラメータの調整が行われれば、中央制御部の制御器は構造が変更されても同等の応答性能を保持できる。

では、この仮想のロボットが協調把持動作を実現するために必要な制御器の設計例を示す。2 体のロボットが指定した軌道で物体へ近づき、ロボットと物体が接触したと同時に把持力の制御を行う作業を考える。この作業を実現するためには、ロボットの手先や把持物体の中心などの位置を制御する機能と、物体の把持力を制御する機能が必要となる。これらの位置、力を制御する機能に対して、文献 [7] にならい、それぞれ図 5.4 に示される PD, P 制御器を設計する。

図 5.4 (a) に示される位置制御器は下添え字 \bullet_{fi} で示される機能座標における制御器である。位置を制御することを目的とするので、 x_{fi}^{res} と指令 x_{fi}^{cmd} の偏差に対して PD 制

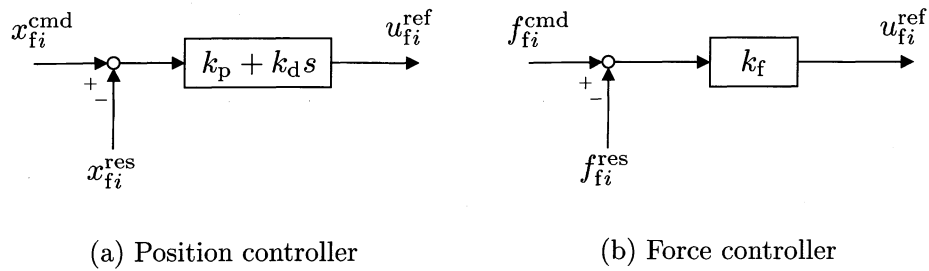


図 5.4: 機能に基づく制御器

御を行うものを設計する。

図 5.4 (b) に示される力制御器は下添え字 \bullet_{fj} で示される機能座標における制御器である。力を制御することを目的とするので、 f_{fj}^{res} と指令 f_{fj}^{cmd} の偏差に対して P 制御を行うものを設計する。

以上が今回想定した協調把持に必要な制御器となる。

そして、これらの制御器を状況に応じて適切に組み合わせていき、図 5.3 中の制御器 1, 2, 3, 4 へ取り付け、目的とする制御システムを設計する。例として、 x 軸方向の反力を制御しつつ、物体の中心位置を制御する協調把持では、機能に基づく制御器 1 に力制御器を取り付け、機能に基づく制御器 2, 3, 4 に対して位置制御器を付加すれば達成できる。このときの各制御器の動作の詳細については、第 6 章で述べる。

このように、あらかじめ必要となる制御器を設計しておき、状況に応じて適切な制御器を取り付ける、または取り外していくことで、変化する様々な仕様に柔軟に対応可能となる。また、あらかじめ複数の制御器を設計しておくことが必要となるが、これらはタスクを構成する最小要素の機能に対して設計されるため、単純かつ明白な構造のものとなる。すなわち、単純化された設計によって、多種多様な環境やタスクに対して柔軟に対応できる枠組みをもつ制御システムが構成できる。

5.4 構造可変型ロボットに適した姿勢の制御方法

前節までの機能性に基づく制御システムは、作業空間における手先のタスクのみを制御するために構成されたものである。そのため、作業空間における自由度より、関節空間における自由度が多くなる冗長なロボットの場合、その余分な自由度を有効に活用することができない。構造可変型ロボットは構造の組み替えにより、冗長なロボットが構成されることがあると考えられ、関節空間における冗長自由度を有効に活用できる制御システムが望ましいといえる。本節では関節空間における冗長自由度を関節空間における機能を付加して有効に利用できる構造可変型ロボットに適した冗長指令の生成法について述べる。

5.4.1 ヤコビ行列を用いた冗長指令の生成方法 [15]

冗長なロボットは、手先位置を制御しても無数に姿勢を選ぶことが可能である。本小節では、まず一般的なヤコビ行列 $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})$ を用いた冗長指令の生成方法について説明する。

姿勢を制御するための関節空間の参照入力 $\dot{\boldsymbol{\phi}}^{\text{ref}}$ が与えられたとする。参照入力 $\dot{\boldsymbol{\phi}}^{\text{ref}}$ を関節空間の速度参照入力として各関節を速度制御すれば、目的の姿勢へロボットは制御される。しかし、そのまま参照入力 $\dot{\boldsymbol{\phi}}^{\text{ref}}$ を付加した場合、手先に速度外乱 $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})\dot{\boldsymbol{\phi}}^{\text{ref}}$ が生じ、手先の制御に誤差が加わってしまう。そこで、図 5.5 のように、手先の速度外乱を相殺するように姿勢制御による手先の速度外乱 $\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})\dot{\boldsymbol{\phi}}^{\text{ref}}$ を手先の速度参照入力 $\dot{\mathbf{x}}^{\text{ref}}$ にあらかじめ加えておくことで、手先の制御には影響を与えずにロボットの姿勢を制御することができる。すなわち、このとき (5.4) 式のように関節空間の参照入力を計算する。

$$\begin{aligned}\boldsymbol{\theta}^{\text{ref}} &= \mathbf{J}_{\text{aco}}^{\dagger}(\boldsymbol{\theta})\{\dot{\mathbf{x}}_{\text{r}}^{\text{ref}} - \mathbf{J}_{\text{aco}}(\boldsymbol{\theta})\dot{\boldsymbol{\phi}}^{\text{ref}}\} + \dot{\boldsymbol{\phi}}^{\text{ref}} \\ &= \mathbf{J}_{\text{aco}}^{\dagger}(\boldsymbol{\theta})\dot{\mathbf{x}}_{\text{r}}^{\text{ref}} + \{\mathbf{I} - \mathbf{J}_{\text{aco}}^{\dagger}(\boldsymbol{\theta})\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})\}\dot{\boldsymbol{\phi}}^{\text{ref}}\end{aligned}\quad (5.4)$$

次に、姿勢を制御するための参照入力 $\dot{\boldsymbol{\phi}}^{\text{ref}}$ の生成方法について、評価関数の勾配を利用した方法を一例として説明する。ロボットの姿勢を評価関数 $U(\boldsymbol{\theta})$ で評価するとき、その勾配を用いた $\dot{\boldsymbol{\phi}}^{\text{ref}}$ の生成は (5.5) 式ようになる。

$$\dot{\boldsymbol{\phi}}^{\text{ref}} = [k_1 \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_1} \quad k_2 \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_2} \quad \cdots \quad k_n \frac{\partial U(\boldsymbol{\theta})}{\partial \theta_n}]^T \quad (5.5)$$

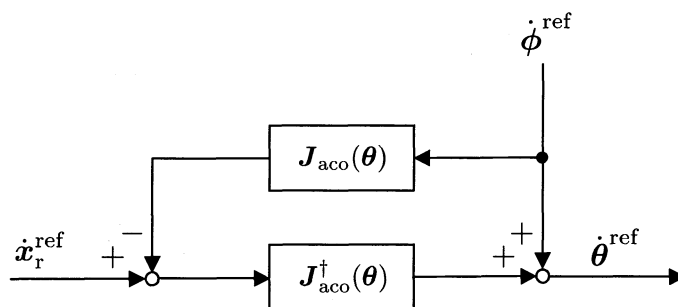


図 5.5: 手先に影響を与えない冗長指令の付加

ここで、 $\mathbf{k} = [k_1 \ k_2 \ \dots \ k_n]^T$ は比例ゲインであり、この値が正のとき、 $\dot{\phi}^{\text{ref}}$ は評価関数を大きく保つように働き、負のときは評価関数を小さく保つように働く。

このように、勾配法を用いて $\dot{\phi}^{\text{ref}}$ を生成し、(5.4) 式を用いて関節への参照入力を与えれば手先の制御に影響なく、姿勢を制御することができる。すなわち、関節空間の冗長自由度を有効に扱うことができる。

しかし、第 3 章でも述べたように、ヤコビ行列の擬似逆行列を用いた計算は、構造可変型ロボットの制御システムには不向きとなる欠点が多い。そこで、次の小節では、ヤコビ行列の擬似逆行列を用いずにロボットの姿勢を制御する方法を説明する。

5.4.2 簡便な冗長指令生成方法 [4]

前小節で説明した問題を解決するため、本小節では、文献 [13] で提案されている力ベースの制御システムを速度ベースへ変更し、第 3 章での図 3.1 の点線で囲まれた速度制御システムを図 5.6 に拡張した構造可変型ロボットに適した簡便な冗長指令生成方法を提案する。この制御システムでは、構造可変型ロボットの姿勢を制御するための冗長制御器が新たに付加されている。冗長制御器は、前小節と同様の勾配法により、 $\dot{\phi}^{\text{ref}}$ を出力する。大きな特徴は、 $\dot{\phi}^{\text{ref}}$ を (5.4) 式では必要とした $\mathbf{I} - \mathbf{J}_{\text{aco}}^{\dagger}(\theta)\mathbf{J}_{\text{aco}}(\theta)$ を通さずに角速度参照入力 $\dot{\theta}^{\text{ref}}$ へ加えていることにある。この場合、冗長指令 $\dot{\phi}^{\text{ref}}$ がロボットの手先の制御に関しては外乱となることが懸念されるが、この影響はロボットの手先に生じる速度偏差を推定し補償する作業空間オブザーバによって解決できる。つまり、図 3.1 における作業空間オブザーバは仮想速度伝播アルゴリズムで生じる誤差を補償するために付加されたものであるのに対し、図 5.6 における作業空間オブザーバは、それに加えて冗長指令を与えることによる手先誤差をも補償するものとなる。したがって、この制御システムは、 $\mathbf{I} - \mathbf{J}_{\text{aco}}^{\dagger}(\theta)\mathbf{J}_{\text{aco}}(\theta)$ を導出する計算を回避して冗長指令が与

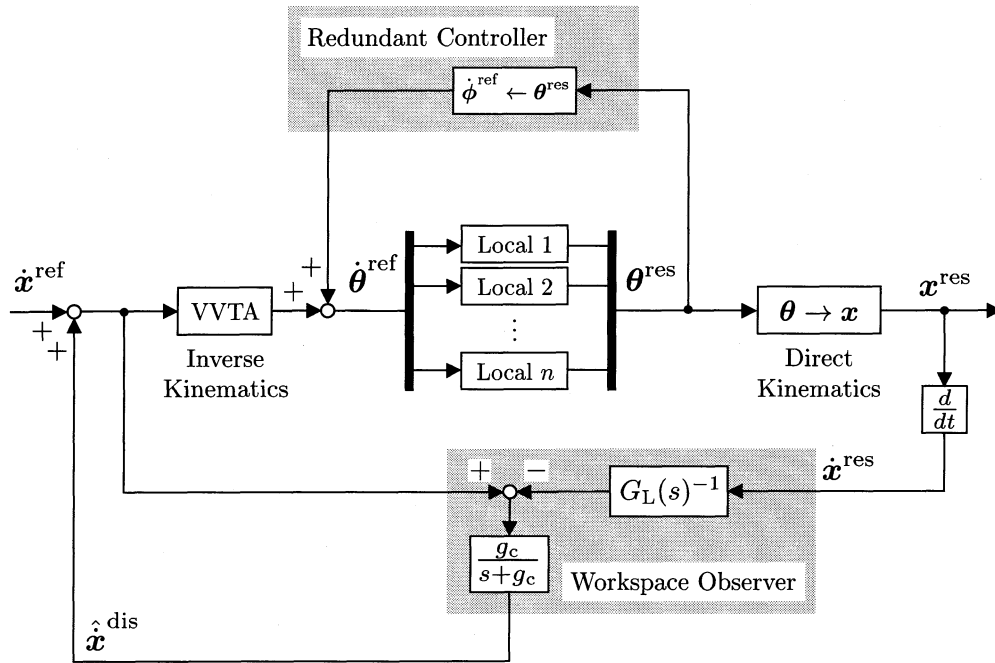


図 5.6: 冗長指令生成コントローラ

えられるという利点がある。

このように、作業空間オブザーバを利用した制御システムを構成することで、ヤコビ行列の計算を回避しながらもロボットの姿勢を制御することができる。したがって、構造可変型ロボットに適した姿勢の制御システムであるといえる。

5.4.3 勾配法の比例ゲインの一決定法

本節では、勾配法により冗長指令 $\dot{\phi}^{ref}$ を (5.5) 式のように生成するときに用いる比例ゲイン k の一決定法を示す。

前節で説明した冗長指令の生成方法は、作業空間オブザーバの補償が成立することを前提にしている。したがって、冗長指令により、作業空間オブザーバで補償できないほどの大きな外乱が手先に加われば、制御システムは不安定化する。この観点から、手先から離れた関節の回転ほど、ロボットの手先には大きな速度を与え、冗長指令が加えられた場合には大きな外乱となることに注目する。つまり、作業空間オブザーバの負担を軽減するため、手先から離れた関節ほど小さな冗長指令を与えることが望ましい。この考えを元にした比例ゲイン k の一決定法として、各関節から手先までの距離の逆比を用いることが有効である。例として、図 5.7 に表されるような 3 軸のロボッ

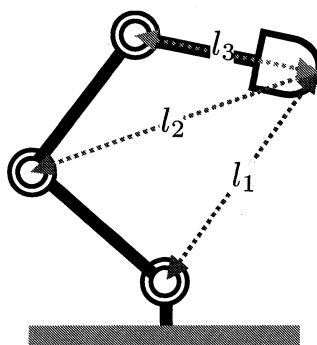


図 5.7: 各関節から先端までの距離

トの場合，各関節から手先までの距離を l_1 , l_2 , l_3 とすると (5.6) 式で制御ゲイン k_i が与えられる。

$$k_1 = k_{\text{const}}/l_1, \quad k_2 = k_{\text{const}}/l_2, \quad k_3 = k_{\text{const}}/l_3 \quad (5.6)$$

ここで， k_{const} は冗長指令が大きな値になりすぎないように定める比例定数であり，正負のどちらかに定めることで，評価関数の値を所望の値へと近づける。このような方法で評価関数の勾配に乗じる制御ゲインを与えることで，根元などの手先から離れた関節が大きく動作してしまうことを防ぐことができる。

5.4.4 機能性に基づく制御システムの拡張

本節では，5.2 節で提案した図 5.1 に示される手先の機能性に基づく制御システムと 5.4 節で提案した図 5.6 に示される姿勢の制御システムを組み合わせる。手先を機能性に基づいて制御し，かつ姿勢の制御が可能となる制御システムを図 5.8 に示す。この制御システムは手先を機能に基づく制御器で制御し，冗長制御器を用いて冗長自由度を利用して姿勢を制御することができる。また，作業空間オブザーバの補償が行き届いているとき，機能に基づく制御器と冗長制御器は互いに独立である。したがって，個々の制御器パラメータの調整は互いを意識しないで設計することができる。

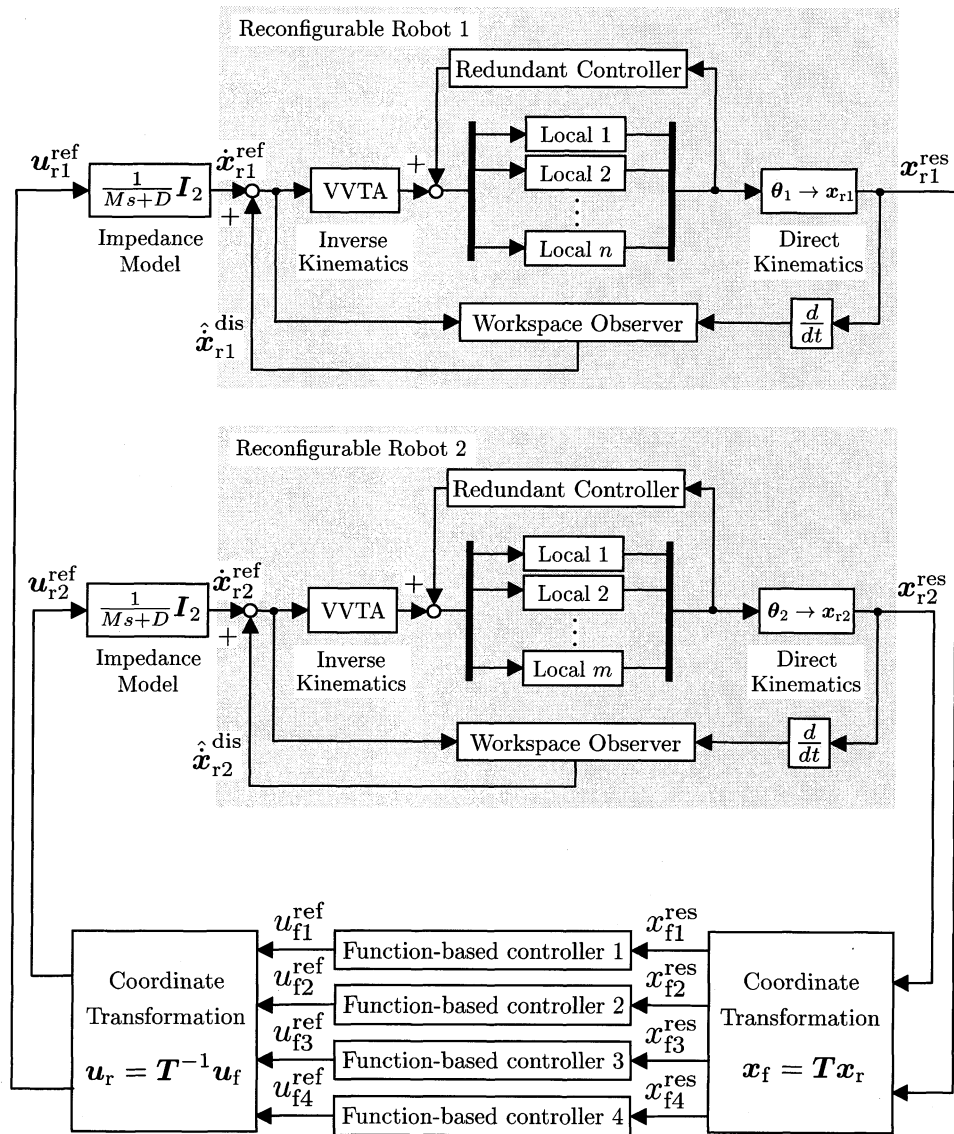


図 5.8: 姿勢制御が可能な機能性に基づく制御システム

第6章 カセンサレス制御システムの構築

ロボットが作業を行う上で力制御は必須であるが、高価であり壊れやすい力センサを必要とする力制御システムは産業システムにおいては好まれていない。これを背景に、構造可変型ロボットの力制御に関しても、力センサを用いずに行うことが望ましい。本章では、構造可変型ロボットの力センサレス制御システムについて検討する。

6.1 トルク反力の推定

6.1.1 トルク反力推定オブザーバ [8]

力センサを用いず力情報の検出のために外乱オブザーバをトルク反力推定に応用したトルク反力推定オブザーバを用いる。トルク反力推定オブザーバを図 6.1 に示す。ここで τ_i^{ref} はトルク参照入力、 τ_i^{fric} は摩擦による外乱トルク、 τ_i^{grav} は重力による外乱トルク、 τ_i^{ext} は外部から加わる力による外力トルク、 $\hat{\tau}_i^{\text{reac}}$ はトルク反力推定値を示す。 $J_{n,i}$ は設計者が定める慣性のノミナル値、 η_i はアクチュエータ i に備え付けられた減速機の効率である。重力、摩擦トルク、減速機の効率については次の小節で述べるものとする。

図 6.1 のブロック図よりトルク反力推定値 $\hat{\tau}_i^{\text{reac}}$ は (6.1) 式より導出される。

$$\hat{\tau}_i^{\text{reac}} = \frac{g_{\text{reac}}}{s + g_{\text{reac}}} \{ \tau_i^{\text{dis}} + \tau_i^{\text{ext}} + (J_i - J_{n,i}) \ddot{\theta}_i^{\text{res}} - \tau_i^{\text{fric}} - \tau_i^{\text{grav}} \} \quad (6.1)$$

図 3.3 の外乱オブザーバと異なり、重力、摩擦トルクの影響を推定値から取り除いている。重力、摩擦トルクの推定を精度よく行うことができ、実際の慣性とノミナル慣性の値に誤差が少なく ($J_i \simeq J_{n,i}$)、相互慣性力、遠心コリオリ力などの同定が困難な干渉トルクが十分小さい条件下で駆動していると考えられるのであれば、トルク反力推定オブザーバのカットオフ周波数 g_{reac} の範囲内で $\hat{\tau}_i^{\text{reac}} \simeq \tau_i^{\text{ext}}$ となり、外力によるトルク反力を推定することができる。

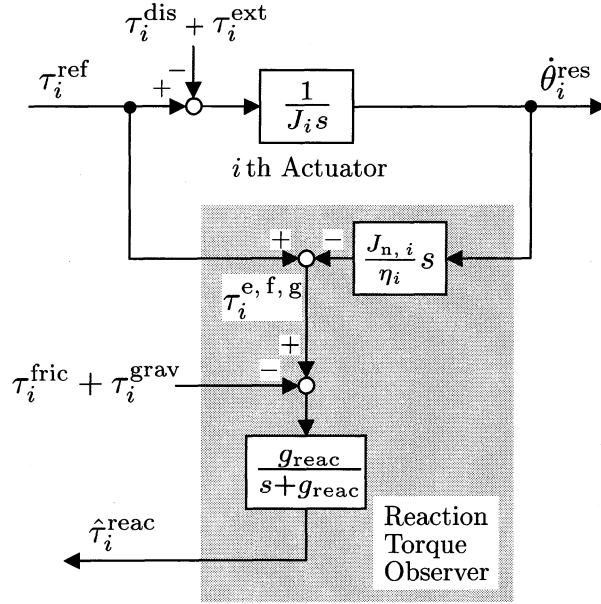


図 6.1: トルク反力推定オブザーバ

6.1.2 重力トルク，摩擦トルク，減速機の効率

本研究ではトルク反力推定オブザーバを構成するために必要な重力トルク τ_i^{grav} ，摩擦トルク τ_i^{fric} ，減速機の効率 η_i を以下のように考える。

重力トルク

重力トルクは，順運動学で使用する DH 法の同次変換行列を用いることで簡単に求めることができる。 n 関節のロボットの関節 i の重力トルク τ_i^{grav} の一般式は (6.2) 式となる。

$$\tau_i^{\text{grav}} = - \sum_{j=i}^n m_j \tilde{\mathbf{g}}^T \frac{\partial^0 \mathbf{T}_{j,j}}{\partial \theta_i} \hat{\mathbf{s}}_j \quad (6.2)$$

ここで， $^j \hat{\mathbf{s}}_j$ は座標系 Σ_i の原点 O_i からリンク i の質量中心までのベクトルを座標系 Σ_i で表したものを意味する。 $\tilde{\mathbf{g}}$ は基準座標系 Σ_0 で表した重力加速度 $\tilde{\mathbf{g}} = [\tilde{g}_x \ \tilde{g}_y \ \tilde{g}_z \ 0]^T$ とする。また，(6.2) 式中の $\partial^0 \mathbf{T}_j / \partial \theta_i$ は (6.3) 式のように簡単に求めることができ，実際に偏微分したものを計算する必要はないことに注意されたい。

$$\frac{\partial^0 \mathbf{T}_j}{\partial \theta_i} = {}^0 \mathbf{T}_1 {}^1 \mathbf{T}_2 \cdots {}^{i-1} \mathbf{T}_i \Delta_i^i \mathbf{T}_{i+1} \cdots {}^{j-1} \mathbf{T}_j \quad (6.3)$$

したがって、(6.3) 式より (6.8), (6.9), (6.10) 式のように各関節変数で偏微分した同次変換行列が求められる。

$$\frac{\partial {}^0\mathbf{T}_1}{\partial \theta_1} = {}^0\mathbf{T}_1 \mathbf{\Delta}_2 {}^1\mathbf{T}_2 = \begin{bmatrix} -S_{\theta_1} & -C_{\theta_1} & 0 & 0 \\ C_{\theta_1} & -S_{\theta_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.8)$$

$$\frac{\partial {}^0\mathbf{T}_2}{\partial \theta_1} = {}^0\mathbf{T}_1 \mathbf{\Delta}_2 {}^1\mathbf{T}_2 = \begin{bmatrix} -S_{\theta_1+\theta_2} & -C_{\theta_1+\theta_2} & 0 & -l_1 S_{\theta_1} \\ C_{\theta_1+\theta_2} & -S_{\theta_1+\theta_2} & 0 & l_1 C_{\theta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.9)$$

$$\frac{\partial {}^0\mathbf{T}_2}{\partial \theta_2} = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \mathbf{\Delta}_2 = \begin{bmatrix} -S_{\theta_1+\theta_2} & -C_{\theta_1+\theta_2} & 0 & 0 \\ C_{\theta_1+\theta_2} & -S_{\theta_1+\theta_2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.10)$$

(6.2) 式に、(6.8), (6.9), (6.10) 式を代入することにより、各関節にかかる重力トルク τ^{grav} は (6.11), (6.12) 式のように求められる。

$$\begin{aligned} \tau_1^{\text{grav}} &= -m_1 \tilde{\mathbf{g}}^T \frac{\partial {}^0\mathbf{T}_{11}}{\partial \theta_1} \hat{\mathbf{s}}_1 - m_2 \tilde{\mathbf{g}}^T \frac{\partial {}^0\mathbf{T}_{22}}{\partial \theta_1} \hat{\mathbf{s}}_2 \\ &= m_1 g l_{g1} C_{\theta_1} + m_2 g l_2 C_{\theta_1+\theta_2} + m_2 g l_{g2} C_{\theta_1+\theta_2} \end{aligned} \quad (6.11)$$

$$\begin{aligned} \tau_2^{\text{grav}} &= -m_2 \tilde{\mathbf{g}}^T \frac{\partial {}^0\mathbf{T}_{22}}{\partial \theta_2} \hat{\mathbf{s}}_2 \\ &= m_2 g l_{g2} C_{\theta_1+\theta_2} \end{aligned} \quad (6.12)$$

このように、重力トルクに関しては、順運動学で用いる DH 法に必要な同次変換行列の組み合わせで求めることができる。すなわち、ロボットが組み替えられても、各分散知能からあらかじめ送信される同次変換行列と、リンク長 l_i 、質量中心までの距離 l_{gi} 、質量パラメータ m_i を更新するのみで制御システムは対応できる。

摩擦トルク

摩擦トルク τ_i^{fric} を (6.13) 式のようにモデル化する。

$$\begin{aligned} \tau_i^{\text{fric}} &= D_i \dot{\theta}_i^{\text{res}} + \tau_i^{\text{coul}} \cdot \text{sign}(\dot{\theta}_i^{\text{res}}) \\ \text{sign}(\dot{\theta}_i^{\text{res}}) &= 1 : \dot{\theta}_i^{\text{res}} \geq 0 \\ \text{sign}(\dot{\theta}_i^{\text{res}}) &= -1 : \dot{\theta}_i^{\text{res}} < 0 \end{aligned} \quad (6.13)$$

ここで、 D_i は粘性摩擦係数、 τ_i^{coul} はクーロン摩擦力である。 $\text{sign}(\dot{\theta}_i^{\text{res}})$ は速度の正負で 1 と -1 が離散的に切り替わる数であり、これによりクーロン摩擦を表現する。これらのパラメータは文献 [14] より、図 3.3 に示されるような外乱オブザーバを用いて同定する方法が提案されている。本研究では、文献 [14] より示された同定方法を用いて求めた D_i 、 τ_i^{coul} を用い、トルク反力推定オブザーバを構成する。

また、例外として図 6.1 において、 $0 \leq \tau_i^{\text{e,f,g}} < \tau_i^{\text{grav}} + \tau_i^{\text{fric}}$ である場合と、 $\tau_i^{\text{grav}} + \tau_i^{\text{fric}} \leq \tau_i^{\text{e,f,g}} < 0$ である場合は、重力、摩擦トルクの影響を取り除くときにトルク反力の符号が反転し、逆向きのトルク反力が推定されてしまう矛盾が生じる。この場合、摩擦トルクを (6.13) 式から (6.14) 式のように変更する。

$$\tau_i^{\text{fric}} = \tau_i^{\text{e,f,g}} - \tau_i^{\text{grav}} \quad (6.14)$$

(6.14) 式のように摩擦トルクを考えることにより、上記の条件下では $\tau_i^{\text{e,f,g}}$ と $\tau_i^{\text{fric}} + \tau_i^{\text{grav}}$ がつりあっていると考え、トルク反力推定オブザーバの出力 $\hat{\tau}_i^{\text{reac}} = 0$ と推定する。

減速機の効率 [17]

減速機の効率 η_i は、使用する減速機の速比、入力回転数、潤滑剤の温度、負荷トルクと定格トルクの比で決定できる数であり、(6.15) 式より計算される。

$$\eta_i = K_{e,i} \cdot \eta_{e,i} \quad (6.15)$$

ここで、 $K_{e,i}$ は効率補正係数、 $\eta_{e,i}$ は温度と入力回転数から決定される効率である。速比 50 の温度効率曲線を図 6.3 に、効率補正曲線を図 6.4 に示す。

例として、速比 50 の減速機 (CSF-8-50-2XH) の入力回転数 500 r/min、負荷トルク 0.90 Nm、潤滑剤の温度 10 °C の場合の減速機の効率 η_i を求める。温度効率は、図 6.3 のより、 $\eta_{e,i} = 77\%$ と読み取れる。トルク比 α_i は、減速機 (CSF-8-50-2XH) の定格トルクは 1.8 Nm であるので、 $\alpha_i = 0.90/1.8 = 0.5$ である。したがって、効率補正係数 $K_{e,i}$ は図 6.4 より $K_{e,i} = 0.90$ と読み取れる。以上から、減速機の効率 η_i は (6.15) 式より、 $\eta_i = 0.9 \cdot 77 = 69.3\%$ と求まる。

本研究では、使用する実験機器の条件を考慮して、潤滑剤の温度を 10 °C、入力回転数 500 r/min のときの温度効率特性 $\eta_{e,i} = 77\%$ をノミナル値として使用した。そして、効率補正係数 $K_{e,i}$ は、ロボットの制御中に逐次、以下の方法で算出した。

アクチュエータ i に与えるトルク参照入力 τ_i^{ref} に定格トルク 1.8 を除することでトルク比 α_i を算出する。そして、0.1 から 1.0 までの 10 点を等間隔に線形補間して表した効率補正曲線を使用し、算出したトルク比 α における効率補正係数 $K_{e,i}$ を求める。

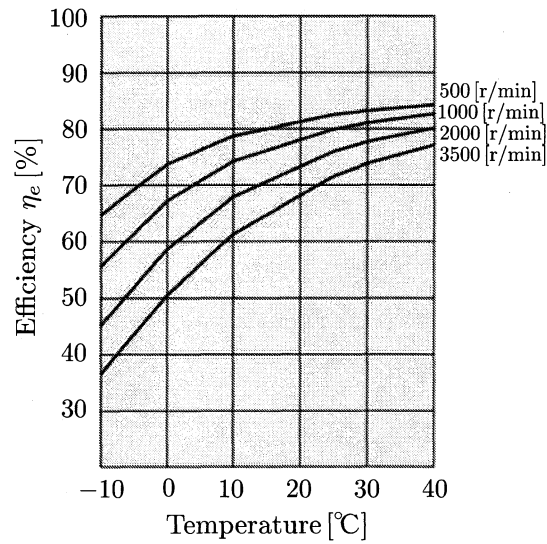


図 6.3: 減速機 (速比 50) の温度効率曲線

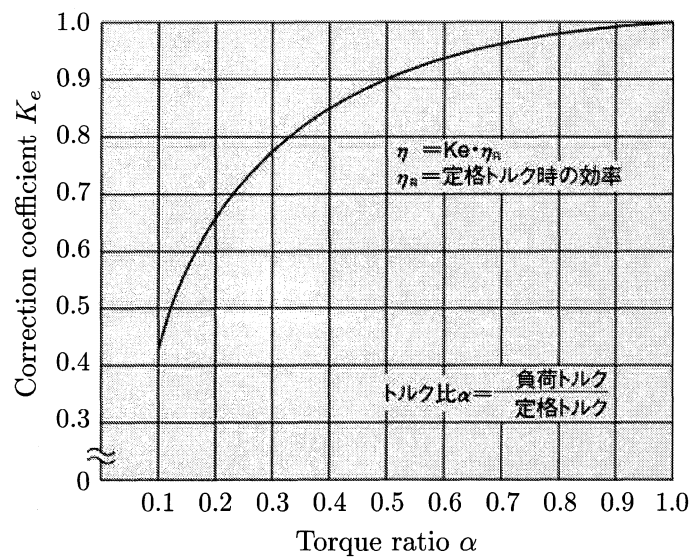


図 6.4: 減速機 (速比 50) の効率補正曲線

6.1.3 分散知能の拡張

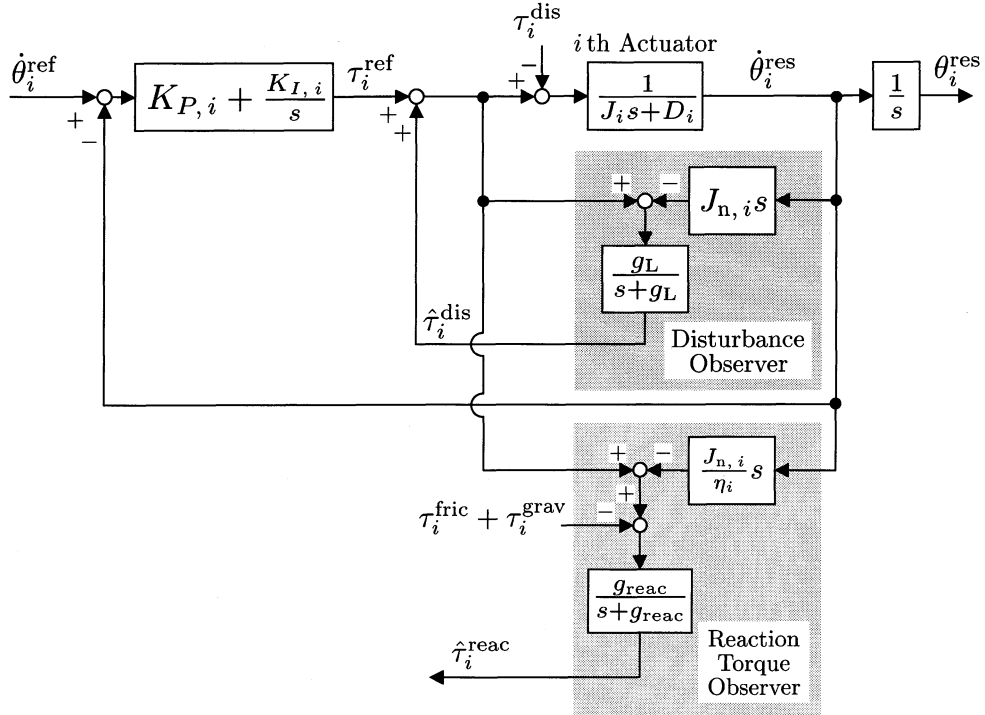


図 6.5: トルク反力推定器付き分散知能の制御システム

3.3 節の図 3.2 に示される分散知能に対して、新たにトルク反力推定オブザーバを付加し、図 6.5 のように拡張する。この制御システムより、各モジュールの関節におけるトルク反力が推定できる。

6.1.4 トルク反力推定の実験

トルク反力推定オブザーバの動作を確かめるため、図 6.6 に示される 3 関節ロボット (根元から 1 番目の関節は使用しない) を用いて簡単な実験を行った。文献 [14] の方法より得られたトルク反力推定オブザーバに必要なパラメータを表 6.1 に示す。

また、図 6.6 に示す実験機器のパラメータを表 6.2, 6.3 に示す。

各関節に対してそれぞれの角度を維持する位置制御中に、操作者が図 6.6 に示されるように手先に外乱を加える。そのときに、手先に取り付けられた力センサの検出値 $\mathbf{f}_r^{\text{res}}$ を (6.16) 式に示される仮想仕事の原理より、各トルク反力応答値 $\boldsymbol{\tau}^{\text{res}}$ へ変換する。

$$\boldsymbol{\tau}^{\text{res}} = \mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta}) \mathbf{f}_r^{\text{res}} \quad (6.16)$$

表 6.1: トルク反力推定オブザーバのパラメータ (1)

symbol	value
τ_1^{coul}	0.23850 Nm
τ_2^{coul}	0.26032 Nm
τ_3^{coul}	0.25942 Nm
D_1	0.16635 Ns/m
D_2	0.17428 Ns/m
D_3	0.13610 Ns/m

表 6.2: サーボモータ仕様 (1)

Servomotor Type	SGMMJ-A2BAB
Encoder	Incremental encoder 2048P/R
Rated Output	20 W
Rated Torque	0.0638 Nm
Rated Current	0.66 A
Instantaneous Peak Torque	0.191 Nm
Rated Speed	3000 r/min
Max. Rotation Speed	5000 r/min
Moment of Inertia	$5.48 \times 10^{-7} \text{ kg} \cdot \text{m}^2$

このトルク反力応答値 τ^{res} を理想の値とし、トルク反力推定値 $\hat{\tau}^{\text{reac}}$ と比較する。

実験結果を図 6.7 に示す。実験結果より、手先に外乱が加わり、トルク反力応答が値を持つとき、トルク反力推定値に関しても力が加わっていることから、力センサを用いずに力の情報が検出できていることがわかる。ただし、理想のトルク応答値 τ^{res} に比べて、トルク反力推定値 $\hat{\tau}^{\text{reac}}$ は若干小さな値で推定されている。この理由に、摩擦トルクなどの同定誤差や、静止摩擦トルク、トルクの大きさによって変化するクーロン摩擦をうまく考慮できていないことが考えられる。このような問題を解決するため、トルク反力の推定精度の向上を提案している文献 [8] などがある。しかし、高精度なパラメータの推定や複雑な摩擦モデルを使用すると、構造の変更に対して行わなければならない対応が複雑化する。そのため、本研究では現状での単純なモデル扱ったトルク反力の推定機構を用いて、力センサレス制御システムを構築していき、どれほどのタスクが行えるかを検討することを目的とする。

表 6.3: 減速機仕様 (1)

Reduction Gears Type	CSF-8-50-2XH-F-SP
Gear Ratio	50
Rated Torque (200 r/min)	1.8 Nm
Instantaneous Peak Torque	6.6 Nm
Peak Torque (Start and Stop)	3.3 Nm

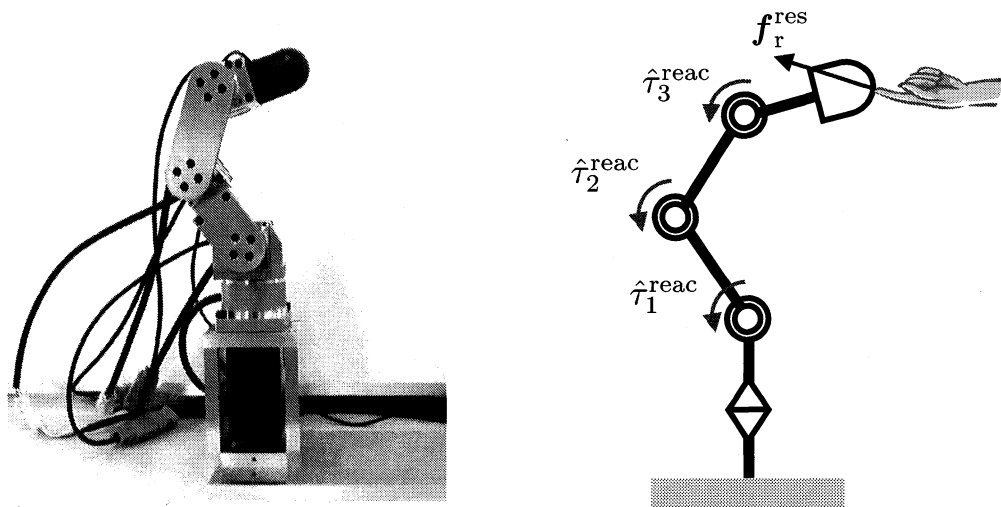


図 6.6: 実験機器 (1)

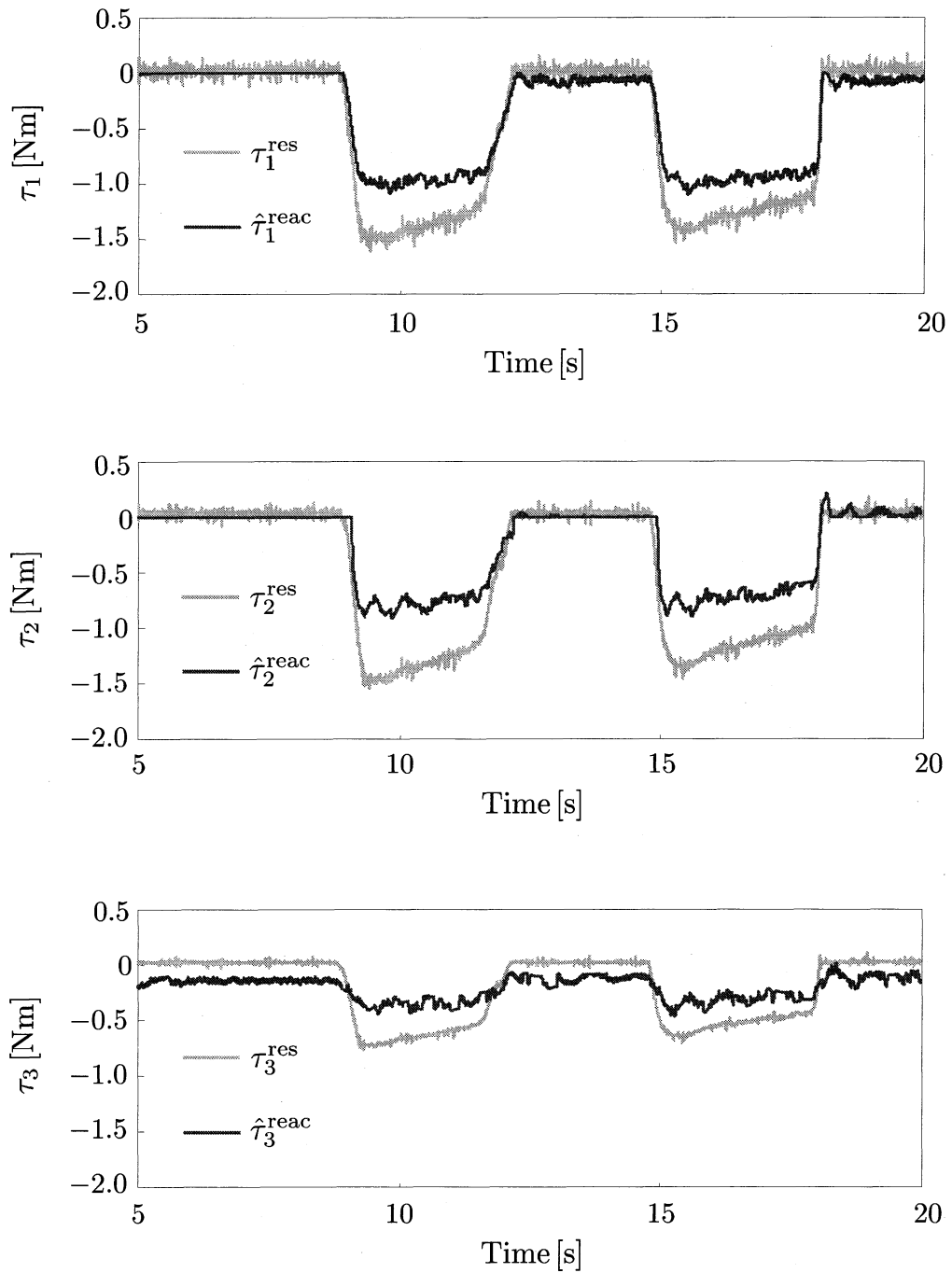


図 6.7: トルク反力推定の実験結果

6.2 座標変換の問題

前節では、トルク反力推定オブザーバにより力センサを用いずに推定トルク反力 $\hat{\tau}_i^{\text{reac}}$ が得られることを説明した。しかし、一般的にロボットの力制御は回転座標系で定義される関節空間ではなく、直交座標系で定義される作業空間で行われる。したがって、関節空間における推定トルク反力 $\hat{\tau}^{\text{reac}}$ から作業空間における手先の推定反力 $\hat{\mathbf{f}}_r^{\text{reac}}$ への座標変換が必要となる。通常、静力学における力の作業空間から関節空間への座標変換は (6.16) 式に示される仮想仕事の原理より定義される。

トルク反力推定値 $\hat{\tau}^{\text{reac}}$ から手先の推定反力 $\hat{\mathbf{f}}_r^{\text{reac}}$ への座標変換は (6.16) 式の逆変換に相当するので、(6.17) 式のように変換することができる。

$$\hat{\mathbf{f}}_r^{\text{reac}} = \{\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})\}^{-1} \hat{\tau}^{\text{reac}} \quad (6.17)$$

また、 $\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})$ が正方行列ではなく、逆行列が存在しないのであれば、擬似逆行列 $\{\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})\}^\dagger$ を用いて (6.18) 式で座標変換を行う。

$$\begin{aligned} \hat{\mathbf{f}}_r^{\text{reac}} &= \{\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})\}^\dagger \hat{\tau}^{\text{reac}} \\ \{\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})\}^\dagger &= \{\mathbf{J}_{\text{aco}}(\boldsymbol{\theta})\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})\}^{-1} \mathbf{J}_{\text{aco}}(\boldsymbol{\theta}) \end{aligned} \quad (6.18)$$

このように (6.17), (6.18) 式を用いれば手先の推定反力 $\hat{\mathbf{f}}_r^{\text{reac}}$ が求まり、 $\hat{\mathbf{f}}_r^{\text{reac}}$ をフィードバックする制御システムを組むことで作業空間の反力の力センサレス制御が可能となる。

しかしながら、 $\mathbf{J}_{\text{aco}}^T(\boldsymbol{\theta})$ の擬似逆行列の導出はロボットの構造 (関節数) の増加に伴い、計算負荷が過大となり、大幅な数式の変更が必要となる。したがって、構造可変型ロボットの制御には不向きであるといえる。

6.3 計算の省力化の提案

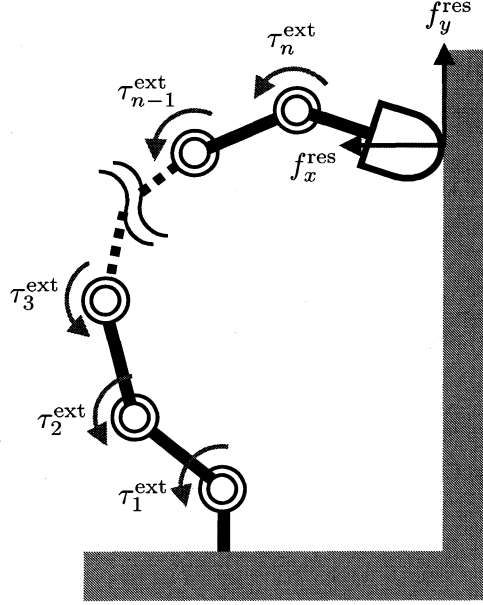


図 6.8: n 関節ロボット

本節では、前節で説明した問題を解決するため、容易な計算でトルク反力から手先反力を得る方法を考える。まず、仮想仕事の原理について考察する。

図 6.8 に示される 2 次元平面における n 関節のロボットを例にする。このとき、ヤコビ行列を (6.19) 式のように定義すると、仮想仕事の原理は (6.20) 式のような関係を表す。

$$J_{\text{aco}}(\theta) = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \dots & \frac{\partial x}{\partial \theta_n} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \dots & \frac{\partial y}{\partial \theta_n} \end{bmatrix} \quad (6.19)$$

$$\begin{aligned} \tau_1 &= \frac{\partial x}{\partial \theta_1} f_x + \frac{\partial y}{\partial \theta_1} f_y \\ \tau_2 &= \frac{\partial x}{\partial \theta_2} f_x + \frac{\partial y}{\partial \theta_2} f_y \\ &\vdots \\ \tau_n &= \frac{\partial x}{\partial \theta_n} f_x + \frac{\partial y}{\partial \theta_n} f_y \end{aligned} \quad (6.20)$$

(6.20) 式は手先の力と関節のトルクのつり合いを表す関係式であり、 n 関節のロボットであれば n 個のつり合いの式が成立する。前節で説明したヤコビの転置行列の擬似逆行列を用いる計算方法は、 n 個の力のつりあいを示す (6.20) 式をすべて満たす手先の

力 f_x , f_y が存在しない場合でも, (6.20) 式の 2 乗誤差の和を最小化する f_x , f_y の解が得られるものである。すなわち, トルク反力の推定を劣化させる動的な外乱である相互慣性力や遠心コリオリ力が働くような動作の速い運動下でも, その劣化を抑えるように計算することが期待できる。

しかし, 静的な条件下では, 理論上, (6.20) 式のすべてが成立する。したがって, (6.20) 式をすべて考慮する必要はなく, 手先の力 f_x , f_y の 2 次元の情報を得たいのであれば (6.20) 式のうち, 2 つの式を選択し, 連立方程式を解くだけで手先反力を求めることができる。例えば, 第 1, 2 関節のトルクと手先の力のつり合いの式から (6.21), (6.22) 式のように変形でき, f_x , f_y を求めることができる。

$$f_x = \frac{1}{\frac{\partial x}{\partial \theta_1} \frac{\partial y}{\partial \theta_2} - \frac{\partial y}{\partial \theta_1} \frac{\partial x}{\partial \theta_2}} \left(\frac{\partial y}{\partial \theta_2} \tau_1 - \frac{\partial y}{\partial \theta_1} \tau_2 \right) \quad (6.21)$$

$$f_y = \frac{1}{\frac{\partial x}{\partial \theta_1} \frac{\partial y}{\partial \theta_2} - \frac{\partial y}{\partial \theta_1} \frac{\partial x}{\partial \theta_2}} \left(-\frac{\partial x}{\partial \theta_2} \tau_1 + \frac{\partial x}{\partial \theta_1} \tau_2 \right) \quad (6.22)$$

このように手先反力を求めることで, 構造可変型ロボットの制御に関して以下の利点が生まれる。

- すべてのトルク反力を推定する必要がない。

ロボットの関節数に関係なく, 2 次元の手先反力を導出したいのであれば, 2 つのトルク反力を推定するのみでよい。トルク反力の推定は, ロボットの摩擦や重力トルクを計算する必要があるため, これが簡単化できることは大きな利点である。

- ヤコビ行列を導出しなくてよい。

(6.19) 式に示されるように n 軸のヤコビ行列の導出には $2n$ 個の構造に依存した勾配を求める必要がある。しかし, 提案手法では関節数に関係なく, 2 次元の手先反力を導出したいのであれば, x , y 軸それぞれ 2 つの勾配を求めるのみで手先反力が求められる。

以上の利点から, 本研究では, この静力学における計算方法の省力化方法を構造可変型ロボットに適したカセンサレス制御システムとして組み込んだものを提案する。

6.4 中央制御部の拡張

構造可変型ロボット単体の位置と力の制御を可能とするセンサレス制御システムを図 6.9 に示す。図 6.9 中の分散知能は図 6.5 に示されるトルク反力推定オブザーバが付加されたものである。この制御システムは, 位置と力の両方の制御器を持ち合わせて

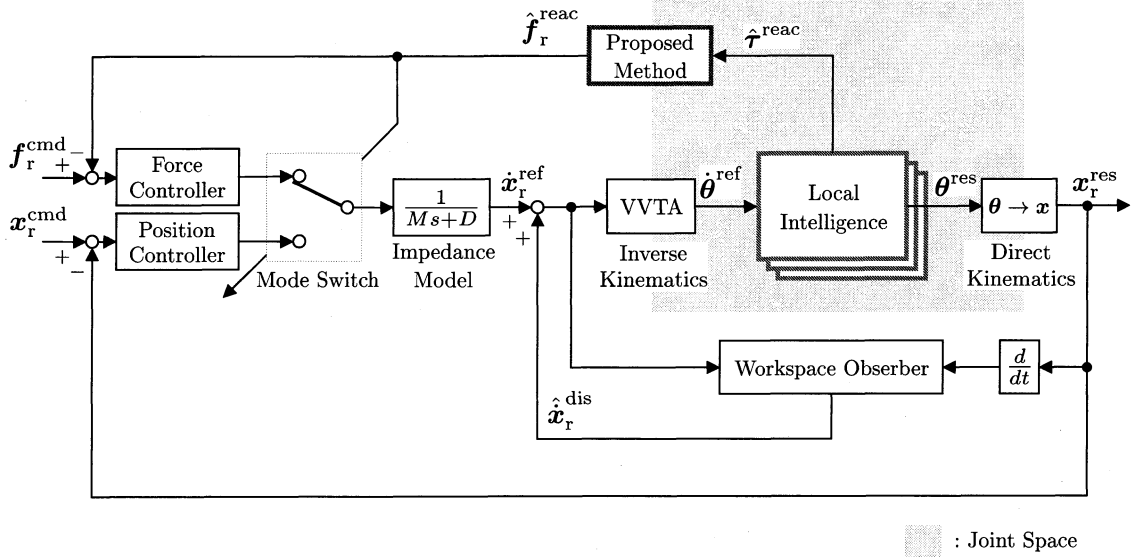


図 6.9: 拡張された中央制御部

おり、制御モードを各作業空間において Mode switch によって切り替えることによって、どちらの制御入力を反映させるかを決定する。切り替え条件は、前節の方法で計算された手先反力推定値 \hat{f}_r^{reac} があらかじめ定めた閾値を越える場合に、その座標系で力制御モードに移行し、そうでない場合は位置制御モードとなる。この制御システムにより、ある方向の反力を制御して壁を押しながら、別の方向で位置制御するなどのない動作や、はめ込み動作が可能となる。

6.5 機能性に基づく制御システムのセンサレス化

機能空間の情報 \bullet_f は、作業空間の情報 \bullet_r から座標変換されて求められる情報である。したがって、関節空間の情報であるトルク反力推定値 $\hat{\tau}^{reac}$ を導出するだけでは、その推定値を機能空間の情報に座標変換できないため、機能性に基づく制御システムを構築することはできなかった。

しかし、6.3 節で提案した計算方法により、簡単に作業空間における手先反力推定値 \hat{f}_r^{reac} を計算することができる。これにより、(4.5) 式によって、機能空間における手先反力推定値 \hat{f}_f^{reac} を計算することが可能となる。そして、 \hat{f}_{fi}^{reac} をフィードバックする機能性に基づく制御器を構成すれば、図 5.1 に示される構造可変型ロボットに適した機能性に基づく分散制御システムをカセンサレス制御システムとして構成することができる。

第7章 シミュレーションと実験

本章では、提案する構造可変型ロボットに適した制御システムの有効性を確認するために以下の実験とシミュレーションを行う。

実験 1

第5章で提案した協調制御システムの有効性を構造の変化を模擬した実機システムが協調把持を行う実験により検証する。ここでは、提案する制御システムが構造の変更に容易に対応でき、かつ物体への接近から各ロボットの反力を制御する協調把持へとタスクの内容が切り替わるものを行わせることで多様な作業に対応できることを検証する。

実験 2

第5章で提案した簡易な冗長指令が生成可能な制御システムの有効性を構造の変化を模擬した実機システムが協調把持を行う実験により検証する。実験1と同様のタスクを手先に行わせながら、手先を水平に保つという姿勢の制御を冗長指令として加えることで、提案する制御システムの有効性を検証する。

実験 3 (シミュレーション)

第6章で提案した力センサレス制御システムの有効性をシミュレーション上での実験により確認する。ロボット単体が壁を押し込む作業を想定し、力センサレスで壁からの反力を制御する。

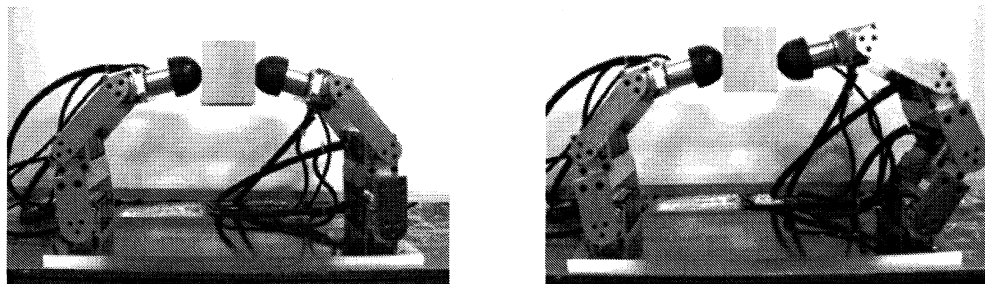
実験 4

第6章で提案した力センサレス制御システムの有効性をマニピュレータを用いた実験により確認する。実験3と同様に、ロボット単体が壁を押し込む作業を想定し、力センサレスで壁からの反力を制御する。

実験 5

6.5節で提案した力センサレスな機能性に基づく協調制御システムをマニピュレータを用いた実験により確認する。

7.1 実験1 (構造の組み替えとタスク変化の伴う協調作業)



(a) 3－3 関節ロボット

(b) 3－4 関節ロボット

図 7.1: 実験機器 (2)

表 7.1: サーボモータ仕様 (2)

Servomotor Type	SGMM-A2C312
Encoder	Incremental encoder 2048P/R
Rated Output	20 W
Rated Torque	0.0637 Nm
Rated Current	1.3 A
Instantaneous Peak Torque	0.191 Nm
Rated Speed	3000 r/min
Max. Rotation Speed	5000 r/min
Moment of Inertia	$5.48 \times 10^{-7} \text{ kg} \cdot \text{m}^2$

表 7.2: 減速機仕様 (2)

Reduction Gears Type	CSF-5-100-2XH-F-SP
Gear Ratio	100
Rated Torque (200 r/min)	0.6 Nm
Instantaneous Peak Torque	2.7 Nm
Peak Torque (Start and Stop)	1.4 Nm

5.2節で提案した図5.1に示される制御システムの有効性を確認するため、2次元平面における2体のロボットによる協調把持動作を図7.1に示す機器を用いて実験を行った。各関節に用いられるサーボモータ、減速機の仕様を表7.1、7.2に示す。図6.6の実験機器とはパラメータが若干異なっていることに注意されたい。

これらの実験機器は提案する制御システムの有効性を確認するために構造可変型ロボットを模擬したものであり、実際には第2章で述べられた非接触伝送は用いられていない。各ロボットの手先には力センサが取り付けられており、手先に加わる反力を検出することができる。この一般的なロボットアームを組み替えることで構造変化を模擬し、提案する制御システムの有効性を検証する。

この実験では、物体への接近、物体に接触してからの把持力制御という2つのタスクを想定し、協調把持中は把持物体を並進運動させる。そのタスクを2体の3軸ロボット(case1)と、3軸と4軸ロボット(case2)によって行う。

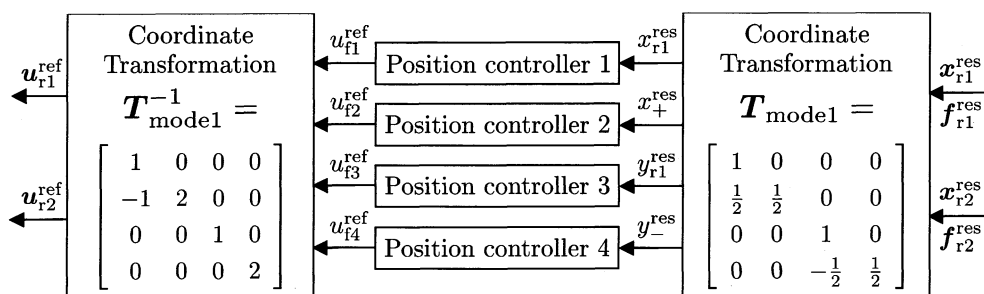
case1とcase2において、図5.1で示される制御システムの変更が必要な部分は、順運動学と逆運動学のブロックと、備え付けられる分散知能の数のみである。順運動学は簡単な加算や乗算の変更でありDH法を用いているので対応は簡単である。また、逆運動学も構造可変型ロボットに適した逆運動学であるVVTAを採用しているため、容易に変更できる。分散知能に関しても、単純な速度制御システムをもつ演算器を新たに付加するのみである。以上から、簡単な制御システムの更新で、case1からcase2への構造の変更に対応することができる。そして、ロボットの手先が行う作業を制御する中央制御部の制御器1, 2, 3, 4は、構造に依存しないため、case1, 2に関係なく設計することが可能である。

物体への接近動作と接触してからの協調把持という2つのタスクを行う機能の組み合わせを説明する。ロボットが物体に接近しているときをmode1、物体に接触し、各ロボットに取り付けられた力センサが反力を検出しているときをmode2とし、各modeで機能の組み合わせを変更して作業の変更に対応する。ここで機能座標系として使用する和の座標系と差の座標系を(7.1), (7.2)式のように定める。

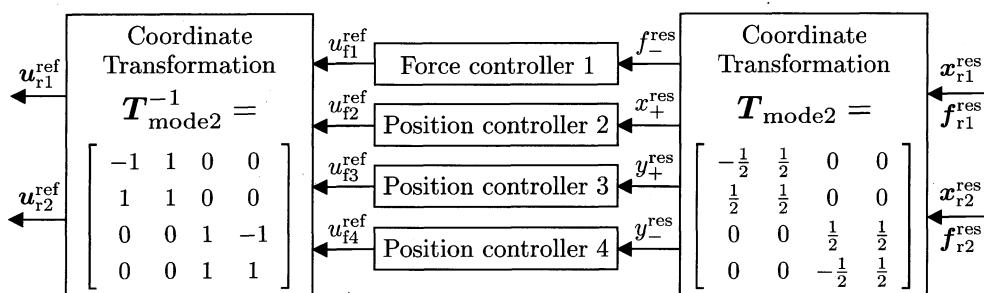
$$x_+ = \frac{1}{2}(x_{r1} + x_{r2}) \quad (7.1)$$

$$x_- = \frac{1}{2}(x_{r2} - x_{r1}) \quad (7.2)$$

下添え字 \bullet_+ は和の座標系を表し、 \bullet_- は差の座標系を表す。また、 y, u, f に対しても同様に和と差の座標系が定義される。 x_+ は定義より、各ロボットの位置の平均値を与えるため、mode2の把持物体の中心位置の制御などに利用でき、差の座標系は把持力



(a) mode 1



(b) mode 2

図 7.2: 各モードの機能の組み合わせ

の大きさ f_{-} の制御などに利用できる。これらの座標系を利用した各 mode の機能に基づく制御器と変換行列 \mathbf{T} の構成例を図 7.2 に示し、それらの詳細について説明する。

mode 1 では物体へ接近する軌道を制御するため、図 5.4 (a) に示される位置制御器を図 5.1 の制御器 1, 2, 3, 4 に与える。 $\mathbf{T}_{\text{mode1}}$ の 1, 3 行目の座標系に対応する位置制御器には、把持物体に接近するための位置指令 $\mathbf{x}_{r1}^{\text{cmd}} = [x_{r1}^{\text{cmd}} \ y_{r1}^{\text{cmd}}]^T$ が加えられ、ロボット 1 の手先の位置応答 $\mathbf{x}_{r1}^{\text{res}} = [x_{r1}^{\text{res}} \ y_{r1}^{\text{res}}]^T$ を制御する。2 行目の和の座標系に対応する位置制御器は $x_{r1}^{\text{res}} = -x_{r2}^{\text{res}}$ を達成し、4 行目の差の座標系に対応する位置制御器は、 $y_{r1}^{\text{res}} = y_{r2}^{\text{res}}$ を達成する。すなわち、これらの機能を組み合わせた全体の動作は、 y 軸に対して対称に、ロボット 1 とロボット 2 が物体に接近する。

mode 2 では把持力を制御するため、図 5.4 (b) に示される力制御器を図 5.1 中の制御器 1 へ与え、ロボットの手先位置や物体の中心位置を制御するため、図 5.4 (a) に示される位置制御器を、図 5.1 中の制御器 2, 3, 4 へ与える。mode 2 では、 $\mathbf{T}_{\text{mode2}}$ の 1 行目の座標系に対応する把持力制御器が差の座標系における反力 f_{-}^{res} の大きさを制御する。2, 3 行目の和の座標系に対応する位置制御器は、物体の中心位置 $\mathbf{x}_{+}^{\text{res}} = [x_{+}^{\text{res}} \ y_{+}^{\text{res}}]^T$ を

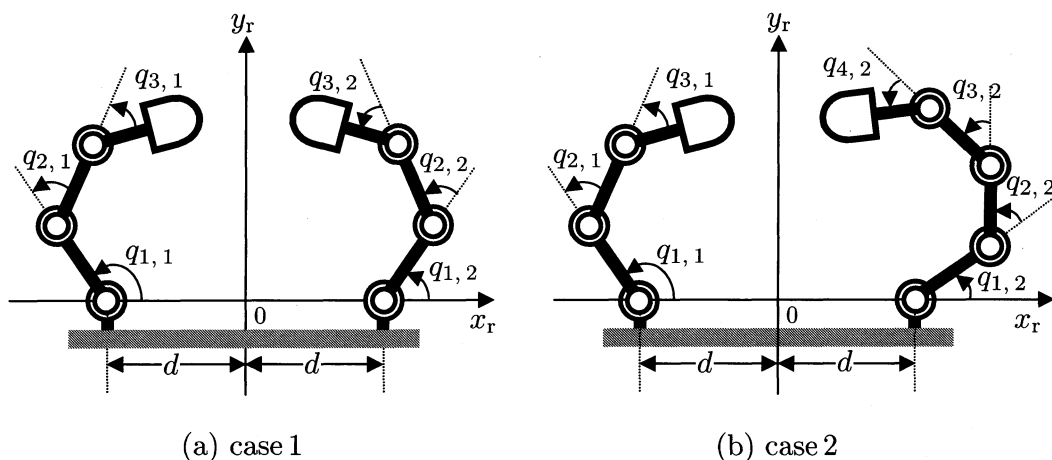


図 7.3: 実験機器の座標系

表 7.3: 実験パラメータ

symbol	description	value
k_p	propotional gain for position controller	900 N/m
k_d	differential gain for position controller	50 Ns/m
k_f	force gain for force controller	0.025
f_{-}^{cmd}	force command in differential coordinate	3.0 N
M	impedance parameter for mass	1 kg
D	impedance parameter for damping	50 Ns/m
τ_v	equivalent time constant for CDM	0.015 s
g_L	cut-off frequency for disturbance observer	30 rad/s
g_c	cut-off frequency for workspace observer	25 rad/s

制御する。また、この制御器に実験開始から 12 秒後に x 軸方向のみに正弦波状の並進指令を加え、物体の中心位置を並進運動させる。4 行目の差の座標系に対応する位置制御器は、mode 1 と同様の機能で、 $y_{r1}^{\text{res}} = y_{r2}^{\text{res}}$ を達成する。すなわち、これらの機能を組み合わせた全体の動作は、把持力、接触点を一定に保ちながら、物体を x 軸方向に並進させるものとなる。

実験で用いる座標系を図 7.3 のように設定し、case 1, 2 で共通する実験パラメータを表 7.3 に示す。各 case で用いる実験機器のリンク長と、各ロボットの初期角度を表 7.4 に示す。ここで、 $L_{i,j}$ はロボット j の第 i リンクの長さを表している。各ロボットの根元から y 軸までの距離 d は 0.117 m とした。各機能に基づく制御器のパラメータ

表 7.4: 各 case の各モジュールの初期角度

case 1				case 2			
link length [m]		angle [deg]		link length [m]		angle [deg]	
$L_{1,1}$	0.06	$q_{1,1}$	105	$L_{1,1}$	0.06	$q_{1,1}$	105
$L_{2,1}$	0.06	$q_{2,1}$	-30	$L_{2,1}$	0.06	$q_{2,1}$	-30
$L_{3,1}$	0.0575	$q_{3,1}$	-30	$L_{3,1}$	0.0575	$q_{3,1}$	-30
$L_{1,2}$	0.06	$q_{1,2}$	75	$L_{1,2}$	0.06	$q_{1,2}$	45
$L_{2,2}$	0.06	$q_{2,2}$	30	$L_{2,2}$	0.06	$q_{2,2}$	45
$L_{3,2}$	0.0575	$q_{3,2}$	30	$L_{3,2}$	0.06	$q_{3,2}$	45
—	—	—	—	$L_{4,2}$	0.0575	$q_{4,2}$	45

k_p , k_d , k_f の値は, 実験で行う作業が可能な応答特性が得られるように調整したものである。インピーダンスパラメータ M , D は, 以下の指針に従い設定した。まず M はロボット全体の質量に近い値として $M = 1 \text{ kg}$ とした。 D は制御入力から手先速度までの特性が実現可能な時定数 $\tau \equiv M/D = 0.02 \text{ s}$ を持つように $D = 50 \text{ Ns/m}$ とした。分散知能の外乱オブザーバで用いる慣性のノミナル値は, 実際の慣性値とモータのギア比を考慮し, 3 軸のロボットの場合は根元のモジュールから $J_{n,1} = 0.006 \text{ kgm}^2$, $J_{n,2} = 0.004 \text{ kgm}^2$, $J_{n,3} = 0.002 \text{ kgm}^2$ と設定し, 4 軸のロボットの場合は根元のモジュールから $J_{n,1} = 0.008 \text{ kgm}^2$, $J_{n,2} = 0.006 \text{ kgm}^2$, $J_{n,3} = 0.004 \text{ kgm}^2$, $J_{n,4} = 0.002 \text{ kgm}^2$ と設定した。

実験結果を図 7.4 に示す。位置応答結果より, case 1, 2 のどちらの場合においても mode 1 における接近指令 $\mathbf{x}_{r1}^{\text{cmd}}$ に対してロボット 1 の手先位置応答 $\mathbf{x}_{r1}^{\text{res}}$ はよく追従している。そして, mode 2 に関しても実験開始から 12 秒後, 物体の中心位置を並進運動させるための指令 x_+^{cmd} に応答 x_+^{res} はほぼ一致していることがわかる。また, 反力応答も case 1, 2 ともにおおむね指令値どおりの応答を示している。そして, case 1, 2 の協調把持動作の両者の応答結果を比較すると, 追従性能が大きく変化していないことがわかる。このことから提案する制御システムは, 構造変化に対して中央制御部の制御器パラメータ等の再調整を行うことなく, 自律的な対応が可能であることが示された。また, 物体への接近から把持力の制御というタスク内容の変更に対しても, 中央制御部の制御器を組み替えることで実現することができた。以上から, 提案する制御システムは, タスクの変更や構造の変更に対して柔軟に対応可能であることがわかる。

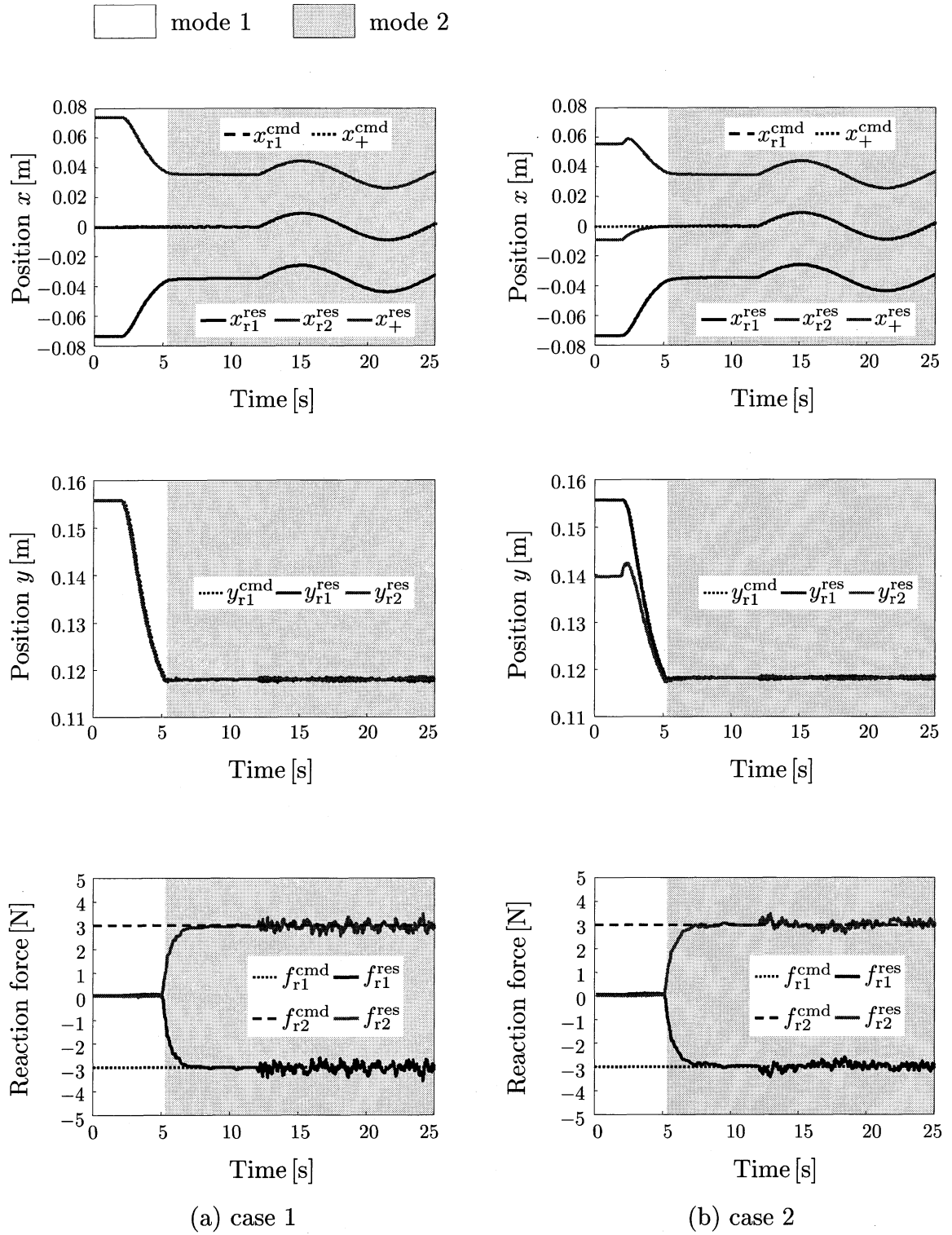


図 7.4: 実験 1 の結果

7.2 実験2 (冗長指令を付加した協調作業)

5.4節で提案した図5.8で示される制御システムの有効性を確認するため、前節と同様の構造可変型ロボットを模擬した実験機器を用いて実験を行った。この実験では、手先の作業に関しては物体への接近 (x_{r1}^{res} , x_{r2}^{res} の制御), 接触してからの把持力制御 (f_{r1}^{res} , f_{r2}^{res} の制御) という2つの作業を想定する。使用する機能の組み合わせは前節と同様の図7.2示される組み合わせとした。

そして、これらの手先の作業中に、図7.5に示される手先のリンクの角度を水平に保つ ($\theta_{tip1}^{cmd} = 0\text{rad}$, $\theta_{tip2}^{cmd} = \pi\text{rad}$) 姿勢の冗長制御を加える。評価関数を (7.3), (7.4) 式のように、指令と応答との2乗誤差を評価する関数に設定した。この評価関数を小さくする姿勢制御を行うように、(5.7) 式の比例ゲイン $k_{const} = -0.10$ と負の値に設定する。

$$V(\theta_{tip1}) = (0 - \theta_{tip1})^2 \quad (7.3)$$

$$V(\theta_{tip2}) = (\pi - \theta_{tip2})^2 \quad (7.4)$$

この実験は、構造の組み替えに対して提案手法が有効であることを示すため、3-3関節のロボット (case 1) と3-4関節のロボット (case 2) を用いて実験を行った。そして、各 case で冗長指令が有効に働いていることを示すため、冗長指令を付加する場合と付加しない場合の実験結果を示し、比較する。

得られた case 1, 2 の実験結果をそれぞれ図7.6, 7.7に示す。図7.6, 7.7の結果より、手先の位置、反力応答はほぼ同等の結果が得られながら、手先のリンク角 θ_{tip1}^{res} , θ_{tip2}^{res} は、冗長指令が加えられた場合の応答が水平 ($0, \pi\text{rad}$) に近い位置で保たれていることがわかる。すなわち、手先のタスクに影響なくロボットの姿勢が制御された。また、これらの結果は case 1, 2 を比較してもほぼ同等の結果が得られているため、この制御系は構造の組み替えに関しても適切に対応できたことがわかる。以上から、提案手法の有効性が確認できた。

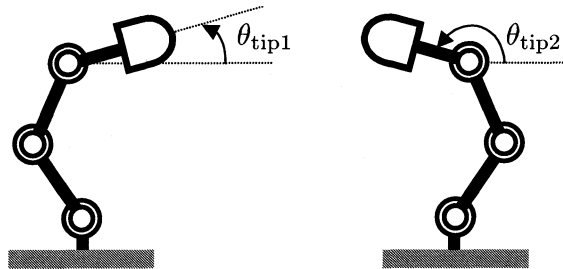


図 7.5: ロボットの手先のリンク角度

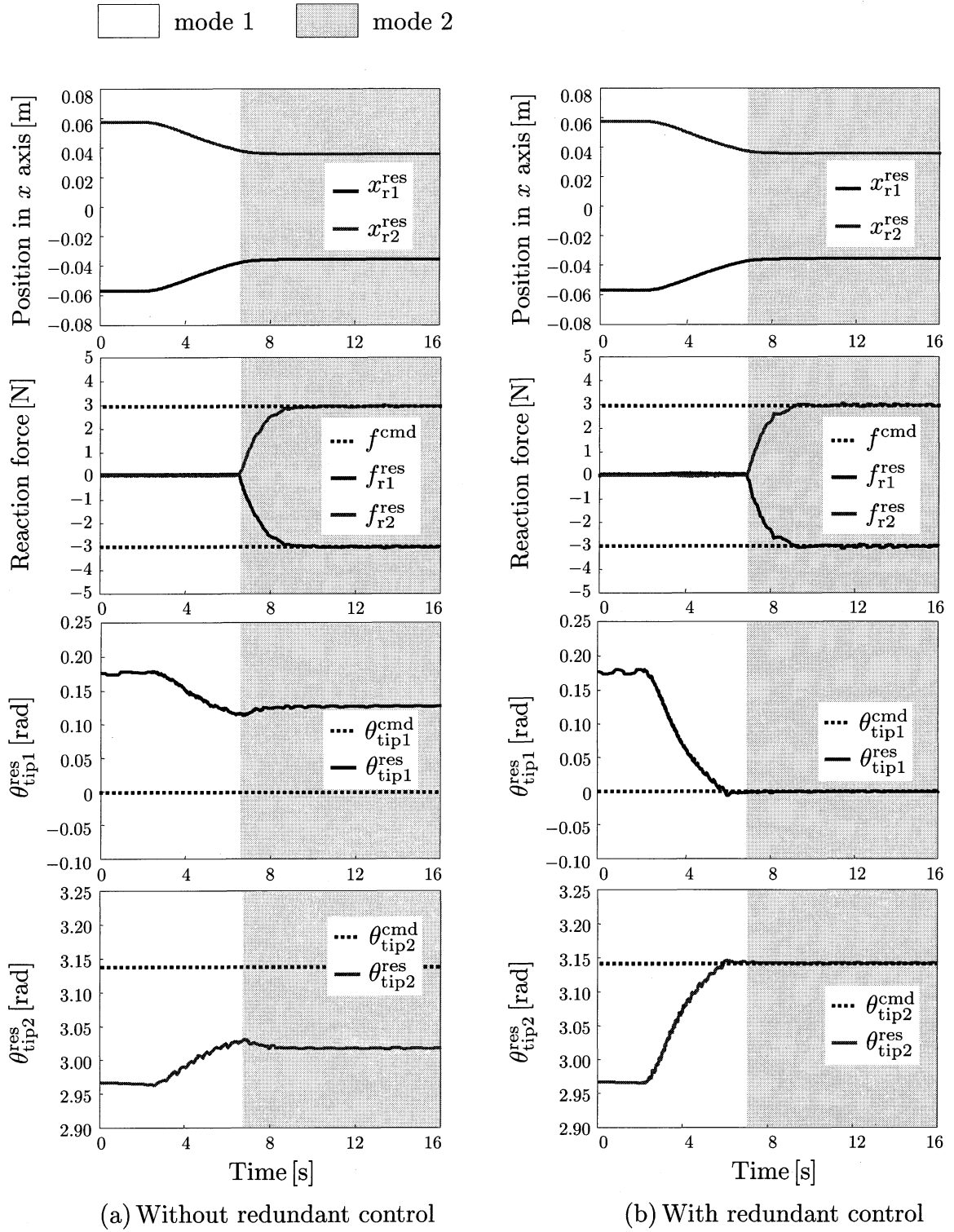


図 7.6: 実験 2 の結果 (case 1)

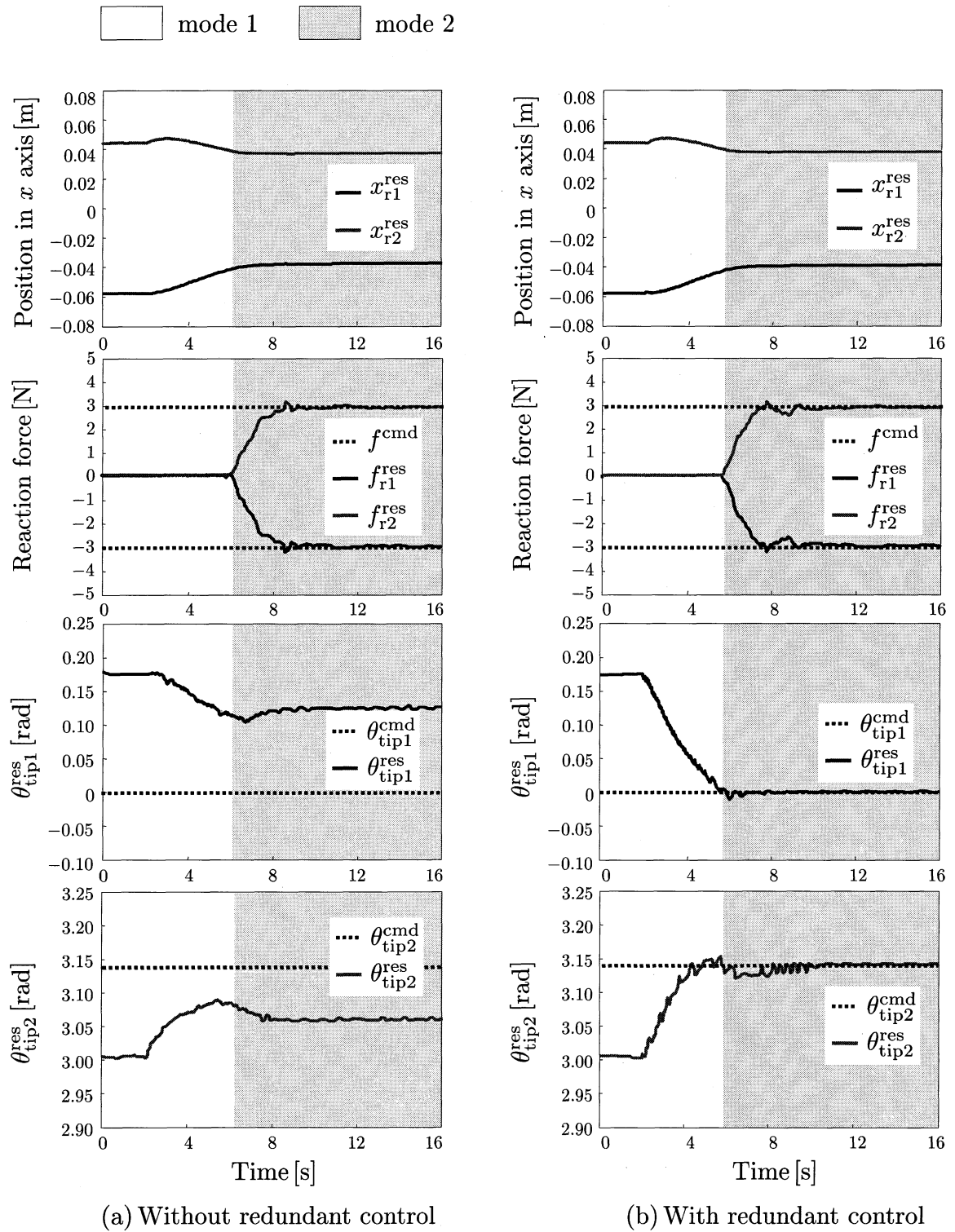


図 7.7: 実験 2 の結果 (case 2)

7.3 実験3 (カセンサレス制御システムのシミュレーション)

6.4節で提案した図6.9に示される制御システムの有効性を確認するため、シミュレーション上で実験する。シミュレーション上では図7.8に示される鉛直型3関節ロボットが壁に近づき、壁に接触した後に壁からの反力を制御するタスクを想定する。このタスク中に実際に壁から加わる反力 $f_{r,x}^{\text{res}}$ と反力推定値 $\hat{f}_{r,x}^{\text{reac}}$ を比較し、提案手法の有効性を検証する。なお、今回のシミュレーションでは、図7.8における $\hat{\tau}_1^{\text{reac}}$, $\hat{\tau}_2^{\text{reac}}$ のみをトルク反力推定オブザーバにより推定し、 $\hat{\tau}_3^{\text{reac}}$ の値は用いずに (6.21) 式より手先の反力推定値 $\hat{f}_{r,x}^{\text{reac}}$ を求めた。

ロボットは壁から図7.8のように0.07 mだけ離れているとし、壁はバネ $K = 70000 \text{ N/m}$ とダンパ $D = 200 \text{ Ns/m}$ で表されるものとしてモデル化した。位置制御器は比例ゲイン k_p 、微分ゲイン k_d とするPD制御器、力制御器は比例ゲイン k_f とするP制御器とした。

シミュレーションパラメータを表7.5に示し、シミュレーション結果を図7.9, 7.10に示す。

図7.9の結果より、壁に接近する姿勢の変化に関して、良好な結果を示している。図7.10(a), (b)の位置応答結果より x , y 軸方向のどちらの応答も指令値によく追従している。なお、約8秒から x 軸方向の位置応答が指令値と若干ずれていることに関して

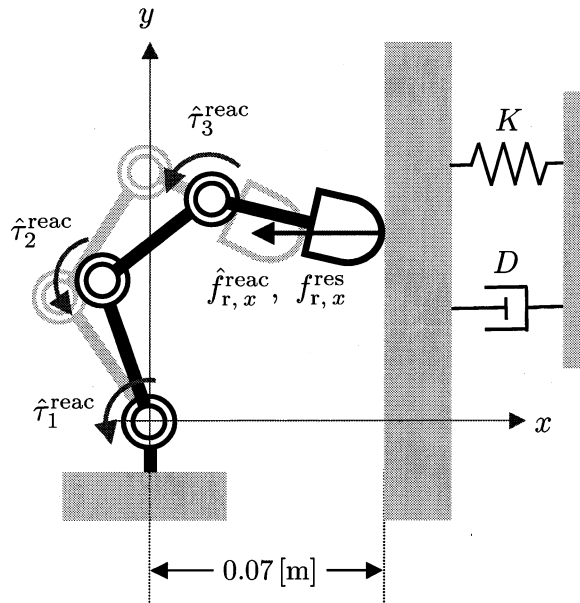


図 7.8: 3 関節ロボットの壁押し作業

表 7.5: シミュレーションパラメータ

symbol	description	value
k_p	propotional gain for position controller	1800 N/m
k_d	differential gain for position controller	200 Ns/m
k_f	force gain for force controller	0.050
f_r^{cmd}	force command	5.0 N
M	impedance parameter for mass	1 kg
D	impedance parameter for damping	50 Ns/m
τ_v	equivalent time constant for CDM	0.015 s
g_L	cut-off frequency for disturbance observer	30 rad/s
g_c	cut-off frequency for workspace observer	10 rad/s
g_{reac}	cut-off frequency for reaction torque observer	50 rad/s

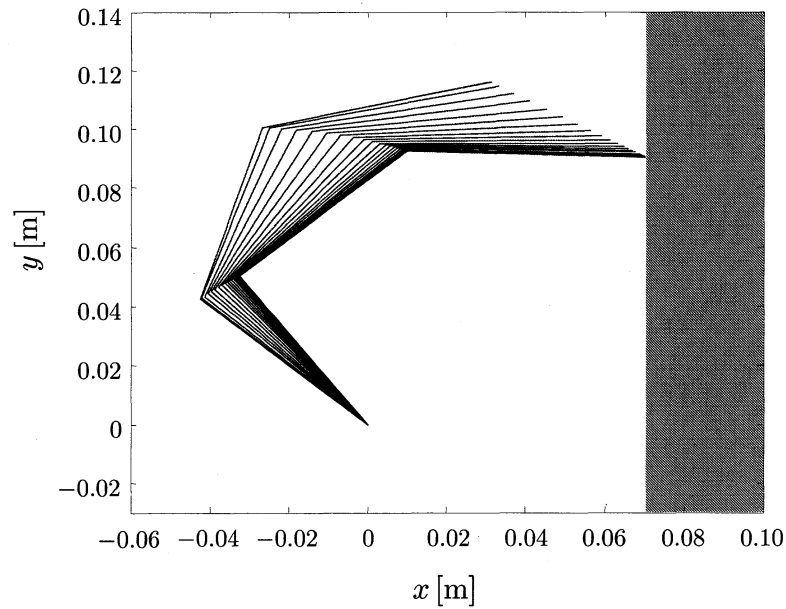


図 7.9: 姿勢の変化

は、制御途中に位置制御モードから力制御モードに移行しており、力制御モード中に x 軸方向の位置は制御されていないためである。それに対して、 y 軸方向では常に位置制御モードで動作するため、指令値と応答値がずれることなく追従している。

図 7.10 (c), (d) の反力応答結果について 6.3 節で提案した方法により計算された推定反力 $\hat{f}_{r,x}^{\text{reac}}$ をフィードバックすることで、実際の壁からの反力 $f_{r,x}^{\text{res}}$ を指令値 $f_{r,x}^{\text{cmd}} = 5 \text{ N}$

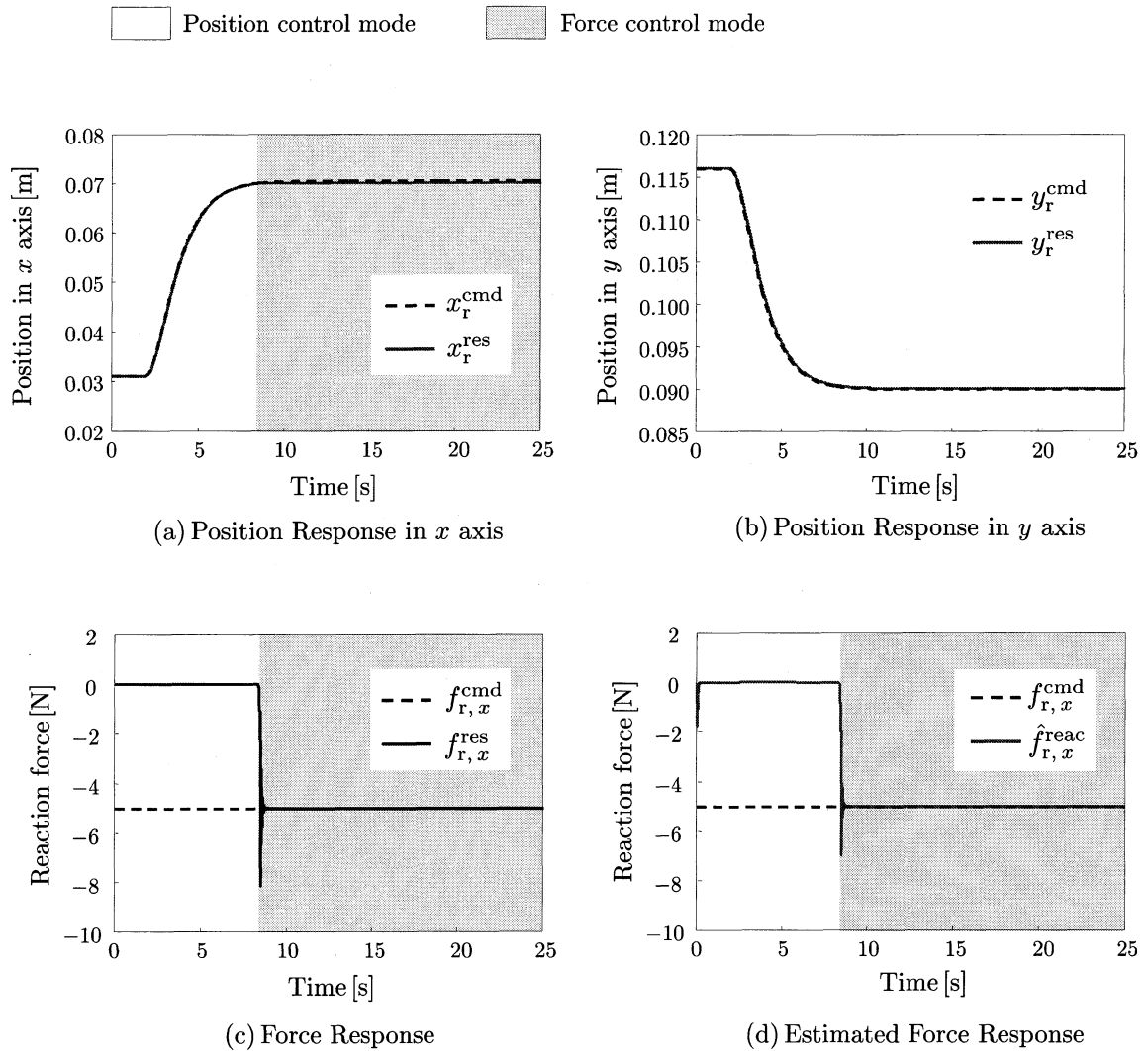


図 7.10: 実験 3 (シミュレーション) の結果

に制御できている。壁に接触する瞬間に、推定反力 $\hat{f}_{r,x}^{\text{reac}}$ と実際の壁からの反力 $f_{r,x}^{\text{res}}$ に誤差があるのは、推定反力 $\hat{f}_{r,x}^{\text{reac}}$ がトルク反力推定オブザーバのカットオフ周波数 g_{reac} 内でしか推定されないことと、座標変換に用いる仮想仕事の原理が静的な条件下でしか成立しないためであると考えられる。

以上から、本研究で提案する力センサレス制御システムは、静的な条件下において、有効であることがシミュレーションにより示されたといえる。

7.4 実験 4 (力センサレス制御システムの実機実験)

6.4 節で提案した図 6.9 に示される制御システムの有効性を確認するため、図 6.6 に示す鉛直型 3 関節ロボットを用いて実験した。

想定するタスクは、前節と同様、壁への接近から接触した後、壁からの反力を制御し、実際に壁から加わる反力 $f_{r,x}^{\text{res}}$ と反力推定値 $\hat{f}_{r,x}^{\text{reac}}$ を比較し、提案手法の有効性を検証する。実験環境、実験パラメータは、前節と同様に図 7.8、表 7.5 のように設定した。

実験結果を図 7.11 に示す。図 7.11 (a), (b) の位置応答結果は前節と同様よく追従していることがわかる。図 7.11 (d) の推定反力 $\hat{f}_{r,x}^{\text{reac}}$ の制御結果は、指令値 $f_{r,x}^{\text{cmd}} = 5\text{ N}$ によく追従しており、力制御に関しても良好な結果が得られたといえる。しかし、図 7.10 (c) より、推定反力 $\hat{f}_{r,x}^{\text{reac}}$ を 5 N に制御できても、実際の反力 $f_{r,x}^{\text{res}}$ は正確に 5 N に制御できてはいないことがわかる。これは 6.1.4 節で述べたように、トルク反力が正確に推定できておらず、その誤差の残る推定値より推定反力 $\hat{f}_{r,x}^{\text{reac}}$ を求めるため、当然ながら実際の手先反力からはずれてしまう。

しかし、一般に精度の高い力制御を要求するタスクは研磨作業などに限られる。それに対して、はめあい作業や簡単な把持などのタスクに関しては、ロボットが力の加減を調整できる程度の力制御を行うことができれば可能となる。このことから、本研究で提案する力センサレス制御システムによって、はめあい作業や簡単な把持などのタスクを実現することは可能であると考えられる。次節では、力センサレス制御で簡単な把持を実験する。

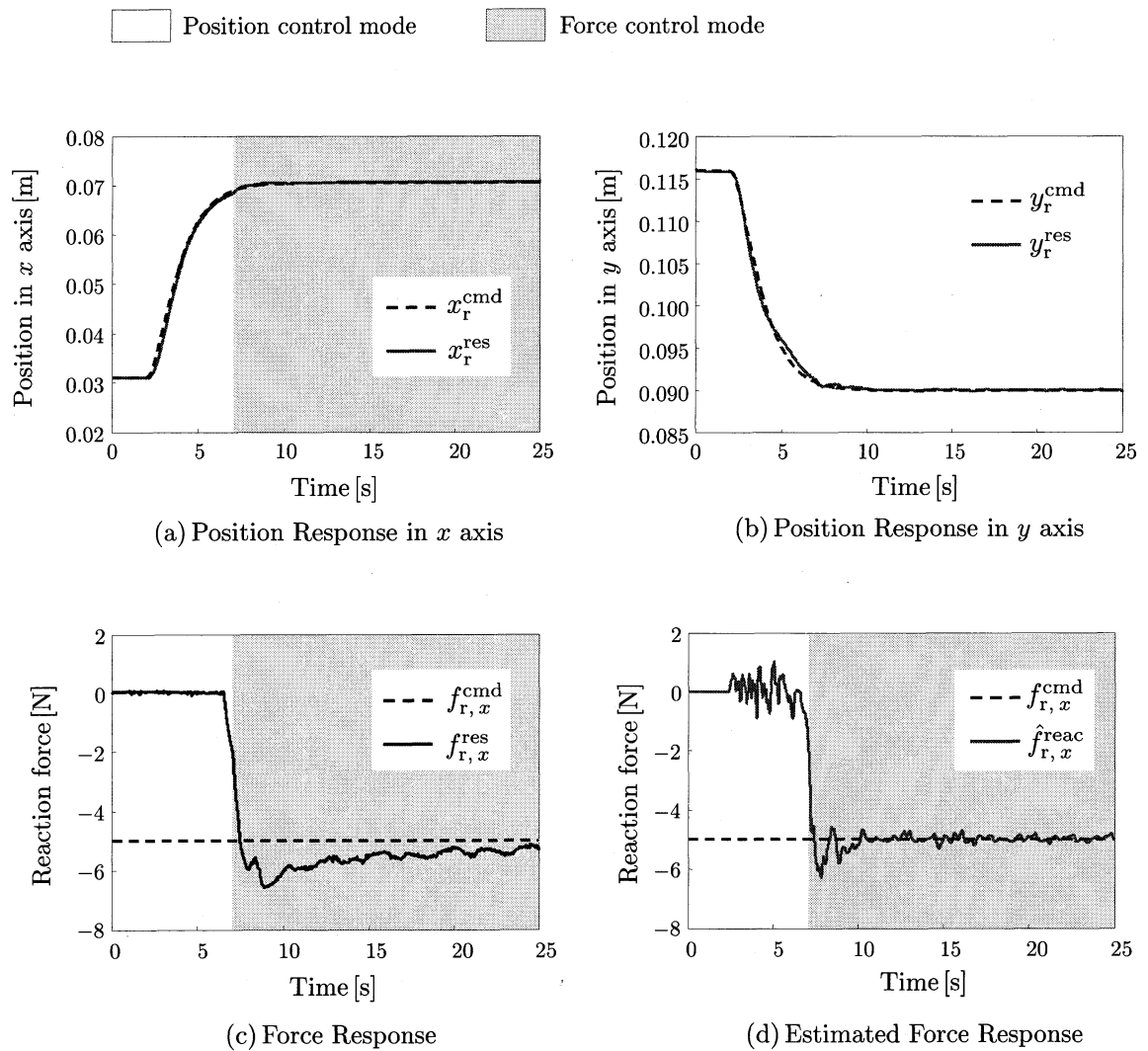


図 7.11: 実験 4 の結果

7.5 実験5 (協調把持の力センサレス制御)

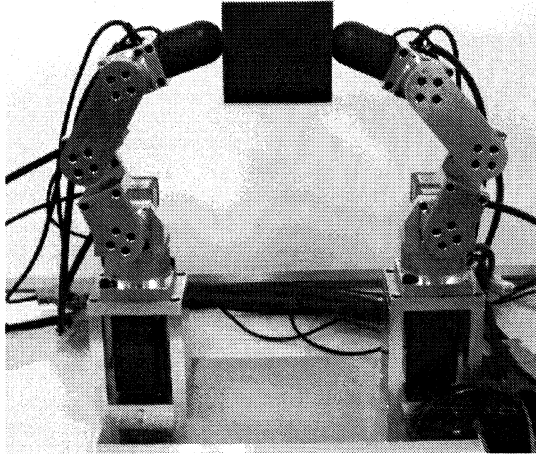


図 7.12: 実験機器 (3)

6.5 節で述べたように、図 5.1 に示される構造可変型ロボットに適した機能性に基づく分散制御システムを力センサレス制御システムとして構成することができる。力センサレス化した機能性に基づく分散制御システムの有効性を示すため、図 7.12 に示される実験機器を用いて実験を行った。なお、図 7.12 の 2 体のロボットは、それぞれが 4 関節のロボットであるが、根元の関節は用いずに、手先に 2 自由度をもつ冗長ロボットとして扱うものとする。各ロボットのパラメータは、6.1.4 節の表 6.2, 6.3 に示されるとおりであり、図 7.1 の実験機器とはパラメータが若干異なっていることに注意されたい。

実験 1 の case 1 の協調把持のタスクにおいて、把持力を制御する機能を、推定反力 $\hat{f}_{r1,x}^{\text{reac}}$, $\hat{f}_{r2,x}^{\text{reac}}$ から差の座標系における推定反力 $\hat{f}_-^{\text{reac}} = \frac{1}{2}(\hat{f}_{r2,x}^{\text{reac}} - \hat{f}_{r1,x}^{\text{reac}})$ へ座標変換したものをを用いるように変更する。残りの機能は図 7.2 同様の組み合わせを用いて、物体への接近 (mode 1) から把持力を制御する協調把持 (mode 2) を行う。

実験結果を図 7.13 に示す。これらの結果より、位置、反力応答は指令に応答がよく追従しているため、良好な制御結果が得られたといえる。また、図 7.13 (c), (d) の実際の反力応答と、推定反力応答は概ね一致しており、センサレスで良好な把持力の制御が行えたといえる。このことから、前節で述べたように、簡単な把持に関しては力センサレス制御が可能であるといえる。以上から、力センサレスな機能性に基づく制御システムの有効性が確認できた。

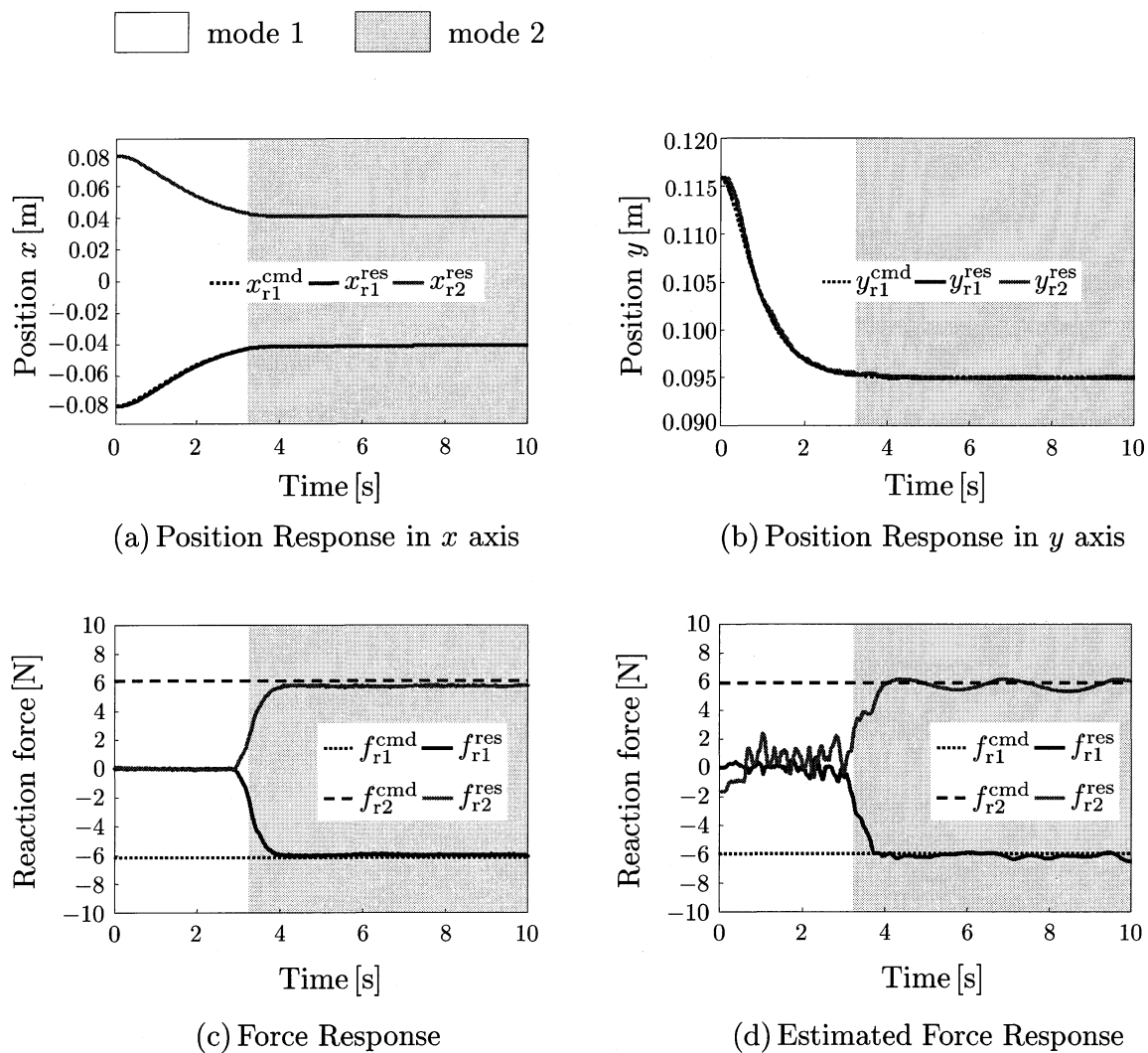


図 7.13: 実験 5 の結果

第8章 まとめと今後の課題

8.1 まとめ

本研究では、構造可変型ロボットに適した制御システムの構築を目的とした。先行研究では、構造可変型ロボットに適した中央制御部と分散知能を構成した分散制御システムを提案しており、演算の集中を回避することで構造の組み替えにより対応を簡単化する制御システムを提案している。また、その制御システムによって、ロボット単体の手先の位置制御、および力制御を可能とし、シミュレーション上での実験で有効性を確認している。

これらを背景に、本研究では、上記の制御システムをさらに発展させるため、作業の多様化が図れる制御システムへの拡張と、制御システムの力センサレス化の2点を主眼として研究をすすめた。

まず、作業の多様化に関しては、構造可変型ロボットに適した分散制御システムに「機能性」を導入することを提案した。その制御システムは、複数台の構造可変型ロボットによる協調作業を可能とし、かつ複数の機能に基づく制御器の組み合わせによりロボットの手先の作業を制御するため、それらの多様な組み合わせにより作業の多様化が図れる。また、構造可変型ロボットに適した速度ベースの制御システムが手先の応答特性を構造に依存しないように正規化することで、機能性に基づく分散制御システムの構成に必要な機能に基づく制御器の設計は、ロボットの構造に依存しなくなる。そのため、ロボットの構造を意識せずに制御器を設計することが可能となり、構造が組み替えられても手先の作業の制御は同等の性能が保たれるという利点が生まれる。このことから、提案する制御システムは、構造や作業の変更に対して柔軟に対応可能な制御システムであるといえる。本研究では、構造可変型ロボットを模擬した実機システムを用いた実験において、その制御システムによる協調把持動作を実現することで、提案する制御システムが構造や作業の変更に対応できることを示した。また、簡便な姿勢の冗長制御システムと機能性に基づく制御システムを組み合わせ、ロボットの姿勢を制御しながら手先の作業を制御する実験を行った。そして、その実験結果から、冗長自由度を有効に活用できる制御システムであることを示した。

これに加えて、本研究では、産業用システムとして非常に有利となる力センサレス制御システムの構築を目指した。力の情報を力センサなしで検出するため、トルク反力推定オブザーバを利用した制御システムを提案した。また、関節空間におけるトルク反力推定値から作業空間における手先反力推定値を容易な計算から得る方法を提案することで、運動学の計算が問題となり構造の組み替えの対応が複雑化することを回避できる構造可変型ロボットに適した力センサレス制御システムを構築した。本研究では、提案する力センサレス制御システムの有効性を、ロボット単体が壁を押し込むタスクを制御する実験によって検証した。簡単な同定のみでトルク反力を推定するため、推定精度には限界があるものの、実験結果では概ねロボットの力制御をセンサレスで行えることを確認した。そして最後に、力センサレス化した機能性に基づく制御システムを構築し、実験において力センサレスで簡単な把持が可能であることを示し、そのシステムの有効性を確認した。

8.2 今後の課題について

今後の研究について、現状での考えを以下の点にまとめる。

まず一点目に、本研究では問題を簡単化するため、制御システムの適応環境を 2 次元平面に限定してきたが、実際に 3 次元空間においても適応可能であることを検証していく必要があると考えられる。2 次元平面のシステムの場合、設計者は直感的な計算でロボットの幾何学的構造を把握できる。しかし、3 次元空間における多自由度ロボットに関しては、直感的な計算は困難となるため、系統的な手法が必要となる。系統的な計算で 3 次元空間を考慮できる方法の代表例として、順運動学で説明した DH 法が挙げられる。この手法のように、構造可変型ロボットに適した逆運動学である仮想速度伝播アルゴリズム (VVTA) を系統的な計算で 3 次元空間へ拡張できることを示す必要がある。

二点目に、分散知能の演算能力を積極的に利用する付加価値をつけていくことが挙げられる。本研究は、主に指令生成の役割を担う中央制御部についてを扱っており、指令を受け取り各モジュールを制御する分散知能に関しては、大幅な制御システムの変更を加えてはいない。しかし、提案している分散制御システムでは、各関節レベルで演算能力をもつ知能を配置するため、関節レベルでその能力を積極的に利用できる付加価値を与えることが望ましい。例としては以下の価値を付加できれば制御システムの発展が望めると考えている。

- 各関節レベルのリミッタ

分散知能を各モジュールに配置することで各関節レベルで制御ループを持つため、そのレベルで故障検知が行えるという利点がある。そして、各分散知能でトルクや速度のリミッタを設定し、停止信号を送ることが可能になる。現状では、故障を検知したときその場停止を行うなどの付加価値を与えやすいという考えから、各関節レベルでは力次元ではなく、速度次元の制御ループを持たせているが、実際にその価値を積極的に活かすように制御システムを設計できてはいない。そのため、今後は故障検知からリミッタを構成するアルゴリズムに関して、構造可変型ロボットに適したものを構成できれば、分散知能を配置する利点がより大きくなる。

- 各関節のノミナル慣性のオンライン推定

現状では、2次元平面で幾何学的な構造を把握しているため、各関節のノミナル慣性を求めることは難しくはない。しかし、3次元空間における多自由度ロボットとなれば、各関節のノミナル慣性の計算は複雑化する。また、モータに取り付けられた減速機の減速比が大きい場合、負荷慣性は限りなく小さくなるため、モータ側の慣性と減速比のみで、大まかにノミナル慣性を見積もる方法はあるものの、その値が実際の慣性からずれるほど制御性能は劣化する。この問題の一解決策が各関節のノミナル慣性のオンライン推定である。各関節レベルで制御ループをもつため、各関節の慣性を制御中の制御入力と実際の応答から、逐次その関節のノミナル慣性を計算することができる。オンライン推定により、ノミナル慣性の設定の精度向上が望めるのであれば、簡単な計算により大まかなノミナル慣性を見積もり、あらかじめ慣らし運転を行い、正確なノミナル慣性を求めるような手順を踏めば、構造の変更に対してノミナル慣性を再設定することが組み替え時の対応の足かせにはならなくなる。さらに、ノミナル慣性の設定の精度向上は、本研究で述べたトルク反力の推定の精度向上にもつながる。

参考文献

- [1] 神村明哉, 村田智, 吉田英一, 黒河治久, 富田康治, 小鍛冶繁: 「自己組立可能なモジュール型ロボットシステムに関する研究 (複数ユニットによる変形・移動実験)」, 日本機械学会論文集 (C 編), Vol. 68, No. 667, pp. 174–180 (2002)
- [2] 植山剛, 福田敏男, 新井史人, 加藤雄平, 松村司朗, 上杉武弘: 「動的再編成可能ロボットシステムに関する研究 (第 10 報, CEBOT の群構造に対するネットワーク・エネルギー評価を用いた組織の分散制御構造化)」, 日本機械学会論文集 (C 編), Vol. 58, No. 549, pp. 132–139 (1992)
- [3] N. Miwa, K. Yubai and J. Hirai: “A Fundamental Study on Reconfigurable Robot System Construction with Central- and Local Intelligence”, *Proc. of 9th IEEE International Workshop on Advanced Motion Control*, Vol. 1, pp. 90–93 (2006)
- [4] N. Miwa, K. Yubai, N. Kataoka and J. Hirai: “Inverse Kinematics and Redundant Control Best-suited for Reconfigurable Robot”, *Proc. of International Conference on Mechatronics*, WA2-A-3 (2007)
- [5] N. Kataoka, K. Yubai, and J. Hirai: “Position and Force Control System for Reconfigurable Robot”, CACS2007, T-108 (2007)
- [6] T. Tsuji and K. Ohnishi: “A Controller Design Method of Decentralized Control System”, *Proc. of 5th International Power Electronics Conference*, pp. 1378–1385 (2005)
- [7] T. Tsuji, K. Ohnishi and A. Sabanovic: “A Controller Design Method Based on Functionality”, *Proc. of 9th IEEE International Workshop on Advanced Motion Control* (2006)
- [8] 稲玉哲, 駒田諭, 大西公平: 「外乱推定オブザーバを用いたロボットハンドのバイラテラルサーボ制御」, 電気学会論文誌 (産業応用部門誌), Vol. 109-D, No. 4, pp. 281–287 (1989)

- [9] 村上俊之, 大西公平:「ロバスト制御に基づいた多自由度ロボットの安定性および作業空間での非干渉制御に関する一考察」, 電気学会論文誌 (産業応用部門誌), Vol.113-D, No.5, pp.639-646 (1993)
- [10] S.Manabe: “Coefficient Diagram Method”, *Proc. of 14th IFAC Symposium on Automatic Control in Aerospace*, pp.322-327 (1998)
- [11] 真鍋舜治:「係数図法による2慣性共振系制御器の設計」, 電気学会論文誌 (産業応用部門誌), Vol.118-D, No.1, pp.58-66 (1998)
- [12] 矢向高弘, 伊藤正尚, 大西公平:「仮想力伝播に基づく協調マニピュレータの分解制御」, 精密工学会誌, Vol.67, No.2, pp.241-245 (2001)
- [13] 柴田剛慶, 村上俊之:「冗長マニピュレータにおける受動性に基づく零空間制御」, 精密工学会誌, Vol.73, No.11, pp.1238-1245 (2007)
- [14] 栗屋伊智郎, 加藤 義樹, 大田裕二, 三宅岩夫, 伊藤正美:「クーロン摩擦が作用する機械運動系のパラメータ同定法」, 日本機械学会論文集 (C編), Vol.59, No.567, pp.108-114 (1993)
- [15] 吉川 恒夫, ロボット制御基礎論, コロナ社 (1988)
- [16] 松日楽 信人, 大明 準治, ロボットシステム入門, オーム社 (2004)
- [17] ニューハーモニックドライブ CSF ミニシリーズ カタログ, ハーモニックドライブシステムズ (2004)

謝辞

本論文は、筆者の三重大学大学院工学研究科博士前期課程電気電子工学専攻在学中における研究活動の成果を纏めたものであります。

本研究の遂行ならびに修士学位論文の作成にあたって、熱心なご指導とご鞭撻を賜りました、三重大学教授工学博士 平井 淳之先生，同大学准教授 駒田 諭先生，同大学准教授 弓場井 一裕先生に深く感謝いたします。また，多くの助力をいただいた，技術職員 中村 勝氏に深く感謝いたします。

また，電機システム研究室と制御システム研究室の合同検討会において，貴重なご意見をいただきました同大学教授 石田 宗秋先生，同大学准教授 残間忠直先生に深く感謝いたします。

本論文の作成にあたり，査読頂き貴重な御意見を頂きました同大学准教授 早川 聡一郎先生に深く感謝いたします。

そして，他大学でありながら，多くの親切なご指導を頂きました，埼玉大学助教 辻 俊明先生に心から感謝いたします。

構造可変型ロボット研究グループの先輩として，親身なご指導を頂きました三重大学大学院卒業生，片岡 典剛氏に深く感謝いたします。また，同研究グループの同期として，共に研究を遂行した同大学卒業生 久保西 真弘君に感謝いたします。そして同じ研究グループとして共に研究を進め，貴重な経験を頂いた研究生 北村 政仁君，4年生 川北 将大君に感謝いたします。両君の今後の躍進を心より期待しております。

電機システム研究室の同期として，時には切磋琢磨し合い，時には互いを励まし合いながら研究を遂行してきた宇佐見 秀徳君，木下 弘輔君，灰屋 和勇君，村上 洋輔君には，共に研究室での生活が楽しく有意義に過ごすことができたことに深く感謝いたします。

そして，同じ制御理論グループの一員として，日頃から共に研究に励んだ藤井 宏樹君，水谷 彰孝君，寺田 真也君，長坂 太郎君，山本 真資君に感謝致します。

また，何かと至らない自分を受け入れ，多くのご協力を頂き，学部から大学院までの三年間の研究生活を充実したものにしてくれた電機システム研究室の皆さんに深く感謝いたします。電機システム研究室の更なる発展を心より願っております。

論文目録

- (1) 上村, 弓場井, 平井:「構造可変なロボットアームを用いた協調把持動作の実現」, 平成 19 年 SICE 三重地区計測制御研究講演会講演論文集, B7 (2007.12)
- (2) 上村, 弓場井, 平井:「構造可変型ロボットに適した機能性に基づく分散制御システムの構築」, 平成 20 年 電気学会産業応用部門大会論文集, R2-6 2-44 (2008.8)
- (3) A.Uemura, K.Yubai, J.Hirai: “A Function-based Decentralized Control System for Cooperative Motions by Multiple Reconfigurable Robots”, CACS International Automatic Control Conference, FA08-84 (2008.11) (The Best Student Paper Award 受賞)
- (4) 上村, 弓場井, 平井:「構造可変型マニピュレータによる協調作業と姿勢の制御」, 産業計測制御研究会論文集, II C-09-125 (2009.3)
- (5) 上村, 弓場井, 平井:「複数の構造可変型ロボットによる協調作業の制御」, 平成 21 年 電気学会産業応用部門大会論文集, Y-55 (2009.8) (YPC 優秀発表賞 受賞)
- (6) 上村, 弓場井, 平井:「構造可変型ロボットに対する機能性に基づく分散制御システムによる協調作業の実現」, 平成 21 年 電気学会産業応用部門誌, Vol.129-D No.10 pp.1004-1010 (2009.10)
- (7) 上村, 弓場井, 平井:「構造可変型ロボットに適した力センサレス制御システムの構築」, 産業計測制御研究会論文集 (2010.3 発表予定)