



Title	産業用ロボットアームの解析と制御に関する研究
Author(s)	後藤, 聡
Citation	大阪大学, 1995, 博士論文
Version Type	VoR
URL	<a href="https://doi.org/10.11501/3106813">https://doi.org/10.11501/3106813</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

産業用ロボットアームの解析と制御に  
関する研究

1995年5月

後藤 聡

産業用ロボットアームの解析と制御に  
関する研究

1995年5月

後藤 聡

# 目次

第1章 序論	1
1.1 産業用ロボットアームとは	1
1.2 産業用ロボットアームの歴史	2
1.3 産業用ロボットアームの使用例	3
1.4 産業用ロボットアームの制御法の現状	4
1.5 本論文の目的と概要	6
第2章 ソフトウェアサーボ系の要求制御性能に基づいたトルク分解能決定法	9
2.1 まえがき	9
2.2 ソフトウェアサーボ系のトルク分解能解析のための数式モデル	10
2.3 トルク量子化誤差による位置決め精度の劣化	11
2.3.1 ソフトウェアサーボ系の位置決め	11
2.3.2 位置決め誤差と角加速度分解能との関係	12
2.4 トルク量子化誤差によるランプ応答の劣化	13
2.4.1 ソフトウェアサーボ系のランプ応答	13
2.4.2 角速度出力が目標角速度 $V_{ref}$ 以下の状態	13
2.4.3 角速度出力が目標角速度 $V_{ref}$ 以上の状態	14
2.4.4 ふらつきの周期と振幅	14
2.5 トルク分解能の決定	15
2.5.1 位置決め精度	15
2.5.2 ランプ応答のふらつき	16
2.5.3 トルク分解能のビット数の計算	16
2.5.4 トルク分解能決定のための数値例	17
2.5.5 制御性能とトルク分解能, 制御パラメータの関係	17
2.6 まとめ	18
第3章 産業用多関節ロボットアームの作業線形モデルの近似誤差の評価とモデルの妥当性	19
3.1 まえがき	19
3.2 産業用多関節ロボットアーム制御系の作業線形モデル	20
3.2.1 産業用多関節ロボットアーム制御系	20
3.2.2 2軸ロボットアームの実軌道	20

3.2.3	2軸ロボットアームの作業線形近似軌道	23
3.3	作業線形モデルの妥当性と適用範囲	24
3.3.1	作業線形モデルの近似誤差	24
3.3.2	作業線形モデルの定量的評価	25
3.3.3	作業線形モデルの誤差の累積	28
3.3.4	多軸ロボットアームへの拡張	30
3.4	計算機シミュレーションと実験結果	31
3.5	まとめ	33
<b>第4章</b>	<b>産業用多関節ロボットアームの高速輪郭制御における軌跡誤差の解析</b>	<b>35</b>
4.1	まえがき	35
4.2	産業用多関節ロボットアームの高速動作時における問題点	35
4.2.1	産業用多関節ロボットアームの制御手順	35
4.2.2	高速動作時における軌跡誤差	36
4.3	産業用多関節ロボットアームの数式モデル	37
4.3.1	キネマティクス	37
4.3.2	ダイナミクス	37
4.4	産業用多関節ロボットアームの軌跡誤差	38
4.4.1	軌跡誤差の定義	38
4.4.2	直線目標軌道に対するロボットアームの動作	38
4.4.3	軌跡誤差の原因	41
4.4.4	軌跡誤差と目標速度の関係	41
4.5	軌跡誤差の目標速度と目標軌跡依存性	41
4.5.1	目標速度依存性	41
4.5.2	目標軌跡依存性	42
4.6	まとめ	43
<b>第5章</b>	<b>極配置レギュレータを用いた教示信号修正法</b>	<b>44</b>
5.1	まえがき	44
5.2	教示信号修正法	45
5.2.1	教示信号修正法の概念	45
5.2.2	1次系モデルに基づく教示信号修正法	45
5.2.3	2次系モデルに基づく教示信号修正法	47
5.3	教示信号修正法の特性解析	51
5.3.1	1次系モデル	51
5.3.2	2次系モデル	54
5.4	産業用ロボットアームによる実験結果	55
5.4.1	ロボットアームへの教示	55
5.4.2	実験結果	55
5.5	まとめ	57

<b>第 6 章</b>	<b>速度情報に着目した教示信号修正法</b>	<b>58</b>
6.1	まえがき	58
6.2	教示速度修正法	58
6.2.1	教示速度修正法の概念	58
6.2.2	教示速度修正法のアルゴリズム	59
6.2.3	パラメータの選定	61
6.3	教示速度修正法の評価	62
6.3.1	操作時間と精度	62
6.3.2	実験結果	63
6.4	まとめ	64
<b>第 7 章</b>	<b>ガウシアンネットを用いた教示信号修正法</b>	<b>65</b>
7.1	まえがき	65
7.2	ガウシアンネットの構造と初期値決定のための数式モデル	65
7.3	ガウシアンネットを用いた教示信号修正法	66
7.3.1	教示信号修正法の考え方	66
7.3.2	ガウシアンネットによる逆システムの構成	66
7.3.3	ガウシアンネットの適用	70
7.4	XY テーブルによる輪郭制御実験	70
7.4.1	実験条件	70
7.4.2	学習データ生成	71
7.4.3	輪郭制御実験結果	72
7.5	まとめ	73
<b>第 8 章</b>	<b>従軸逆ダイナミクスを用いた主軸位置同期制御法</b>	<b>74</b>
8.1	まえがき	74
8.2	主軸位置同期制御対象の数式モデルと主軸位置同期制御の必要性	75
8.2.1	主軸位置同期制御対象の数式モデル	75
8.2.2	位置同期を図らない場合の問題点	76
8.3	主軸位置同期制御法	77
8.3.1	主軸位置同期制御法の導出	77
8.3.2	主軸位置同期制御法の特性解析	77
8.4	シミュレーション結果と実験結果	78
8.4.1	主軸位置同期制御シミュレーション	78
8.4.2	主軸位置同期制御実験	83
8.5	まとめ	86
<b>第 9 章</b>	<b>結論</b>	<b>87</b>
	謝辞	90

目次

*iv*

参考文献

91

研究発表

94

# 第1章

## 序論

### 1.1 産業用ロボットアームとは

ロボットとは知能、感覚器および汎用性にとんだ操作部で構成される装置であり、少なくとも人間の腕に相当する部分を有し、人間の手に似た作業を行なう。ロボットの特徴として、従来の自動機械と比較して滑らかで複雑な生物に似た動きをする。ロボットの知能は人間の命令を解読し、各種センサからの信号を処理して、作業を実行する操作信号を発生する。ロボットの感覚器は、外界や作業対象物の状態を検知し、作業状況に応じた適応的動作を行なうためのものである。ロボットの操作部は、アーム、ハンド、移動機構などから構成され、その特徴は、多様な目的に対応できる汎用性にある<sup>[1]</sup>。人間の労働を自動機械に代替する究極の姿としてのロボットは、唯一の人道的に許された奴隷として人間の仕事を代替するものである<sup>[2]</sup>。現実の環境をあまり変えることなく、人間と共存し得る形でのロボットを開発していくためには、人間に近い知能や技能をロボットに持たせる必要がある<sup>[3]</sup>。現時点で利用可能な知能や技術を前提とすると、マン・ロボットインターフェースの部分にシワ寄せする形でロボットを使う側の人間が苦勞するか、ロボットの利用目的とその範囲を限定するか、あるいは、ロボットの能力に合わせてある程度環境を作り変えるしか手がない。ロボット研究の目的は、これらの現状を緩和するため、あるいは、将来へのさらなる飛躍を目指して、知能と技術のレベルの向上を図ることにある。

産業用ロボットは、人間にとって危険であったり、過酷であったり、単調な作業を代行するために誕生した。産業用ロボットが製造業に導入された初期の頃は、その目的は作業工程内の省力・省人化であったが、近年のコンピュータ技術やエレクトロニクス技術の飛躍的な発展にともなって、産業用ロボットの機能や性能も急速に向上し、その利用される分野も、人間にとっての苦勞作業のみではなく、人間にとって困難な精密作業や悪環境下作業、人間による汚染を嫌うクリーン環境での作業にも利用されるようになってきている。ロボットの価格の下落もあって、製品のコストダウンのために生産性を向上させることと、その品質均一化や向上にも大きく寄与している。若年労働人口の減少や、高齢化による作業力低下を補う手段として、産業用ロボットの使用は欠かせないものとなっている<sup>[4]</sup>。最近では、産業用ロボットを産業用ロボット自身が製造する自動工場も現れている(安川電機モートマンセンターなど)。

産業用ロボットが他の産業機械と異なる点は、それを動かすのに教示という手法が採用されたことにある。この点をもっとも端的に表しているのが、ティーチングプレーバックロボットで用いられている教示再生制御方式であり、ロボットに行なわせたい作業を、あらかじめオペレータがロボットを動かして、その位置や経路、作業条件、作業順序などの情報を教示し記憶させる。実行時にはそれを再生することにより、その作業を繰り返し行なわせることができる<sup>[5]</sup>。

産業用ロボットの生産現場における位置づけは、生産ラインでのある程度汎用的な自動化の手段(機器)の一つとして捉えられている。ここで、ある程度の汎用性を追求することは、産業用ロボットでは二つの意味で重要である。まず、第一には専用機械としての自動化装置では生産ラインに使用する上で多くの種類の機械が必要となることであり、第二にはあまりの汎用性を追求した自動化装置では、その経済性が問題となることである。すなわち、産業用ロボットは、経済性を十分考慮した上で汎用性を追求していく必要がある。

## 1.2 産業用ロボットアームの歴史

ロボットを作ることに対する人間の発想は古く、B.C.3000年頃にエジプト人になる道具のなかに見い出される<sup>[6]</sup>。ロボットという言葉自身は、1920年にチェコスロバキアのカレル・チャペックの戯曲「ロッサム・ユニバーサル・ロボット会社」で初めて使われた<sup>[4]</sup>。先端技術としてのロボットは、1960年代にはじまるNASA(米国航空宇宙局)の研究プロジェクトとして世に出たものが多く、月面などでの作業に遠隔操作の技術を導入し、各種の人間の作業をロボットに代行させようとするものであった。

今日の産業用ロボットは、テレオペレータと数値制御工作機械にその起源を持つ。テレオペレータは、操縦者が遠くはなれたところから作業を行なうことを可能にする装置であり、数値制御工作機械は、デジタル符号化した切削データによって、自動的に金属材料を望み通りの形に削るものである<sup>[7]</sup>。1962年、初めての産業用ロボットとして、アメリカのAMF社の「バーサトラン」とユニメーション社の「ユニメート」が登場している<sup>[4]</sup>。

産業用ロボットは、その制御手法の特徴から、第1世代、第2世代、第3世代という分類がなされている<sup>[2]</sup>。1960年代からの第1世代ロボット(ティーチングプレーバックロボット)は、仕事の前のティーチングというプロセスで教え込まれた自分の位置や姿勢を繰り返し再生(プレーバック)する繰り返し機械である。オペレータがコントローラを介してか、あるいは直接にアクチュエータを駆動し、関節角センサで検出したその動きをレコーダに記録する。作業実行時には、起動信号を与えることにより、レコーダの記録がコントローラを通じて再現される。これらがもっとも効果的に利用できるのは、第2次産業の製造業である。

1970年代から、外界の状況を知るためのセンサを付加して、その状況に適応させる工夫が始まる。第2世代ロボット(数値制御ロボット)は感覚付適応ロボットであり、外界の状況を知るためのセンサを付加して、その状態に応じて自分自身の行動

を変化させる。オペレータがコンピュータに数値情報を提供すると、コンピュータはこれをアクチュエータの動きに換算してコントローラに送り、アクチュエータがコンピュータの指令に従って動く。この第2世代ロボットは、第1世代ロボットのようにあらかじめ与えられて作業を繰り返すのみでなく、外界の情報に対して適応的に動作することが可能である。

1980年代から開発が始まった第3世代ロボット(知能ロボット)は、外界認識の3次元自然環境への拡大と、人間とロボットのコミュニケーションが確立されたロボットである。オペレータはコンピュータに簡単な作業仕様のみを指令すれば良く、コンピュータは指令を解読するとともに、視覚、触覚などの各種センサ信号とを総合してコントローラに実行信号を送り、環境や作業対象に適応した動作が実行される。この知能ロボットの開発が、2000年以降の産業用ロボットの需要拡大の鍵を握っている<sup>[8]</sup>。

### 1.3 産業用ロボットアームの使用例

1993年12月現在における世界の産業用ロボットの稼働台数は610,605台であり、このうち、日本は368,514台で全体の60.2%を占めている。このことから、日本が産業用ロボットの中心的役割を担っていることがわかる。また、ロボット台数の伸び率も、世界では前年比6.3%増であり、日本においても同5.3%の伸びとなっている<sup>[9]</sup>。

日本産業用ロボット工業会では1980年をロボット元年としている。国内産業用向けの産業用ロボットの出荷額推移を見ると、1980年から1985年では、年平均25.4%の高い成長を示しており、1986年から1991年までも年平均17.2%の伸びを示している。1992年と1993年には、平成不況の影響を受け、それぞれマイナス36%とマイナス22%の大幅減となっているが、国内製造業向けと輸出向けの産業用ロボットの需要は、1995年以降、年平均8~11%が見込まれている<sup>[8]</sup>。

産業用ロボットアームは、その機構上の特徴から、直交座標ロボット、円筒座標ロボット、極座標ロボット、多関節ロボットの四つのタイプに分けられている<sup>[10]</sup>。直交座標ロボットは、各軸の運動方向が互いに直交するように構成されており、それぞれの軸の運動方程式が独立に扱える上に線形になるので、もっとも制御しやすいものとなる。また、位置決め精度も姿勢によらず一定となるため、高精度を出しやすい。欠点としては、作業空間に対してアームの占有空間が大きく、動作速度も大きくとれないことがある。円筒座標ロボットは、 $x, y$ 平面の代わりに、基底部の垂直軸回りの回転と水平方向の突き出し(直動)によって極座標の運動を実現させたもので、これによって作業領域を大きくとることができる。極座標ロボットは、さらに第2軸も回転軸としたもので、3次元極座標に基づく運動で表されるロボットである。多関節ロボットは、さらに第3軸も回転軸としたものをいう。多関節ロボットの中にも水平多関節(スカラ)型や平行リンク型などさまざまなタイプがある。多関節ロボットは一般にもっとも自由な3次元運動が可能とされており、実際に使用されている産業用ロボットの多くも多関節ロボットである。この多関節ロボットの

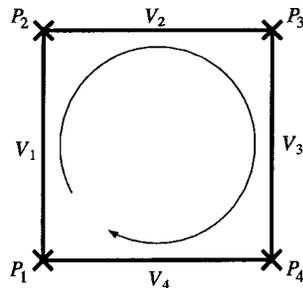


Fig. 1.1 Motion of industrial robot arms of teaching playback type.

位置決め精度を高めるのは、他の機構と比較して難しくなる。なぜならば、一般に作業精度はアームの先端の位置決め精度に依存するが、多関節ロボットの場合には、各軸の関節角の測定誤差が積み重なった値が先端部の位置決め誤差となり、誤差の拡大が起こりやすいからである。

産業用ロボットアームは、IC、LSIのプリント基盤への自動挿入、自動車などの製造ラインのスポット溶接、アーク溶接、塗装、組立、機械加工業での加工部品のハンドリングなど様々な用途に使用されている。産業用ロボットアームの使用例を分野別に見ると<sup>[11]</sup>、マテリアルハンドリング用としては、加工機へのワークの着脱、プレス作業などがあげられる。悪環境下作業用としては、スポット溶接、アーク溶接、塗装作業、鋳造品やモールド品のバリ取り、機械加工部品の面取りなどに用いられている。組立作業用としてはピックアンドプレイス作業、はめあい作業、タイプライタレバーの本体への組付け、IC、LSIのプリント基盤への自動挿入、軽合金の穴あけ作業などに用いられている。

#### 1.4 産業用ロボットアームの制御法の現状

ティーチングプレーバックロボットアームの動作は、まず、ロボットアームを動作させる目標軌道を、アームを実際に動作させたり、キー入力などによって与え、その入力された軌道を微小区間に分割し、その分割点ごとに、各軸ごとのサーボモータへの指令を計算する。この指令に追従するように、サーボコントローラによってサーボモータが制御される。そのサーボモータの回転に従って、アームが目標軌道に追従するように動作する。例えば、Fig. 1.1に示すような軌道でロボットアームを動作させる場合には、1. 目標位置  $P_1 \sim P_4$ へアームを移動させ記憶、2. 目標速度  $V_1 \sim V_4$ の入力、3. 動作確認、4. 繰り返し動作、の手順にしたがって実際の作業を行なうことになる。

Fig. 1.2に産業用ロボットアーム制御系の構成図を示す。産業用ロボットアームは、管理部、指令生成部、各軸サーボ系(位置制御部、速度制御部、電流制御部、パワーアンプ、モータ)と機構部から構成される。産業用ロボットアームは、通常、アクチュエータであるサーボモータが独立に制御されている<sup>[12]</sup>。すなわち、ロボット

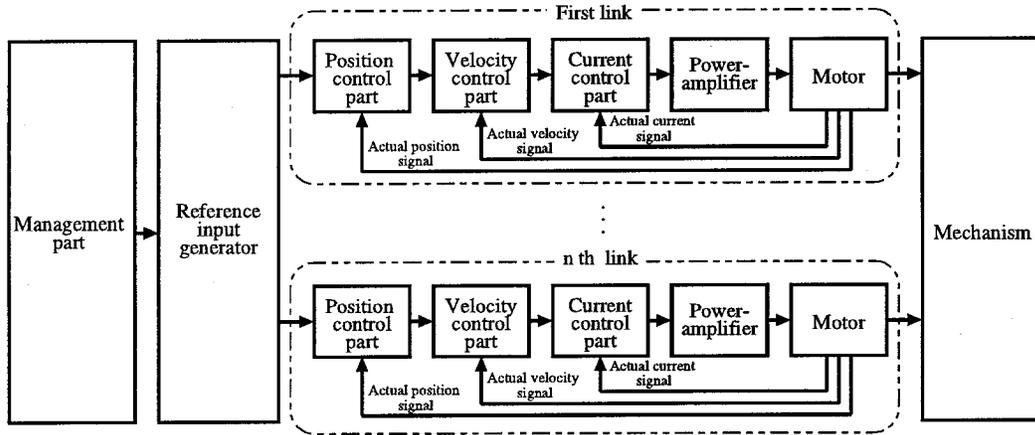


Fig. 1.2 Control structure of the industrial robot arms.

アームの各関節にポテンショメータ、タコジェネレータ、エンコーダなどのセンサを取り付けて、関節位置や速度を検出する。そして、ロボットアームの制御装置は、これらのデータを用いて、現実の手先位置が目標軌道にできるだけ沿うように、ロボットアームの各関節のアクチュエータへの入力を決する。このように、ロボットアームの先端の位置、速度、加速度情報を検出してループを構成するのではなく、ロボットアームのアクチュエータであるサーボモータの回転角、回転速度、回転加速度を検出してループを構成する制御法をセミクロードループタイプ制御法という<sup>[13]</sup>。各軸の制御装置は関節位置を制御変数とするサーボ機構であり、通常、産業用ロボットアームでは定数ゲインを用いた位置・速度フィードバックループが組まれている。このセミクロードループタイプ制御法では、ロボットアームの先端に新たにセンサを付加する必要がなく、コストが安くなるという利点があるが、モータの回転軸についてしか検出しないために、アームの機構のガタやアームのたわみなどによる、アームの先端の実際の位置とモータの軸から換算したアームの先端の位置とのずれを補償することはできない。また、この各軸独立のセミクロードループ制御法は、ロボットの姿勢による動特性の変化や各関節間の干渉を、各軸サーボ機構に対する外乱として対処していることになる。直交座標ロボットアームの場合には、アームの位置により慣性行列が変化しないために、各軸のサーボモータ独立の制御であっても問題は生じないが、多関節ロボットアームの場合には、アームの位置によって慣性行列が変化するため、サーボモータにかかる負荷が変化することになる。この負荷変化による制御性能の劣化を抑えるために、産業用ロボットアームでは、減速比を大きく取り、また、平行リンク機構を用いてアームの位置による負荷変化を小さくしている<sup>[10]</sup>。

セミクロードループタイプ制御法で問題となるのは、各軸独立に制御するために、ロボットの動作速度、位置精度などに対する要求が厳しくなるにつれて、ロボットの位置による負荷変化や各関節間の干渉などが無視できなくなり、目標追従特性が不

十分になる点や、ロボットアームを構成する各軸のサーボ系の遅れなどにより、目標軌道から追従軌道の劣化が生じる点である。

## 1.5 本論文の目的と概要

本論文で対象とする産業用ロボットアームは、第1世代の産業用ロボットアーム、すなわち、ティーチングプレーバックロボットアームを中心としている。このティーチングプレーバックロボットアームは、あらかじめティーチング作業により、動作に必要な情報をすべてロボットアームに与え、これらの情報に基づいてロボットアームの動作の制御が行われる。産業用ロボットアームの制御の基本は第1世代の産業用ロボットアームであり、第2、第3世代の産業用ロボットアームは、第1世代の産業用ロボットにセンサや知能を付加した物である。また、現在産業界で実際に用いられている産業用ロボットアームの多くは第1世代の産業用ロボットアームである。

本論文の目的は、現場で使用されている産業用ロボットアームの様々な経験則の解析と理論的裏付けを行い、また、実機の産業用ロボットアームに容易に適用することができる高精度制御法を開発することにある。現状の産業用ロボットアームは、工作機械の延長上にあり、ロボットアームを構成する各軸が独立に制御されており、ロボット制御理論と産業用ロボットアームで実際に用いられている制御の現状との間は、大きな隔たりがあるといわざるを得ない。産業用ロボットアームの特性の解析についてもあまり行なわれることがなく、実際に作業を行なう上において問題が生じた場合においても、その場の調整や今までの経験則により解決を図ってきており、問題点の原因究明や特性解析はなされていない。また、産業用ロボットアームの高速高精度化に対する要求は、年々高まってきており、経験則だけで対応するには限界がある。

本論文では、産業用ロボットアームの解析と制御に主眼をおき、特に、制御においては、実機にすぐに応用することが可能である手法を開発することを念頭においている。すなわち、産業用ロボットアームのハードウェアや制御系自身に変更を加えるのではなく、産業用ロボットの入力信号である教示信号を修正することによって、制御性能を向上させる手法を採っている。提案している手法の多くは、フィードバックタイプではなくフィードフォワードタイプの制御法である。産業用ロボットアームのダイナミクスは、各軸独立の1次系または2次系と簡単なモデル化を行なっている。これは、産業用ロボットアームの現状の制御法に対応したものであり、現在実際に稼働している産業用ロボットアームの作業条件下においては妥当なモデル化である。また、1次系と2次系の使い分けは、作業速度が比較的低速な場合を想定しているとき(モータの定格速度の1/100程度)には1次系モデルを用いており、作業速度が高速の場合(モータの定格速度の1/100~1/20程度)には2次系モデルを用いている。基本的には1次系モデルを用いて解析や制御法の開発を行なうが、1次系モデルでは問題が生じなかったり、作業速度が高速な場合を想定するときにはモデル化誤差が無視できなくなるため2次系モデルを用いている。また、産業用ロボットアームのモデル化において、ロボットの姿勢による動特性の変化や各関節

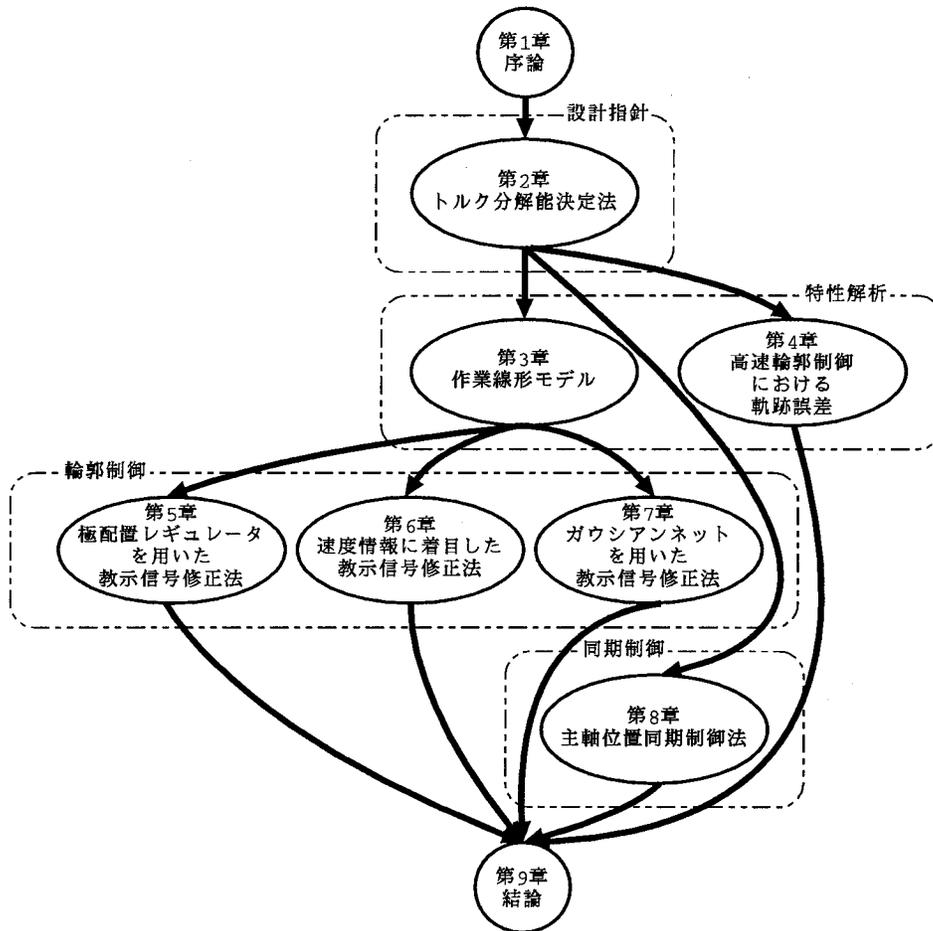


Fig. 1.3 Flow chart of the organization of the thesis.

間の干渉は、各軸サーボ機構に対する外乱として適切に補償されているため無視している。

本論文の構成について述べる。Fig. 1.3に本論文を構成する各章の関係のフローチャートを示す。2章では、産業用ロボットアームのアクチュエータの解析を行なう。産業用ロボットのアクチュエータとして使用され始めているソフトウェアサーボ系におけるトルク分解能の決定について、現状では経験則によりトルク分解能が決定されているが、ソフトウェアサーボ系のモデルに基づいて、系統的にトルク分解能を決定する手法を導出する。3章では、産業用多関節ロボットアームの作業座標系における線形モデルの妥当性を評価する。多関節ロボットアームにおいて各軸ごとに独立に制御されている場合には、その関節座標系における各軸のダイナミクスをそのまま作業座標系に適用することはできないが、実際の作業現場においては、作業座標系において関節座標系のダイナミクスをそのまま用いて、新しい作業への適用可否判断などが行なわれている。そこで、多関節ロボットアームのダイナミクスを作業座標系において線形近似(作業線形モデル)を行った場合の近似誤差の評価

とモデルの妥当性について検討する。4章では、作業線形モデルで表現できないような、高速動作時における産業用多関節ロボットアームの制御性能劣化の原因と評価を行なう。産業用多関節ロボットアームを高速で動作させる場合に、目標軌跡と追従軌跡とのあいだにずれが生じることが経験されているが、このずれの原因の究明と、ずれと目標速度や目標軌跡との関係を導出する。5章、6章、7章では、産業用ロボットアームの高精度輪郭制御を実現するための制御手法である教示信号修正法を提案する。提案する手法は、産業用ロボットアームの入力信号である教示信号を、ロボットアームを動作させる目標軌道をそのまま入力するのではなく、修正を加えることによって、ロボットアームの追従軌道と目標軌道を近づける手法である。5章では目標軌道の修正に極配置レギュレータ理論を用いる。5章で提案した極配置レギュレータを用いた教示信号修正法は、産業用ロボットに教示する位置と速度を適切に修正することによって、高速高精度を目指すものであるが、産業用ロボットアームのモデルにモデル化誤差が含まれる場合には、ロボットアームの追従軌跡にオーバーシュートが生じるという問題点があったために、6章では、教示信号の位置情報には修正を加えずに、速度情報のみで修正を施す、速度情報に着目した教示信号修正法を提案する。また、7章ではロボットアームのモデル化誤差を小さくするために、学習機能を持つニューラルネットの一つであるガウシアンネットを用いて教示信号を修正する教示信号修正法を提案する。8章では、産業用ロボットアームを用いる作業の一つであるネジ切り作業などで必要となる、主軸(回転軸)と従軸(直動軸)との位置を同期させるための、従軸逆ダイナミクスを用いた主軸位置同期制御法を提案する。最後に9章で結論を述べる。

## 第 2 章

# ソフトウェアサーボ系の要求制御性能に基づいたトルク分解能決定法

### 2.1 まえがき

近年、産業界では、サーボモータを駆動するサーボコントローラの制御回路が、すべてマイクロコンピュータのソフトウェアで組まれた、ソフトウェアサーボが利用されつつある<sup>[14, 15, 16]</sup>。ソフトウェアサーボは制御回路にデジタル回路を用いるために、温度変化などによる特性の変化がない、外乱に強い、制御回路をソフトウェアで容易に変更可能、メカニズム、検出器、パワーアンプなどの好ましくない特性を補正して制御することが可能、上位システムとの情報伝達が容易といった利点がある<sup>[13, 17, 18]</sup>。一方で、ソフトウェアサーボ系においては、位置情報であるパルス信号から速度情報を推定するとき、パルス出力が疎になる低速時における速度情報の推定の問題や、位置情報、トルク情報の量子化による量子化誤差の問題が存在する。位置情報の量子化は、サーボモータに取り付けられたエンコーダによって決まる。トルク情報の量子化とは、モータが出力するトルクに相当するモータの電流フィードバックもマイコンで行なうために、モータを動作させるパワーアンプとマイコンとの間は A/D, D/A 変換を介して結合されていて、この A/D, D/A 変換による量子化を意味する。速度情報推定の問題を解決する手段として、速度信号をオブザーバで推定したり<sup>[19, 20]</sup>、ロバストな設計を行う手法<sup>[21]</sup>が提案されている。また、位置情報の量子化、すなわち、エンコーダの選定は、装置の位置決め精度などから決定されている。近年の高精度制御のためのエンコーダの高分解能化によって、従来はあまり考慮されることのなかったトルク情報の量子化による制御性能への影響が問題となってきた。このトルク情報の量子化誤差の問題については、適切な方策は提案されておらず、その制御性能への影響も考慮されていない。現状ではトルク分解能の決定は、経験値を用いているが、トルク分解能とサーボ系の制御性能の関係を導くことによって、要求精度の制御を実現するために必要なトルク分解能を理論的に決定する必要がある。

本章では、トルク量子化を含むソフトウェアサーボ系を表す数式モデルを提案し、その数式モデルのもとで、トルク分解能に対する位置決め精度やランプ応答の位置のふらつきなどの制御性能を明らかにする。これらの関係式を使用することによって、サーボ系の制御性能があらかじめ推測でき、また、その関係式を逆に用いるこ

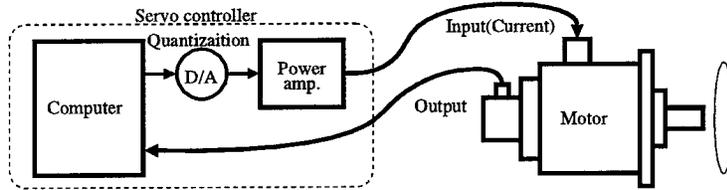


Fig. 2.1 Software servo system.

とによって、ソフトウェアサーボ系に必要な制御性能を実現するためのトルクを出力するためには、最低何ビットで A/D, D/A 変換を行なう必要があるかを決定することができる。

## 2.2 ソフトウェアサーボ系のトルク分解能解析のための数式モデル

本章で考察するソフトウェアサーボ系の概念図を Fig. 2.1 に示す。ソフトウェアサーボ系では、Fig. 2.1 に示すように、サーボコントローラ内の制御回路がマイコンのソフトウェアで構成されるために、制御回路からの出力であるトルク（電流）指令がデジタル量となる。それゆえ、電流指令をパワーアンプへ入力する際に D/A 変換器が必要となる。トルク量子化を含むサーボ系の 2 次系モデルのブロック線図を Fig. 2.2 に示す。 $K_p[1/s]$ ,  $K_v[1/s]$  はそれぞれサーボコントローラの位置ループゲイン、速度ループゲインを表す。また、速度ループのサンプリング間隔を  $\Delta t_v[s]$  とおく。サーボ系は、通常、位置フィードバック、速度フィードバック、電流フィードバックから構成される。位置フィードバックと速度フィードバックについては実際のモータの出力がサーボコントローラにフィードバックされるが、電流フィードバックについてはパワーアンプの出力がフィードバックされるのみであり、モータが実際に発生するトルクに変換した電流をフィードバックするものではない。そこで、Fig. 2.2 のブロック線図ではサーボ系の数式モデルとして位置フィードバックと速度フィードバックのみを考慮し、電流フィードバックは、指令電流値がそのままパワーアンプから出力されるものと仮定して省略している。速度ループの制御手法は比例制御や比例積分制御が行なわれているが、ここでは速度ループ全体の特性を 1 次系で表して、位置制御と速度制御とを合わせて 2 次系モデルで表現する<sup>[22]</sup>。

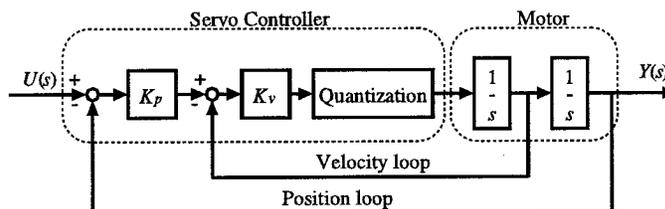


Fig. 2.2 Second order model for the software servo system containing torque quantization.

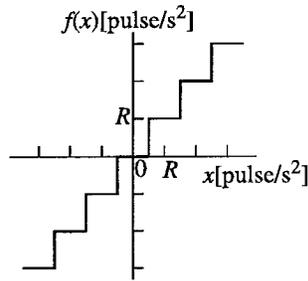


Fig. 2.3 Quantization of angular acceleration.

本章において問題としている A/D, D/A 変換に伴うトルク量子化を, Fig. 2.2 中の量子化要素によって表現する. トルクの量子化を行なう関数を  $f(\cdot)$  とすると, トルク量子化を含むサーボ系の数式モデルは

$$\ddot{y} = f(-K_v \dot{y} - K_p K_v y + K_p K_v u) \quad (2.1)$$

となる. サーボモータの回転角度はエンコードによってパルス [pulse] で検出されるため, 位置指令としてモータの回転角度  $u$  をパルスで表現し, 速度指令として角速度入力  $K_p(u - y)$  [pulse/s], トルク量子化に入力するトルク指令に相当するものとして角加速度入力  $K_v\{K_p(y - u) - \dot{y}\}$  [pulse/s<sup>2</sup>] を考える. 角加速度の量子化  $f(x)$  は, Fig. 2.3 の階段関数で示すように, 入力角加速度  $x$  [pulse/s<sup>2</sup>] が角加速度分解能  $R$  [pulse/s<sup>2</sup>] ごとに量子化されて出力されるものとする.

また, トルクの量子化による制御性能への影響のみを考慮するため, 位置や速度は量子化せずに連続値としてフィードバックされるものと仮定して取り扱う. これは, 実際のソフトウェアサーボ系において, サーボモータのエンコードの分解能が無限小, すなわち, 空間的に連続に位置と速度情報が得られる理想状態を意味する. エンコードが存在する実際のソフトウェアサーボ系と比較すると, この仮定のもとでの制御性能は最高のものであり, 導出するトルク分解能の条件は, 必要条件を与えることになる. つまり, ソフトウェアサーボ系において要求される制御性能を実現するためには, 最低限この条件を満足するようなトルク分解能を持つ A/D, D/A 変換を行なう必要がある. また, この仮定を導入することにより, トルク分解能の制御性能への影響の解析が容易になり, 2.5.1, 2.5.2 節でトルク分解能条件式の導出が可能となる. また, 2.5.4 節ではその条件式の妥当性をサーボモータのエンコードを考慮して実系を忠実に表す計算機シミュレーション上で示す.

## 2.3 トルク量子化誤差による位置決め精度の劣化

### 2.3.1 ソフトウェアサーボ系の位置決め

ソフトウェアサーボ系の位置決めに対する, トルク量子化誤差の影響について考える. 目標位置  $P_{ref}$  [pulse] と位置出力の定常値  $y(\infty)$  [pulse] の差である位置決め誤

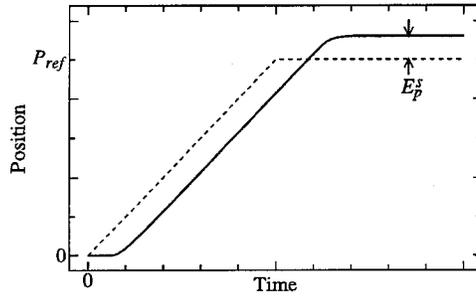


Fig. 2.4 Deterioration of positioning of the software servo system.

差  $E_p^s = P_{ref} - y(\infty)$ [pulse] は、制御パラメータ  $K_p$ ,  $K_v$  と角加速度分解能  $R$  によって定まるが、その関係式を理論的に導出する。ここでは、Fig. 2.4に示すように、サーボモータを目標位置  $P_{ref}$  まで一定速度入力で回転させて、位置決めを行なうものとする。角加速度を  $R$  ごとに量子化すると、サーボモータの速度も速度ループのサンプリング間隔  $\Delta t_v$  ごとの時刻においては量子化されることになる。すなわち、角加速度量子化を含むサーボ系においては、速度ループのサンプリング間隔  $\Delta t_v$  ごとでは、速度出力が  $R\Delta t_v$ [pulse/s] 単位でしか変化しない。この量子化の分解能を角速度分解能と呼ぶ。このことから、角加速度が量子化されたサーボ系においては、角速度出力が  $0$ [pulse/s] になるまで速度フィードバックが働き、角速度出力が  $0$  となった時点で速度フィードバックが切れて、位置出力が一定値となる定常状態が続くことになる。

### 2.3.2 位置決め誤差と角加速度分解能との関係

入力が目標位置  $P_{ref}$  と等しくなった時点において、Fig. 2.2の量子化要素への入力は  $K_v\{K_p(P_{ref} - y) - \dot{y}\}$  と表される。この値の大きさが角加速度分解能  $R$  以上では、位置及び速度フィードバックがかかり、角加速度分解能未満となり、かつ、 $\dot{y} = 0$ [pulse/s] となった時点で量子化要素の出力が零となり、位置出力が一定値で停止する。

位置出力が一定値となる定常状態において、量子化要素への入力の大きさは、Fig. 2.2から、位置決め誤差  $E_p^s$  を用いて  $|K_p K_v E_p^s|$  と表され、この値が角加速度の分解能  $R$  未満となるため、位置決め誤差  $E_p^s$  は  $K_p$ ,  $K_v$ ,  $R$  を用いて

$$|E_p^s| < \frac{R}{K_p K_v} \tag{2.2}$$

と表すことができる。(2.2) 式から、位置決め誤差  $E_p^s$  の上限は角加速度分解能  $R$  に比例し、位置、速度ループゲイン  $K_p$ ,  $K_v$  に反比例することがわかる。

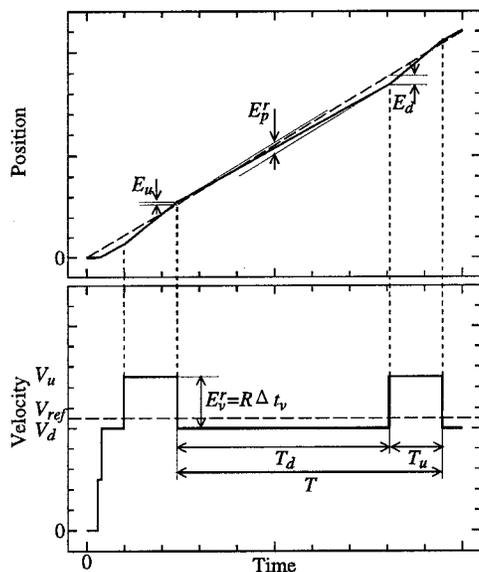


Fig. 2.5 Deterioration of ramp response

## 2.4 トルク量子化誤差によるランプ応答の劣化

### 2.4.1 ソフトウェアサーボ系のランプ応答

次に、ソフトウェアサーボ系のランプ応答に対する、トルク量子化誤差の影響について考える。サーボモータを一定速度入力  $V_{ref}$  [pulse/s] で回転させるものとする。角加速度を  $R$ ごとに量子化すると、目標角速度が角速度分解能の整数倍であれば、目標角速度と角速度出力が一致するため、角速度出力にふらつきは生じない。しかし、目標角速度が角速度分解能の整数倍でなければ、目標角速度と角速度出力が一致しないため、角速度出力にふらつきが生じる。

Fig. 2.5に角速度出力がふらつく様子を示す。Fig. 2.5の上図が位置のふらつきを表し、下図が角速度のふらつきを表す。Fig. 2.5から、応答は角速度出力が目標角速度以下の状態 ( $T_d$ [s]の区間)と目標角速度以上の状態 ( $T_u$ [s]の区間)の二つの状態に分けることができる。

### 2.4.2 角速度出力が目標角速度 $V_{ref}$ 以下の状態

角速度出力が目標角速度  $V_{ref}$  以下の状態では、出力角速度は角速度の量子化により  $V_d = \lceil V_{ref}/(R\Delta t_v) \rceil R\Delta t_v$  [pulse/s] となる (ここで、 $\lceil x \rceil$  は  $x$  以下の最大整数を表す)。目標角速度と角速度出力との差  $V_{ref} - V_d$  が積分されて出力位置誤差となり、角加速度入力の大きさが角加速度分解能の半分  $R/2$  (Fig. 2.3参照) を越えると、角速度分解能に相当する正のパルスが生じる。パルスが生じる時点における位置の出力誤差を  $E_d$  [pulse] とすると、このときの角加速度入力、Fig. 2.2のループをたどることにより  $K_v(K_p E_d - V_d)$  と表される。この値が角加速度分解能の半分  $R/2$  とな

ることにより

$$K_v(K_p E_d - V_d) = \frac{R}{2} \quad (2.3)$$

という関係式が成立する。(2.3)式を  $E_d$  について解くと

$$E_d = \frac{R + 2K_v V_d}{2K_p K_v} \quad (2.4)$$

となる。(2.4)式が正方向の位置の誤差の振幅となる。また、このときの速度の誤差の振幅は  $V_{ref} - V_d$  である。

#### 2.4.3 角速度出力が目標角速度 $V_{ref}$ 以上の状態

角加速度分解能に相当するパルスが生じると、角速度出力は  $R\Delta t_v$  増加し  $V_u = [V_{ref}/(R\Delta t_v) + 1]R\Delta t_v$  [pulse/s] となる。目標角速度と角速度出力との差  $V_u - V_{ref}$  が積分されて出力位置誤差となり、角加速度入力大きさが角加速度分解能の半分  $R/2$  (Fig. 2.3参照) を越えると、角加速度分解能に相当する負のパルスが生じる。パルスが生じる時点における位置の出力誤差を  $E_u$  [pulse] とすると、角加速度入力は  $-K_v(K_p E_u + V_u)$  と表される。この値が角加速度分解能の半分  $R/2$  となることにより

$$-K_v(K_p E_u + V_u) = -\frac{R}{2} \quad (2.5)$$

という関係式が成立する。(2.5)式を  $E_u$  について解くと

$$E_u = \frac{R - 2K_v V_u}{2K_p K_v} \quad (2.6)$$

となる。(2.6)式が正方向の位置の誤差の振幅となる。また、このときの速度の誤差の振幅は  $V_u - V_{ref}$  である。この負のパルスが生じた時点で、角速度出力は 2.4.2節の角速度出力が目標角速度以下の状態に戻り、この二つの状態が繰り返されることによってランプ応答のふらつきとなる。

#### 2.4.4 ふらつきの周期と振幅

角速度出力が目標角速度以下の状態が継続する時間  $T_d$  は、Fig. 2.5において、目標速度と追従速度との差  $V_{ref} - V_d$  が時間  $T_d$  続いたときの位置のずれが  $E_u + E_d$  となるということから、(2.4)式と(2.6)式を用いて

$$\begin{aligned} T_d &= \frac{E_d + E_u}{V_{ref} - V_d} \\ &= \frac{R(1 - K_v \Delta t_v)}{K_p K_v (V_{ref} - V_d)} \end{aligned} \quad (2.7)$$

となる。同様にして、角速度出力が目標角速度以上の状態が継続する時間  $T_u$  は

$$\begin{aligned} T_u &= \frac{E_d + E_u}{V_u - V_{ref}} \\ &= \frac{R(1 - K_v \Delta t_v)}{K_p K_v (V_u - V_{ref})} \end{aligned} \quad (2.8)$$

となる。ふらつきの周期  $T$ [s] は (2.7) 式の  $T_d$  と (2.8) 式の  $T_u$  を加え合わせた

$$\begin{aligned} T &= T_d + T_u \\ &= \frac{R^2 \Delta t_v (1 - K_v \Delta t_v)}{K_p K_v (V_{ref} - V_d)(V_u - V_{ref})} \end{aligned} \quad (2.9)$$

となる。位置のふらつきの振幅  $E_p^r$ [pulse] は、(2.4) 式と (2.6) 式を加え合わせた

$$\begin{aligned} E_p^r &= E_d + E_u \\ &= \frac{R + K_v(V_d - V_u)}{K_p K_v} \\ &= \frac{R(1 - K_v \Delta t_v)}{K_p K_v} \end{aligned} \quad (2.10)$$

となる。速度のふらつきの振幅  $E_v^r$ [pulse/s] は角加速度分解能より

$$E_v^r = R \Delta t_v \quad (2.11)$$

となる。

以上導出した (2.9) 式から (2.11) 式が、ふらつきの周期  $T$ 、位置のふらつきの振幅  $E_p^r$ 、速度のふらつきの振幅  $E_v^r$  と角加速度分解能  $R$  の関係を表す関係式である。これらの関係式からその特性を調べると、位置のふらつきの振幅  $E_p^r$  と速度のふらつきの振幅  $E_v^r$  は角加速度分解能  $R$  と比例し、ふらつきの周期  $T$  は角加速度分解能  $R$  の 2 乗に比例することがわかる。また、ふらつきの周期  $T$  は目標速度  $V_{ref}$  に依存するが、位置のふらつきの振幅  $E_p^r$  と速度のふらつきの振幅  $E_v^r$  は目標速度  $V_{ref}$  に依存せずに、角加速度分解能  $R$  とサーボコントローラのパラメータ  $K_p$ 、 $K_v$ 、 $\Delta t_v$  のみに依存することがわかる。

## 2.5 トルク分解能の決定

### 2.5.1 位置決め精度

位置決め誤差  $E_p^s$  がエンコーダ出力の 1[pulse] 未満になれば、エンコーダの分解能に比べてトルク分解能による影響は無いものと取り扱うことができる。位置決め誤差  $E_p^s$  が 1 未満になるための角加速度分解能  $R$  の条件は、(2.2) 式の右辺が 1 未満になることから

$$|E_p^s| < \frac{R}{K_p K_v} < 1 \quad (2.12)$$

を  $R$  について解くことにより

$$R < K_p K_v \quad (2.13)$$

と表される。すなわち、サーボモータの位置決め精度  $E_p^s$  を 1 未満にするためには、角加速度分解能  $R$  は (2.13) 式を満たすように決定する必要がある。

### 2.5.2 ランプ応答のふらつき

ランプ応答では、角速度出力劣化の振幅  $E_v^r$  と位置出力劣化の振幅  $E_p^r$  が、それぞれの許容範囲  $E_{limv}^r$  [pulse/s],  $E_{limp}^r$  [pulse] 以下になるようにトルク分解能を決定する。位置出力劣化の振幅条件を満たす角加速度分解能の上限  $R_p$  [pulse/s<sup>2</sup>] は、(2.10) 式を用いて

$$R_p = \frac{K_p K_v E_{limp}^r}{1 - K_v \Delta t_v} \quad (2.14)$$

と計算され、角速度出力劣化の振幅条件を満たす角加速度分解能の上限  $R_v$  [pulse/s<sup>2</sup>] は、(2.11) 式を用いて

$$R_v = \frac{E_{limv}^r}{\Delta t_v} \quad (2.15)$$

と計算される。角加速度分解能  $R$  は (2.14) 式と (2.15) 式から

$$R \leq \min(R_v, R_p) \quad (2.16)$$

と与える必要がある。すなわち、(2.16) 式を満たすように角加速度分解能  $R$  を決定すると、要求される許容範囲内にランプ応答の劣化を抑えることができる。

### 2.5.3 トルク分解能のビット数の計算

角加速度分解能  $R$  とソフトウェアサーボの電流フィードバックの際に用いられる A/D, D/A 変換のビット数との対応をとる。まず、角加速度分解能  $R$  を、モータの慣性モーメント  $J$  [kgm<sup>2</sup>] とエンコーダの一回転当たりのパルス数  $P$  [pulse/rev] を用いてトルク分解能  $R_\tau$  [Nm] に換算すると

$$R_\tau = \frac{2\pi R J}{P} \quad (2.17)$$

となる。

次に、A/D, D/A 変換のビット数に変換するわけであるが、ソフトウェアサーボ系で使用される A/D, D/A 変換のビット数は、符号ビットを除いた最大ビットでモータの最大トルクが出力できるようにとられる。すなわち、A/D, D/A 変換の分解能のビット数  $B$  [bit] とモータの最大トルク  $\tau_{max}$  [Nm] とトルク分解能  $R_\tau$  [Nm] との関係式は

$$\begin{aligned} 2^{B-1} &= \frac{\tau_{max}}{R_\tau} \\ &= \frac{\tau_{max} P}{2\pi R J} \end{aligned} \quad (2.18)$$

によって与えられる。ただし、最後の式変形は (2.17) 式を用いた。(2.18) 式を A/D, D/A 変換の分解能のビット数  $B$  について解くと

$$B = \log_2 \frac{\tau_{max} P}{\pi R J} \quad (2.19)$$

となる。この (2.19) 式で与えられるビット数  $B$  を用いると、要求するランプ応答の精度を満たすように A/D, D/A 変換の規格を決定できる。

#### 2.5.4 トルク分解能決定のための数値例

導出したソフトウェアサーボの制御性能と A/D, D/A 変換のビット数との関係をソフトウェアサーボ系に適用して, その有効性を確認する. 想定したサーボコントローラの位置ループゲインと速度ループゲインは, それぞれ  $K_p = 40[1/s]$ ,  $K_v = 200[1/s]$ , 速度ループのサンプリング間隔  $\Delta t_v = 50[\mu s]$ , サーボモータの規格は  $J = 0.13 \times 10^{-4}[\text{kgm}^2]$ ,  $\tau_{max} = 1.47[\text{Nm}]$ ,  $P = 5000[\text{pulse/rev}]$  とする. 位置決め精度を (2.13) 式, ランプ応答の劣化を  $E_{limp}^r = 1[\text{pulse}]$ ,  $E_{limv}^r = 1[\text{pulse/s}]$  となるように (2.16) 式を用いて角加速度分解能を決定し, (2.19) 式を用いてトルク分解能のビット数  $B$  を計算すると  $B = 15[\text{bit}]$  となる.

位置の量子化が存在する実際のソフトウェアサーボ系に対して, 導出したトルク分解能でどの程度の制御性能が得られるかを見るために, このトルク分解能を用いて位置情報の量子化までも考慮した計算機シミュレーションを行なった. 目標軌道は  $u(t) = 10000t[\text{pulse}]$  ( $0 \leq t < 1[\text{s}]$ ),  $u(t) = 10000[\text{pulse}]$  ( $1 \leq t \leq 2[\text{s}]$ ) とし, 速度情報の推定は位置情報の差分を用いた. このとき, 位置決め精度は  $E_p^s = 1[\text{pulse}]$ , ランプ応答の位置のふらつきは  $E_p^r = 2[\text{pulse}]$ , 速度のふらつきは  $E_v^r = 200[\text{pulse/s}]$  であった. また, トルク分解能を考慮せずに, 位置の量子化のみを考慮した場合においても, 位置決め精度, ランプ応答の位置のふらつき, 速度のふらつきすべて同じ値であった. 位置の量子化が存在する場合において, トルク量子化を考慮した結果と考慮しない結果が一致することから, 導出したトルク分解能においてはトルク量子化による影響が無視できることがわかる. また, 位置の量子化を考慮しない理想状態における設計値とシミュレーション結果を比較すると, 位置決め精度とランプ応答の位置のふらつきについては, 導出したトルク分解能で十分所望の制御性能が得られている. ランプ応答の速度のふらつきについては, 位置の量子化による影響が大きく出ている. この速度のふらつきが設計値からかなり劣化している原因は, 速度情報の推定に位置情報からの単純な差分を用いていることにあると考えられる. つまり, 差分によって速度情報を推定する場合の速度推定値の分解能が  $1[\text{pulse}]/\Delta t_v[\text{s}] = 1/(50 \times 10^{-6}) = 20000[\text{pulse/s}]$  となることが原因である. この速度推定値の分解能を基準とすると, 速度のふらつきは 1% であり非常に小さな値である.

次に, より一般的な制御性能の要求を満たすトルク分解能のビット数を求めるために, (2.2) 式で表される位置決め誤差  $E_p^s$ , (2.10) 式で表される位置のふらつきの振幅  $E_p^r$ , (2.11) 式で表される速度のふらつきの振幅  $E_v^r$  と角加速度分解能  $R$  の関係式を, (2.19) 式を用いてトルク分解能のビット数  $B$  に変換した関係を Fig. 2.6 に示す.

Fig. 2.6 を用いることによって, 要求する制御性能からトルク分解能のビット数を求めるのみならず, 実際に動いているソフトウェアサーボ系のトルク分解能のビット数から位置決め精度とランプ応答の制御性能を求めることができる.

#### 2.5.5 制御性能とトルク分解能, 制御パラメータの関係

ソフトウェアサーボ系の制御性能とトルク分解能, 制御パラメータの関係を整理する.

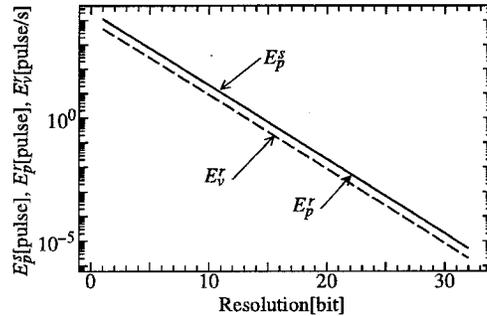


Fig. 2.6 Relationship between the control performance and the resolution of A/D and D/A converter.

1. Fig. 2.6で示されるように、位置決め誤差  $E_p^s$ 、位置のふらつきの振幅  $E_p^r$ 、速度のふらつきの振幅  $E_v^r$  の対数とトルク分解能のビット数  $B$  は1次関数で表される
2. (2.2) 式と (2.10) 式から位置決め誤差  $E_p^s$ 、ランプ応答の位置のふらつきの振幅  $E_p^r$  は位置ループゲイン  $K_p$  に反比例する
3. (2.11) 式から、速度のふらつきの振幅  $E_v^r$  は制御パラメータ  $K_p, K_v$  には依存しない
4. (2.9) 式から、ランプ応答のふらつきの周期  $T$  は目標速度  $V_{ref}$  に依存する。

## 2.6 まとめ

ソフトウェアサーボ系における制御性能はトルク分解能に依存する。そこでトルク量子化を含むソフトウェアサーボ系の数式モデルを導入し、ソフトウェアサーボ系の位置決め精度、ランプ応答の劣化とトルク分解能の関係を明らかにした。また、要求する制御性能から、トルク分解能のビット数を決定するための関係式を導き、グラフに表した。このグラフを用いることにより、ソフトウェアサーボのマイコンからパワーアンプへの出力の際の D/A 変換と、パワーアンプの出力をマイコンに取り込むための A/D 変換のビット数を、経験値に頼ることなく理論的に決定でき、また、使用されているソフトウェアサーボ系のビット数から、位置決め精度とランプ応答のトルク量子化による劣化を求めることができる。

## 第 3 章

# 産業用多関節ロボットアームの作業線形モデルの近似誤差の評価とモデルの妥当性

### 3.1 まえがき

多関節ロボットアームのダイナミクスは、その慣性行列がロボットアームの姿勢によって変化するため、ロボットアームを構成する各軸が干渉し、また、遠心力、コリオリ力や重力などの非線形項を含む複雑な微分方程式で表される。このような複雑なダイナミクスを持つ多関節ロボットアームの高速高精度制御には、一般に非常に複雑な制御器が必要とされる<sup>[23, 24, 25, 26]</sup>。しかし、産業界で用いられている多関節ロボットアームにおいては、減速比を大きくとることにより負荷慣性の影響を小さくし、また、平行リンク機構を用いて慣性行列の非対角成分の影響を小さくしているため、各関節のアクチュエータであるサーボモータをそれぞれ独立に制御する手法がとられている<sup>[5]</sup>。産業用ロボットアームの位置決めや輪郭制御では、作業座標系での目標軌道を細かく分割し、その分割点に対応する関節角を各軸に対して計算し、それぞれの軸の関節角が、分割された関節座標系における目標関節角まで一定速度で動くように、サーボモータの回転角の制御が行われる<sup>[10]</sup>。この制御方式において、作業座標系におけるロボットアームのダイナミクスは、経験的に作業座標系の各座標軸で線形独立とみて、ロボットアームの輪郭制御における誤差解析や、新しい作業への適用可否判断が行われているが、その線形独立の仮定の是非を理論的に解明する必要がある。

本章では、作業座標における各座標軸独立の線形近似モデル(以下、作業線形モデルと呼ぶ)の線形近似誤差を理論的に導出し、ロボットアームの可動領域内で線形近似誤差の少ない領域である線形近似可能領域(以下、作業線形近似可能領域と呼ぶ)を明らかにする。作業線形近似可能領域内では、作業座標系における目標軌道の分割や、各軸ごとの制御器を含むロボットアーム制御系を、作業座標系の各座標軸で線形独立の制御対象としてとらえることができる。そのため、関節座標系における特性、すなわち、サーボモータとその制御器の特性だけで新しい作業における適用可否判断ができ、また、制御性能を向上させるための指令値の修正などを作業座標系において各座標軸独立に行うことができ、さらに、より高速高精度のロボットアーム制御理論を開発した際に、その理論の実系への適用の限界が明らかにできる利点を有する。

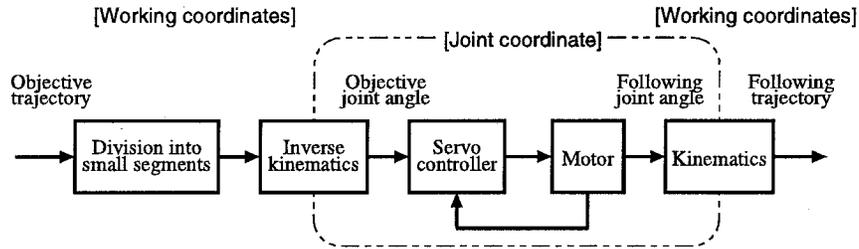


Fig. 3.1 Block diagram of a servo system for an industrial articulated robot arm

## 3.2 産業用多関節ロボットアーム制御系の作業線形モデル

### 3.2.1 産業用多関節ロボットアーム制御系

産業用多関節ロボットアームの輪郭制御のブロック線図を Fig. 3.1 に示す。まず、作業座標系での目標軌道を微小区間に分割する。その分割点に対応する関節角を各軸に対して計算し、それぞれの軸の関節角が分割された関節座標系での目標関節角まで一定速度で動くようにサーボモータの回転角の制御を行い、各軸のサーボモータをその移動量だけ回転させる。すると、アームの機構による座標変換でアーム先端が作業座標系の目標軌道に沿って動く。

目標軌道は作業座標系で与えられ、ロボットアームの制御は非線形変換を伴う関節座標系で各軸ごとに独立に行われ、追従軌道は再び非線形変換を伴う作業座標系で評価される。この制御方式でロボットアームを制御するとき、産業用ロボットアームの制御系は、通常、作業座標系で各座標軸線形独立として近似している。この作業座標系での線形近似の妥当性を議論する(3.3節)ための準備として、このロボットアーム制御系の作業座標系における実軌道と作業線形モデルによる作業線形近似軌道を導出する。

### 3.2.2 2軸ロボットアームの実軌道

多軸ロボットの特性を議論するに当たって、2軸の部分でその本質的な議論を行い、3.3.4節で多軸へと議論を拡張する。Fig. 3.2に、二つの剛体リンクが◎で表す関節でつながって、先端がこの平面上を動く2軸ロボットアームの概略図を示す。Fig. 3.2中の $(\alpha, \beta)$ は関節座標系における関節角、 $(x, y)$ は作業座標系におけるアームの先端の位置、 $l_1, l_2$ はそれぞれ1軸と2軸の長さを表す。この2軸ロボットアームは多軸ロボットアームの基本的な構成をなしており、スカラ型ロボットアームでは、平面の位置決めをこの2軸で行っている。

初めに、作業座標系と関節座標系との関係を明確にするために、関節座標 $(\alpha, \beta)$ から作業座標 $(x, y)$ への変換(キネマティクス)と、作業座標 $(x, y)$ から関節座標 $(\alpha, \beta)$ への変換(逆キネマティクス)を示す。キネマティクスは、Fig. 3.2から

$$x = l_1 \cos \alpha + l_2 \cos(\alpha + \beta) \quad (3.1a)$$

$$y = l_1 \sin \alpha + l_2 \sin(\alpha + \beta) \quad (3.1b)$$

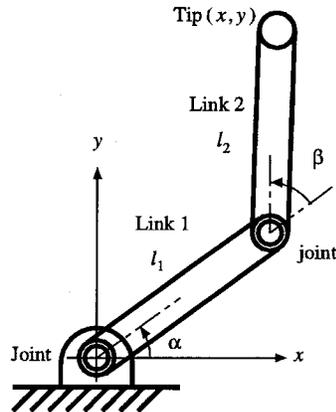


Fig. 3.2 Structure of an articulated robot arm of two-degree-of-freedom.

であり、逆キネマティクスは、(3.1) 式を  $(\alpha, \beta)$  について解くことにより

$$\alpha = \sin^{-1}\left(\frac{y}{\sqrt{x^2 + y^2}}\right) - \sin^{-1}\left(\frac{l_2 \sin \beta}{\sqrt{x^2 + y^2}}\right) \quad (3.2a)$$

$$\beta = \pm \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \quad (3.2b)$$

で与えられる。ここで、(3.1b) 式の符号は、作業座標系の 1 点を指定するのに関節座標系では 2 通りの可能性があることを表している。

次に、関節座標系におけるロボットアームのダイナミクスを与える。産業用ロボットアームでは、減速比を大きくとって負荷慣性を小さくし、また、平行リンク機構を用いて、慣性行列の非対角成分の影響を小さくしているため、アクチュエータであるサーボモータを各軸ごとに独立に制御する手法でロボットアームの制御を行っている。実際に稼働している産業用ロボットアームにおいては、ロボットアームの移動速度が 0.25[m/s] 以下程度のそれほど大きくない場合、負荷慣性による各軸の干渉や、コリオリ力などの非線形項を無視することができ、ロボットアームのダイナミクスは、アクチュエータであるサーボモータとその制御器を合わせて、各軸独立に 1 次系

$$\dot{\alpha}(t) = -K_p \alpha(t) + K_p u_\alpha(t) \quad (3.3a)$$

$$\dot{\beta}(t) = -K_p \beta(t) + K_p u_\beta(t) \quad (3.3b)$$

で表現できる<sup>[13]</sup>。(3.3) 式で表されるモデルを関節線形モデルと呼ぶ。ここで、 $u_\alpha(t)$ 、 $u_\beta(t)$  はそれぞれ 1 軸、2 軸の角度入力を表し、 $K_p$  は位置ループゲインである。Fig. 3.3 に 1 次系モデルのブロック線図を示す。本章では、各軸のダイナミクスは関節座標系で (3.3) 式で表せる、すなわち、関節線形モデルは実ロボットダイナミクスを表現することを認め<sup>[10, 13]</sup>、この仮定の下で以下のすべての議論を展開する。

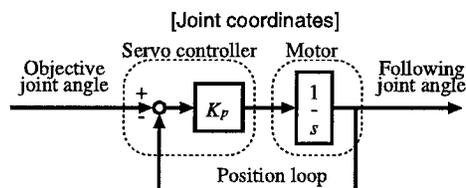


Fig. 3.3 Block diagram of a simplified first order model for a servo motor and its controller in the joint coordinates.

目標軌道を細分割した微小区間に対して，ロボットアームがどのように追従するかを調べる．目標軌道の細分割によって分割されたそれぞれの軌道に対して，その分割された一つの微小区間の作業座標系での始点と終点を，それぞれ  $(x_0, y_0), (x_T, y_T)$ ，関節座標系での始点と終点を，それぞれ  $(\alpha_0, \beta_0), (\alpha_T, \beta_T)$  で表す．この微小区間における関節座標と作業座標との関係は Fig. 3.4 で与えられる． $(x_0, y_0)$  と  $(\alpha_0, \beta_0)$ ， $(\alpha_T, \beta_T)$  と  $(x_T, y_T)$  の関係は，作業座標系と関節座標系との関係が (3.1) 式で表せることから

$$x_0 = l_1 \cos \alpha_0 + l_2 \cos(\alpha_0 + \beta_0) \quad (3.4a)$$

$$y_0 = l_1 \sin \alpha_0 + l_2 \sin(\alpha_0 + \beta_0) \quad (3.4b)$$

$$x_T = l_1 \cos \alpha_T + l_2 \cos(\alpha_T + \beta_T) \quad (3.5a)$$

$$y_T = l_1 \sin \alpha_T + l_2 \sin(\alpha_T + \beta_T) \quad (3.5b)$$

と表される．

産業用ロボットアームにおいては，分割された微小区間に対して各軸ごとに一定角速度入力  $(v_\alpha, v_\beta)$  が与えられることから，ロボットアームのダイナミクス (3.3) 式

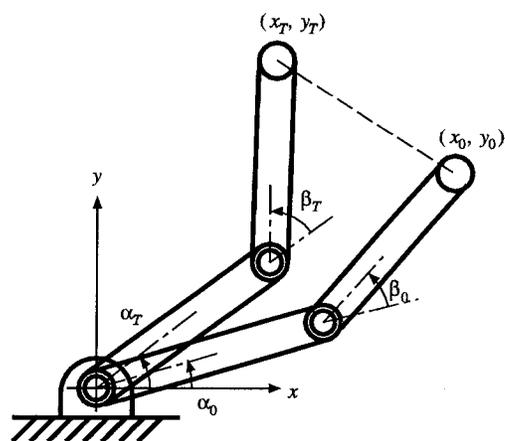


Fig. 3.4 A single segment of a divided objective trajectory.

の各軸に対する角度入力  $(u_\alpha(t), u_\beta(t))$  は

$$u_\alpha(t) = \alpha_0 + v_\alpha t, \quad v_\alpha = \frac{\alpha_T - \alpha_0}{T} \quad (3.6a)$$

$$u_\beta(t) = \beta_0 + v_\beta t, \quad v_\beta = \frac{\beta_T - \beta_0}{T} \quad (3.6b)$$

と与えられる。ここで、 $T$ は目標軌道の細分割における微小時間であり、分割の始点における時刻は0とする。

(3.6)式で表される角度入力の際の、作業座標系におけるロボットアームの位置を導出する。ロボットアームの初期時刻における目標軌道と実軌道の位置は等しい  $(\alpha(0), \beta(0)) = (\alpha_0, \beta_0)$  とすると、ロボットアームの関節座標系での位置は、(3.6)式の角度入力を(3.3)式に代入して微分方程式を解くことにより

$$\alpha(t) = \alpha_0 + v_\alpha \delta(t) \quad (3.7a)$$

$$\beta(t) = \beta_0 + v_\beta \delta(t) \quad (3.7b)$$

となる。ここで

$$\delta(t) = t + \frac{e^{-K_p t} - 1}{K_p} \quad (3.8)$$

である。このときのロボットアームの作業座標系の位置は、(3.7)式を非線形変換である(3.1)式に代入することにより

$$x(t) = l_1 \cos(\alpha_0 + v_\alpha \delta(t)) + l_2 \cos(\alpha_0 + \beta_0 + (v_\alpha + v_\beta) \delta(t)) \quad (3.9a)$$

$$y(t) = l_1 \sin(\alpha_0 + v_\alpha \delta(t)) + l_2 \sin(\alpha_0 + \beta_0 + (v_\alpha + v_\beta) \delta(t)) \quad (3.9b)$$

と計算できる。この(3.9)式が、ロボットアーム先端の作業座標系における実軌道を表す。この実軌道に対応させて、本章で問題とする、作業座標系において各座標軸独立に線形近似を行った、作業線形モデルでの作業線形近似軌道を導出する。

### 3.2.3 2軸ロボットアームの作業線形近似軌道

作業座標系で、ロボットアーム制御系を  $x$  軸  $y$  軸それぞれ独立に線形近似すると

$$\dot{\hat{x}}(t) = -K_p \hat{x}(t) + K_p u_x(t) \quad (3.10a)$$

$$\dot{\hat{y}}(t) = -K_p \hat{y}(t) + K_p u_y(t) \quad (3.10b)$$

となる。ここで、 $(\hat{x}(t), \hat{y}(t))$  は線形近似されたロボットアームの作業座標系での位置を表し、 $(u_x(t), u_y(t))$  は作業座標系での位置入力を表す。この(3.10)式が本章で議論の対象となる作業線形モデルである。線形近似された(3.10)式で、目標軌道をFig. 3.4のように細分割したときの、微小区間に対するロボットアームの応答を導出する。ここで、ロボットアームの初期時刻における目標軌道と作業線形近似軌道の位置は等しい  $(\hat{x}(0), \hat{y}(0)) = (x_0, y_0)$  とする。厳密には、関節座標系での入力である(3.6)式に対応する作業座標系における入力は、(3.1)式の座標変換によって導出す

る必要があるが、この作業座標系における入力、作業座標系では一定速度の入力とはならないが、作業座標系での入力も一定速度であるものとして

$$u_x(t) = x_0 + v_x t, \quad v_x = \frac{x_T - x_0}{T} \quad (3.11a)$$

$$u_y(t) = y_0 + v_y t, \quad v_y = \frac{y_T - y_0}{T} \quad (3.11b)$$

と近似し、その近似はほとんど無視できる。この(3.11)式の入力を(3.10)式の作業線形モデルに代入し、微分方程式を解くことによりロボットアームの作業線形近似軌道は

$$\hat{x}(t) = x_0 + v_x \delta(t) \quad (3.12a)$$

$$\hat{y}(t) = y_0 + v_y \delta(t) \quad (3.12b)$$

となる。すなわち、ロボットアームの作業座標系における実軌道(3.9)式に対応する作業線形近似軌道は(3.12)式で与えられる。

### 3.3 作業線形モデルの妥当性と適用範囲

#### 3.3.1 作業線形モデルの近似誤差

ロボットアーム制御系の実軌道(3.9)式と作業線形近似軌道(3.12)式を比較することにより、本章での議論の対象となる作業線形モデルの近似精度を評価する。作業座標系における近似誤差は(3.9)式と(3.12)式の差

$$e_x(t) = \hat{x}(t) - x(t) \quad (3.13a)$$

$$e_y(t) = \hat{y}(t) - y(t) \quad (3.13b)$$

である。(3.13)式の $(e_x(t), e_y(t))$ を作業線形近似誤差と呼ぶ。(3.13)式を空間に関する項と時間に関する項に分けて評価するために、ロボットアームの実際の作業座標系の位置を表す(3.9)式を、 $(\alpha_0, \beta_0)$ からの移動量が小さいものとして、テイラー展開を用いて1次近似すると

$$\begin{aligned} \tilde{x}(t) = & l_1 \{ \cos(\alpha_0) - \sin(\alpha_0) v_\alpha \delta(t) \} \\ & + l_2 \{ \cos(\alpha_0 + \beta_0) - \sin(\alpha_0 + \beta_0) (v_\alpha + v_\beta) \delta(t) \} \end{aligned} \quad (3.14a)$$

$$\begin{aligned} \tilde{y}(t) = & l_1 \{ \sin(\alpha_0) + \cos(\alpha_0) v_\alpha \delta(t) \} \\ & + l_2 \{ \sin(\alpha_0 + \beta_0) + \cos(\alpha_0 + \beta_0) (v_\alpha + v_\beta) \delta(t) \} \end{aligned} \quad (3.14b)$$

と計算される。実軌道と(3.14)式のテイラー展開1次近似軌道の間には

$$\begin{aligned} x(t) = & \tilde{x}(t) + l_1 o(v_\alpha \delta(t)) + l_2 o((v_\alpha + v_\beta) \delta(t)) \\ = & \tilde{x}(t) + o(\delta(t)) \end{aligned} \quad (3.15a)$$

$$\begin{aligned} y(t) = & \tilde{y}(t) + l_1 o(v_\alpha \delta(t)) + l_2 o((v_\alpha + v_\beta) \delta(t)) \\ = & \tilde{y}(t) + o(\delta(t)) \end{aligned} \quad (3.15b)$$

の関係がある<sup>[27]</sup>. ここで, (3.15) 式中の  $o(\delta(t))$  は  $\delta(t)$  よりも高位の微小数を表す. 三角不等式を用いると, 実軌道と作業線形近似軌道の誤差の大きさは, (3.12) 式と (3.14) 式から

$$\begin{aligned} |\hat{x}(t) - x(t)| &\leq |\hat{x}(t) - \tilde{x}(t)| + |\tilde{x}(t) - x(t)| \\ &= |\varepsilon_x \delta(t)| + |o(\delta(t))| \end{aligned} \quad (3.16a)$$

$$\begin{aligned} |\hat{y}(t) - y(t)| &\leq |\hat{y}(t) - \tilde{y}(t)| + |\tilde{y}(t) - y(t)| \\ &= |\varepsilon_y \delta(t)| + |o(\delta(t))| \end{aligned} \quad (3.16b)$$

と抑えることができる. ここで,  $(\varepsilon_x(t), \varepsilon_y(t))$  は

$$\varepsilon_x = v_x + y_0 v_\alpha + l_2 \sin(\alpha_0 + \beta_0) v_\beta \quad (3.17a)$$

$$\varepsilon_y = v_y - x_0 v_\alpha - l_2 \cos(\alpha_0 + \beta_0) v_\beta \quad (3.17b)$$

であり, ロボットアームの位置と速度に依存し, 時間には依存しない誤差項である.  $\delta(t)$  が十分小さいときには, (3.15) 式の  $o(\delta(t))$  の項が無視できる. 従って作業線形近似誤差は

$$e_x(t) \approx \varepsilon_x \delta(t) \quad (3.18a)$$

$$e_y(t) \approx \varepsilon_y \delta(t) \quad (3.18b)$$

と近似することができる. すなわち,  $\delta(t)$  が十分小さい, 言い替えると, 目標軌道の分割間隔が十分小さければ, 作業線形近似誤差は (3.18) 式で表すことができる. (3.18) 式は, (3.17) 式のロボットアームの位置に依存する項  $(\varepsilon_x, \varepsilon_y)$  と時間に依存する項  $\delta(t)$  との積で与えられる. (3.17) 式の  $(\varepsilon_x, \varepsilon_y)$  はロボットアームの位置  $(x_0, y_0)$ ,  $(\alpha_0, \beta_0)$  と移動速度  $(v_x, v_y)$ ,  $(v_\alpha, v_\beta)$  との関数となっている. ここで, ロボットアームの位置を関節座標で表現した  $(\alpha_0, \beta_0)$  は, (3.1) 式のキネマティクスにより作業座標系で表現することが可能であり, また, 関節座標系での移動速度も, (3.6) 式により  $(v_\alpha, v_\beta) = ((\alpha_T - \alpha_0)/T, (\beta_T - \beta_0)/T)$  で表され, (3.1) 式のキネマティクスにより作業座標系  $(x_0, y_0)$ ,  $(x_T, y_T)$  で表現できるため, (3.18) 式は作業座標系におけるロボットアームの位置  $(x_0, y_0)$ ,  $(x_T, y_T)$  で表現することができる. この (3.18) 式が着目する作業線形近似誤差を表す式であり, この大きさを評価することにより, ロボットアーム制御系の作業線形モデルの妥当性と作業線形近似可能領域を導出することができる.

### 3.3.2 作業線形モデルの定量的評価

ロボットアームの作業座標系における (3.10) 式の作業線形モデルの作業線形近似誤差が小さい領域, すなわち, 作業線形近似可能領域を定量的に評価する. Fig. 3.5 に, 作業座標系で点線で囲まれたロボットアームの可動領域内において, 0.2[m] ごとに分割された 188 点のそれぞれの始点  $(x_0, y_0)$  (図中の●) から矢印の方向にロボットアームを動かした場合の, 作業線形近似誤差の位置依存項  $(\varepsilon_x, \varepsilon_y)$  の値 (図中の●

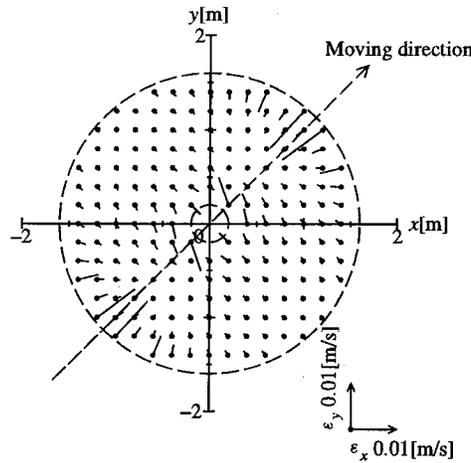


Fig. 3.5 Linearization errors in the working coordinates for various initial points (dots: initial positions of the robot arm; bars from the dots: linearization error vector  $(\varepsilon_x, \varepsilon_y)$ ).

からでていた線分)を(3.17)式で計算した結果を示す。アームの長さは  $l_1 = 0.7[\text{m}]$ ,  $l_2 = 0.9[\text{m}]$ , 移動速度は  $v_x = 0.1[\text{m/s}]$ ,  $v_y = 0.1[\text{m/s}]$  とし, ロボットアームの逆キネマティクス(3.2b)式の符号は常に正とした。Fig. 3.5から, 作業線形モデルの近似精度は, ロボットアームの移動方向において, 可動領域の限界近くになるほど悪くなるのがわかる。また, アームが縮んでいる領域では, 作業線形近似誤差が大きくなっている。作業線形近似誤差は, アームの作業座標系における絶対的な位置に依存するのではなく, アームの姿勢に依存しているため, Fig. 3.5の作業線形近似誤差の結果は, 矢印の方向にロボットアームを動かした場合を表すのみではなく, どの方向に移動させる場合においても, Fig. 3.5を原点について回転させ, アームの移動方向を図中の矢印の方向に合わせると, その移動方向に対する作業線形近似誤差の位置依存項を表す。

次に, 視点を変えて, ロボットアームの一つの始点(原点からアームの先端位置までの距離  $r = \sqrt{x_0^2 + y_0^2}$ で記述される)から, いろいろな移動方向によって作業線形近似誤差がどのように変化するかをみる。  $r = 0.25[\text{m}]$ ,  $0.38[\text{m}]$ ,  $1.5[\text{m}]$ ,  $1.55[\text{m}]$  の4点において, 移動速度  $v = \sqrt{v_x^2 + v_y^2} = \sqrt{0.02} \approx 0.141[\text{m/s}]$  で, 初期位置を中心として一周  $2\pi$ の各移動方向にアームを移動させた場合の, 作業線形近似誤差の位置依存項の大きさ  $\sqrt{\varepsilon_x^2 + \varepsilon_y^2}$ を計算した結果を Fig. 3.6に示す。 Fig. 3.6の横軸  $\theta$ がアームを移動させる角度を表しており, アームを伸ばす方向を角度の基準  $\theta = 0[\text{rad}]$ にとり,  $\theta = \pi[\text{rad}]$ がアームを縮める方向を表す。 Fig. 3.6から, アームの可動領域 ( $0.2[\text{m}] \leq r \leq 1.6[\text{m}]$ )の限界近くである  $r = 0.25[\text{m}]$ と  $1.55[\text{m}]$ において, アームが伸縮する動作において作業線形近似誤差が大きくなり, 伸縮方向に垂直方向に移動する場合には, 作業線形近似誤差は比較的小さいことがわかる。

作業線形近似誤差(3.18)式は時間にも依存しているため, 時間に依存する項  $\delta(t)$  の

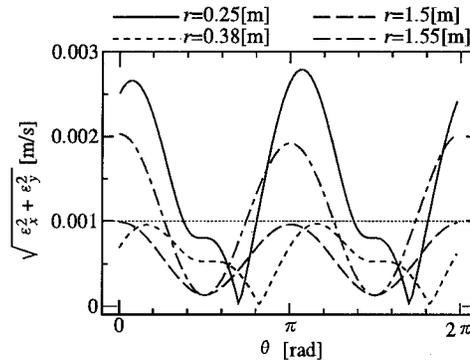


Fig. 3.6 Linearization errors for the different moving directions  $\theta$  and the positions  $r$  of the robot arm ( $r = 0.25[\text{m}]$ ,  $r = 0.38[\text{m}]$ ,  $r = 1.5[\text{m}]$ ,  $r = 1.55[\text{m}]$ ).

$K_p = 15[1/\text{s}]$ とした場合の時間推移を Fig. 3.7に示す. 微小時間  $T = 0.02[\text{s}]$ での $\delta(t)$ は $0.0027[\text{s}]$ であり, Fig. 3.6より $0.38[\text{m}] \leq r \leq 1.5[\text{m}]$ の範囲ではアームをどの方向に移動させても, 作業線形近似誤差の位置依存項の大きさ $\sqrt{\epsilon_x^2 + \epsilon_y^2}$ は $0.001[\text{m/s}]$ 以下であるので, 作業線形近似誤差の最大値を計算すると $0.0027[\text{mm}]$ となり, この値は微小時間  $T = 0.02[\text{s}]$ での微小区間の長さ $0.141[\text{m/s}] \times 0.02[\text{s}] = 0.00282[\text{m}]$ の約 $0.1\%$ と非常に小さい値である. すなわち, ロボットアームの移動速度が $0.141[\text{m/s}]$ で目標軌道の分割における微小時間が $0.02[\text{s}]$ の場合に, 目標軌道の分割の1区間において作業線形近似誤差が目標軌道の $0.1\%$ 以内である作業線形近似可能領域は $0.38[\text{m}] \leq r \leq 1.5[\text{m}]$ ということができる.

一般的なロボットアームに対する作業線形近似可能領域の導出手順を整理する. ロボットアームのリンク長 $l_1, l_2$ , 位置ループゲイン $K_p$ , 目標軌道の分割における微小時間 $T$ , アームの移動の速さ $v$ , 原点からアームの先端位置までの距離 $r$ (一般性を失うことなく $x$ 軸上にアームの先端を置くものとする), 移動方向を $\theta$ としたときの作業線形近似誤差の大きさは, 以下に示す方法で求めることができる.

1.  $(x_0, y_0) = (r, 0)$ ,  $(x_T, y_T) = (x_0 + vT \cos \theta, y_0 + vT \sin \theta)$ とする.
2. 逆キネマティクス(3.2)式を用いて $(\alpha_0, \beta_0)$ ,  $(\alpha_T, \beta_T)$ を求める.
3. 作業座標系の移動速度 $(v_x, v_y) = (v \cos \theta, v \sin \theta)$ , 関節座標系の移動速度 $(v_\alpha, v_\beta) = ((\alpha_T - \alpha_0)/T, (\beta_T - \beta_0)/T)$ を計算する.

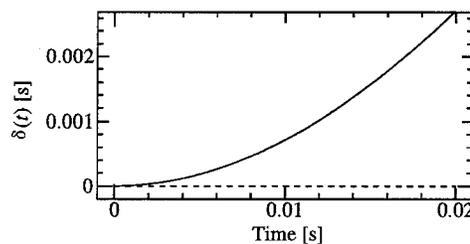


Fig. 3.7 Time dependency of the linearization error  $\delta(t)$ .

4. (3.17) 式を用いて作業線形近似誤差の位置依存項  $(\varepsilon_x, \varepsilon_y)$  を求め、その大きさ  $\sqrt{\varepsilon_x^2 + \varepsilon_y^2}$  を計算する.
5. (3.8) 式を用いて作業線形近似誤差の時間依存項  $\delta(T)$  を計算する.
6. 作業線形近似誤差の大きさ  $\sqrt{\varepsilon_x^2 + \varepsilon_y^2} \delta(T)$  を計算する.

作業線形近似可能領域を、目標軌道の1区間において作業線形近似誤差が目標軌道の分割された微小区間  $vT$  の  $p\%$  以下となる領域と定義すると、作業線形近似可能領域は (i) の原点からアームの先端位置までの距離  $r$  をアームの可動領域で変化させたもとで、(ii) のアームの移動方向  $\theta = 0 \sim 2\pi$  で作業線形近似誤差の大きさ  $\sqrt{\varepsilon_x^2 + \varepsilon_y^2} \delta(T)$  を計算し、その大きさが設定した許容範囲  $pvT/100$  以下であるか否かを判定することにより求めることができる.

### 3.3.3 作業線形モデルの誤差の累積

今までの評価は、目標軌道の細分割において、分割された1区間について線形近似が妥当であるかを評価し、作業線形近似可能領域の導出法を示した. 実際の目標軌道は分割軌道が繋ぎ合わさったものからなり、軌道全体としてみたときの区間ごとの作業線形近似誤差が、時間の経過と共にどのように累積するかを調べる.

(a) 1区間での作業線形近似誤差の推移

3.2.2節と3.2.3節においては、分割された微小区間の始点において目標軌道、実軌道と作業線形近似軌道の位置がすべて同じであるものとして、実軌道と作業線形近似軌道の導出を行ったが、この節では、始点での目標軌道、実軌道と作業線形近似軌道がそれぞれ違う値をとっている場合の、1区間における作業線形近似誤差を調べる. 初期時刻の目標軌道の位置を、作業座標と関節座標でそれぞれ  $(u_{x_0}, u_{y_0})$ ,  $(u_{\alpha_0}, u_{\beta_0})$ , 実軌道を、作業座標と関節座標でそれぞれ  $(x_0, y_0)$ ,  $(\alpha_0, \beta_0)$  とし、作業線形近似軌道の位置を作業座標で  $(\hat{x}_0, \hat{y}_0)$  とする.

ロボットアームの実軌道は、(3.3) 式に (3.6) 式を代入し、初期条件  $(\alpha_0, \beta_0)$  を用いて  $(\alpha(t), \beta(t))$  について解き、(3.1) 式に代入すると

$$x(t) = l_1 \cos(\alpha_0 + (u_{\alpha_0} - \alpha_0)\sigma(t) + v_\alpha \delta(t)) + l_2 \cos(\alpha_0 + \beta_0 + (u_{\alpha_0} + u_{\beta_0} - \alpha_0 - \beta_0)\sigma(t) + (v_\alpha + v_\beta)\delta(t)) \quad (3.19a)$$

$$y(t) = l_1 \sin(\alpha_0 + (u_{\alpha_0} - \alpha_0)\sigma(t) + v_\alpha \delta(t)) + l_2 \sin(\alpha_0 + \beta_0 + (u_{\alpha_0} + u_{\beta_0} - \alpha_0 - \beta_0)\sigma(t) + (v_\alpha + v_\beta)\delta(t)) \quad (3.19b)$$

となる. ここで

$$\sigma(t) = 1 - e^{-K_p t} \quad (3.20)$$

である. 作業線形近似軌道は、(3.10) 式に (3.11) 式を代入し、初期条件  $(\hat{x}_0, \hat{y}_0)$  を用いて  $(\hat{x}(t), \hat{y}(t))$  について解くと

$$\hat{x}(t) = \hat{x}_0 + (u_{x_0} - \hat{x}_0)\sigma(t) + v_x \delta(t) \quad (3.21a)$$

$$\hat{y}(t) = \hat{y}_0 + (u_{y_0} - \hat{y}_0)\sigma(t) + v_y \delta(t) \quad (3.21b)$$

と計算される。実軌道と作業線形近似軌道の誤差を求めると、(3.21)式と(3.19)式との差から

$$\begin{aligned}
 e_x(t) &= \hat{x}(t) - x(t) \\
 &= \hat{x}(t) - \tilde{x}(t) + o(\sigma(t)) \\
 &= (\hat{x}_0 - x_0)e^{-K_p t} + \varepsilon_x \delta(t) + \{(u_{x_0} - x_0) + y_0(u_{\alpha_0} - \alpha_0) \\
 &\quad + l_2 \sin(\alpha_0 + \beta_0)(u_{\beta_0} - \beta_0)\}\sigma(t) + o(\sigma(t)) \tag{3.22a}
 \end{aligned}$$

$$\begin{aligned}
 e_y(t) &= \hat{y}(t) - y(t) \\
 &= \hat{y}(t) - \tilde{y}(t) + o(\sigma(t)) \\
 &= (\hat{y}_0 - y_0)e^{-K_p t} + \varepsilon_y \delta(t) + \{(u_{y_0} - y_0) + x_0(u_{\alpha_0} - \alpha_0) \\
 &\quad - l_2 \cos(\alpha_0 + \beta_0)(u_{\beta_0} - \beta_0)\}\sigma(t) + o(\sigma(t)) \tag{3.22b}
 \end{aligned}$$

となる。ここで、式変形中  $(\tilde{x}(t), \tilde{y}(t))$  は、実軌道(3.19)式のテイラー展開1次近似

$$\begin{aligned}
 \tilde{x}(t) &= x_0 + l_1 \sin \alpha_0 \{(u_{\alpha_0} - \alpha_0)\sigma(t) + v_\alpha \delta(t)\} \\
 &\quad + l_2 \sin(\alpha_0 + \beta_0) \{(u_{\alpha_0} + u_{\beta_0} - \alpha_0 - \beta_0)\sigma(t) + (v_\alpha + v_\beta)\delta(t)\} \tag{3.23a}
 \end{aligned}$$

$$\begin{aligned}
 \tilde{y}(t) &= y_0 - l_1 \cos \alpha_0 \{(u_{\alpha_0} - \alpha_0)\sigma(t) + v_\alpha \delta(t)\} \\
 &\quad - l_2 \cos(\alpha_0 + \beta_0) \{(u_{\alpha_0} + u_{\beta_0} - \alpha_0 - \beta_0)\sigma(t) + (v_\alpha + v_\beta)\delta(t)\} \tag{3.23b}
 \end{aligned}$$

である。(3.22)式における第1項は、初期時刻での実軌道  $(x_0, y_0)$  と作業線形近似軌道  $(\hat{x}_0, \hat{y}_0)$  との違いによる項で、第2項は、3.3.1節において目標軌道と実軌道と作業線形近似軌道が等しいものとして導出した作業線形近似誤差(3.18)式であり、第3項は、目標軌道と実軌道との初期時刻における位置が違ふことによって生じた誤差項で、第4項は、(3.23)式のテイラー展開1次近似による誤差項である。

初期時刻における作業線形近似誤差  $(\hat{x}_0 - x_0, \hat{y}_0 - y_0)$  は、(3.22)式中の最後の式における第1項により、時間と共に指数的に減衰していることがわかる。

#### (b) 作業線形近似誤差の累積

先の議論により、作業線形近似誤差の時間推移の特性は  $x$  座標と  $y$  座標共に等しいので、ここでは  $x$  座標についてのみ考察する。作業線形近似誤差が微小間隔  $T$  で増加するときの上限値を、三角不等式を用いて調べる。目標軌道の分割の微小区間  $T$  での作業線形近似誤差の大きさは、(3.22)式より

$$\begin{aligned}
 |e_x(T)| &= |(\hat{x}_0 - x_0)e^{-K_p T} + \varepsilon_x \delta(T) \\
 &\quad + \{(u_{x_0} - x_0) + y_0(u_{\alpha_0} - \alpha_0) + l_2 \sin(\alpha_0 + \beta_0)(u_{\beta_0} - \beta_0)\}\sigma(T) + o(\sigma(T))| \\
 &\leq E_0 e^{-K_p T} + E_1 \sigma(T) \tag{3.24}
 \end{aligned}$$

で抑さえられる。ここで、 $E_0 = |\hat{x}_0 - x_0|$ 、 $E_1$  は1区間で新たに生じた作業線形近似誤差の大きさを表す正の定数であり、最後の式変形においては、 $\delta(T) = o(\sigma(T))$  を用いている。

同様にして、目標軌道の分割の第  $N$  区間における作業線形近似誤差の大きさは

$$|e_x(NT)| \leq |e_x((N-1)T)|e^{-K_p T} + E_N \sigma(T) \quad (3.25)$$

で抑えることができる。(3.25) 式を逐次用いることによって、第  $N$  区間における作業線形近似誤差の大きさの上限を、初期値からの作業線形近似誤差の累積によるものとして表現すると

$$\begin{aligned} |e_x(NT)| &\leq |e_x((N-1)T)|e^{-K_p T} + E_N \sigma(T) \\ &\leq (|e_x((N-2)T)|e^{-K_p T} + E_{N-1} \sigma(T))e^{-K_p T} + E_N \sigma(T) \\ &\leq E_0 e^{-NK_p T} + E_1 \sigma(T) e^{-(N-1)K_p T} + \dots + E_N \sigma(T) \\ &\leq E_0 e^{-NK_p T} + E_{max} \sigma(T) (e^{-(N-1)K_p T} + e^{-(N-2)K_p T} + \dots + 1) \\ &= E_0 e^{-NK_p T} + E_{max} \sigma(T) \frac{1 - e^{-NK_p T}}{1 - e^{-K_p T}} \\ &= E_0 e^{-NK_p T} + E_{max} (1 - e^{-NK_p T}) \end{aligned} \quad (3.26)$$

となる。ここで、 $E_{max} = \max(E_1, E_2, \dots, E_N)$  であり、導出過程で (3.20) 式の  $\sigma(T)$  の値を用いた。(3.26) 式の第 1 項が初期時刻における作業線形近似誤差の影響を表しており、第 2 項が目標軌道の分割の区間ごとに生じた作業線形近似誤差の累積値を表している。(3.26) 式より、分割数  $N$  が大きくなっても、作業線形近似誤差は発散することではなく、有限確定値  $E_{max}$  に収束することがわかる。この  $E_{max}$  は、3.3.1 節において、目標軌道と実軌道と作業線形近似軌道が等しいものとして導出した作業線形近似誤差 (3.18) 式と、目標軌道と実軌道との違いによって生じた誤差によって定まる定数である。また、ロボットアームのダイナミクス特性を良くする、すなわち、位置ループゲイン  $K_p$  を大きくしても、作業線形近似誤差の累積値の上限は  $E_{max}$  となることがわかる。また、目標軌道の時間  $NT$  を一定としたもとの、目標軌道の分割数を多くし、分割時間を短くする、すなわち、 $N \rightarrow \infty, T \rightarrow 0$  としても、作業線形近似誤差の累積値の上限は、(3.26) 式により変化しないことがわかる。この誤差の上限を評価すると、3.3.2 節と同じ条件でロボットアームを  $x$  軸の正方向に  $(-0.8[\text{m}], 0.8[\text{m}])$  から  $(-0.7[\text{m}], 0.8[\text{m}]) \rightarrow 0.1[\text{m/s}]$  で移動させた場合に、終点における実際の作業線形近似誤差は  $6.89 \times 10^{-3}[\text{mm}]$  で、(3.26) 式から計算される誤差の上限は  $6.10 \times 10^{-2}[\text{mm}]$  となる。誤差の上限で見積もった場合においても作業線形近似誤差は無視できる大きさである。

### 3.3.4 多軸ロボットアームへの拡張

今までの議論では、2 軸ロボットアームについて作業線形近似可能領域を導出した。この節では、2 軸ロボットアームから多軸ロボットアームに拡張した場合についての、作業線形近似可能領域について考察する。第 3 軸が  $z$  軸方向の直動となるスカラ型ロボットについては、作業線形近似可能領域は 2 軸ロボットアームの作業線形近似可能領域を  $z$  軸方向に平行移動させた領域となる。第 4 軸はエンドエフェクタ自身の回転であるので、作業線形近似する必要はない。

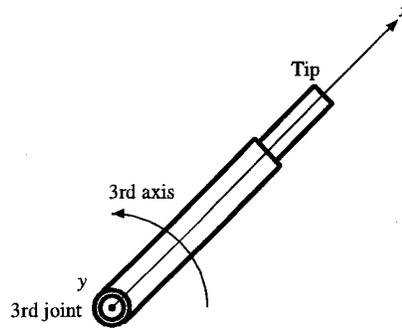


Fig. 3.8 The third axis and  $x$  axis of a three-degree-of-freedom robot arm.

次に、6軸ロボットアームの作業座標における位置決めを行なう、ベースからの3軸について考える。この6軸ロボットアームの第3軸は、Fig. 3.2で表される2軸ロボットアームの $y$ 軸を軸として回転するように採られる。従って、この3軸からなるロボットアームの可動領域は、中心部が空洞の球となる。第3軸と $x$ 軸で作られる平面でのロボットアームをFig. 3.8に示す。この第3軸と $x$ 軸で作られる平面における線形近似と、前節で議論した2軸ロボットアームの線形近似の違いは、前者は1軸が回転でもう1軸が直動であるのに対して、後者は2軸とも回転であるという点のみである。すなわち、前節で論じた2軸ロボットアームの線形近似というのは、2軸の回転から2軸の直動への変換が可能であることを示したことになるので、第3軸と $x$ 軸で作られる平面での1軸が回転でもう1軸が直動から2軸の直動への変換は、前節で論じた2軸ロボットアームの線形近似の議論と同様になる。すなわち、第3軸と $x$ 軸で作られる平面でのロボットアームは作業座標系で線形近似可能であり、3軸のロボットアームの作業線形近似可能領域は、2軸ロボットアームの作業線形近似可能領域を $y$ 軸を軸として回転した領域となる。ハンドに近い3軸については、動作する作業空間が手首を中心とする球の表面と、エンドエフェクタ自身の回転となるため、作業線形近似する必要はない。

### 3.4 計算機シミュレーションと実験結果

今まで議論してきた作業線形近似が、ロボットアームの制御性能に如何に関わるかを見るために、計算機シミュレーションを行った。シミュレーションに用いたロボットアームは  $l_1 = 0.7[\text{m}]$ ,  $l_2 = 0.9[\text{m}]$ ,  $K_p = 15[1/\text{s}]$  とし、目標軌道は速度  $0.25[\text{m/s}]$  で  $y$ 軸方向に  $0.15[\text{m}]$  移動し、その後  $x$ 軸方向に  $0.15[\text{m}]$  移動する直線軌道とした。目標軌道として作業線形近似誤差が  $0.2\%$ 以内である作業線形近似可能領域  $0.5[\text{m}] \leq r \leq 1.45[\text{m}]$  内の  $(x_0, y_0) = (-0.8[\text{m}], 0.65[\text{m}])$  と可能領域外  $(x_0, y_0) = (-1.13137[\text{m}], 0.98137[\text{m}])$  の二通りについてシミュレーションを行った。目標軌道の分割間隔は  $T = 20[\text{ms}]$  とした。作業線形近似可能領域内の頂点  $(x, y) = (-0.8[\text{m}], 0.8[\text{m}])$  における作業線形近似誤差は  $(\varepsilon_x, \varepsilon_y) = (0.00068[\text{m/s}], -0.00009$

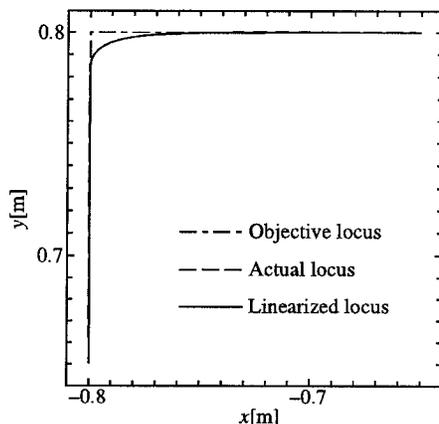


Fig. 3.9 A simulation result of the actual trajectory and the linearization model trajectory of the two-degree-of-freedom robot arm inside of the linearizable region.

[m/s]) であり，目標軌道 1 区間で生じる作業線形近似誤差は，分割された目標軌道 0.002[m] の 0.037% と非常に小さい値である．作業線形近似可能領域外の頂点  $(x, y) = (-1.13137[m], 1.13137[m])$  における作業線形近似誤差は  $(\varepsilon_x, \varepsilon_y) = (0.00000[m/s], 0.25000[m/s])$  であり，目標軌道 1 区間で生じる作業線形近似誤差は，分割された目標軌道 0.002[m] の 13.5% と大きな値となっている．

Fig. 3.9に，作業線形近似可能領域内における実軌跡と線形近似した際の応答軌跡，Fig. 3.10に，作業線形近似可能領域外における実軌跡と線形近似した際の応答軌跡を示す． Fig. 3.9の作業線形近似可能領域内においては，実軌跡と線形近似した際の応答軌跡は図上では一致しており，両者の誤差は最大でも 0.2[mm] であり，無視

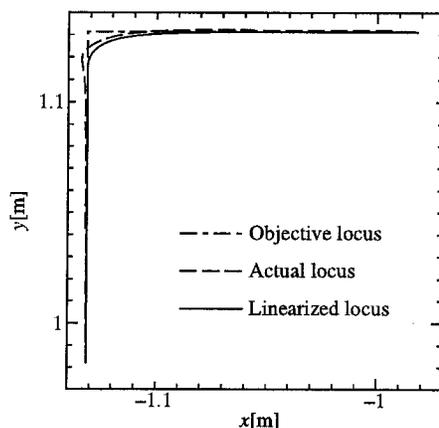


Fig. 3.10 A simulation result of the actual trajectory and the linearization model trajectory of the two-degree-of-freedom robot arm outside of the linearizable region.

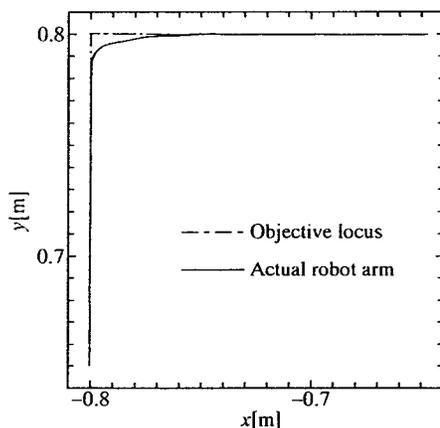


Fig. 3.11 An experimental result of the actual trajectory and the linearization model trajectory of the six-degree-of-freedom robot arm inside of the linearizable region.

できる大きさである。それに対して、Fig. 3.10の作業線形近似可能領域外においては、実軌跡と線形近似した際の応答軌跡はずれており、両者の誤差は最大2.7[mm]でありかなり大きくなっている。また、作業線形近似可能領域外においては、実軌道にオーバーシュートが生じており、ロボットアーム自身の制御性能も劣化している。すなわち、作業線形近似可能領域内においては、ロボットアームの実軌道と作業線形近似軌道がほぼ等しく取り扱うことができるため、ロボットアームの制御性能は作業座標系において評価することが可能であるが、作業線形近似可能領域外においては、ロボットアームの作業座標系での評価は困難となり、制御性能も作業線形モデルで表される制御性能から劣化し、オーバーシュートなどが生じる。

次に、線形近似モデルの妥当性を示すために、6軸産業用ロボットアーム(モートマンK3S, 最大負荷3[kg])で輪郭制御実験を行った。Fig. 3.11に実験結果を示す。この実験結果を見ると、Fig. 3.9の作業線形モデルでのシミュレーション結果とほぼ一致していることがわかる。このことから、本章で導いた作業線形近似可能領域内では、産業用ロボットアームは作業線形モデルで表し得ることを、実験でも検証したことになる。

### 3.5 まとめ

産業界で用いられている多関節ロボットアームが、作業座標系において各座標軸で線形独立なモデルである作業線形モデルで表現するときの作業線形近似誤差を理論的解析により求め、作業線形近似可能領域の導出法を明らかにした。作業線形近似可能領域内においては、産業用ロボットアーム制御系は、作業座標系において各座標軸独立の線形近似可能であり、作業線形近似誤差は非常に小さく、作業線形近似誤差が時間と共に累積して発散しないことが理論的に導出でき、作業線形近似可能領域外ではロボットアームの制御性能が劣化することがシミュレーションにより

確認された。また、実機ロボットアームによる実験結果においても、作業線形近似可能領域内では、作業線形モデルによる計算機シミュレーション結果とほぼ一致することが確認された。

ロボットアームの作業線形近似可能領域においては、ロボットアーム制御系を作業座標系で各座標軸独立の線形の制御対象としてとらえ、その制御対象に対する制御器の構成において既存の線形制御理論を適用することにより、より高速高精度のロボットアーム制御を行うことが可能となる。作業線形近似可能領域外で高精度制御を行うためには、作業線形モデルを補正するような非線形制御の必要性を示唆している。

## 第4章

# 産業用多関節ロボットアームの高速輪郭制御における軌跡誤差の解析

### 4.1 まえがき

近年、産業用ロボットアームの作業速度の高速化に対する要求が高くなってきている。産業用ロボットアームの輪郭制御において、低速動作時には目標軌跡と追従軌跡が一致するが、高速動作時には目標軌跡から追従軌跡がずれることを経験する。実際の産業用ロボットアームを動作させると、目標速度が1[m/s]のときに追従軌跡の目標軌跡からの誤差(軌跡誤差)は2~4[mm]程度生じることが確認されている。

高速時におけるロボットアームの制御については、ダイレクトドライブマニピュレータの逆システムを用いてダイナミクスの遅れを補償する手法<sup>[28]</sup>や、教示信号修正法(5章, 6章, 7章参照)など様々な手法が提案されている<sup>[24, 29, 30, 31]</sup>。しかしながら、産業用多関節ロボットアームの軌跡誤差の原因や、軌跡誤差と作業速度との関係についての研究はなされていない。

本章では、産業用多関節ロボットアームの軌跡誤差の原因を究明し、また、軌跡誤差の大きさと目標速度、目標軌跡の関係について明らかにする。

### 4.2 産業用多関節ロボットアームの高速動作時における問題点

#### 4.2.1 産業用多関節ロボットアームの制御手順

産業用多関節ロボットアームの輪郭制御は、以下の手順にしたがって行なわれる(Fig. 4.1参照)。ここで、本章においては、問題を単純化するために、作業座標系における目標軌道の分割については、十分小さいものとして無視することにする。

- (i) 作業座標系において目標軌道を与える。
- (ii) 逆キネマティクスを用いて、作業座標系における目標軌道を関節座標系における目標関節角に変換する。
- (iii) アクチュエータであるサーボモータが目標関節角に追従するように、各軸独立に制御される。
- (iv) キネマティクスによって、関節座標系における追従関節角が作業座標系における追従軌道に変換される。

上の4ステップにしたがって、ロボットアームが目標軌道に追従する。

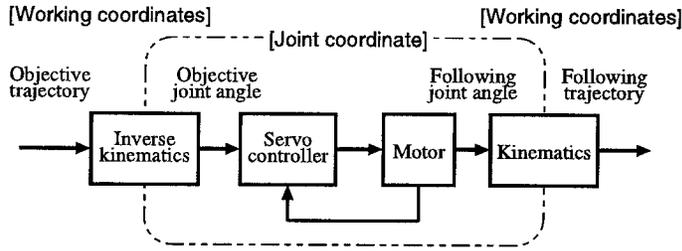


Fig. 4.1 Control structure of a servo motor of actuators of the industrial articulated robot arm; the servo motor of the robot arm are controlled for each link, independently.

#### 4.2.2 高速動作時における軌跡誤差

実際の産業用ロボットアームの輪郭制御を高速動作で行なう場合に、輪郭制御性能が劣化する例を示す。Fig. 4.2は、6自由度の産業用ロボットアーム（モートマン K10S, 安川電機製; 最大負荷 10[kg], 最大高さ 1940[mm], 最大稼働半径 1555[mm], 重量 300[kg]) で、目標速度 1.0[m/s] で直線軌道を描かせたときの、目標軌跡と追従軌跡との誤差である軌跡誤差を示している。ここで、軌跡とは時間概念を取りロボットアームが移動した経路を指し、軌道はロボットアームの先端の位置の時間経過を指す。Fig. 4.2を見ると、目標軌跡が直線であるにもかかわらず、高速動作時には追従軌跡が目標軌跡から劣化して曲線になっていることがわかる。このときの最大軌跡誤差は 2.5[mm] であった。このような軌跡誤差は、産業用ロボットアームで作業を行なう際に、製品の精度に直接悪影響を与えるため、非常に大きな問題となる。軌跡誤差は輪郭制御での作業精度を表す最も重要な指標となるために、本章においては、制御性能の指標として軌跡誤差を用いる。また、本章での目標軌道は、産業用ロボットアームにおいて最も基本的な軌道である直線軌道とした。

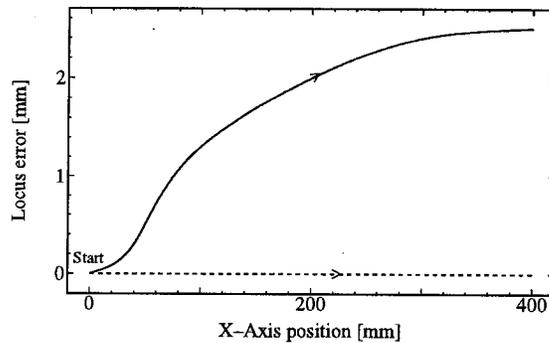


Fig. 4.2 Experimental result of the locus error of an industrial articulated robot arm of six-degree-of-freedom; the dashed line is the objective locus and the solid line was the following locus when the objective locus was a straight line and the objective velocity was 1[m/s]

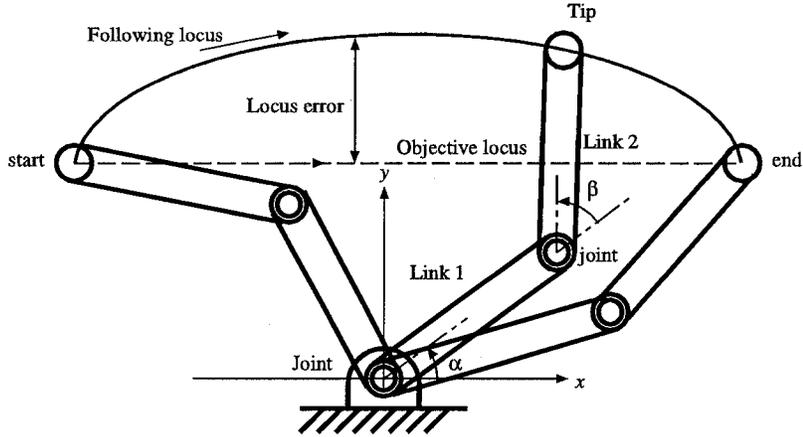


Fig. 4.3 Locus error of an industrial articulated robot arm of two-degree-of-freedom; bold line: links of the robot arm, ⊙: joints of the robot arm,  $(\alpha, \beta)$ : joint angle.

### 4.3 産業用多関節ロボットアームの数式モデル

#### 4.3.1 キネマティクス

Fig. 4.3に示す2軸の多関節ロボットアームに基づいて解析を行なう。図中の⊙が関節，破線が目標軌跡，実線が追従軌跡を示す。1軸と2軸の長さをそれぞれ  $l_1$  と  $l_2$  とする。関節座標系における関節角  $(\alpha, \beta)$  は，作業座標系における位置  $(x, y)$  からキネマティクス

$$x = l_1 \cos \alpha + l_2 \cos(\alpha + \beta) \tag{4.1a}$$

$$y = l_1 \sin \alpha + l_2 \sin(\alpha + \beta) \tag{4.1b}$$

で変換され，作業座標系における位置  $(x, y)$  は，関節座標系における関節角  $(\alpha, \beta)$  から逆キネマティクス

$$\alpha = \sin^{-1} \left( \frac{y}{\sqrt{x^2 + y^2}} \right) - \sin^{-1} \left( \frac{l_2 \sin \beta}{\sqrt{x^2 + y^2}} \right) \tag{4.2a}$$

$$\beta = \pm \cos^{-1} \left( \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \tag{4.2b}$$

で変換される。

#### 4.3.2 ダイナミクス

産業用多関節ロボットアームのアクチュエータであるサーボモータは，関節座標系において各軸独立に制御される。Fig. 4.4に，関節座標系におけるロボットアームのダイナミクスのブロック線図を示す。高速動作時におけるロボットアームの挙動を調べるために，関節座標系におけるダイナミクスは2次系モデル

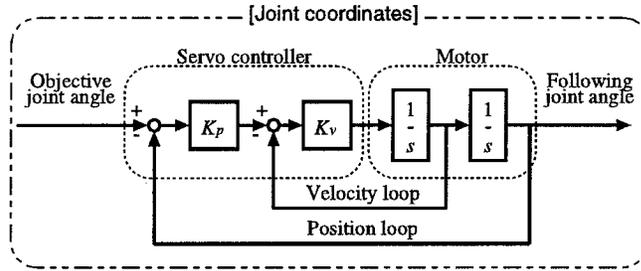


Fig. 4.4 Robot arm dynamics in the joint coordinates;  $K_p$ : position loop gain,  $K_v$ : velocity loop gain.

$$\ddot{\alpha}(t) = -K_v \dot{\alpha}(t) - K_p K_v \alpha(t) + K_p K_v u_\alpha(t) \quad (4.3a)$$

$$\ddot{\beta}(t) = -K_v \dot{\beta}(t) - K_p K_v \beta(t) + K_p K_v u_\beta(t) \quad (4.3b)$$

で表現する．ここで， $(\alpha(t), \beta(t))$  は追従関節角， $(u_\alpha(t), u_\beta(t))$  は目標関節角， $K_p$  と  $K_v$  はサーボコントローラの位置ループゲインと速度ループゲインを示す<sup>[22]</sup>．

産業用多関節ロボットアームにおいては，機械的遊び，指令時間間隔，重力やコリオリ力などの非線形項，負荷イナーシャの変化など，制御性能を劣化させる多くの原因がある．しかし，上に挙げた原因がない理想的な状況においても，高速動作時における軌跡誤差が存在する．次の節で，(4.1) 式と (4.3) 式で表される理想的な数式モデルを基に，軌跡誤差の原因を究明する．

## 4.4 産業用多関節ロボットアームの軌跡誤差

### 4.4.1 軌跡誤差の定義

作業座標系における追従軌跡  $(x, y)$  は，追従軌道  $(x(t), y(t))$  から時間  $t$  を消去することによって得られる．軌跡誤差は目標軌跡  $(u_x, u_y)$  と追従軌跡  $(x, y)$  の距離によって定義する．作業座標を回転させることによって，一般性を失うことなく目標軌跡を X 軸に平行におく．このとき，目標軌跡の Y 軸の位置は  $u_y = y_0$  と一定となるため，軌跡誤差は

$$\delta y = y - y_0, \quad x_0 \leq x \leq x_T \quad (4.4)$$

となる．ここで， $x_0$  と  $x_T$  は，それぞれ目標軌跡の始点と終点を表す．以下において軌跡誤差 (4.4) 式を導出する．

### 4.4.2 直線目標軌道に対するロボットアームの動作

追従誤差の原因を調べるために，直線目標軌道に対するロボットアームの動作の計算機シミュレーションを行なう．ロボットアームのリンクの長さは  $l_1 = 0.7[\text{m}]$ ， $l_2 = 0.9[\text{m}]$  で，位置ループゲインと速度ループゲインはそれぞれ  $K_p = 10[1/\text{s}]$ ， $K_v = 50[1/\text{s}]$  とする．目標軌道は

$$u_x(t) = \begin{cases} 1.0t - 0.5 & [\text{m}] \quad (0 \leq t < 1[\text{s}]) \\ 0.5 & [\text{m}] \quad (1 \leq t \leq 1.5[\text{s}]) \end{cases} \quad (4.5a)$$

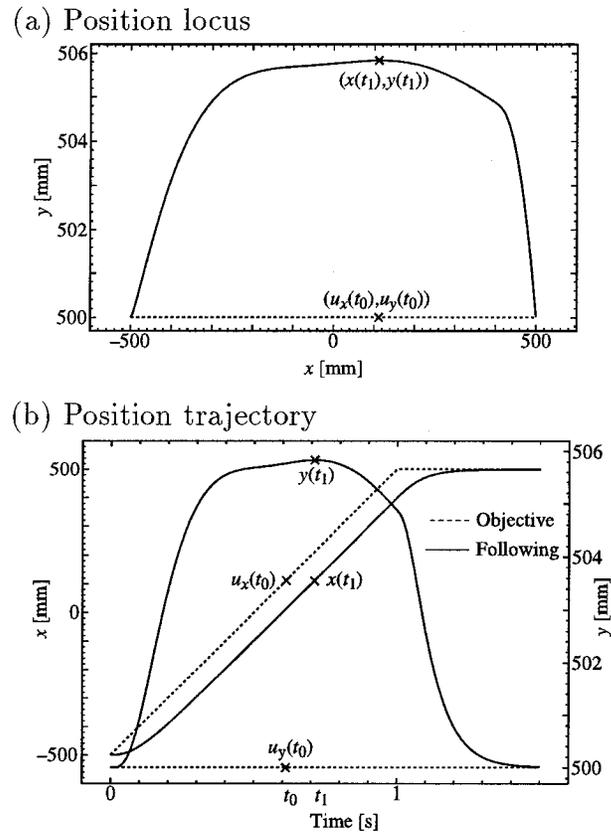
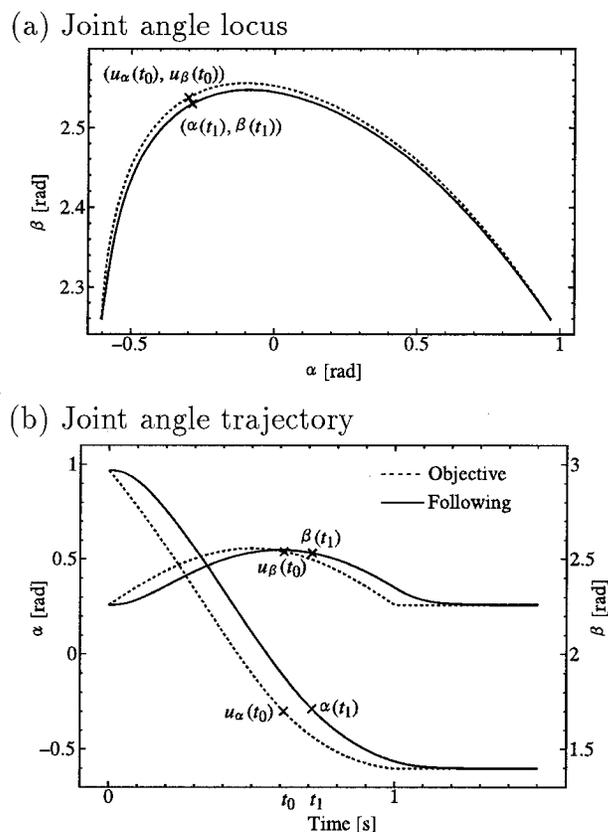


Fig. 4.5 Position trajectory and locus of the two-degree-of-freedom robot arm in the working coordinates; the maximum locus error was the distance between  $(x(t_1), y(t_1))$  and  $(u_x(t_0), u_y(t_0))$ ,  $\delta y_{max} = y(t_1) - u_y(t_0) = 5.8[\text{mm}]$  at  $t_0 = 0.615$  and  $t_1 = 0.714[\text{s}]$ .

$$u_y(t) = 0.5 \text{ [m]} \tag{4.5b}$$

と与える. Fig. 4.5(a), (b)に, それぞれ, 作業座標系における追従軌跡と追従軌道を示す. Fig. 4.5(a)から, 最大軌跡誤差は  $(x(t_1), y(t_1))$  と  $(u_x(t_0), u_y(t_0))$  の距離, すなわち,  $\delta y_{max} = y(t_1) - u_y(t_0) = 5.8[\text{mm}]$  であることがわかる. ここで,  $t_0 = 0.615$ ,  $t_1 = 0.714[\text{s}]$  であり, 最大軌跡誤差が生じる追従軌道の時間  $t_1$  は, 目標軌道の時間  $t_0$  とは異なっている. これは, 軌道誤差ではなく軌跡誤差を輪郭制御性能の指標として用いているためである.

Fig. 4.5(b)において, X軸の追従軌道は, (4.3)式で表される2次系モデルのランプ応答とほぼ等しいが, Y軸の追従軌道は, 目標軌道が  $u_y(t) = 500[\text{mm}], 0 \leq t \leq 1.5[\text{s}]$  と一定であるにも関わらず, 一定とはなっていない. これは, ロボットアームのダイナミクス(4.3)式は, 作業座標系ではなく関節座標系におけるダイナミクスを表し, 作業座標系においては, 作業座標系と関節座標系との非線形変換により非線形のダイナミクスとなるためである.



**Fig. 4.6** Joint angle trajectory and locus of the two-degree-of-freedom robot arm in the joint coordinates; the maximum locus error occurred when the following joint angle was  $(\alpha(t_1), \beta(t_1))$  and the objective joint angle was  $(u_\alpha(t_0), u_\beta(t_0))$ .

追従関節角の軌跡と軌道を Fig. 4.6(a), (b) に示す。Fig. 4.6(a) から、関節座標系において、追従軌跡は目標軌跡から劣化していることがわかる。これは、関節座標系において追従軌跡が目標軌跡から遅れているためである (Fig. 4.6(b) 参照)。

計算機シミュレーション結果から、軌跡誤差発生メカニズムは以下のように説明できる。

- (i) 作業座標系において目標軌道  $u_x(t)$  と  $u_y(t)$  の関係が線形であっても、関節座標系において目標関節角  $u_\alpha(t)$  と  $u_\beta(t)$  の関係は非線形となる。これは、作業座標系から関節座標系への変換 (4.2) 式が非線形変換であるためである。
- (ii) 関節座標系におけるロボットアームのダイナミクス (4.3) 式は、ゲイン劣化と位相遅れが存在する。すなわち、定常状態において追従関節角の軌道の時間を  $t_1 - t_0$  だけ進ませても、目標関節角の軌道とは一致しない。このゲイン劣化によって、関節座標系において追従軌跡が目標軌跡から劣化する (Fig. 4.6(a) 参照)。
- (iii) 関節座標系における軌跡誤差の座標変換によって、作業座標系における軌跡誤

差が生じる (Fig. 4.5(a) 参照).

#### 4.4.3 軌跡誤差の原因

作業座標系と関節座標系の座標変換が線形であれば, 追従軌道は目標軌道から遅れても目標軌跡と追従軌跡は一致する. また, ロボットアームのダイナミクスに遅れが存在しなければ, 座標変換が非線形変換であっても, 目標軌跡と追従軌跡は一致する. このことから, 軌跡誤差の原因は以下のようにまとめることができる.

- (i) 作業座標系と関節座標系の座標変換 (4.1) 式, (4.2) 式の非線形性
- (ii) 関節座標系におけるロボットアームのダイナミクス (4.3) 式のゲイン劣化

#### 4.4.4 軌跡誤差と目標速度の関係

最大軌跡誤差  $\delta y_{max}$  と目標速度  $V$  との関係を導出する. ここで, 最大軌跡誤差は定常状態であるものと仮定する. この仮定は, 最大軌跡誤差が, ロボットアームのダイナミクス (4.3) 式の時定数の3倍以降に生じると言い替えることができる. 4.4.2節の計算機シミュレーションにおいて, 時定数が  $0.036[s]$  であったので,  $0.108[s]$  以降はロボットアームは定常状態であるとみなすことができる. この計算機シミュレーションにおいては, 最大軌跡誤差は  $t_1 = 0.714[s]$  で生じているため, 最大軌跡誤差は定常誤差とみなすことができる.

関節座標系における目標軌道は, 作業座標系における目標軌道から, 逆キネマティクス (4.2) 式を用いて導出されるため, 目標速度  $V$  に比例する角速度  $\omega$  の正弦波で近似することができる. ロボットアームのダイナミクス (4.3) 式によって, 追従関節角は目標関節角から

$$1 - |G(j\omega)| = 1 - \frac{1}{\sqrt{1 + \frac{(K_v - 2K_p)\omega^2}{K_p^2 K_v} + \frac{\omega^4}{(K_p K_v)^2}}} \approx \frac{1}{2} \left( \frac{(K_v - 2K_p)\omega^2}{K_p^2 K_v} + \frac{\omega^4}{(K_p K_v)^2} \right) \quad (4.6)$$

にしたがって劣化する. ここで,  $G(j\omega)$  はロボットアームのダイナミクス (4.3) 式の周波数伝達関数を表す. 作業座標系における追従軌道は, キネマティクス (4.1) 式によって追従関節角から導出される. (4.6) 式のテイラー展開1次近似によって, 最大軌跡誤差は

$$\delta y_{max} \approx AV^2 + BV^4 \quad (4.7)$$

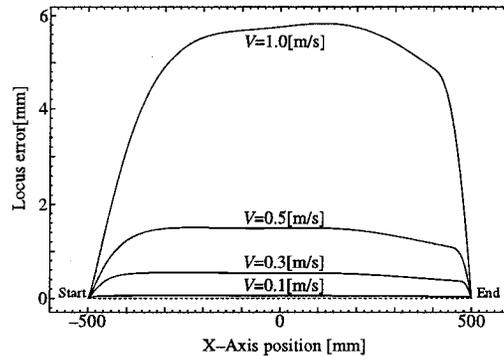
と近似することができる. ここで  $A, B$  は係数である. (4.7) 式により, 最大軌跡誤差は目標速度  $V$  に大きく依存し, その関係は  $V^2$  と  $V^4$  の線形結合で近似できることがわかる.

### 4.5 軌跡誤差の目標速度と目標軌跡依存性

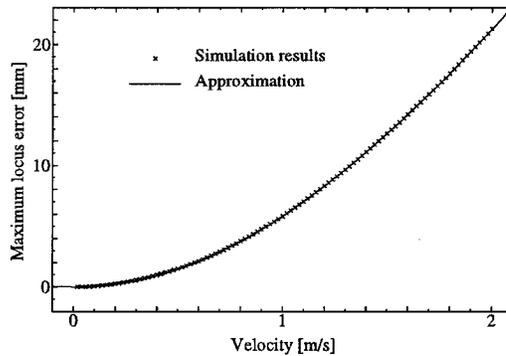
#### 4.5.1 目標速度依存性

目標速度と軌跡誤差の関係を, 計算機シミュレーションによって調べる. シミュレーション条件は, 目標速度を除いて4.4.2節と同じである. 目標速度が  $V = 0.1$ ,

(a) Locus



(b) Maximum locus error as a function of objective velocity

Fig. 4.7 Maximum locus error for different objective velocities ( $V = 0.2 \sim 2$ [m/s])

0.3, 0.5, 1.0[m/s] の下でのシミュレーション結果を Fig. 4.7(a) に示す. Fig. 4.7(b) は, 目標速度が  $V = 0.02 \sim 2$ [m/s] 間の 100 点に対する目標軌道を用いて, 最大軌跡誤差を計算した結果と, (4.7) 式の最大軌跡誤差の目標速度依存性の係数  $A, B$  を最小二乗法を用いて求めた曲線  $\delta y_{max} = 5.99V^2 - 0.17V^4$ [mm] を示す. この結果から, (4.7) 式で表される関係は最大軌跡誤差を非常に良く近似することがわかる. (4.7) 式の係数を実験結果から求めることによって, 最大軌跡誤差と目標速度の関係式を導出することができ, その関係式を用いることにより, 最大軌跡誤差が指定された精度を持つような目標速度を適切に決定することができる.

#### 4.5.2 目標軌跡依存性

次に, 軌跡誤差が目標軌跡の違いによってどのように変化するかを, 計算機シミュレーションによって調べる. 目標速度  $V = 1.0$  [m/s] の下でのシミュレーション結果を Fig. 4.8 に示す. 目標軌跡の Y 軸の位置は  $y_0 = 0.3, 0.5, 1.0, 1.5$ [m/s] で, X 軸の目標軌跡は 4.5.1 節のシミュレーションと同じである. シミュレーション結果から, 目標軌跡の Y 軸の位置が作業座標系の原点に近づくほど軌跡誤差が大きくなっていることがわかる. このシミュレーション結果は, 軌跡誤差が目標軌跡にも依存し, 原点付近において作業を行なうと, 軌跡誤差が大きくなり, 制御性能が劣化することを示している.

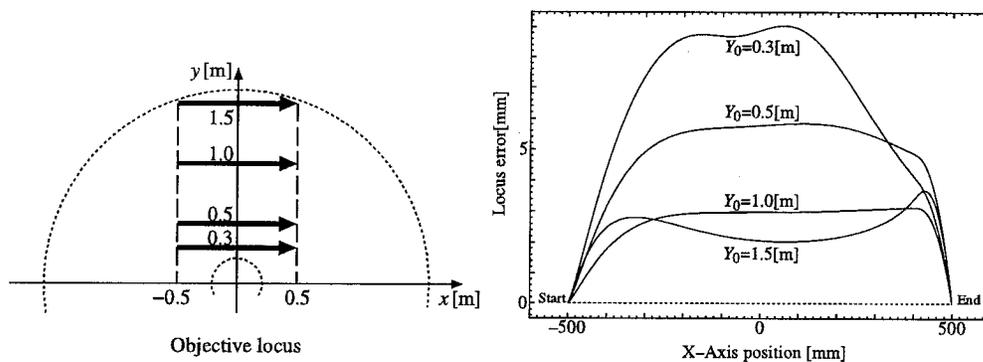


Fig. 4.8 Locus error as a function of x-axis position for several objective locus

## 4.6 まとめ

産業用多関節ロボットアームの高速動作時での輪郭制御における軌跡誤差は、多関節ロボットアームの非線形座標変換とダイナミクスのゲイン劣化によるものであることを究明した。最大軌跡誤差は目標速度の2乗と4乗の線形結合で近似され、また、軌跡誤差は目標軌跡の位置に依存することを示した。導出した最大軌跡誤差と目標速度との関係式により、要求する精度を満たすような目標速度を適切に決定することができる。

## 第5章

# 極配置レギュレータを用いた教示信号修正法

### 5.1 まえがき

産業用ロボットアームの輪郭制御を行なう場合において、ティーチング作業の際に用いる教示信号として、目標軌道の値をそのまま用いると、プレーバック時のロボットの応答動作に遅れが生じたり、構成各軸の応答動作にばらつきがでて、目標軌道と実際のロボット動作の軌道との間にずれが生じる。そこで、ロボットを様々な目的に対して正確に動作させるための研究が数多くなされてきている。例えば、ロボットの適応制御<sup>[32, 33]</sup>、VSSを用いたトルク算出制御<sup>[25]</sup>、線形制御則を用いたロバスト制御<sup>[34]</sup>、未知ダイナミクスを持つロバスト制御<sup>[35]</sup>、PD制御器を用いたロバスト制御<sup>[29]</sup>、直交座標系におけるロバスト制御<sup>[36]</sup>などがある。これらの研究の多くは、フィードバック型の制御法であり、産業応用を考えると必ずしも実現が容易でない。フィードフォワード型のロボットアーム制御の研究としては、フィードフォワードデジタル制御器<sup>[37]</sup>や、フレキシブルロボットに対するフィードフォワード制御器<sup>[38]</sup>が提案されている。このフィードフォワード制御器は、specific position commandと呼ばれる2次スプラインや多項式の目標軌道に対するの制御器であり、一般的な目標軌道に対するものではない。

本章では、教示信号として、目標軌道そのままの値を用いずに、目標軌道に修正を施した信号を教示信号として用いることによって、既存の装置のハードに変更を加えずにロボットアームの制御系の特性を改善する教示信号修正法を提案する。この教示信号修正法を用いることによって、ロボットアームをできるだけ速やか、かつ正確に目標軌道に応答させることが可能となる。この教示信号修正法によるロボットアームの制御の特性を、時間領域と周波数領域で解析し、教示信号修正法によって制御系の特性が改善されていることを理論的に確認した。

提案する教示信号修正法を、実際に稼働している産業用ロボットアームに適用して輪郭制御実験を行い、本手法の有用性を検討した。この結果から、教示信号として目標軌道の値をそのまま入力する従来法の場合と比較して、本手法を用いて教示信号を修正した場合の方がロボットアームの追従軌道が著しく目標軌道に近いことが確認された。本手法は、ソフト的に教示信号を修正するだけで、既存の装置のハードの構成はまったく変更せずに、制御対象の動作の応答特性を改善して、ハードの持つ最高の特性を発揮させることが可能となるため、工学的に有用な方法といえる。

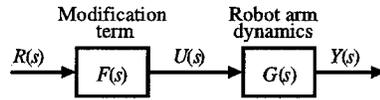


Fig. 5.1 Block diagram of the modified taught data method.

## 5.2 教示信号修正法

### 5.2.1 教示信号修正法のご概念

産業用ロボットアームの作業座標系における各座標軸ごとの入出力関係を、各座標軸独立に

$$Y(s) = G(s)U(s) \quad (5.1)$$

で表す。ここで、 $U(s)$ は教示信号、 $Y(s)$ はロボットアームの追従軌道、 $G(s)$ はロボットアームのダイナミクスを表す。ティーチングプレーバックロボットは、ロボットアームの先端の位置や速度を計測しないセミクローズドタイプ制御系で、かつ、ロボットアーム制御系のハードの変更を避けたいため、制御手法はフィードフォワードタイプである必要がある。そこで、目標軌道  $R(s)$  に修正要素  $F(s)$  を通して教示信号  $U(s)$  を作成する。すなわち、教示信号  $U(s)$  は

$$U(s) = F(s)R(s) \quad (5.2)$$

と表現される。Fig. 5.1に教示信号修正法のブロック線図を示す。理想的な制御性能  $Y(s) = R(s)$ 、すなわち、ロボットアームが目標軌道と一致するためには、修正要素  $F(s)$  はロボットアーム制御系の逆システム  $G^{-1}(s)$  であることが要求される。しかし、修正要素を  $F(s) = G^{-1}(s)$  と設計することは、逆システム  $G^{-1}(s)$  がプロパーでなくなるため、目標軌道が微分不可能である場合には教示信号が発散する。そこで、ロボットアーム制御系を状態空間表示し、極配置レギュレータを用いて、修正後のロボットアーム制御系  $F(s)G(s)$  が望ましい閉ループ制御系となるように修正要素  $F(s)$  を設計する。

### 5.2.2 1次系モデルに基づく教示信号修正法

#### (a) 数式モデル

まず、修正要素  $F(s)$  の導出が容易な、ロボットアーム制御系の1次系モデルに基づく教示信号修正法を導出する。ロボットアームのアクチュエータであるサーボモータの速度が、定格の1/100程度の速度でロボットアームを動作させる場合には、サーボモータの動特性や速度制御の特性をほぼ定数とみなし、位置制御の特性のみを考えれば良い。したがって、制御装置とサーボモータ、メカニズムを含めたロボットアーム全体の制御系は Fig. 5.2に示されるように、作業座標系において各座標軸独立に1次系

$$G_1(s) = \frac{K_p}{s + K_p} \quad (5.3)$$

で表現できる(3章参照)。ここで、 $K_p$ は位置制御器の位置ループゲインである<sup>[13]</sup>。

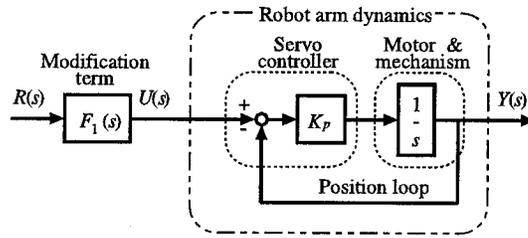


Fig. 5.2 Block diagram of the modified taught data method based on the first order model.

(b) 修正要素

ロボットアーム制御系 (5.3) 式を状態空間表示して、極配置レギュレータを用いて修正要素  $F_1(s)$  を導出する。ここで、目標軌道  $r(t)$  に対して、 $\dot{r}(t) \simeq 0$  という仮定を用いる。(5.3) 式から、仮定  $\dot{r}(t) \simeq 0$  の下で、ロボットアーム制御系を状態空間表示すると

$$\dot{x}(t) = -K_p x(t) + K_p r^*(t) \tag{5.4}$$

となる。ここで、 $x(t) = y(t) - r(t)$ 、 $r^*(t) = u(t) - r(t)$  である。この  $\dot{r}(t) \simeq 0$  という仮定により (5.4) 式の形が導かれ、この状態方程式には極配置レギュレータが適用でき、教示信号の修正項  $r^*(t)$  の導出が容易になる。いま設けた仮定  $\dot{r}(t) \simeq 0$  が以下の良好な結果を得るに至った一つの鍵となっているが、その意義に関しては 5.3.1 節で述べる。

操作量  $r^*(t)$  は、極配置レギュレータにより

$$r^*(t) = K_s x(t) \tag{5.5}$$

と与えられる。ここで  $K_s$  はレギュレータのフィードバックゲインである。フィードバックゲイン  $K_s$  とレギュレータの極  $\gamma$  との関係は

$$\gamma = -K_p(1 - K_s) \tag{5.6}$$

と表される。(5.5) 式と (5.6) 式を (5.4) 式に代入することにより、教示信号  $u(t)$  は

$$\dot{u}(t) - \gamma u(t) = -\frac{\gamma}{K_p}(\dot{r}(t) + K_p r(t)) \tag{5.7}$$

と与えられる。(5.7) 式のラプラス変換により、修正要素  $F_1(s)$  は

$$F_1(s) = -\frac{\gamma(s + K_p)}{K_p(s - \gamma)} \tag{5.8}$$

のように与えられる。(5.7) 式の微分方程式を  $u(t)$  について解くことにより、ロボットアーム制御系の 1 次系モデルに基づく修正教示信号  $u(t)$  が計算できる。

修正要素  $F_1(s)$  をロボットアーム制御系  $G_1(s)$  に適用すると、修正後のロボットアーム制御系は

$$Y(s) = \frac{-\gamma}{s - \gamma} R(s) \quad (5.9)$$

となる。元のロボットアーム制御系 (5.3) 式と修正後のロボットアーム制御系 (5.9) 式を比較すると、修正要素はロボットアーム制御系の極を  $-K_p$  から  $\gamma$  へ変換していることがわかる。

### (c) 極の選定

修正要素 (5.8) 式中の設計者が与えるレギュレータの極  $\gamma$  の選定について述べる。まず、ロボットアームの制御性能を向上させるためには、修正後のロボットアームの制御系が修正前よりも応答を早くする必要があるため

$$\gamma \leq -K_p \quad (5.10)$$

を満足する必要がある。また、実際のロボットアームに教示信号修正法を適用するに当たって、ロボットアームのアクチュエータであるサーボモータの速度制限を考慮する必要がある。サーボモータの最大速度を  $V_{max}$  とすると、速度制限は

$$|K_p(u(t) - y(t))| \leq V_{max} \quad (5.11)$$

と表すことができる。(5.11) 式の左辺はサーボモータへの速度入力を表す。実際には、(5.11) 式の左辺をいくらかの違った極で教示信号修正法の計算機シミュレーションを行ない、(5.10) 式と (5.11) 式の条件を満足する最も小さい極を選定すればよい。

## 5.2.3 2次系モデルに基づく教示信号修正法

### (a) 数式モデル

ロボットアームの動作速度が高速になり、サーボモータの速度が定格の  $1/20 \sim 1/100$  程度となる場合には、サーボモータの速度制御の特性までを考慮して、ロボットアーム全体の制御系を、Fig. 5.3 に示されるように、各座標軸独立に2次系

$$G_2(s) = \frac{K_p K_v}{s^2 + K_v s + K_p K_v} \quad (5.12)$$

で表現する必要がある<sup>[22]</sup>。ここで、 $K_v$  は速度ループゲインである。

### (b) 修正要素

2次系モデル (5.12) 式に基づくロボットアーム制御系を状態空間表示し、極配置レギュレータと最小次元オブザーバを用いて修正要素を導出する。2次系モデルは速度ループを含むため、修正要素  $F_2(s)$  の導出は1次系モデルの場合と比較して複雑となる。

2次系モデル (5.12) 式から、 $\ddot{r}(t) + K_v \dot{r}(t) \simeq 0$  の仮定の下で、ロボットアーム制御系を

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + b r^*(t), \quad y^*(t) = c\mathbf{x}(t) \quad (5.13)$$

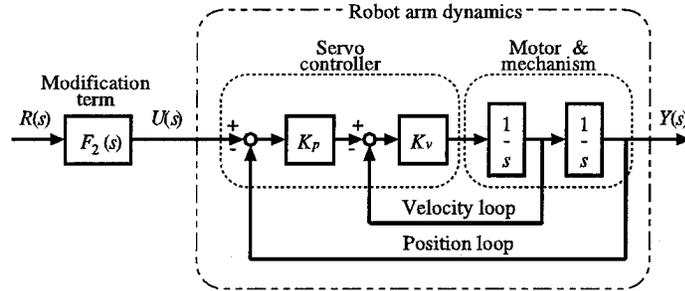


Fig. 5.3 Block diagram of the modified taught data method based on the second order model.

と状態空間表示する．ここで

$$A = \begin{pmatrix} -K_v & 1 \\ -K_p K_v & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ K_p K_v \end{pmatrix}, \quad c = (1 \quad 0) \quad (5.14a)$$

$$\mathbf{x}(t) = \begin{pmatrix} y^*(t) \\ \dot{y}^*(t) + K_v y^*(t) - K_p K_v r^*(t) \end{pmatrix} \quad (5.14b)$$

$$y^*(t) = y(t) - r(t), \quad r^*(t) = u(t) - r(t) \quad (5.14c)$$

である．仮定 $\ddot{r}(t) + K_v \dot{r}(t) \simeq 0$ は，レギュレータ理論を用いるために導入された．この仮定 $\ddot{r}(t) + K_v \dot{r}(t) \simeq 0$ の導入の意義に関しては5.3.2節で述べる．

ロボットアーム制御系の状態空間表示(5.13)式に対して，極配置レギュレータと最小次元オブザーバを適用する．極配置レギュレータは

$$r^*(t) = (f_1 \quad f_2) \hat{\mathbf{x}}(t) \quad (5.15)$$

と表される．ここで，

$$f_1 = 1 - \frac{K_v}{K_p} - \frac{\gamma_1 + \gamma_2}{K_p} - \frac{\gamma_1 \gamma_2}{K_p K_v} \quad (5.16a)$$

$$f_2 = \frac{1}{K_p} + \frac{\gamma_1 + \gamma_2}{K_p K_v} \quad (5.16b)$$

である．また，最小次元オブザーバは

$$\dot{z}(t) = \mu z(t) - (K_p K_v + \mu K_v + \mu^2) y^*(t) + K_p K_v r^*(t) \quad (5.17a)$$

$$\hat{\mathbf{x}}(t) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} z(t) + \begin{pmatrix} 1 \\ -\mu \end{pmatrix} y^*(t) \quad (5.17b)$$

となる． $\hat{\mathbf{x}}(t)$ を(5.15)式に代入することにより，操作量 $r^*(t)$ は

$$r^*(t) = (f_1 - \mu f_2) y^*(t) + f_2 z(t) \quad (5.18)$$

と導出される。修正要素  $F_2(s)$  を得るために、(5.14c) 式、(5.17) 式と (5.18) 式を周波数領域に変換すると

$$Y^*(s) = Y(s) - R(s), \quad R^*(s) = U(s) - R(s) \quad (5.19a)$$

$$Z(s) = \frac{-K_p K_v - \mu K_v - \mu^2}{s - \mu} Y^*(s) + \frac{K_p K_v}{s - \mu} R^*(s) \quad (5.19b)$$

$$R^*(s) = (f_1 - \mu f_2) Y^*(s) + f_2 Z(s) \quad (5.19c)$$

となる。(5.19b) 式を (5.19c) 式へ代入すると、 $R^*(s)$  と  $Y^*(s)$  の関係

$$R^*(s) = \frac{(s - \mu)f_1 - (\mu s + K_p K_v + \mu K_v)f_2}{s - \mu - f_2 K_p K_v} Y^*(s) \quad (5.20)$$

が得られる。(5.19a) 式と (5.20) 式から、 $U(s)$  は  $R(s)$  と  $Y(s)$  により

$$U(s) = (1 - P(s))R(s) + P(s)Y(s) \quad (5.21)$$

と与えられる。ここで

$$P(s) = \frac{(s - \mu)f_1 - (\mu s + K_p K_v + \mu K_v)f_2}{s - \mu - f_2 K_p K_v} \quad (5.22)$$

である。目標軌道  $R(s)$  とロボットアームの追従軌道  $Y(s)$  の関係は、(5.12) 式と (5.21) 式から

$$Y(s) = \frac{G_2(s)(1 - P(s))}{1 - G_2(s)P(s)} R(s) \quad (5.23)$$

となる。最終的に、修正要素  $F_2(s)$  は、(5.23) 式から

$$F_2(s) = \frac{1 - P(s)}{1 - G_2(s)P(s)} \quad (5.24)$$

と導出される。 $f_1$  と  $f_2$  を代入すると、修正要素  $F_2(s)$  はレギュレータの極  $\gamma_1, \gamma_2 (< 0)$ 、オブザーバの極  $\mu (< 0)$  とフィードバックゲイン  $K_p, K_v$  を用いて

$$F_2(s) = \frac{\alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0}{(s - \gamma_1)(s - \gamma_2)(s - \mu)} \quad (5.25)$$

と表現される。ここで

$$\alpha_0 = -\mu \gamma_1 \gamma_2 \quad (5.26a)$$

$$\alpha_1 = (K_v + \mu)(\gamma_1 + \gamma_2) + K_v^2 + \gamma_1 \gamma_2 + K_v \mu - \frac{\mu \gamma_1 \gamma_2}{K_p} \quad (5.26b)$$

$$\alpha_2 = \frac{1}{K_p} \{ (K_v + \mu)(\gamma_1 + \gamma_2) + K_v^2 + \gamma_1 \gamma_2 + K_v \mu \} - \frac{\mu \gamma_1 \gamma_2}{K_p K_v} \quad (5.26c)$$

$$\alpha_3 = \frac{1}{K_p K_v} \{ (K_v + \mu)(\gamma_1 + \gamma_2) + K_v^2 + \gamma_1 \gamma_2 + K_v \mu \} \quad (5.26d)$$

である。修正要素  $F_2(s)$  は時間領域に

$$(D - \gamma_1)(D - \gamma_2)(D - \mu)u(t) = (\alpha_3 D^3 + \alpha_2 D^2 + \alpha_1 D + \alpha_0)r(t) \quad (5.27)$$

と変換される。ここで、 $D$ は微分演算子である。微分方程式(5.27)式を  $u(t)$  について解くことによって、2次系モデルに基づく修正教示信号  $u(t)$  を計算することができる。

修正要素  $F_2(s)$  とロボットアーム制御系(5.12)式から、修正後のロボットアーム制御系は

$$Y(s) = \frac{\beta_1 s + \beta_0}{(s - \gamma_1)(s - \gamma_2)(s - \mu)} R(s) \quad (5.28)$$

と表される。ここで

$$\beta_0 = -\mu\gamma_1\gamma_2 \quad (5.29a)$$

$$\beta_1 = (K_v + \gamma_1 + \gamma_2)(K_v + \mu) + \gamma_1\gamma_2 \quad (5.29b)$$

である。

### (c) 極の選定

修正要素(5.25)式的设计において、レギュレータの極  $\gamma_1, \gamma_2$  とオブザーバの極  $\mu$  を適切に選定する必要がある。オブザーバの極はレギュレータの極より小さく選ぶ必要があるから

$$\mu < \min\{\gamma_1, \gamma_2\} \quad (5.30)$$

とし、レギュレータの極に関しては、一般性を失うことなく  $\gamma_1 \leq \gamma_2$  と仮定する。教示信号修正法を実際のロボットアームに適用するに当たって、ロボットアームの追従軌道にオーバーシュートは避けなければならない。一つの零点をもつ3次系(5.28)式において、オーバーシュートを生じさせない条件は、最大極が零点以下であれば良い<sup>[39]</sup>。そこで、レギュレータの極を

$$\gamma_2 \geq \frac{\mu\gamma_1\gamma_2}{(K_v + \gamma_1 + \gamma_2)(K_v + \mu) + \gamma_1\gamma_2} \quad (5.31)$$

を満足するように選定する。(5.31)式を簡単にすると

$$\gamma_2 \geq -K_v \quad (5.32)$$

となる。条件(5.32)式の中で応答を最も早くするように極を  $\gamma_2 = -K_v$  とすると

$$Y(s) = \frac{\mu\gamma_1}{(s - \gamma_1)(s - \mu)} R(s) \quad (5.33)$$

となる。元のロボットアーム制御系(5.12)式と修正後のロボットアーム制御系(5.33)式から、修正要素はロボットアーム制御系の極を  $\{-K_v \pm \sqrt{K_v^2 - 4K_v K_p}\}/2$  から  $\gamma_1$  と  $\mu$  に変換していることがわかる。1次系の場合と同様に、ロボットアームの制御性

能を向上させるためには、修正後のロボットアームの制御系が修正前よりも早くなるということから、 $\gamma_1$ は

$$\gamma_1 \leq \frac{-K_v - \sqrt{K_v^2 - 4K_v K_p}}{2} \quad (5.34)$$

を満足すればよい。また、極 $\gamma_1$ と $\mu$ の選定において、サーボモータの速度制限の条件(5.11)式とサーボモータのトルク制限を考慮する必要がある。サーボモータのトルク制限は

$$C|K_v\{K_p(u(t) - y(t)) - \dot{y}(t)\}| \leq \tau_{max} \quad (5.35)$$

と表される。ここで $\tau_{max}$ はサーボモータの最大トルクを表し、 $C$ は加速度からトルクへ変換する係数を表す。これらのパラメータは装置固有の値である。計算機シミュレーションを通じて、極 $\gamma_1$ と $\mu$ が条件(5.11)式、(5.34)式と(5.35)式を満足する最も小さい極を選定する。

### 5.3 教示信号修正法の特性解析

本章で提案している教示信号修正法は、極配置レギュレータ理論に基づいている。レギュレータ理論は、通常、目標点にシステムの出力を到達させるために用いられる。しかし、ロボットアームの制御は追従制御であり、目標軌道は一定ではなく時間変化する。また、修正教示要素の導出において、極配置レギュレータ理論を用いるために、1次系モデルに基づく場合には仮定 $\dot{r}(t) \simeq 0$ を導入し、2次系モデルに基づく場合には仮定 $\ddot{r}(t) + K_v \dot{r}(t) \simeq 0$ を導入したが、ロボットアームの使用条件を考えると、実際には、目標軌道に対して、これらの仮定が常に成り立つものではない。そこで、これらの仮定を導入した意味づけとその妥当性に関して検討し、教示信号修正法を用いた場合の応答特性が、教示信号に目標軌道そのままの値を用いた従来法と比較してどのように改善するかを、時間領域と周波数領域から調べる。

#### 5.3.1 1次系モデル

1次系モデルに基づく教示信号修正法の特性解析を行なう。まず、時間領域において解析する。教示信号修正法の目標軌道 $r(t)$ と制御系の出力 $y(t)$ との時間領域における関係式は、(5.9)式で表される伝達関数から、逆ラプラス変換を用いて

$$\dot{y}(t) = \gamma y(t) - \gamma r(t) \quad (5.36)$$

となる。一方、目標軌道の値をそのまま教示信号として用いる従来法 $u(t) = r(t)$ の場合、目標軌道 $r(t)$ と出力 $y(t)$ の特性を表す式は、(5.3)式で表される伝達関数から逆ラプラス変換を用いて

$$\dot{y}(t) = -K_p y(t) + K_p r(t) \quad (5.37)$$

で表される。教示信号修正法の特性を表す(5.36)式を、従来法の特性を表す(5.37)式と比較すると、 $-y(t)$ と $r(t)$ の係数が $K_p$ から $-\gamma$ に変わっていることがわかる。つ

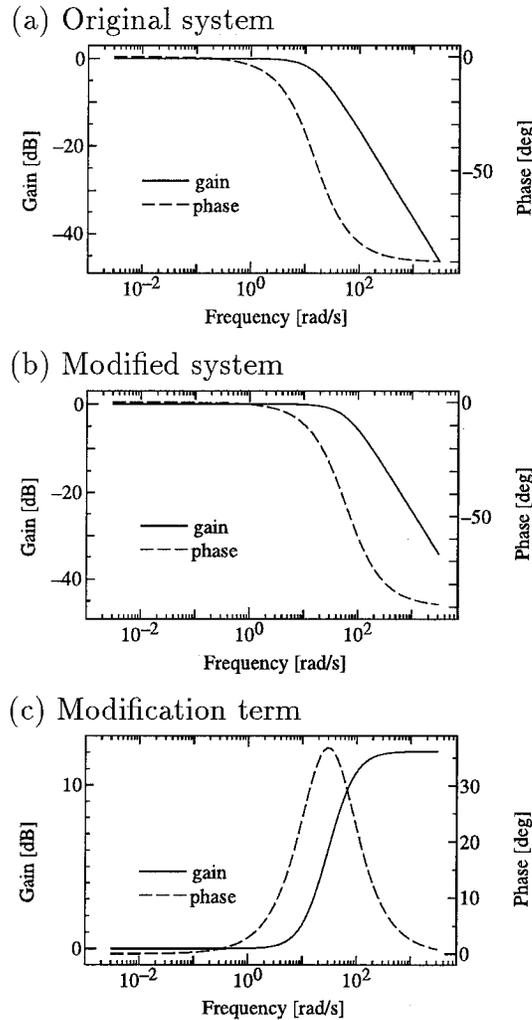


Fig. 5.4 Bode diagram of the modified taught data method based on the first order model ( $K_p = 15 [1/s]$ ,  $\gamma = -60 [1/s]$ )

まり、教示信号修正法では、適切な教示信号の修正によって、系の特性が  $K_p$  から  $-\gamma$  に変換されている。(5.36) 式の時定数は  $-1/\gamma$  であり、 $\gamma < -K_p$  の範囲でレギュレータの極  $\gamma$  を適切に設計することにより、従来法の (5.37) 式の時定数  $1/K_p$  より小さくすることができる。つまり、教示信号修正法では、小さな時定数で出力  $y(t)$  を目標軌道  $r(t)$  に速やかに追従させ得ることがわかる。輪郭制御の精度を同一にすれば、本手法は従来法に比べて目標軌道の速度を  $-\gamma/K_p$  倍上げることができる。

次に、周波数領域において解析する。Fig. 5.4 に  $K_p = 15 [1/s]$ ,  $\gamma = -60 [1/s]$  の場合の (a) 修正前のシステム, (b) 修正後のシステム, (c) 修正要素のブロック線図を示す。Fig. 5.4(a) 修正前のシステムと (b) の修正後のシステムのボード線図を比較する。Fig. 5.4(b) に示す修正後のシステムのボード線図から、ゲイン特性が 0 [dB] で一定とみなせる限界の周波数が、 $\omega = 30 [rad/s]$  であることがわかる。この周波数は

Fig. 5.4(a) のロボットアームの制御系のゲイン特性の $\omega = 7$  [rad/s]と比較して高くなっている。位相特性についても、ほぼ遅れが生じない限界の周波数が $\omega = 1$  [rad/s]となり、Fig. 5.4(a)の $\omega = 0.02$  [rad/s]と比較すると高くなっていることがわかる。これらの特性の改善は、教示信号を修正することによって、遮断周波数が $K_p$ から $-\gamma$ へと高くなっているためである。

修正要素のゲイン特性は、(5.8)式から

$$|F_1(j\omega)| = -\frac{\gamma}{K_p} \sqrt{\frac{\omega^2 + K_p^2}{\omega^2 + \gamma^2}} \quad (5.38)$$

となる。Fig. 5.4(c)のゲイン特性から、教示修正要素のゲインは、 $\omega = 7$  [rad/s]付近から周波数の増加と共に増加をはじめ、 $\omega = 500$  [rad/s]で約12 [dB]に達している。この修正教示要素のゲインが増加し始める周波数 $\omega = 7$  [rad/s]は、Fig. 5.4(a)のロボットアームの制御系のゲインが落ち始める周波数と一致している。このことは教示修正要素が、元のロボットアームの制御系のゲインを補償していることを表している。

また、教示修正要素の位相特性は(5.8)式より

$$\arg F_1(j\omega) = -\arctan \frac{(\gamma + K_p)\omega}{\omega^2 + \gamma^2} \quad (5.39)$$

となる。Fig. 5.4(c)の位相特性から、 $\omega = 0.02$  [rad/s]より高い周波数帯域から教示修正要素が位相を進めていることがわかる。この周波数はFig. 5.4(a)のロボットアームの制御系の位相が遅れ始める周波数に一致している。修正要素の最大位相は

$$\sin \phi_m = \frac{\gamma + K_p}{\gamma - K_p} \quad (5.40)$$

と計算され、そのときの周波数は

$$\omega_m = \sqrt{-K_p \gamma} \quad (5.41)$$

となる<sup>[40]</sup>。以上により、修正要素 $F_1(s)$ は位相進み補償を行っていることがわかる。このことから、修正要素によってロボットアームは、教示信号に目標軌道の値をそのまま用いた従来法と比較して、より高周波成分を含む目標軌道に対しても、ゲインの劣化や位相の遅れが生じず、目標軌道に早く追従することがわかる。

この修正要素 $F_1(s)$ を、従来から用いられている逆システムによるフィードフォワード制御と比較すると、逆システムによるフィードフォワード制御においては、目標軌道が微分不可能な場合に修正教示信号が発散することが生じる。それに対して、教示信号修正法では、(5.8)式から修正要素はプロパーであり、修正教示信号が発散するといったことは生じない。また、 $\gamma \rightarrow -\infty$ の極限においては、教示信号修正法は逆システムによるフィードフォワード制御と一致する。

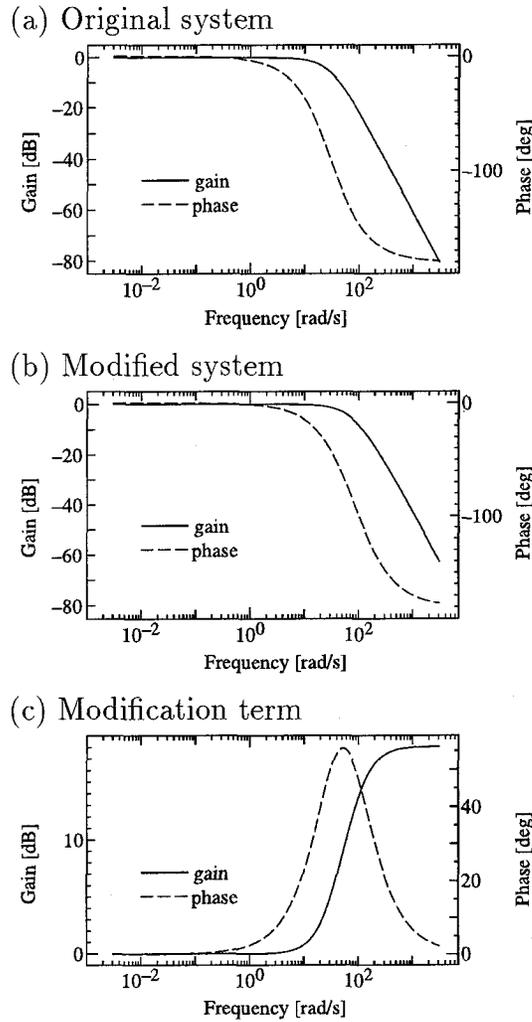


Fig. 5.5 Bode diagram of the modified taught data method based on the second order model ( $K_p = 15 [1/s]$ ,  $K_v = 60 [1/s]$ ,  $\gamma_1 = \gamma_2 = -60 [1/s]$ ,  $\mu = -120 [1/s]$ ).

### 5.3.2 2次系モデル

2次系モデルに基づく教示信号修正法の特性格解析を行なう。2次系モデルにおける特性も、1次系モデルに基づく特性とほぼ同じであり、時間領域においては、元のロボットアーム制御系 (5.12) 式と修正後のロボットアーム制御系 (5.33) 式とを比較することにより、修正要素はロボットアーム制御系の極を  $\{-K_v \pm \sqrt{K_v^2 - 4K_v K_p}\}/2$  から  $\gamma_1$  と  $\mu$  に変換している。

周波数領域においても、 $K_p = 15 [1/s]$ ,  $K_v = 60 [1/s]$ ,  $\gamma_1 = \gamma_2 = -60 [1/s]$ ,  $\mu = -120 [1/s]$  の場合の、2次系モデルに基づく教師信号修正法のボード線図を、Fig. 5.5に示すと、これは、1次系モデルに基づく場合の特性 (Fig. 5.4) とほぼ同じであり、2次系モデルに基づく教示信号修正法も位相進み補償器とみなすことがで

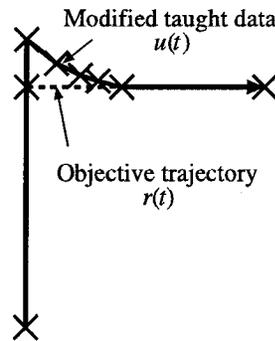


Fig. 5.6 Taught data by using the modified taught data method.

きる。

## 5.4 産業用ロボットアームによる実験結果

教示信号修正法の有効性を確認するために、6自由度の産業用ロボットアーム(モートマン K10S)を用いて実験を行なった。モートマンの位置ループゲインと速度ループゲインは、それぞれ  $K_p = 15$  [1/s] と  $K_v = 60$  [1/s] である。サーボモータの速度制限は  $V_{max} = 1$  [m/s], でトルク制限は  $\tau_{max} = 1.0$  [Nm], そして、加速度をトルクに変換する係数は  $C = 5.3 \times 10^{-3}$  [kgm] である。ロボットアームの先端にペンを取り付け、ロボットアームに二次元軌道を描かせる実験を行なった。

### 5.4.1 ロボットアームへの教示

Fig. 5.6に示すように、産業用ロボットアームは教示位置の系列とその間の速度を指定することによって動作する<sup>[41]</sup>。修正教示信号の作成の仕方は、まず、1次系モデルに基づく場合には微分方程式(5.7)式、2次系モデルに基づく場合には微分方程式(5.27)式を解くことによって、教示信号  $u(t)$  を計算する。微分方程式を解く際には、オイラー法を用いた。教示位置は、サンプリング間隔 20 [ms] で教示信号  $u(t)$  をサンプリングすることにより導出した。また、教示速度は連続する教示位置の差分を取ることによって計算した。ティーチングボックスを用いてロボットアームを教示位置(Fig. 5.6の×)に移動させ、位置を記憶させる。教示速度はオペレーションパネルによるキー入力で教示する。すべての教示位置と教示速度を教示し終わった後に、オペレーションパネルのプレイキーを押すことによってロボットアームは動作する。

### 5.4.2 実験結果

目標軌道は、Fig. 5.7の左上部に示されるような、三本の直線と二つの角からなる軌道を与えた。目標軌道の速度は 250 [mm/s] である。三つの手法による実験結果を Fig. 5.7 (a) 従来法, (b) 1次系モデルに基づく教示信号修正法, (c) 2次系モデルに基づく教示信号修正法にそれぞれ示す。レギュレータとオブザーバの極は、計算機シミュレーションにより、1次系に基づく場合には  $\gamma = -60$  [1/s], 2次系モデルに基

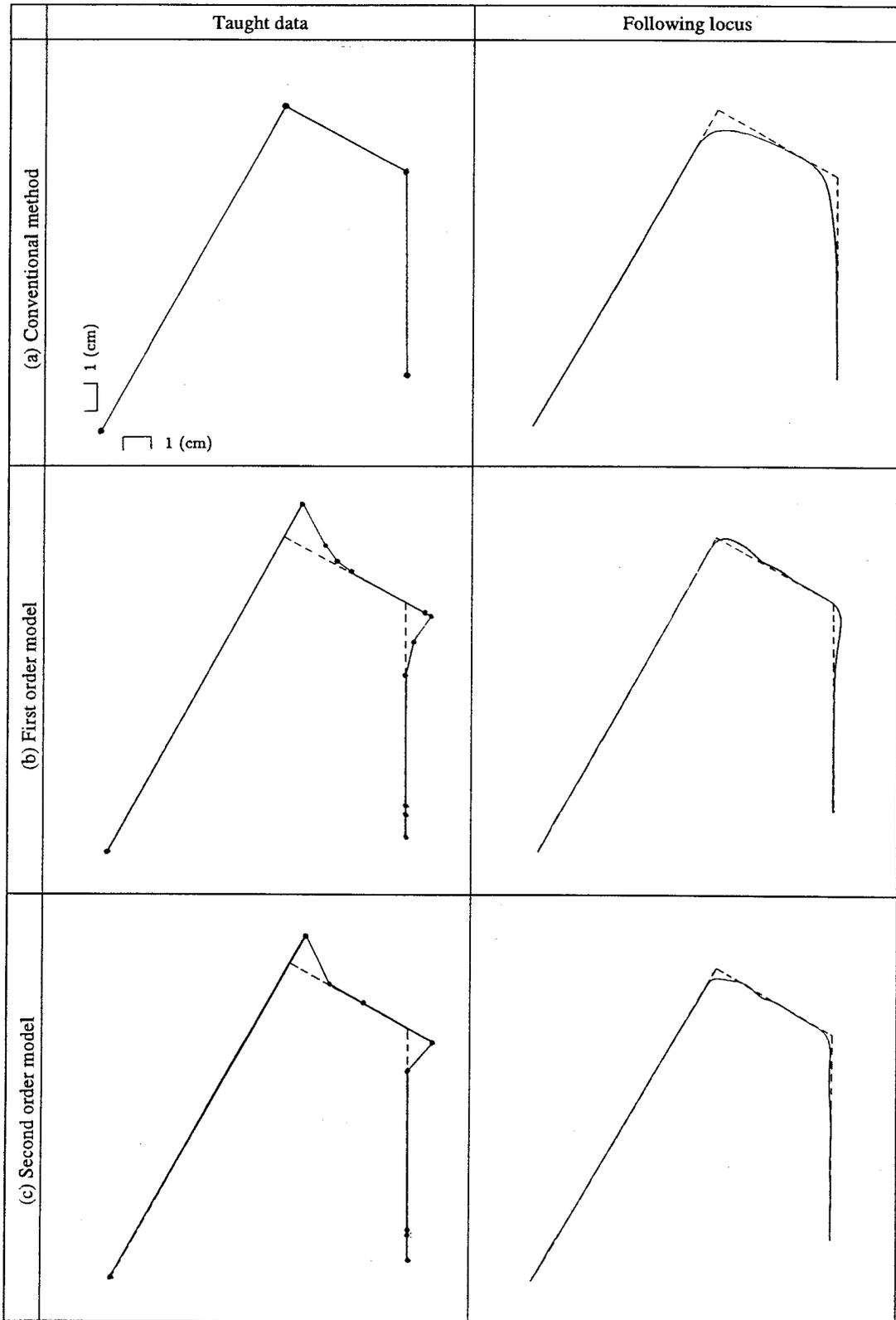


Fig. 5.7 Experimental results for contour control of the industrial robot arm: (a) Conventional method, (b) First order model, (c) Second order model.

づく場合には $\gamma_1 = -60$  [1/s],  $\gamma_2 = -60$  [1/s],  $\mu = -120$  [1/s]とした。

従来法を用いた場合の追従軌跡 (a) では、ロボットアームの動作の遅れにより角が取れてなまってしまっている。1次系モデルに基づく教示信号修正法を用いた場合の追従軌跡 (b) と2次系モデルに基づく場合の追従軌跡 (c) では、ロボットアームの遅れが適切に補償されているために、角も正確に追従していることがわかる。しかし、1次系モデルに基づく結果を見るとわずかなオーバーシュートが見られる。産業用ロボットアームの輪郭制御において、このようなオーバーシュートは極力さける必要がある。そこで、2次系モデルに基づく結果を見ると、オーバーシュートがなく、目標軌跡と追従軌跡が一致していることがわかる。1次系モデルに基づく結果においてオーバーシュートが生じた原因は、目標速度 250 [mm/s] では、実際のロボットアームを1次系モデルで表現したときのモデル化誤差が無視できないためである。

目標軌跡と追従軌跡との誤差の面積を比較すると、従来法においては 136[mm<sup>2</sup>]、1次系モデルに基づく教示信号修正法では 60[mm<sup>2</sup>]、2次系モデルに基づく教示信号修正法では 40[mm<sup>2</sup>]であった。この結果からも、教示信号修正法の有効性が確認できる。

## 5.5 まとめ

ロボットアームの高速高精度制御のための、極配置レギュレータに基づく教示信号修正法を提案した。本手法は、ロボットアームのダイナミクスを状態空間表示し、極配置レギュレータ理論を用いてロボットアームの特性を改善させる手法であり、また、本手法を使用する際のパラメータの選定手順も示した。提案した教示信号修正法の特性を時間領域と周波数領域で解析することにより、この教示信号修正法によってロボットアーム制御系の特性が改善されることを理論的に確認した。本手法を実際の産業用ロボットアームに適用し、その実験結果から教示信号修正法の有効性が確認された。教示信号修正法は、教示信号を修正するのみで、装置のハードの変更を必要としないために、産業界において有効に利用可能である。

## 第 6 章

# 速度情報に着目した教示信号修正法

### 6.1 まえがき

産業用ロボットアームの輪郭制御には、高精度かつ高速動作が要求される。通常、産業用ロボットアームでは、目標軌道、すなわち、目標位置と速度はロボットアームの入力にそのまま用いられる。この従来法では、ロボットアームの動作速度が低速である場合には正確な位置制御を行なうことができる。しかし、ロボットアームの動作速度は、作業目的に応じて与えられるため、常にロボットアームの高精度制御を行なうことは困難である。ロボットアームの高精度位置制御を行なう手法の一つとして、学習制御法が提案されている<sup>[30][42][43]</sup>。また、5章で述べた教示信号修正法も高精度制御手法の一つに挙げられる。教示信号修正法は、教示信号の位置と速度の両方を修正して高精度制御を実現する手法であり、ロボットアームのモデルが正確である場合には高精度制御性能を得ることができる。しかし、モデル化誤差が存在する場合には、制御性能は劣化し、オーバシュートが生じることがある。産業用ロボットアームの輪郭制御においては、作業精度に直接悪影響を与えるオーバシュートを極力避ける必要がある。

本章では、産業用ロボットアームの制御性能を向上させるために、速度情報に着目した教示信号修正法(以下、教示速度修正法と呼ぶ)を提案する。本手法は、教示位置は目標位置をそのまま入力し、教示速度のみを目標速度から修正して教示することによって制御性能を向上させる手法であり、モデル化誤差が存在する場合においても、オーバシュートが生じることはない。

### 6.2 教示速度修正法

#### 6.2.1 教示速度修正法の内容

産業用ロボットアームを用いて作業を行なう際には、多くの場合、目標軌跡は折れ線近似されることが多い。5章で示した教示信号修正法は、一般的な目標軌道に対して制御性能を向上させる手法であるが、教示速度修正法は、産業応用での輪郭制御では目標軌跡が折れ線近似される点を積極的に利用している。教示速度修正法の基本的な考え方は、追従軌跡の精度を改善するために、目標速度を修正して教示することである。従来法における輪郭制御性能は、主に動作速度に依存し、動作速度が速くなるほど、ロボットアームのダイナミクスの遅れにより、輪郭制御性能が

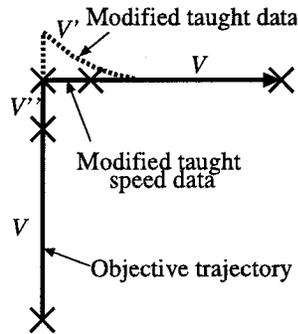


Fig. 6.1 Concept of the modified taught speed data method.

劣化する。そこで、折れ線近似された目標軌跡の角における速度を遅くすることによって、ロボットアームのダイナミクスの遅れによる悪影響を低減させ、追従軌跡を目標軌跡と一致させる。Fig. 6.1に教示速度修正法と極配置レギュレータ理論に基づく教示信号修正法の場合の教示の仕方の違いを示す。目標軌道は速度  $V$  で直角に曲がる軌道である。教示信号修正法の場合には、教示位置と教示速度ともに修正するために、教示位置は点線で示すようになり、教示速度は  $V'$  に修正され、教示の仕方も複雑になるが、教示速度修正法の場合には教示位置はそのまま教示速度のみを修正するために、Fig. 6.1の角の部分において教示速度を  $V'$  に下げるのみであるため、比較的容易に教示することができる。

### 6.2.2 教示速度修正法のアルゴリズム

目標軌跡を折れ線近似し、その中の長さ  $L$  の線分について考える。産業用ロボットアームは、各々の折れ線近似目標軌道に対して独立に1入力1出力系で表現することができる(3章参照)。そこで、ロボットアーム制御系を2次系モデル

$$Y(s) = \frac{K_p K_v}{s^2 + K_v s + K_p K_v} U(s) \quad (6.1)$$

で表現する。ここで、 $U(s)$  はロボットアームの入力軌道で  $Y(s)$  は出力軌道である。また、 $K_p$ 、 $K_v$  はそれぞれ位置ループゲインと速度ループゲインを表す。

まず、折れ線近似目標軌跡の中に含まれる線分  $L$  の終点について考える。終点まわりで減速を開始する時刻を0として、減速する時間間隔を  $T_a$  とする。すなわち、 $T_a$  は目標速度  $V$  から、最小速度  $V_{min}$  に減速する時間間隔を表す。ロボットアームの入力速度は

$$\dot{u}(t) = V - at \quad (6.2)$$

と与えられる。ここで、加速度は  $a = (V - V_{min})/T_a$  である。ロボットアームの入力軌道  $u(t)$  は(6.2)式の入力速度  $\dot{u}(t)$  の積分により

$$u(t) = Vt - \frac{a}{2}t^2 \quad (6.3)$$

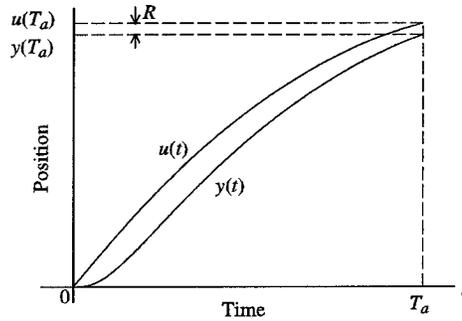


Fig. 6.2 Determination of the parameter  $T_a$ .

と計算される. ここで, 減速の時間間隔  $T_a$  を決定するために, 簡単のため  $u(0) = 0$  としている. (6.3) 式をラプラス変換して (6.1) 式に代入し, 逆ラプラス変換すると

$$y(t) \approx -\frac{a}{2}t^2 + \left(V + \frac{a}{K_p}\right)t + \frac{a}{K_p K_v} - \frac{a}{K_p^2} - \frac{V}{K_p} \quad (6.4)$$

となる. ここで,  $T_a > 3\tau$  の仮定の下で過渡項を無視する.  $\tau = 2/(K_v - \sqrt{K_v^2 - 4K_p K_v})$  は, ロボットアーム制御系 (6.1) 式の最大時定数を意味する. 時刻  $T_a$  の決定は, Fig. 6.2 に示すように, 時刻  $T_a$  においてロボットアームの出力位置が指定された点に到達するようにする. すなわち

$$y(T_a) = u(T_a) - R \quad (6.5)$$

となるように  $T_a$  を決定する. ここで,  $R$  は許容パラメータで, 時刻  $T_a$  における目標位置と出力位置の許容される誤差を意味する. (6.3) 式と (6.4) 式を (6.5) 式に代入し,  $T_a$  について解くことにより

$$T_a = \frac{(K_v - K_p)(V - V_{min})}{K_p K_v (K_p R - V_{min})} \quad (6.6)$$

となる. このように時間間隔  $T_a$  で終点付近における減速を行なうことにより, 産業用ロボットアームの輪郭制御性能を向上できる.

次に, 線分  $L$  の始点における入力速度を求める. 折れ線近似目標軌道の角の点における精度を上げるために, 始点における  $3\tau$  の時間間隔は最小速度  $V_{min}$  を与える. それから, 時間間隔  $T_a$  で最小速度  $V_{min}$  から目標速度  $V$  まで加速度  $a$  で加速する. 速度が目標速度  $V$  となった時点で, 時間間隔  $T_v$  の間, 入力速度を目標速度  $V$  で固定する. 時間間隔  $T_v$  は, 入力速度の積分が線分の長さ  $L$  に等しくなるように

$$T_v = \frac{L - 3V_{min}\tau - (V + V_{min})T_a}{V} \quad (6.7)$$

と決定される.

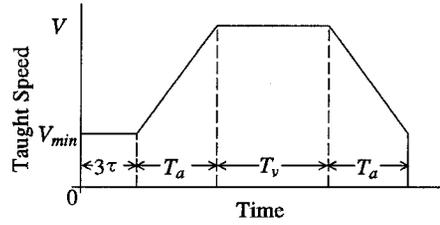


Fig. 6.3 Taught speed data by using the modified speed data method for each line of the approximated locus.

結果として、入力速度は

$$\dot{u}(t) = \begin{cases} V_{min} & (0 \leq t < 3\tau) \\ V_{min} + a(t - 3\tau) & (3\tau \leq t < 3\tau + T_a) \\ V & (3\tau + T_a \leq t < 3\tau + T_a + T_v) \\ V - a(t - 3\tau - T_a - T_v) & (3\tau + T_a + T_v \leq t \leq 3\tau + 2T_a + T_v) \end{cases} \quad (6.8)$$

と与えられる。

教示速度修正法によって修正された入力速度を Fig. 6.3に示す。教示速度修正法は、教示位置は目標軌跡そのまま与え、教示速度は(6.8)式により折れ線近似目標軌跡に含まれる各線分に対して教示する。そのことにより、目標軌跡に追従軌跡に近づけることができる。

### 6.2.3 パラメータの選定

(6.6)式の $T_a$ を計算するために、パラメータ $V, V_{min}, R$ を選定する必要がある。目標速度 $V$ はロボットアームの作業目的によって決定される。最小速度 $V_{min}$ は作業効率、作業精度、作業対象の材質などによって決定される。許容パラメータ $R$ は精度と制約条件から選定される。最初の制約条件は、折れ線近似目標軌跡の角における加速度が、ロボットアームのアクチュエータであるサーボモータの最大加速度を越えないようにすることである。すなわち、

$$\frac{V_{min}^2}{R} \leq \alpha_{max} \quad (6.9)$$

を満たすように許容パラメータ $R$ を選定する必要がある。ここで $\alpha_{max}$ はサーボモータの最大加速度を表す。次の制約条件は、許容パラメータ $R$ は最小速度 $V_{min}$ の定常速度偏差 $V_{min}/K_p$ 以上でなければならないことである。すなわち

$$R \geq \frac{V_{min}}{K_p} \quad (6.10)$$

と表される。結果として、許容パラメータ $R$ は制約条件(6.9)式と(6.10)式を満足するように

$$\max \left\{ \frac{V_{min}}{K_p}, \frac{V_{min}^2}{\alpha_{max}} \right\} \leq R \quad (6.11)$$

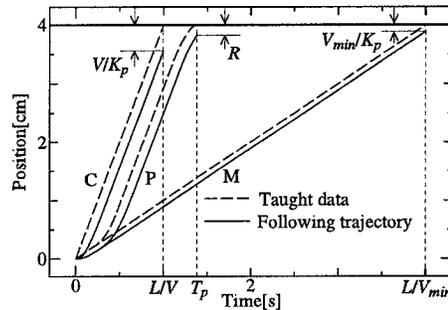


Fig. 6.4 Comparison of the proposed method (P), the conventional method (C) and the method (M) by computer simulations.

と選定する必要がある。許容パラメータ  $R$  を小さく選ぶと操作時間が長くなり、許容パラメータを  $R = V_{min}/K_p$  と選定した場合には、教示速度は最小速度  $V_{min}$  を与えることになる。一方、許容パラメータ  $R$  を大きく選ぶと、操作時間は短くなる。

## 6.3 教示速度修正法の評価

### 6.3.1 操作時間と精度

提案した教示速度修正法の有効性を確認するため、教示速度修正法 (P) と他の 2 つの手法 (C), (M) との比較を行なう。手法 (C) は従来法、すなわち、教示位置と速度ともに目標軌道をそのまま入力する手法を表し、手法 (M) は、教示位置は目標軌跡と同じであるが、教示速度を最小速度  $V_{min}$  で与える手法を表す。

最初に、線分  $L$  の終点における目標軌跡と追従軌跡との誤差を評価する。教示速度修正法 (P) では、誤差は許容パラメータ  $R$  で決定される。手法 (C) と (M) における誤差は、ロボットアーム制御系 (6.1) 式の定常速度偏差によって、それぞれ、 $V/K_p$  と  $V_{min}/K_p$  で与えられる。結果として、手法 (M) が最も正確な追従軌跡を与える。

次に、三つの手法の操作時間を比較する。教示速度修正法 (P) の操作時間は  $T_p = T_v + 2T_a + 3\tau$  で与えられる。手法 (C) と (M) における操作時間は、距離  $L$  を速度で割ることにより、それぞれ  $L/V$  と  $L/V_{min}$  と計算できる。手法 (C) が最も短い操作時間を与え、手法 (M) が最も長い操作時間を与える。ただし、目標軌跡の長さが十分長いときには、教示速度修正法 (P) と手法 (C) の操作時間はほとんど等しくなる。

Fig. 6.4 に三つの手法の操作時間と精度を表す。計算機シミュレーション条件は以下の通りである。位置ループゲイン  $K_p$  と速度ループゲイン  $K_v$  はそれぞれ  $10[1/s]$  と  $58[1/s]$  とし、目標軌跡の線分の長さ  $L$  は  $4[cm]$ 、目標速度  $V$  は  $4[cm/s]$  で、最小速度  $V_{min}$  は  $1[cm/s]$  とした。また、アクチュエータの最大加速度  $\alpha_{max}$  は  $42.2[cm/s^2]$  とした。教示速度修正法の許容パラメータ  $R$  は  $0.2[cm]$  と選んだ。教示速度修正法 (P) の誤差 ( $0.2[cm]$ ) は、従来法 (C) の誤差 ( $0.4[cm]$ ) より小さく、手法 (M) の誤差 ( $0.1[cm]$ ) より少し大きい。教示速度修正法 (P) の操作時間 ( $1.37[s]$ ) は、手法 (M) の操作時間 ( $4.0[s]$ ) よりかなり短く、従来法 (C) の操作時間 ( $1.0[s]$ ) とほぼ等しい。こ

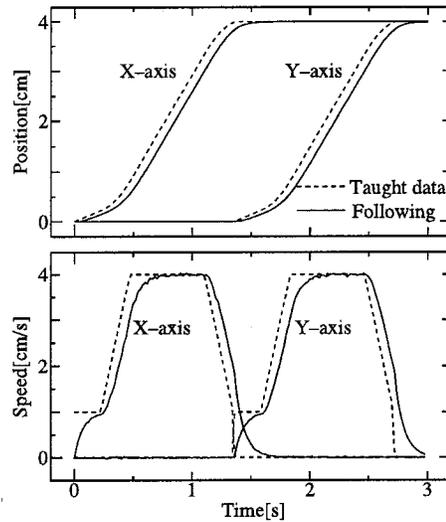


Fig. 6.5 Experimental results for contour control of an XY table (Following trajectory).

の結果，教示速度修正法 (P) は，精度と操作時間との総合的判断から，従来法 (C) や手法 (M) と比較して優れているといえる。

次に，教示速度修正法のロバスト性について考察する．産業用ロボットアーム制御系 (6.1) 式は，オーバシュートが生じないように，適切に調整されている．ロボットアーム制御系の 2 次系モデル (6.1) 式にモデル化誤差が存在する場合，(6.8) 式中のパラメータ  $\tau$ ,  $T_a$ ,  $T_v$ ,  $a$  が最適値からずれるために，教示速度は最適とはならない．しかし，教示速度修正法は教示位置には修正を加えないため，モデル化誤差が存在しても，オーバシュートが生じることはない。

### 6.3.2 実験結果

教示速度修正法の有効性を確認するために，直交型の産業用ロボットと考えることができる XY テーブルを用いて輪郭制御実験を行なった．XY テーブルは，X 軸 Y 軸それぞれ独立にサーボモータによって動作する．位置ループゲインと速度ループゲインは，それぞれ  $K_p = 10[1/s]$ ,  $K_v = 58[1/s]$ ，サーボモータの最大加速度は  $\alpha_{max} = 42.2[cm/s^2]$  である．目標軌跡の線分の長さは  $L = 4[cm]$ ，目標速度は  $V = 4[cm/s]$ ，最小速度は  $V_{min} = 1[cm/s]$  とした．許容パラメータは  $R = 0.2[cm]$  とし，これは制約条件 (6.11) 式を満足する．実験結果の追従軌道を Fig. 6.5，追従軌跡を Fig. 6.6 に示す．また，比較のために，Fig. 6.6 には従来法を用いた実験結果も載せる．点線は目標軌跡，実線は教示速度修正法を用いた追従軌跡，破線は従来法を用いた追従軌跡を表す．Fig. 6.5 から，操作時間は従来法 (2[s]) と比較して，多少時間がかかっている (2.74[s]) ことがわかる．Fig. 6.6 から追従軌跡は目標軌跡と良く一致しており，教示速度修正法の角における誤差は  $0.2[cm]$  であった．教示速度修正法における追従軌跡の誤差は，従来法 ( $0.4[cm]$ ) と比較してかなり改善されている．実験結果から教示速度修正法の有効性が示された．

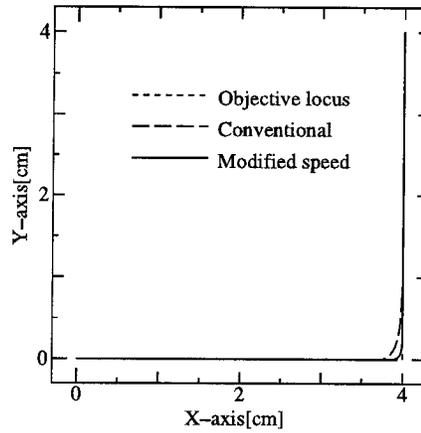


Fig. 6.6 Experimental results for contour control of an XY table (Following locus).

#### 6.4 まとめ

産業用ロボットアームの輪郭制御の制御性能を改善させるための教示速度修正法を提案した。この手法は、追従軌跡を目標軌跡に近づけるために、速度情報のみを修正する手法である。本手法の有効性を、直交座標ロボットアームとみなすことができるXYテーブルの輪郭制御実験に適用して確認した。教示速度修正法は、装置のハードを変更する必要はなく、教示信号を修正するのみであるので、5章で提案した教示信号修正法と比較して、容易に利用可能である。

## 第7章

# ガウシアンネットを用いた教示信号修正法

### 7.1 まえがき

産業用ロボットアームの輪郭制御を行なう際に、通常は目標軌道そのままロボットアームの入力として用いることが多いが、動作が速いときにはロボットアームのダイナミクスの遅れなどにより、目標軌道と追従軌道との間にずれが生じることを経験する。その問題点の対策として、5章ではロボットアームの入力に目標軌道そのまま用いるのではなく、目標軌道に修正を施す教示信号修正法を提案した。この教示信号修正法を用いると、ロボットアームのモデルが正確な場合には、目標軌道と追従軌道が一致することが確認されている。しかし、ロボットアームの厳密なモデルを得ることは常に可能とはいえず、何らかのモデル化誤差が存在することが多い。また、6章で提案した教示速度修正法は、モデル化誤差に対してはロバストであるが、目標軌跡が折れ線近似される場合に限定される。そこで、本章においては、目標軌道の修正にニューラルネットの一つであるガウシアンネット<sup>[44]</sup>を用いる方法を提案する。ニューラルネットは学習機能、関数近似、情報の記憶など多くの魅力的な特徴を備えており、制御の分野にも数多く応用されている<sup>[45, 46, 47]</sup>。ニューラルネットのユニットがガウス関数であるガウシアンネットは、適応制御に応用されていて<sup>[48]</sup>、プラントの非線形性に対する適応補償器として用いられている。目標軌道の修正にガウシアンネットを用いることによって、ネットの構造と初期値の決定が系統的に行なうことができ、学習機能を有するためにロボットアームのモデル化誤差を減らすことが可能となる。

本章は、ガウシアンネットを用いた教示信号修正法を提案する。ガウシアンネットの構造と初期値の決定を、ロボットアーム制御系の2次系モデルに基づいて決定し、実データを用いて学習することにより高精度の輪郭制御を実現する。

### 7.2 ガウシアンネットの構造と初期値決定のための数式モデル

ガウシアンネットの構造と初期値を決定するために必要となる、産業用ロボットアームの数式モデルについて述べる。実際のサーボ系に近い、速度ループまでを考慮した Fig. 7.1 の2次系モデルを、ロボットアーム制御系の数式モデルとして採用する。Fig. 7.1において、 $K_p[1/s]$ 、 $K_v[1/s]$  はそれぞれ位置ループゲイン、速度ループゲインを表し、二つの  $1/s$  の部分はサーボモータとメカニズムを表す。産業界で

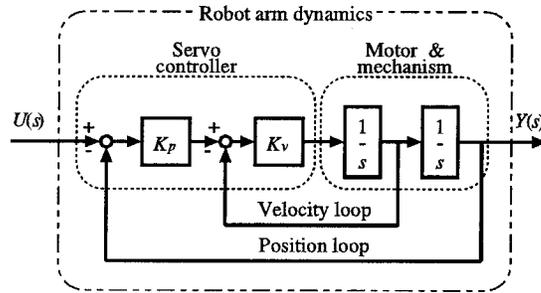


Fig. 7.1 Second order model of a robot arm dynamics

は、速度ループの制御手法には比例制御や比例積分制御が行なわれているが、ここでは速度ループ全体の特性を1次系で表して、位置制御と速度制御とを合わせて近似的に2次系モデルで表現する<sup>[22]</sup>。この2次系モデルの数式は

$$\ddot{y}(t) = -K_v \dot{y}(t) - K_p K_v y(t) + K_p K_v u(t) \quad (7.1)$$

と表される。ここで、 $u(t)$ はロボットアームへの位置入力、 $y(t)$ はロボットアームの位置出力を表す。また、ガウシアンネットで構成するロボットアームの逆システムは、 $r(t)$ を目標軌道とすると、(7.1)式を $y(t) = r(t)$ として逆に解くことにより

$$u(t) = r(t) + \frac{1}{K_p} \dot{r}(t) + \frac{1}{K_v K_p} \ddot{r}(t) \quad (7.2)$$

となる。ただし、目標軌道 $r(t)$ は2階微分可能であると仮定する。この数式モデルはロボットアームの特性の本質を表し得るが、モデル化誤差を含むという仮定で、7.3節のガウシアンネットによりそのロボットアームの学習を行なうことにする。

## 7.3 ガウシアンネットを用いた教示信号修正法

### 7.3.1 教示信号修正法の考え方

教示信号修正法は、ロボットアームへの入力である教示信号を修正することによって、ロボットアームのダイナミクスの遅れを補償する方法である(5章, 6章参照)。教示信号を修正する際に、ロボットアームのモデル化が正確である場合には、そのモデルに基づいて教示信号を修正する修正要素を構成し、高精度な輪郭制御が可能となるが、一般には正確なモデルを得ることは困難であり、(7.2)式中に何らかのモデル化誤差が存在することが多い。そこで、実際のロボットアームの特性を学習することができるガウシアンネットを用いて、修正要素を構成する。Fig. 7.2にガウシアンネットを用いた教示信号修正法の概念図を示す。

### 7.3.2 ガウシアンネットによる逆システムの構成

#### (a) ガウシアンネット

ガウシアンネットは、ユニットにガウス関数(ガウシアンユニット)を用いるニューラルネットのことである<sup>[44]</sup>。本章で採用するガウシアンユニットは、構造と学習を

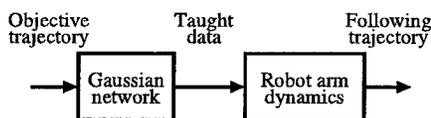


Fig. 7.2 Modification of taught data by using a Gaussian network

簡単化するために1入力のユニットを用いる。

本章で採用するガウシアンネットワークは、複数のユニットの線形結合

$$\phi(\mathbf{x}) = \sum_{i=1}^M w_i \psi_i(x_i) \quad (7.3)$$

からなり、各ユニットは、1入力のガウシアンユニット

$$\psi_i(x_i) = \exp\left(-\frac{(x_i - m_i)^2}{2\sigma_i^2}\right) \quad (7.4)$$

である。ここで、 $\mathbf{x} = (x_1, \dots, x_M)$  はネットへの入力、 $\phi(\mathbf{x})$  はネットの出力、 $M$  はユニット数、 $w_i$  は  $i$  番目のユニットの重み、 $\psi_i(x_i)$  は  $i$  番目のユニットの出力、 $m_i$  は  $i$  番目のユニットの平均値、 $\sigma_i$  は  $i$  番目のユニットの標準偏差を表す。このガウシアンネットワーク (7.3) 式によって、逆システム (7.2) 式の近似を行なう。ただし、ガウシアンネットワークで逆システムを構成するために、目標軌道は2階微分可能である必要がある。

#### (b) 構造決定

採用するガウシアンネットワークは、Fig. 7.3に示すように、3層、3入力、6中間ユニット、1出力とする。ここで、3入力とは、6中間ユニットへの入力  $\mathbf{x} = (x_1, \dots, x_6)$  のうちの2入力ずつを同じ入力とすることで実現している。ネットへの入力は  $(r, \dot{r}, \ddot{r})$ 、すなわち、 $x_1 = x_2 = r$ 、 $x_3 = x_4 = \dot{r}$ 、 $x_5 = x_6 = \ddot{r}$  として、第1, 2ユニットで逆システム (7.2) 式の右辺第1項、第3, 4ユニットで右辺第2項、第5, 6ユニットで右辺第3項を近似する。ガウシアンネットワークの出力  $\phi(\mathbf{x})$  が修正された教示信号となり、ロボットアームへの入力を表す。図7.3中の●はガウシアンユニットを表し、○は線形ユニットを表す。

#### (c) 初期値決定

ガウシアンネットワークで (7.2) 式の逆システムを近似するように、パラメータの初期値を決定する。初期値の決定において、Fig. 7.3で表されるガウシアンネットワークを三つの部分に分けて、1入力、2中間ユニット、1出力のガウシアンネットワークを考える。このガウシアンネットワークで、一般の線形関数  $y = ax$  を近似するために、二つのユニットの平均値だけ符号を変えて

$$\phi(x) = w \exp\left(-\frac{(x - m)^2}{2\sigma^2}\right) - w \exp\left(-\frac{(x + m)^2}{2\sigma^2}\right) \quad (7.5)$$

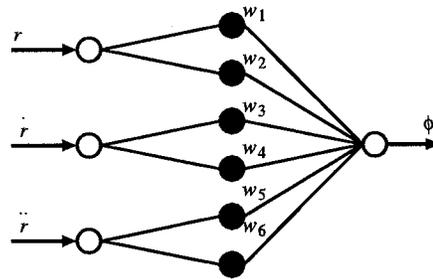


Fig. 7.3 Structure of the Gaussian network

とする. (7.5) 式をテイラー展開1次近似すると

$$\phi(x) \approx \frac{2wm}{\sigma^2} \exp\left(-\frac{m^2}{2\sigma^2}\right) x \quad (7.6)$$

となり, 線形関数の傾きを

$$a = \frac{2wm}{\sigma^2} \exp\left(-\frac{m^2}{2\sigma^2}\right) \quad (7.7)$$

とすれば,  $x = 0$  付近においてはほぼ線形とみなすことができる. 分散 $\sigma$ と平均値 $m$ の関係を変化させたときのグラフを Fig. 7.4に示す. 平均値 $m$ に掛かる係数を0.01ごとに変化させた結果,  $\sigma = 0.57m$ とすれば,  $\phi(x)$ は比較的 $x$ の広い領域において $ax$ を近似できることがわかった. すなわち,  $x_{max}$ をガウシアンネットワーク $\phi(x)$ が線形近似を行なう領域を決定するパラメータとすると, (7.7)式と $\sigma = 0.57m$ を用いて, ガウシアンネットワークのパラメータを

$$m = x_{max}, \sigma = 0.57x_{max}, w = 0.757ax_{max} \quad (7.8)$$

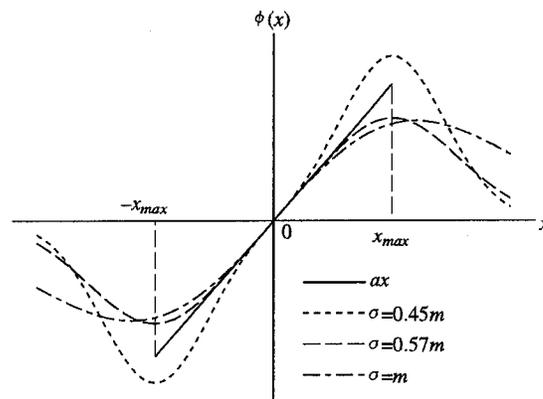


Fig. 7.4 Determination of initial parameters of the Gaussian network

とすれば、線形関数を近似できる。このとき、 $\phi(x)$ は $x = -x_{max}$ で最小値 $-0.755ax_{max}$ 、 $x = x_{max}$ で最大値 $0.755ax_{max}$ をとる。

この関係式を利用して、3入力6ユニット1出力の全体のガウシアンネットの初期値を

$$\left. \begin{aligned} m_1 &= -m_2 = X_{max} \\ \sigma_1 &= \sigma_2 = 0.57X_{max} \\ w_1 &= w_2 = 0.757X_{max} \end{aligned} \right\} \quad (7.9a)$$

$$\left. \begin{aligned} m_3 &= -m_4 = V_{max} \\ \sigma_3 &= \sigma_4 = 0.57V_{max} \\ w_3 &= w_4 = \frac{0.757V_{max}}{K_p} \end{aligned} \right\} \quad (7.9b)$$

$$\left. \begin{aligned} m_5 &= -m_6 = A_{max} \\ \sigma_5 &= \sigma_6 = 0.57A_{max} \\ w_5 &= w_6 = \frac{0.757A_{max}}{K_p K_v} \end{aligned} \right\} \quad (7.9c)$$

と与えると、初期値においてガウシアンネットは、(7.2)式の逆システムを適切に近似することになる。ここで、(7.9)式中の $X_{max}$ 、 $V_{max}$ 、 $A_{max}$ は、それぞれ、位置、速度、加速度に対する線形近似領域を決定する設計パラメータ $x_{max}$ の具体的変数を表す。

#### (d) 学習法

初期パラメータのガウシアンネットにより、(7.2)式で表される逆システムを近似することができる。しかし、一般には(7.1)式で表されるロボットアームの数式モデルにはモデル化誤差が存在するため、(7.2)式の逆システムモデルにも実際のロボットアームからの誤差を含むことになる。そのモデル化誤差を減らすために、実際にロボットアームを動かしたときの実験結果を教師信号に用いて、ガウシアンネットの学習を行なう。ガウシアンネットの学習の際の損失関数として

$$E_{rms} = \sqrt{\frac{2}{K} \sum_{k=1}^K E^k} \quad (7.10)$$

$$E^k = \frac{1}{2}(u^k - \phi(\mathbf{x}^k))^2 \quad (7.11)$$

とする。ここで $(u^k, \mathbf{x}^k) = (u^k, x_1^k, \dots, x_6^k)$ はガウシアンネット学習時の教師信号を表し、 $K$ は教師信号の数を表す。

ガウシアンネットのパラメータの学習には、誤差逆伝搬法を用いる<sup>[49]</sup>。パラメータとその変化分を、それぞれ、 $\mathbf{p}_i = (w_i, m_i, \sigma_i)$ 、 $\Delta \mathbf{p}_i = (\Delta w_i, \Delta m_i, \Delta \sigma_i)$ 、 $i = 1, \dots, 6$ 、学習率を $\eta$ とすると

$$\mathbf{p}_i^{\text{new}} = \mathbf{p}_i^{\text{old}} + \eta \Delta \mathbf{p}_i, \quad i = 1, \dots, 6 \quad (7.12)$$

$$\begin{aligned}\Delta w_i &= -\frac{\partial E^k}{\partial w_i} \\ &= (u^k - \phi(\mathbf{x}^k))\psi_i(x_i^k)\end{aligned}\quad (7.13a)$$

$$\begin{aligned}\Delta m_i &= -\frac{\partial E^k}{\partial m_i} \\ &= \frac{(x_i^k - m_i)}{(\sigma_i)^2}\psi_i(x_i^k)(u^k - \phi(\mathbf{x}^k))w_i\end{aligned}\quad (7.13b)$$

$$\begin{aligned}\Delta \sigma_i &= -\frac{\partial E^k}{\partial \sigma_i} \\ &= \frac{(x_i^k - m_i)^2}{(\sigma_i)^3}\psi_i(x_i^k)(u^k - \phi(\mathbf{x}^k))w_i\end{aligned}\quad (7.13c)$$

で表される。学習は、(7.10)式の損失関数がいき値以下になった時点で終了する。ガウシアンネットの学習は、すべてのパラメータについておこなうため、Fig. 7.3の構造で表し得るすべての関数を表現できる。この学習により、ガウシアンネットは実際のロボットアームの逆システムを学習することが可能になる。例えば、ロボットアームの入力の符号によって、ロボットアームの特性が変わって、入力 of 正負によって線形関数の傾き  $a$  が変化する場合などの、線形ニューラルネットでは表すことのできないような非線形性にも対応できる。

### 7.3.3 ガウシアンネットの適用

学習したガウシアンネットを、Fig. 7.2のガウシアンネット部に用いる。ガウシアンネットがロボットアームの逆システムを表して、ガウシアンネットの出力はロボットアームに対する修正教示信号を作りだし、その教示信号によって動作されたロボットアームは、追従軌道が目標軌道に近づくことが期待できる。ガウシアンネットはロボットアームの逆システムを学習するため、実際の作業時における目標軌道と、教師信号作成時の目標軌道とを一致させる必要はない。言い替えると、一度ロボットアームの逆システムをガウシアンネットが学習すると、そのガウシアンネットを修正要素に用いることによって、任意の目標軌道に対して追従軌道为目标軌道に近づけることができる。

## 7.4 XY テーブルによる輪郭制御実験

### 7.4.1 実験条件

7.3節で提案したガウシアンネットを用いた教示信号修正法の有効性を検証するために、直交座標ロボットと取り扱うことができるXYテーブルを用いて輪郭制御実験を行なった。採用したXYテーブルは、2個のサーボモータそれぞれに取り付けられたボールネジによって、テーブルがX軸とY軸に独立に動くものである。XYテーブルの制御は、X軸Y軸それぞれが独立に行なわれているため、(7.3)式のガ

ウシアンネットも軸ごとに独立に2個構成した。XY テーブルの目標軌道が

$$u_x(t) = \begin{cases} 4.8 & (0 \leq t < 0.5) \\ 4 \cos\left(\frac{\pi(t-0.5)}{2}\right) + \frac{4}{5} \cos\left(\frac{5\pi(t-0.5)}{2}\right) & (0.5 \leq t < 4.5) \\ 4.8 & (4.5 \leq t \leq 5) \end{cases}$$

$$u_y(t) = \begin{cases} 0 & (0 \leq t < 0.5) \\ 4 \sin\left(\frac{\pi(t-0.5)}{2}\right) + \frac{4}{5} \sin\left(\frac{5\pi(t-0.5)}{2}\right) & (0.5 \leq t < 4.5) \\ 0 & (4.5 \leq t \leq 5) \end{cases}$$

のときの輪郭制御実験結果を示す。

#### 7.4.2 学習データ生成

ガウシアンネットの初期パラメータの決定において、(7.9) 式中の位置ループゲインに関しては、装置で設定した値  $K_p = 5[1/s]$  を用いて、速度ループゲインに関しては、直接設定することが不可能であったため、産業界で用いられている臨界条件  $K_v = 4K_p$  から  $K_v = 20[1/s]$  とした。この  $K_v$  は、実際の装置から測定した値ではないため、かなり大きな誤差が存在すると考えられるが、本手法では修正要素にガウシアンネットを用いるために、学習によって実際の装置の逆システムを構成し、高精度輪郭制御が可能となる点が特長である。

また、装置の線形近似領域の条件として、テーブルの可動領域が  $15[\text{cm}]$  であることを考慮して、 $X_{max} = 10[\text{cm}]$  とし、位置に関する二つのガウシアンユニットの出力範囲が  $-7.55 \leq \phi(r) \leq 7.55[\text{cm}]$  となるように設定した。速度に関しても、装置の最高速度が  $9.3[\text{cm/s}]$  であることを考慮して、 $V_{max} = 15[\text{cm/s}]$  とし、速度に関する二つのガウシアンユニットの出力範囲が  $-11.325 \leq \phi(\dot{r}) \leq 11.325[\text{cm/s}]$  となるように設定した。加速度に関しては、装置の安全性を考えて、 $A_{max} = 80[\text{cm/s}^2]$  とし、加速度に関する二つのガウシアンユニットの出力範囲が  $-60.4 \leq \phi(\ddot{r}) \leq 60.4[\text{cm/s}^2]$  として、最大加速度  $84.7[\text{cm/s}^2]$  を越えないように設定した。以上の初期パラメータで構成したガウシアンネットを学習するための教師信号は、ロボットアームの特性が表されている動作であれば任意に与えることができるが、この実験では、目標軌道をそのまま入力してXY テーブルを動作させて、そのときの出力データをパソコンに取り込んで作成した。教師信号を作成する際のサンプリング間隔は、輪郭制御実験時と同じ  $\Delta t = 10[\text{ms}]$  とし、 $(u^k, \mathbf{x}^k) = (u(k\Delta t), y(k\Delta t), y(k\Delta t), \dot{y}(k\Delta t), \dot{y}(k\Delta t), \ddot{y}(k\Delta t), \ddot{y}(k\Delta t))$ ,  $k = 0, \dots, 500$  を教師信号とした。ただし、実際にXY テーブル装置からパソコンに取り込めるデータは、サーボモータに取り付けられたタコジェネレータの速度出力  $\dot{y}$  のみである。そこで、位置出力  $y$  は速度出力の数値積分、加速度出力  $\ddot{y}$  は速度出力の数値微分によって求めた。そこで、タコジェネレータの速度出力  $\dot{y}$  は、 $0 \sim 10[\text{Hz}]$  のローパスフィルターを用いて雑音除去を行なった。ガウシアンネット学習の際の学習率は  $\eta = 0.001$  とし、X 軸 Y 軸共にいき値  $0.35[\text{mm}]$  以下となったときに学習を終了させた。教師信号  $(u^k, \mathbf{x}^k)$ ,  $k = 0, \dots, 500$  のデータセットに対する学習を1回の学習回数と数え

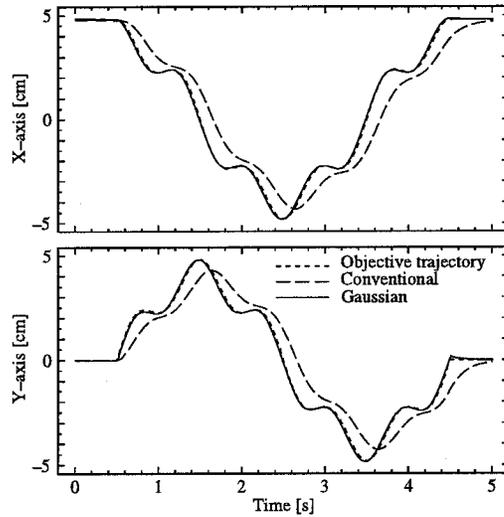


Fig. 7.5 Experimental results by using the Gaussian network (Following trajectory)

ると、このときの学習回数は182回であった。

### 7.4.3 輪郭制御実験結果

学習後のガウシアンネットワークを Fig. 7.3中のガウシアンネットワーク部に用いて、そのガウシアンネットワークで修正された教示信号を、XY テーブルの入力として用いたときの輪郭制御実験結果を示す。Fig. 7.5に学習後のガウシアンネットワークによる実験結果の時間推移を表す追従軌道を示し、Fig. 7.6にXY 平面上の時間のパラメータを消去した追従軌跡を示す。比較のために、修正を施さず目標軌道そのまま入力した場合の実験結果も同じ図中に示す。

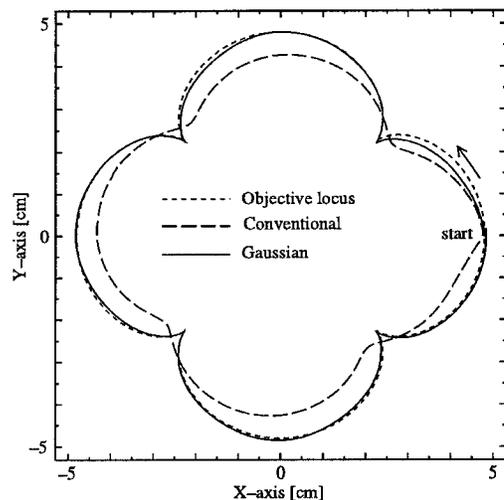


Fig. 7.6 Experimental results by using the Gaussian network (Following locus)

追従軌道と追従軌跡のいずれにおいても、修正を施さない従来法と比較して、ガウシアンネットによって修正を施した場合の方が、明らかに目標に近づいていて、高精度制御が行なわれていることがわかる。これらのことから、ガウシアンネットを用いて修正教示信号を作る本手法の妥当性が示せて、高精度の輪郭制御が達成できた。

## 7.5 まとめ

ガウシアンネットを用いた教示信号修正法を提案した。提案した手法は、ガウシアンネットを用いて逆ダイナミクスを学習し、そのガウシアンネットを用いて目標軌道を修正することにより、輪郭制御性能を向上させることができる。提案した手法の有効性を確認するために、XY テーブルを用いた輪郭制御実験を行なった。輪郭制御実験結果から、目標軌道をそのまま入力する従来手法と比較して、明らかにガウシアンネットを用いた方が制御性能が向上することが確認された。

## 第 8 章

# 従軸逆ダイナミクスを用いた主軸位置同期制御法

### 8.1 まえがき

産業用ロボットアームを用いた作業において、複数のサーボ系の位置同期が様々な分野から要求されている。複数のサーボ系の位置決め制御を考える場合、それぞれのサーボ系の同期を図る必要がないとみなせる場合には、それぞれのサーボ系に独立に制御系を組むことにより高精度な位置決め制御が実現可能である<sup>[36, 42, 43]</sup>。しかし、サーボ系を位置同期させる必要がある場合、それぞれのサーボ系を独立に制御しても位置同期を図ることはできない。そこで、この位置同期を図る手法として次の二つが考えられる。一つは複数のサーボ系の動特性および応答特性を一致させて、互いのサーボ系の指令を同期させることにより、位置出力を同期させる方法、もう一つは動特性や応答特性を一致させることの難しいサーボ系に対して、応答特性の遅いサーボ系の出力を応答特性の速いサーボ系に入力する方法である。前者は、一般的なサーボモータの同期制御に用いられ、動特性や応答特性さえ一致させれば高精度な位置同期制御が実現できる。この例として走行台車の両輪駆動の位置決め及び位置同期が挙げられる。しかし、サーボ系に外乱が加わると高精度な位置同期はできない。後者は、応答特性の遅いサーボ系への指令を出力したのち、その応答を観測しながら、応答特性の速いサーボ系への指令を演算出力するため、応答特性の速いサーボ系の制御性能により同期制御の精度が決定される。この手法を用いている例として、工作機械で行うタップ加工作業が上げられる。タップ加工とは、ドリルにより下穴した後、回転主軸に加工刃を取り付け、回転しながら上下動作を行うことで、ネジを形成する加工であり、ネジピッチを一定にすることが必要となる。タップ加工で使う回転方向と上下方向の 2 軸は、それぞれサーボ系の動特性が異なるために、機械的補正が行われている。しかし、これでは高精度な位置同期ができないので、応答の良い上下方向にむだ時間や 1 次遅れのブロックを通す手法<sup>[50]</sup>やセルフタッピング動作を行わせる手法<sup>[51]</sup>などが用いられている。しかしながら、これらの手法においても、回転方向の僅かな速度低下(外乱)による制御性能劣化や、メカニズムが複雑になるなどの欠点があり、高精度な位置同期を実現することは困難である。そこで、これらの欠点を克服し、主軸位置同期制御が可能な手法として、2 つのサーボ系間の追従制御方法という手法<sup>[52]</sup>が提案されている。この手法は、制御

パラメータが適切であれば、良好に位置同期を行うことができるが、工作機械の用途に応じてパラメータを調節する必要がある、また過渡状態の位置同期が図れないという欠点がある。

本章では、追従させる従軸のサーボ系の逆ダイナミクスを修正要素とし、主軸の位置出力をこの修正要素に通して従軸の入力信号とする、主軸位置同期制御法を提案する。この主軸位置同期制御法は、主軸のサーボ系と従軸のサーボ系の遅れの差を補償して、複数のサーボ系の指令値から出力までの遅れが等しくなるため、主軸と従軸の位置同期を図ることができる。この主軸位置同期制御法は、位置同期の必要な装置のメカニズム、および、制御系のハードウェアにはなんらの影響をも与えず、しかも、それぞれの用途に合わせたパラメータなどの面倒な設定などを必要とせず、どの制御対象にもそのまま利用できるという利点があり、また、過渡状態においても高精度な位置同期が実現できる。

提案する主軸位置同期制御法の特性解析を行い、また、計算機内で作成された様々な外乱下のシミュレーション、および、XY テーブルによる主軸位置同期制御実験とXY テーブルに直接外乱を加える主軸位置同期制御実験から、提案する主軸位置同期制御法の有効性を検討する。

## 8.2 主軸位置同期制御対象の数式モデルと主軸位置同期制御の必要性

### 8.2.1 主軸位置同期制御対象の数式モデル

位置同期の必要な制御対象においては、制御装置とサーボモータを含んだ全体の制御系は、主軸と従軸が独立に制御されていることが多い。そのアクチュエータとしてサーボモータが多く用いられているが、サーボモータとその制御装置には高性能の装置が採用されているため、加工速度がそれほど速くない場合には、サーボモータの速度制御の特性をほぼ定数とみなして、位置制御の特性のみを考えればよい<sup>[13]</sup>。従って、サーボ系の伝達関数は

$$X(s) = \frac{K_x}{s(s + K_x)}U_x(s) + \frac{1}{s + K_x}D_x(s) \quad (8.1a)$$

$$Y(s) = \frac{K_y}{s(s + K_y)}U_y(s) \quad (8.1b)$$

で表される。ここで、X 軸は主軸、Y 軸は従軸を表し、 $X(s)$ 、 $Y(s)$  は X 軸、Y 軸の位置を、 $U_x(s)$ 、 $U_y(s)$  は X 軸、Y 軸における速度入力指令を、 $K_x$ 、 $K_y$  は X 軸、Y 軸のアクチュエータの位置ループゲインをそれぞれ表す。外乱は、タップ加工を想定し、主軸にのみ加わるものとし、 $D_x(s)$  として表す。(8.1a) 式の第1項は速度入力  $U_x(s)$  から X 軸の位置出力への関係を表しており、第2項は X 軸に入る外乱  $D_x(s)$  から X 軸の位置出力への関係を表している。このブロック線図は Fig.8.1 のように表すことができる。制御系の特性は  $K_x$ 、 $K_y$  で表現され、これはハードの構成によってあらかじめ決まっている値である。また、サーボ系の前の  $1/s$  は速度入力を位置入力に変換する積分を意味する。

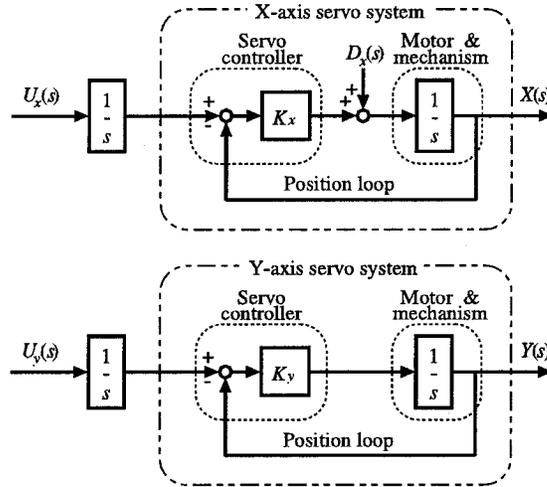


Fig. 8.1 Block diagram of two servo systems composed of X-axis and Y-axis.

主軸位置同期制御の制御目的は、X軸及びY軸の位置出力を同期させる、すなわち

$$Y(s) = kX(s) \tag{8.2}$$

が成立することである。ただし、 $k$ は比例定数である。X軸、Y軸の位置出力が(8.2)式を満たすときに、位置同期されていることになる。

### 8.2.2 位置同期を図らない場合の問題点

X軸とY軸の動特性を考慮せずに、Y軸の速度入力指令  $U_y(s)$  をX軸の速度入力指令  $U_x(s)$  の  $k$ 倍として入力した場合の、Y軸の位置出力は

$$Y(s) = \frac{kK_y}{s(s + K_y)}U_x(s) \tag{8.3}$$

となる。X軸に対するY軸の位置出力誤差は(8.1a)、(8.3)式から

$$kX(s) - Y(s) = \frac{k(K_x - K_y)}{(s + K_x)(s + K_y)}U_x(s) + \frac{k}{s + K_x}D_x(s) \tag{8.4}$$

となる。(8.4)式より、位置同期を図らない場合は、位置出力誤差が0とならないためX軸の位置出力とY軸の位置出力は同期しない。これは、X軸、Y軸の位置ループゲインが異なるために、位置出力がずれるためである。このことから、X軸の速度入力指令をそのまま用いるだけでは、X軸及びY軸の位置ループゲインが同じでない限り同期できない。また、X軸、Y軸の動特性が等しい場合でも、X軸の外乱  $D_x(s)$  に対するY軸の補償を設けていないので、外乱  $D_x(s)$  に対する補償も行う必要がある。

### 8.3 主軸位置同期制御法

#### 8.3.1 主軸位置同期制御法の導出

前節で、X軸の速度入力指令をそのままY軸の速度入力指令とした場合の問題点を述べたが、これから、Y軸の位置をX軸の位置と同期するために、X軸の速度入力指令をX軸とY軸のダイナミクスの違いを補償するように修正を施して、Y軸の速度入力指令とすることによって、位置同期を図ることが考えられる。しかし、X軸の速度入力指令に修正を施してY軸の速度入力指令とするだけでは、X軸に入る外乱に対する補償を行うことができず、高精度な位置同期は実現できない。そこで、X軸の位置出力をフィードバックしてY軸の位置入力信号とすれば、X軸の位置出力をフィードバックしているためX軸に入る外乱の影響を抑えることができる。しかしながら、X軸の位置出力をそのままフィードバックするだけでは、Y軸のダイナミクスに起因する動作の遅れによってX軸との同期は図れない。そこで、Y軸の1次系の逆ダイナミクスを利用して、X軸の位置出力フィードバック信号を修正すれば位置同期を図ることができる。すなわち、Y軸の動特性が1となるように、Y軸の逆ダイナミクスによりフィードフォワード補償を行う。

以上に述べた特性を有するように、修正要素  $F_s(s)$  をY軸の1次系の逆ダイナミクス

$$F_s(s) = \frac{s + K_y}{K_y} \quad (8.5)$$

として構成し、X軸の位置出力を  $F_s(s)$  に通して、その信号をY軸の位置入力信号として採用する、主軸位置同期制御法を提案する。この主軸位置同期制御法は、X軸とY軸との動特性の違いを調整し、X軸に混入する外乱  $D_x(s)$  による悪影響を補償する手法である。そこで、X軸の位置出力をフィードバックする際には、観測ノイズは混入しないものと仮定する(産業用のサーボ系においても、エンコーダでパルスを計測して位置検出が行われるので、観測ノイズなしの仮定に無理はない)。また、以下の議論ではY軸のダイナミクスのモデル化が正確であるという仮定の下で議論を展開する。モデル化誤差が存在する場合には、モデル化誤差を少なくするように(8.5)式中の  $K_y$  の値を正しく調整する必要がある。この提案法のブロック線図を Fig. 8.2 に表す。

#### 8.3.2 主軸位置同期制御法の特性解析

主軸位置同期制御法のY軸の位置出力は

$$Y(s) = \frac{kK_x}{s(s + K_x)} U_x(s) + \frac{k}{s + K_x} D_x(s) \quad (8.6)$$

となり、(8.1a)式と(8.6)式を比較すると、X軸とY軸の位置出力の関係は

$$kX(s) - Y(s) = 0 \quad (8.7)$$

となり、(8.2)式の条件を満たしている。つまり、X軸に如何なる外乱が入っても、Y軸はX軸に位置同期する。ただし、(8.7)式は  $s$  領域に関する等式であり、時間的な応答が一致するためには初期値が同期しているという仮定が必要である。

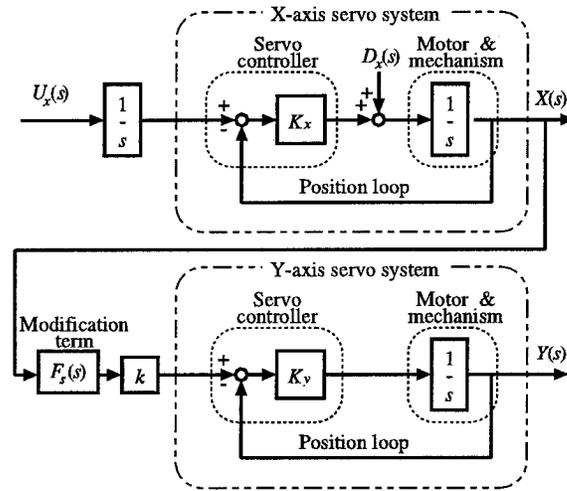


Fig. 8.2 Block diagram of the proposed synchronous position control system.

この提案法の実現性に関して検討してみると、修正要素  $F_s(s)$  は微分を含むため、微分不可能な入力に対して出力が発散する恐れがないことを確認する必要がある。そこで、Y 軸の位置入力信号を計算すると

$$F_s(s)X(s) = \frac{K_x(s + K_y)}{K_y s(s + K_x)} U_x(s) + \frac{s + K_y}{K_y(s + K_x)} D_x(s) \quad (8.8)$$

となり、X 軸の速度入力  $U_x(s)$ 、および、X 軸に入る外乱  $D_x(s)$  に対する Y 軸の位置入力  $F_s(s)X(s)$  への伝達関数は、共にプロパーな伝達関数であるため、Y 軸の位置入力信号が発散する恐れはない。よって修正要素  $F_s(s)$  を (8.6) 式として用いることによる実行上の問題はなく、主軸位置同期制御法の有効性が理論的に確認された。

## 8.4 シミュレーション結果と実験結果

提案した主軸位置同期制御法を用いて、X 軸と Y 軸の位置同期の効果を計算機シミュレーションと XY テーブルを用いた実験で確認する。シミュレーション及び実験の条件は、X 軸の位置ループゲイン  $K_x = 5[1/s]$ 、Y 軸の位置ループゲイン  $K_y = 15[1/s]$ 、比例定数  $k = 1[1]$ 、オイラー法のサンプリング間隔  $T = 0.02[s]$  とする。

### 8.4.1 主軸位置同期制御シミュレーション

位置同期を必要とする装置に起こる外乱を二種類想定し、それらの外乱に対しての (a) 主軸位置同期制御法、比較のため、(b) 位置同期を図らない場合、(c) 2 つのサーボ系間の追従制御方法<sup>[52]</sup>のシミュレーションを行う。この 2 つのサーボ系間の追従制御方法とは、Y 軸への速度入力を X 軸の位置出力フィードバックにより補正する手法である。速度入力波形として、ランプ波とステップ波両方の特性を持ち、しかも、ネジを切った後またもとに戻る Fig. 8.3 のような台形波波形とし、その関数は

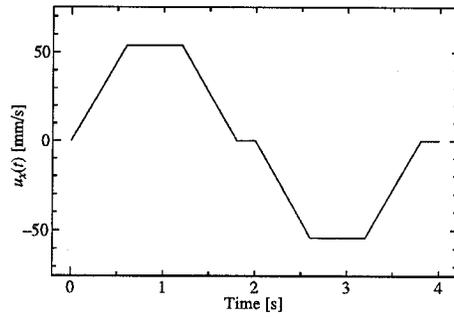


Fig. 8.3 Input signal of trapezoid wave to the servo system.

$$u_x(t) = \begin{cases} 90t & (0 \leq t \leq 0.6) \\ 54 & (0.6 < t \leq 1.2) \\ -90t + 162 & (1.2 < t \leq 1.8) \\ 0 & (1.8 < t \leq 2.0, 3.8 < t \leq 4.0) \\ -90t + 180 & (2.0 < t \leq 2.6) \\ -54 & (2.6 < t \leq 3.2) \\ 90t - 342 & (3.2 < t \leq 3.8) \end{cases} \quad (8.9)$$

とした.

(a) ステップ波外乱

ステップ波外乱は、タップ加工において、加工刃がネジを切り込む時点でステップ状の力がかかるために起こる外乱であり、また、走行台車などの両輪駆動では、主軸であるサーボ系に何らかの影響によって一定に負荷がかかり、速度低下が起こる場合の外乱を想定したものである。その波形は、Fig. 8.4で、その関数は

$$d_x(t) = \begin{cases} 0 & (0 \leq t \leq 0.5, 2.0 < t \leq 4.0) \\ -5 & (0.5 < t \leq 2.0) \end{cases} \quad (8.10)$$

とした.

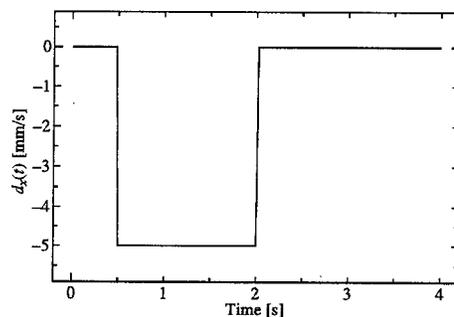


Fig. 8.4 Wave form of step disturbance to the main servo system.

ステップ波外乱における提案した主軸位置同期制御法と、比較のため、位置同期を図らない場合と2つのサーボ系間の追従制御方法のシミュレーション結果を、それぞれ、Fig. 8.5(a), (b), (c)に示す。左からXYテーブルの軌跡、X軸及びY軸の時間推移、X軸に対するY軸の軌道誤差( $e(t) = x(t) - y(t)$ )を表す。

位置同期を図らない場合(b)は、Y軸の応答がX軸の応答より速いために、位置同期が図れていない。これは、X軸、Y軸の位置ループゲインの大きさの違いにより、応答の速さが異なるためであり、軌道誤差は最大8[mm]生じていることがわかる。また、位置ループゲインの違いによる誤差が大きいため、0~2[s]間に加えたステップ波外乱による影響は、軌道誤差のグラフを見る限り、ほとんど見えていない。

提案した主軸位置同期制御法の結果(a)と2つのサーボ系間の追従制御方法による結果(c)を比較すると、XYテーブルの軌跡と時間推移のグラフを見る限り、二つの方法は位置同期している。また、二つの方法とも、0~2[s]間に加えたステップ波外乱による影響が軌道誤差に表れている。しかし、2つのサーボ系間の追従制御方法(c)の0.25[mm]と比較して、主軸位置同期制御法(a)は0.1[mm]とかなり微小であり、ステップ波外乱混入時とそれ以外のときの影響は、2つのサーボ系間の追従制御方法(c)の方が大きい。また、2~4[sec]間のステップ波外乱なしの状況下においても、2つのサーボ系間の追従制御方法(c)は0.25[mm]の軌道誤差があり、主軸位置同期制御法(a)の0.1[mm]よりも大きい。このことにより、主軸位置同期制御法の有効性が確かめられた。また、提案した主軸位置同期制御法においても外乱の影響が僅かに生じたのは、(8.6)式で表される逆ダイナミクス $F_s(s)$ 中の微分を、制御器であるコンピュータ内の差分演算で行なったためによるものである。

#### (b) 歯形周期外乱

歯形周期外乱とは、タップ加工時において、ネジを切り込むときに、加工刃が当たるたびに起こる、周期性のある外乱で、その波形はFig. 8.6で、その関数は

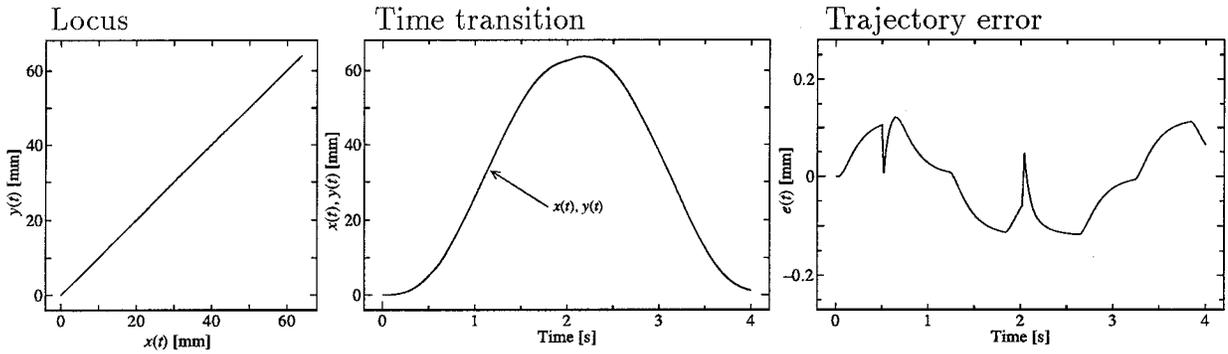
$$d_x(t) = \begin{cases} 0 & (0 \leq t \leq 0.18, 1.98 < t \leq 2.00) \\ -5 \left( 1 + \sin \left( \frac{25\pi t}{9} \right) \right) & (0.36 < t \leq 0.72, 1.08 < t \leq 1.44, 1.80 < t \leq 1.98) \\ -5 \left( 1 - \sin \left( \frac{25\pi t}{9} \right) \right) & (0.18 < t \leq 0.36, 0.72 < t \leq 1.08, 1.44 < t \leq 1.80) \end{cases} \quad (8.11)$$

とする。

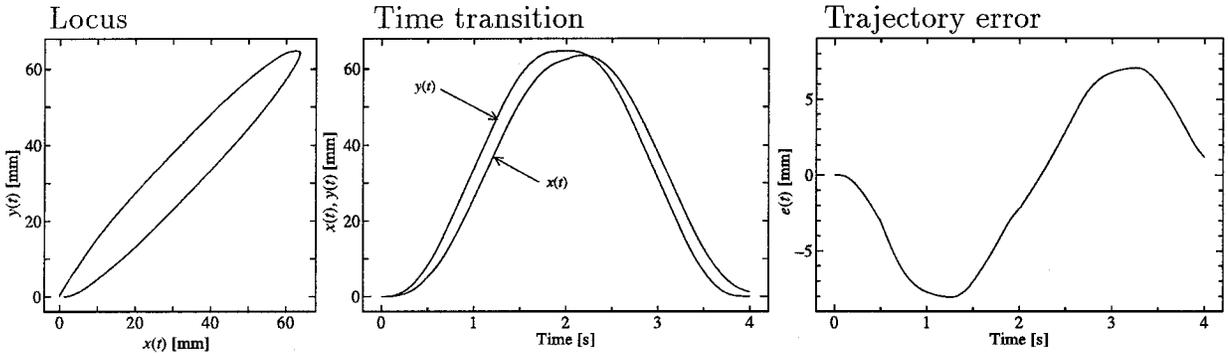
歯形周期外乱下における主軸位置同期制御法、比較のための、位置同期を図らない場合と2つのサーボ系間の追従制御方法のシミュレーション結果をFig. 8.7(a), (b), (c)にそれぞれ示す。ステップ波外乱下のシミュレーション結果との違いが表れた、X軸に対するY軸の軌道誤差( $e(t) = x(t) - y(t)$ )のみを表す。

位置同期を図らない場合の結果(b)は、ステップ波外乱時とほぼ同じ結果となった。提案した主軸位置同期制御法(a)を用いた結果と、2つのサーボ系間の追従制御方法による結果(c)を見ると、歯形周期外乱下と同じように、X軸とY軸は位置同期していることがわかる。しかし、軌道誤差のグラフを見ると、0~2[s]間に加えた歯形周期外乱による影響は、2つのサーボ系間の追従制御方法(c)の方が、提案した

(a) Proposed synchronous position control method



(b) Conventional unsynchronized method



(c) Tracking control method between two servo systems

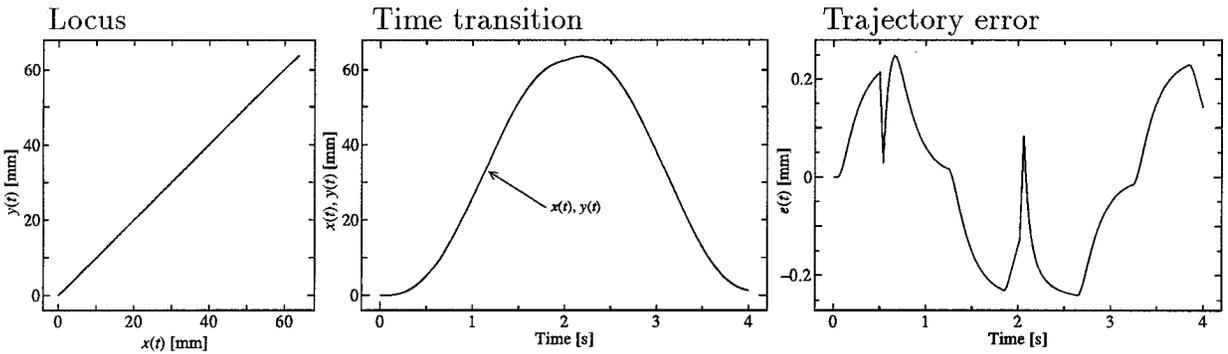


Fig. 8.5 Simulation results of the XY table equipment under the step disturbance.

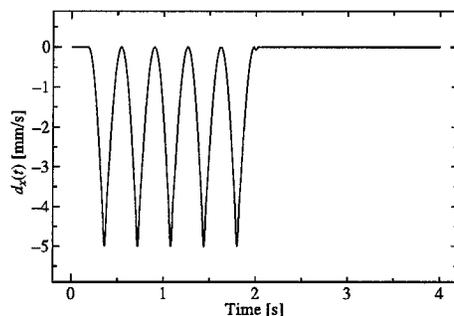
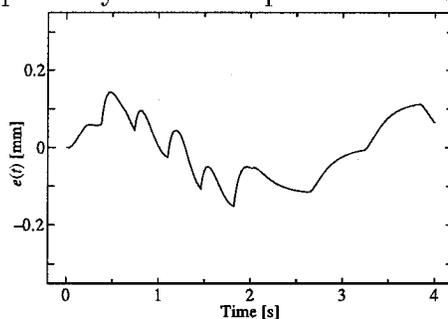
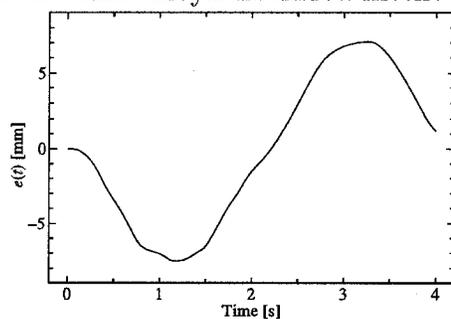


Fig. 8.6 Wave form of sawtooth disturbance to the main servo system.

(a) Proposed synchronous position control method



(b) Conventional unsynchronized method



(c) Tracking control method between two servo systems

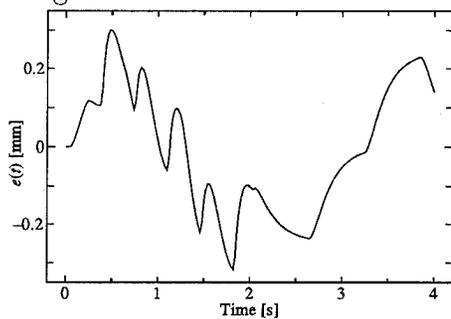


Fig. 8.7 Simulation results of the XY table equipment under the sawtooth disturbance.

主軸位置同期制御法 (a) の  $0.15[\text{mm}]$  と比べ、 $0.3[\text{mm}]$  と 2 倍の大きさの軌道誤差が表れている。また、 $2\sim 4[\text{s}]$  間の歯形周期外乱なしの状況下では、ステップ波外乱と同じ結果となっている。これより、提案した主軸位置同期制御法の有効性が確かめられた。

#### 8.4.2 主軸位置同期制御実験

前節では、計算機内で作られた外乱に対してのシミュレーションを行い、良好な結果を得た。次に、実際に XY テーブルを用いて実験を行った。XY テーブルとは、テーブルが X 軸方向と Y 軸方向とに独立に移動することができる装置である。

外乱  $D_x(s)$  の加え方を二通りの場合について行った。一つは、計算機内で作られる外乱、即ち、XY テーブルの制御器である計算機内で、仮想的に作成する外乱であり、外乱を計算機内で加えることにより、実際の装置に入る可能性のある様々な外乱を作成可能となる。もう一つは、実際の実験装置に物理的に加える外乱であり、より実際の装置に似た状況を作り出すことができる。

##### (a) 計算機内で加えた場合

前節と同じ入力波形を用い、条件も同じにした。ステップ波外乱下の実験結果は、提案した主軸位置同期制御法、位置同期を図らない場合と 2 つのサーボ系間の追従制御方法のシミュレーション結果を、それぞれ、Fig. 8.8(a), (b), (c) に示し、歯形周期外乱下の実験結果も同様に Fig. 8.9(a), (b), (c) に示す。図の見方は、それぞれ Fig. 8.5 と Fig. 8.7 の計算機シミュレーション結果と同じである。全体的にシミュレーション結果と実験結果は一致している。しかし、歯形周期外乱下での軌道誤差を見ると、シミュレーションでは、 $0\sim 2[\text{s}]$  間に加えた歯形周期外乱が表れているが、実験ではほとんど表れていない。これは、XY テーブルと制御器であるパーソナルコンピュータとの間を A/D, D/A 変換器でつないでいるため、その量子化誤差の影響とほとんど変わらないためであると考えられる。

##### (b) 実際の装置に加えた外乱

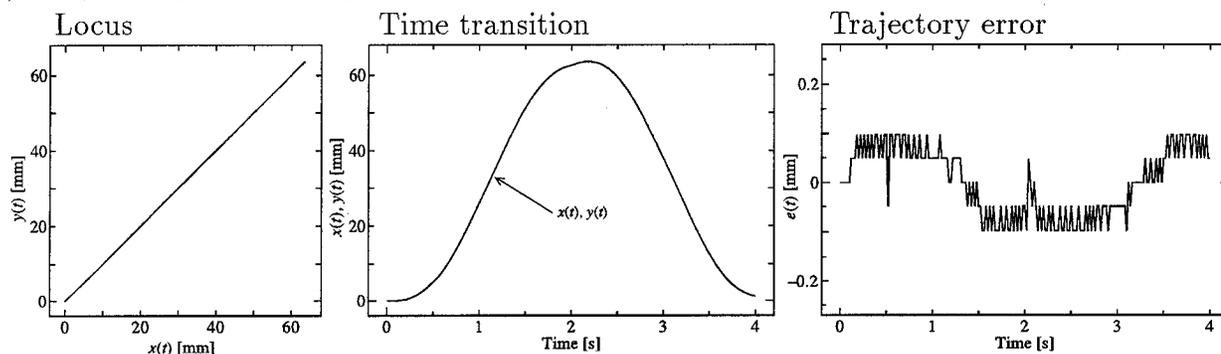
より実際に用いられている装置に近づけるために、実験装置に直接外乱を加える実験を行なった。XY テーブルは、Y 軸が X 軸の上に乗っているために、Y 軸全体が X 軸の進行方向に移動する。そこで、Y 軸を X 軸の進行方向とは反対方向に力を加えることにより、X 軸のみに外乱を加えることが可能である。そこで、XY テーブルが動きだした時点で、XY テーブルの Y 軸を手で力一杯 ( $70\sim 80[\text{kgf}]$  程度) 引っ張ることによって、ステップ波外乱を作る。XY テーブルにおける 2 つのサーボ系間の追従制御方法と主軸位置同期制御法の実験を行った。入力波形は

$$u_x(t) = 28.2 \quad (0 \leq t \leq 5) \quad (8.12)$$

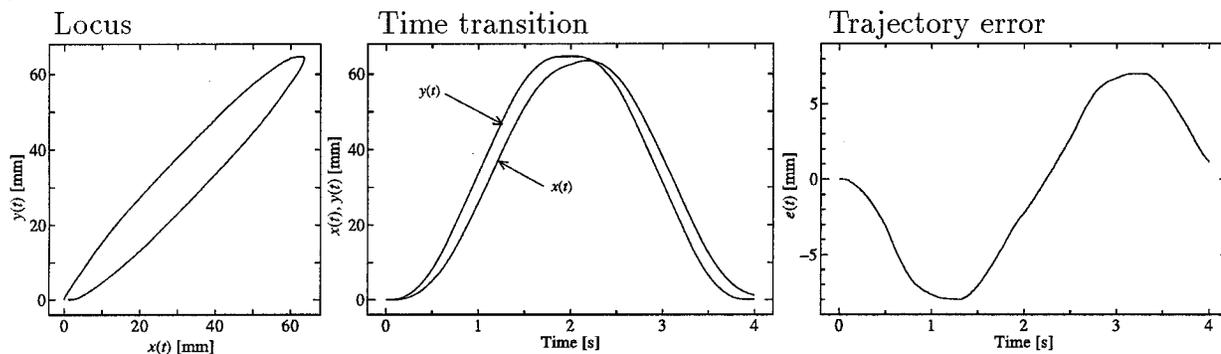
と与えた。

実際の装置に外乱を加えた場合の、提案した主軸位置同期制御法と 2 つのサーボ系間の追従制御方法による実験結果を、それぞれ Fig. 8.10(a), (c) に示す。左から XY テーブルの軌跡、X 軸に対する Y 軸の軌道誤差 ( $e(t) = x(t) - y(t)$ ) の結果を示す。XY テーブルの軌跡を見ると、二つの手法とも Y 軸の出力が X 軸の出力に位置

(a) Proposed synchronous position control method



(b) Conventional unsynchronized method



(c) Tracking control method between two servo systems

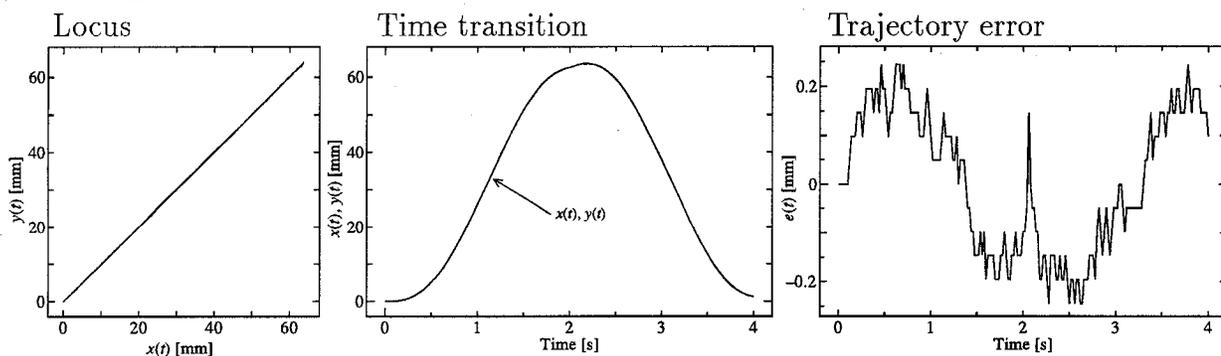
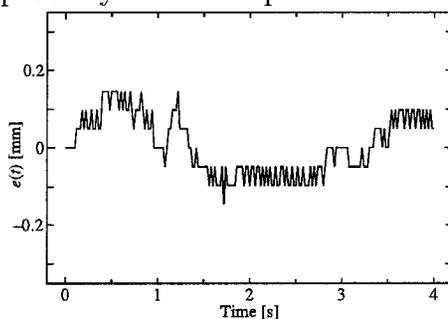
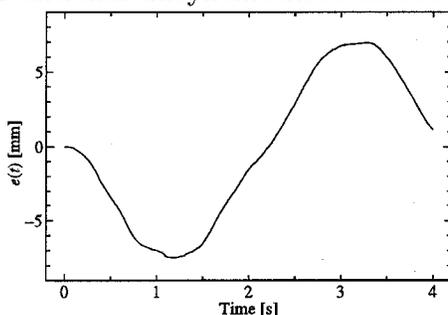


Fig. 8.8 Experimental results of the XY table equipment under the step disturbance by using the proposed synchronous position control method.

(a) Proposed synchronous position control method



(b) Conventional unsynchronized method



(c) Tracking control method between two servo systems

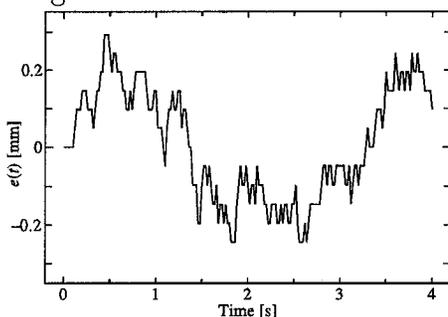
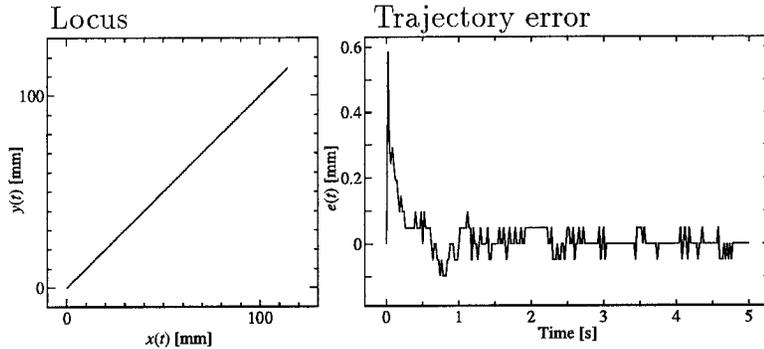


Fig. 8.9 Experimental results of the XY table equipment under the sawtooth disturbance.

同期している。しかし、軌道誤差のグラフを見ると、二つの手法とも実験開始直後に0.6[mm]程度の大きな誤差が現れている。これは、両手法とも、X軸の位置出力フィードバック信号の微分を差分で近似しているため、最初のステップについてはフィードバック信号が入ってこなかったためである。2つのサーボ系間の追従制御方法の結果(c)では、実験開始直後の誤差の減衰が遅く、定常に落ち付いた後の誤差の振幅も、最大0.15[mm]程度あるが、提案した主軸位置同期制御法(a)では、定常誤差は0.07[mm]程度で非常に小さい値である。これより、主軸位置同期制御法は、実際の装置に加えた外乱下での実験でも有効性が確認された。

## (a) Proposed synchronous position control method



## (c) Tracking control method between two servo systems

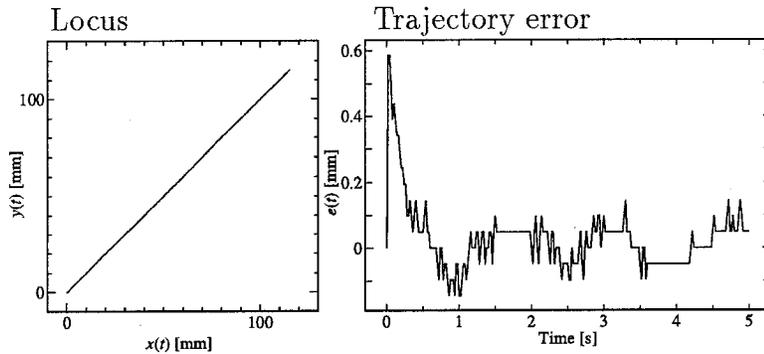


Fig. 8.10 Experimental results of the XY table equipment under the actual disturbance.

## 8.5 まとめ

本章では、動特性を一致させることの難しいサーボ系に対して、応答特性の速い方のサーボ系を遅い方のサーボ系の応答に同期させて制御する方法である主軸位置同期制御法を提案した。本主軸位置同期制御法は、Y軸の逆ダイナミクスを修正要素として、X軸の位置出力をその修正要素に通してY軸の位置入力信号として用いた手法である。この主軸位置同期制御法により、X軸の位置出力とY軸の位置出力を同期させることができる。シミュレーション及びXYテーブルによる実験結果を、位置同期を図らない場合と2つのサーボ系間の追従制御方法と比較検討することにより、本手法の有効性を確認した。この結果から、主軸に外乱が入っても従軸は主軸に追従し、提案した主軸位置同期制御法が有効であるということが確認された。本手法は、主軸に追従させる軸が一軸のみならず、多軸の場合であっても、それぞれの軸のサーボ系の逆ダイナミクスを用いることにより同期を図ることが可能となる。

## 第9章

### 結論

本論文では、今日、産業界の省力・省人化や品質均一化に欠かせないものとなっている産業用ロボットアームについて、現場で用いられている経験則の理論的裏付けと高速高精度制御を目指して、解析と制御法の開発を行なった。以下に本研究の各章における要約を述べる。

1章では、産業用ロボットアームとその歴史、現状の制御法と問題点について説明し、本研究の目的を明らかにした。

2章では、産業用ロボットアームのアクチュエータに用いられつつあるソフトウェアサーボ系の、トルク分解能について解析を行なった。現状では、ソフトウェアサーボ系のトルク分解能は経験的に決定されているが、ソフトウェアサーボ系のモデルを用いて、制御性能とトルク分解能のA/D, D/A変換のビット数との関係式を導出することによって、要求する制御性能から理論的にトルク分解能を決定する手法を提案した。

3章では、産業用多関節ロボットアームにおいて、経験的に、ロボットアームのダイナミクスを作業座標系において各座標軸で線形独立(作業線形モデル)とみて、輪郭制御の誤差解析や、新しい作業への適用可否判断などが行なわれているが、その理論的裏付けと、作業線形モデルを用いることができる領域である作業線形近似可能領域を明らかにした。作業線形近似可能領域においては、産業用多関節ロボットアーム制御系を作業座標系において各座標軸独立の制御系とみなすことができるために、その制御対象に対する制御器の構成においては、既存の線形制御理論を用いることにより、より高精度の制御を行なうことが可能となる。

4章では、産業用多関節ロボットアームを高速に動作させる場合に、目標軌跡と追従軌跡との間にずれ(軌跡誤差)が生じることが経験されているが、この軌跡誤差の原因を明らかにし、軌跡誤差の大きさと目標速度との関係を導出した。その結果、軌跡誤差は目標速度の2乗と4乗の線形結合で近似され、また、軌跡誤差は目標軌跡の位置にも依存することがわかった。導出した軌跡誤差と目標速度との関係式から、要求する精度を満たすような目標速度を適切に決定することが可能となる。

5章では、産業用ロボットアームの高速高精度制御を実現するために、極配置レギュレータ理論を用いて産業用ロボットアームの入力信号である教示信号を修正する教示信号修正法を提案した。この手法は、制御系の特性を改善させるために極配置レギュレータを用いる手法であるが、極配置レギュレータ理論はクローズドルー

Table 9.1 Comparison of the modified taught data method.

	極配置レギュレータ	速度修正	ガウシアンネット
目標軌道 修正 設計 モデル化誤差 適用の容易さ	任意 位置と速度 極 弱い 容易	折れ線近似 速度のみ 許容パラメータ 強い 非常に容易	2階微分可能 位置と速度 ネットの学習 強い やや難易

プ制御法であり、そのまま用いるためにはロボットアームの先端の位置を測定する必要があり、ロボットアームのハードウェアの変更を伴う。そこで、クローズドループからオープンループへの変換を行なうことにより、教示信号の修正のみで制御系の特性を改善することを実現した。

5章で提案した極配置レギュレータを用いた教示信号修正法は、ロボットアームのモデル化が適切に行なわれている場合には、高精度制御が実現できるが、モデル化誤差が含まれる場合には、オーバシュートが生じるなど、制御性能の劣化が見られた。そこで、6章では、教示信号における位置情報には修正を施さず、速度情報についてのみ修正を行なう教示速度修正法を提案した。産業界における輪郭制御では、目標軌跡が折れ線近似によって与えられることに着目し、教示速度修正法は、折れ線近似を行った各線分の始点と終点の速度を適切に減速することによって、高精度制御を実現させた。教示速度修正法は教示信号の位置情報に修正を加えないため、モデル化誤差が存在する場合においても、オーバシュートが生じることはない。

7章では、ロボットアームのモデル化誤差を少なくするために、教示信号の修正に、学習機能を持つガウシアンネットを用いた教示信号修正法を提案した。ガウシアンネットの構造と初期値は、ロボットアームのモデルから系統的に決定し、ガウシアンネットの学習に対する負荷を低減させて、比較的短時間において学習を行なうことが可能となり、高精度制御が実現できる。

5章から7章において提案した教示信号修正法の比較を Table 9.1 に示す。Table 9.1 から、産業用ロボットアームの作業目的に応じて、一番適切な手法を選択すればよい。5章から7章において提案した手法は、すべて、産業用ロボットアームの入力信号である教示信号を修正するのみであり、産業用ロボットアームの制御系の変更や、センサの付加などのハードウェアの変更を必要としないため、実機の産業用ロボットアームに容易に適用可能である。

8章では、産業用ロボットアームを用いる作業の一つである、ねじ切り作業などで用いられる位置同期制御問題に対する手法として、従軸逆ダイナミクスを用いた主軸位置同期制御法を提案した。提案した主軸同期制御法は、主軸の位置を観測して従軸の逆ダイナミクスを用いて修正して従軸へ入力する手法であり、容易に実現可能である。

本論文で考察した産業用ロボットアームは、多関節ロボットアームについては、重

力やコリオリ力などの非線形項，構成する各軸の干渉項などを無視した比較的簡単なモデルに基づいて解析を行なっている。また，関節座標系におけるサーボ系の構成においても，1次系モデルと2次系モデルを基本とし，アームからの反力，摩擦，トルク飽和などの特性は無視できるものとして取り扱っている。しかしながら，実際に稼働している産業用ロボットアームの関節座標系における動特性は，低速時には1次系モデル，高速時には2次系モデルで十分であり，また，多関節ロボットアームにおける非線形項や干渉項は無視できるように工夫されているため，本論文で行なっている研究で基本的な部分においては問題はなく，実機を用いた実験結果においても十分な制御性能が得られている。

本論文で行なっている研究は，すべて，産業界において実際に問題となっている点に対して行なっており，また，その手法に対する検証においても，実機の産業用ロボットアームや，産業界で実際に用いられているサーボ系によって構成されているXYテーブル(これは直交座標ロボットとみなすことができる)を用いている。このことから，本研究で得られた成果は，ただちに産業用ロボットアームに適用することができ，有用性が非常に高いものであると考えられる。

産業用ロボットアームの解析と制御における今後の課題は

- 本研究で採用した産業用ロボットアームのモデルの適用範囲と適用限界
- より厳密な産業用ロボットアームのモデルに基づく解析と制御法の開発
- 2章で述べたトルク分解能決定において，トルク分解能による制御性能への影響と，エンコーダ分解能や摩擦などの他の因子の制御性能への影響との比較
- 産業用ロボットアーム制御系のハードの変更を行なわないと仮定したが，ハードの変更までをも考慮した，より高精度な制御法の開発
- 5章で提案した極配置レギュレータに基づく教示信号修正法の，モデル化誤差に対するロバスト性
- 5章で提案した極配置レギュレータに基づく教示信号修正法は，計算機シミュレーションによって極の選定を行なう必要があるが，制約条件までをも取り込んだ，より直接的なパラメータに基づく設計法

などを解決していく必要がある。

## 謝辞

本研究は、佐賀大学理工学部電気工学科制御工学講座において行なわれたものである。

本研究の遂行ならびに本論文のまとめに際し、懇切な御指導と御鞭撻を頂きました大阪大学 木村英紀教授(現 東京大学教授)に心より感謝致します。

本論文をまとめるにあたり、多くの御助言を頂きました大阪大学 須田信英教授、田村担之教授、藤井隆雄教授ならびに宮崎文夫教授に心より感謝致します。

本研究を遂行するにあたり、終始懇切丁寧な御指導と多大なる御助言を頂きました佐賀大学 中村政俊教授に心よりお礼申し上げます。

本研究を始めるきっかけを作って頂き、研究を進める上において多大なる御助言を頂きました(株)安川電機 久良修郭部長(現 近畿大学教授)に心よりお礼申し上げます。

本研究をまとめるにあたり、多大なる御助言を頂きました鳥取大学 魚崎勝司教授に心よりお礼申し上げます。

本研究を進めるにあたり、貴重な御教示を頂きました佐賀大学 渡邊桂吾教授に感謝の意を表します。

本研究の全般にわたって多くの御助言を頂きました福岡工業大学 西田茂人助教授に感謝の意を表します。

また、この研究の実験にあたり、御協力を頂きました佐賀大学 永淵一成技官、事務処理に際して、御協力を頂きました佐賀大学 江頭美奈子さんに感謝致します。

さらに、本研究を遂行するにあたり、御協力を頂きました佐賀大学理工学部電気工学科制御工学講座の皆さんに感謝致します。

## 参考文献

- [1] 花房 秀郎: ロボット技術の展望, 計測と制御, 第 21 巻, 第 12 号, pp. 1089-1094 (1982).
- [2] 館 暉, 小森谷 清: 第 3 世代ロボット, 計測と制御, 第 21 巻, 第 12 号, pp. 1140-1146 (1982).
- [3] 長田 正: ロボット研究の今後の課題, 日本ロボット学会誌, 第 12 巻, 第 1 号, pp. 2-6 (1994).
- [4] 加藤 一郎 監修 日本産業用ロボット工業会編: 21 世紀のロボット, 工業調査会 (1993).
- [5] 吉川 恒夫: ロボットアームの制御方式, 計測と制御, 第 25 巻, 第 1 号, pp. 37-44 (1986).
- [6] 合田 周平: ロボット事始め, 計測と制御, 第 25 巻, 第 1 号, pp. 1-2 (1986).
- [7] R. P. Paul (吉川 恒夫訳): ロボットマニピュレータ, コロナ社 (1984).
- [8] 米本 完二, 雨宮 正和: 産業用ロボット長期需要予測 (製造業分野) の予測見直し報告, Robot, 第 99 号, pp. 7-10 (1994).
- [9] 日刊工業新聞: 日本、ロボ王国キープも世界全体シェアは低下, 1994 年 10 月 10 日, (1994).
- [10] 有本 卓: ロボットの力学と制御, 朝倉書店 (1990).
- [11] 沼倉 俊郎, 松本 義雄: 産業用ロボットの最近の動向, 計測と制御, 第 18 巻, 第 1 号, pp. 100-106 (1979).
- [12] 吉川 恒夫: ロボット制御基礎論, コロナ社 (1988).
- [13] 安川電機製作所編: メカトロニクスのためのサーボ技術入門, 日刊工業新聞社 (1986).
- [14] 菅野 和宏: ソフトウェアサーボ系第 1 回総論・処理速度の向上で実用期へ, 日経メカニカル, 第 4 巻, 第 4 号, pp. 96-102 (1988).
- [15] 特集 デジタルサーボを使いこなす, オートメーション, 第 36 巻, 第 11 号 (1991).
- [16] 作久田 博司, 吉谷 豊, 米沢 隆広: ソフトウェアサーボによるロボットアームの防振制御 (第 2 報), 日本機械学会論文集 (C 編), 第 54 巻, 第 497 号, pp. 217-220 (1988).
- [17] Canudas de Wit: Robust Control for Servo-mechanism Under Inexact Friction Compensation, *Automatica*, vol. 29, no. 3, pp. 757-761 (1993).
- [18] 武藤 高義: アクチュエータの駆動と制御, コロナ社, pp. 22-30 (1992).
- [19] 久保 謙二: デジタル速度オブザーバ, 機械設計, 第 34 巻, 第 17 号, pp. 158-163

- (1990).
- [20] 藤田 光悦, 海田 英俊: イナーシャ同定機能を付加した瞬時速度検出, 機械設計, 第 34 卷, 第 17 号, pp. 154-171 (1990).
  - [21] 堀 洋一, 今野 雄介: 低精度エンコーダを用いたサーボモータのロバスト制御, ロボティクス・メカトロニクス講演会 '92 講演論文集, vol. A, pp. 167-172 (1992).
  - [22] 久良 修郭: ニューメカ講座, サーボ技術, 位置ループと速度ループの関係, 日経メカニカル, 第 226 卷, 第 8 号, pp. 135-140 (1986).
  - [23] 若松 清司, 田村 浩一郎: ロボットの制御, 計測自動制御学会, pp. 31-78 (1985).
  - [24] R. Ortega and M. W. Spong: Adaptive Motion Control of Rigid Robots: a Tutorial, *Automatica*, vol. 25, no. 6, pp. 877-888 (1989).
  - [25] S. W. Wijesoma, and R. J. Richards: Robust Trajectory Following of Robots Using Computed Torque Structure with VSS, *International Journal of Control*, vol. 52, no. 4, pp. 935-962 (1990).
  - [26] 川崎 晴久: ロボットアームの動力学計算法, 計測と制御, 第 25 卷, 第 1 号, pp. 23-29 (1986).
  - [27] 高木 貞治: 解析概論改訂第三版, 岩波書店, pp. 62-63 (1973).
  - [28] K. Youcef-Toumi and A. T. Y. Kuo, High-Speed Trajectory Control of a Direct-Drive Manipulator, *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 102-108 (1993).
  - [29] D. M. Dawson, Z. Qu, F. L. Lewis, and J. F. Dorsey: Robust Control for the Tracking of Robot Motion, *International Journal of Control*, vol. 52, no. 3, pp. 581-595 (1990).
  - [30] S. Hara, Y. Yamamoto, T. Omata and M. Nakano: Repetitive Control System: A New Type Servo System for Periodic Exogenous Signals, *IEEE Transactions on Automatic Control*, vol. AC-33, no. 7, pp. 659-668 (1988).
  - [31] T. Fukuda, T. Shibata, K. Kosuge, F. Arai, M. Tokita and T. Mitsuoka: Neuromorphic Sensing and Control - Applications to Position, Force, and Impact Control for Robotic Manipulators -, *Proceedings of the 30th Conference on Decision and Control*, pp. 162-169 (1991).
  - [32] R. Ortega and M. W. Spong: Adaptive Motion Control of Rigid Robots: a Tutorial, *Automatica*, vol. 25, no. 6, pp. 877-888 (1989).
  - [33] J. J. E. Slotine and W. Li: Composite Adaptive Control of Robot Manipulators, *Automatica*, vol. 25, no. 4, pp. 509-519 (1989).
  - [34] Z. Qu and J. Dorsey: Robust Tracking Control of Robots by a Linear Feedback Law, *IEEE Transactions on Automatic Control*, vol. AC-36, no. 9, pp. 1081-1084 (1991).
  - [35] R. Shoureshi, M. E. Momot and M. D. Roesler: Robust Control for Manipulators with Uncertain Dynamics, *Automatica*, vol. 26, no. 2, pp. 353-359 (1990).
  - [36] C. Y. Kuo and S. P. T. Wang: Robust Position Control of Robotic Manipulator

- in Cartesian Coordinates, *IEEE Transactions on Robotics and Automation*, vol. RA-7, no. 5, pp. 653-659 (1991).
- [37] D. H. Chyung: Feedforward Digital Tracking Controller, *IEEE Transactions on Automatic Control*, vol. AC-36, no. 10, pp. 1099-1204 (1991).
- [38] R. L. Wells, J. K. Schueller and J. Tlustý: Feedforward and Feedback Control of a Flexible Robotic Arm, *IEEE Control Systems Magazine*, vol. 10, no. 1, pp. 9-15 (1990).
- [39] 片山 徹: フィードバック制御の基礎, 朝倉書店, pp. 62-64 (1987)
- [40] S. M. Shinnars: *Modern Control System Theory and Application*, Massachusetts: Addison-Wesley, pp. 286-289 (1972).
- [41] I. Nagamatsu, T. Tomiyama, M. Tanaka and N. Kyura: An Approach to Software Organization for Control of Industrial Robots, *Journal of Robotic Systems*, vol 2, no. 3, pp. 265-276 (1985).
- [42] 井上 恵, 岩井 真一, 中野 道雄: プレイバックサーボ系の高精度制御, 電気学会論文誌 C, 第 101 巻, 第 4 号, pp. 89-96 (1981).
- [43] 中村 祐司: 予見学習制御とそのサーボモータ位置制御への応用, 計測自動制御学会論文集, 第 27 巻, 第 12 号, pp. 1382-1388 (1991).
- [44] S. Lee and R. M. Kil: A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning, *Neural Networks*, vol. 4, pp. 207-224 (1991).
- [45] *Control Systems Magazine*, Special Issue on Neural Networks in Control Systems, vol. 10, no. 3 (1990).
- [46] *Control Systems Magazine*, Special Issue on Neural Networks in Control Systems, vol. 12, no. 2 (1992).
- [47] K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop: Neural Networks for Control Systems—a Survey, *Automatica*, vol. 28, no. 6, pp. 1083-1112 (1992).
- [48] R. M. Sanner and J. J. E. Slotine: Gaussian Networks for Direct Adaptive Control, *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837-863 (1992).
- [49] B. Widrow and M. A. Lehr: 30 Years of Adaptive Neural Network: Perceptron, Madaline, and Backpropagation, in C. Lau (Ed.), *Neural Networks*, New York, IEEE Press, Part 2, pp. 27-53 (1992).
- [50] 川村 英昭, 藤岡 良基, 佐々木 隆夫: タップ加工制御装置, 日本国特許庁公開特許公報, 昭 63-34018, 昭 63-34019 (1988)
- [51] 下坂 孝次, 美濃羽 貞治, 松田 和彦: タップ盤, 日本国特許庁公開特許公報, 昭 60-186316 (1985)
- [52] 久良 修郭, 平賀 義二: 2 つのサーボ系間の追従制御方式, 日本国特許庁公開特許公報, 昭 63-268011 (1988)

# 研究発表

## A. 学術論文 (掲載済みまたは掲載決定)

- [1] 後藤 聡, 中村 政俊, 久良 修郭: ロボットアームの高速高精度位置決めのための教示信号修正法, 計測自動制御学会論文集, 第 27 巻, 第 12 号, pp. 1396-1404 (1991)
- [2] 後藤 聡, 中村 政俊, 久良 修郭: 産業用多関節ロボットアーム制御系の線形モデルの近似誤差の評価とモデルの妥当性, システム制御情報学会論文誌, 第 7 巻, 第 3 号, pp. 103-112 (1994)
- [3] 後藤 聡, 中村 政俊, 岡 誠司, 久良 修郭: 複数サーボ系における従軸逆ダイナミクスを用いた主軸位置同期制御法, 計測自動制御学会論文集, 第 30 巻, 第 6 号, pp. 669-676 (1994)
- [4] S. Goto, M. Nakamura and N. Kyura: A Modified Speed Data Method for Position Control of Mechanical Servo Systems, *Transaction of the SICE*, vol. 30, no. 6, pp. 641-643 (1994)
- [5] 後藤 聡, 中村 政俊, 久良 修郭: ソフトウェアサーボ系の要求制御性能に基づいたトルク分解能決定法, 電気学会論文誌 C 電子・情報・システム部門誌, 第 114 巻, 第 7/8 号, pp. 783-788 (1994)
- [6] 後藤 聡, 中村 政俊, 久良 修郭: メカトロサーボ系の高精度輪郭制御のためのガウシアンネットを用いた教示信号修正法, 電気学会論文誌 C 電子・情報・システム部門誌, 第 115 巻, 第 1 号, pp. 111-116 (1995)
- [7] S. Goto, M. Nakamura and K. Uosaki: On-line Spectral Estimation of Non-stationary Time Series Based on AR Model Parameter Estimation and Order Selection with A Forgetting Factor, *IEEE Transactions on Signal Processing*, (1994 accepted)

## B. 国際講演論文

- [1] S. Goto and K. Uosaki: An On-Line Identification Procedure for Linear Stochastic Systems, *Proceedings of the 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, pp. 941-946 (1991)
- [2] S. Goto M. Nakamura and N. Kyura: Contour Control of Industrial Robot Arms of Teaching Playback Type by Using a Neural Network, *Proceedings of the IFToMM-jc International Symposium on Theory of Mechines and Mechanisms*, pp. 87-91 (1992)

- [3] S. Goto M. Nakamura and N. Kyura: Relationship between Motion Speed and Working Accuracy of Industrial Articulated Robot Arms, *Proceedings of the 1993 Korean Automatic Control Conference*, pp. 230-233 (1993)
- [4] S. Goto M. Nakamura and N. Kyura: Contour Control Performance of Industrial Robot Arms Operated in Working Coordinates at High Speed Conditions, *Proceedings of the Asian Control Conference*, vol. 3, pp. 635-638 (1994)
- [5] S. Goto M. Nakamura and N. Kyura: Trajectory Generation for Contour Control of Mechatronics Servo Systems Subjected to Torque Constraints, *Proceedings of the 1994 Korean Automatic Control Conference*, pp. 66-70 (1994)
- [6] N. Nanayakkara, M. Nakamura, S. Goto and T. Taniguchi: Modeling and Prediction of Rapid Pollution of Insulators in Substations Based on Weather Information, *Proceedings of the 1994 Korean Automatic Control Conference*, pp. 202-206 (1994)
- [7] S. Goto, M. Nakamura, N. Nanayakkara, T. Matsunaga and T. Taniguchi: Automatic Decision Making of Washing the Rapid Polluted Insulators in Substations, *Proceedings of the 26th ISCIE International Symposium on Stochastic Systems Theory and Its Applications*, pp. 167-171 (1995)
- [8] S. Goto M. Nakamura and N. Kyura: Trajectory Generation of Industrial Mechatronic Systems to Achieve Accurate Contour Control Performance under Torque Constraints, *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, (1995 accepted)
- [9] N. Nanayakkara M. Nakamura S. Goto and T. Taniguchi: Modelling and Simulation of Pollution Deposits in Substation Insulators near Coastal Areas, *IASTD International Conference on Modelling and Simulation*, (1995 accepted)
- [10] S. Goto M. Nakamura N. Nanayakkara and T. Taniguchi: Reliable Automatic Decision Making for Washing the Polluted Insulators in Coastal Substations, *1995 Conference on Electrical Insulation and Dielectric Phenomena*, (1995 submitted)
- [11] M. Nakamura S. Goto and N. Kyura: Accurate Contour Control of Mechatronic Servo Systems by Use of Modified Input Signal: Mathematical Interpretation of Human Skill, *IFAC Workshop Human-Oriented Design of Advanced Robotics Systems*, (1995 submitted)

### C. 国内講演論文

- [1] S. Goto and K. Uosaki: Robust Order Selection of Autoregressive Models and Its Application - Empirical Study -, *Proceedings of the 27th SICE Annual Conference*, pp. 955-958 (1988)
- [2] S. Goto and K. Uosaki: On-Line Identification Scheme for Linear Stochastic Systems, *Preprints of the 21st ISCIE Symposium on Stochastic Systems Theory and Its Applications*, pp. 103-106 (1989)

- [3] 後藤 聡, 魚崎 勝司: 線形確率システムのオンライン同定法とその応用, *Preprints of the Third Signals Systems Control Symposium*, pp. 61-64 (1990)
- [4] 後藤 聡, 中村 政俊, 久良 修郭: ロボットアームの1軸高速高精度位置決めのための教示信号修正法, 平成2年度電気関係学会九州支部連合会大会論文集, p. 532 (1990)
- [5] 後藤 聡, 中村 政俊, 西田 茂人, 柴崎 浩: 線形確率システムモデルを用いた指標追跡運動機能のデータ解析, *Preprints of the 22nd ISCIE Symposium on Stochastic Systems Theory and Its Applications*, pp. 111-114 (1990)
- [6] 後藤 聡, 中村 政俊, 久良 修郭: ロボットアームの1軸位置決め教示信号修正法の特長解析, 平成2年度計測自動制御学会九州支部学術講演会論文集, pp. 243-246 (1990)
- [7] 後藤 聡, 中村 政俊, 久良 修郭: 教示信号修正法に基づく実機ロボットのティーチングと位置決め実験, 平成3年度電気関係学会九州支部連合会大会論文集, p. 533 (1991)
- [8] S. Goto, M. Nakamura and N. Kyura: Accurate High Speed Positioning of Actual Robot Arm using Method for Modifying Taught Data, *Preprints of the 23rd ISCIE Symposium on Stochastic Systems Theory and Its Applications*, pp. 121-124 (1991)
- [9] 後藤 聡, 中村 政俊, 久良 修郭: 2次系表現に基づく教示信号修正法を用いた実機ロボットアームの高精度輪郭制御, 平成3年度計測自動制御学会九州支部学術講演会論文集, pp. 295-298 (1991)
- [10] S. Goto M. Nakamura and N. Kyura: Method for Modifying Taught Data Based on Speed Information for Industrial Robot Arm of Teaching Playback Type, *Proceedings of the 31st SICE Annual Conference*, pp. 1111-1114 (1992)
- [11] 後藤 聡, 中村 政俊, 魚崎 勝司: ARモデルを用いた非定常時系列のオンラインスペクトル推定法, 平成4年度電気関係学会九州支部連合会大会論文集, p. 299 (1992)
- [12] S. Goto, M. Nakamura and K. Uosaki: On-line Spectral Estimation of Non-stationary Time Series Based on Parameter Estimation and Order Selection of AR Models, *Preprints of the 24th ISCIE Symposium on Stochastic Systems Theory and Its Applications*, pp. 111-116 (1992)
- [13] 後藤 聡, 中村 政俊, 久良 修郭: 産業用多関節ロボットアーム制御系の線形近似の妥当性, 平成4年度計測自動制御学会九州支部学術講演会論文集, pp. 187-190 (1993)
- [14] 後藤 聡, 中村 政俊, 久良 修郭: ソフトウェアサーボ系における高精度制御のためのトルク分解能決定法, 平成5年度電気関係学会九州支部連合会大会論文集, p. 544 (1993)
- [15] 中村 政俊, 時田 邦彦, 後藤 聡, 久良 修郭: サーボ系におけるスティックスリップ現象の計算機シミュレーションの方法, 平成5年度電気関係学会九州支部連合会大会論文集, p. 545 (1993)

- [16] 江頭 成人, 中村 政俊, 後藤 聡, 久良 修郭: 産業用ロボットにおける目標軌道分割時間間隔と速度むらとの関係, 平成5年度電気関係学会九州支部連合会大会論文集, p. 546 (1993)
- [17] S. Goto, M. Nakamura and N. Kyura: Deterioration of Contour Control Performance of Industrial Articulated Robot Arms Caused by Nonlinear Coordinate Transformation and Its Compensation, *Proceedings of the 32nd SICE Annual Conference*, pp. 1235-1238 (1993)
- [18] 後藤 聡, 中村 政俊, 久良 修郭: ソフトウェアサーボ系のトルク分解能とランプ応答との関係, 第12回計測自動制御学会九州支部学術講演会予稿集, pp. 335-338 (1993)
- [19] 中村 政俊, 前田 貴信, 後藤 聡, 久良 修郭: サーボ系における機械の作業特性と等価な速度外乱の決定法, 第12回計測自動制御学会九州支部学術講演会予稿集, pp. 301-304 (1993)
- [20] 中村 政俊, 時田 邦彦, 後藤 聡, 久良 修郭: サーボ系の摩擦に伴うスティックスリップの特性解析, 第12回計測自動制御学会九州支部学術講演会予稿集, pp. 311-314 (1993)
- [21] 江頭 成人, 中村 政俊, 後藤 聡, 久良 修郭: サーボ系における目標軌道分割時間間隔と制御性能の関係の実験的検証, 第12回計測自動制御学会九州支部学術講演会予稿集, pp. 339-342 (1993)
- [22] 後藤 聡, 中村 政俊, 原口 憲一, 久良 修郭: メカトロサーボ系の2次系モデルに基づくガウシアンネットを用いた輪郭制御, 平成6年電気学会 電子・情報・システム部門大会講演論文集, pp. 9-12 (1994)
- [23] 中村 政俊, 前田 貴信, 後藤 聡, 久良 修郭: メカトロサーボ系の複数軸の相互誤差フィードバック信号を用いた位置同期制御, 平成6年度電気関係学会九州支部連合会大会論文集, p. 485 (1994)
- [24] S. Goto, M. Nakamura, N. Nanayakkara, T. Matsunaga and T. Taniguchi: Automatic Decision Making of Washing the Rapid Polluted Insulators in Substations, *Preprints of the 26th ISCIE Symposium on Stochastic Systems Theory and Its Applications*, pp. 77-78 (1994)
- [25] S. Goto M. Nakamura and N. Kyura: Trajectory Generation and Inverse Dynamics Compensation for Contour Control of Mechatronics Servo Systems Subjected to Torque Constraints, 第12回日本ロボット学会学術講演会予稿集, pp. 959-960 (1994)
- [26] 中村 政俊, 前田 貴信, 後藤 聡, 久良 修郭: メカトロサーボ系の複数軸相互間の位置同期制御法とその特性解析, 第12回日本ロボット学会学術講演会予稿集, pp. 961-962 (1994)
- [27] 中村 政俊, 郷 俊忠, 後藤 聡, 久良 修郭: メカトロサーボ系の厳密モデルの構成と実験の検証, 第12回日本ロボット学会学術講演会予稿集, pp. 955-956 (1994)
- [28] 後藤 聡, 中村 政俊, 久良 修郭: トルク飽和を考慮したメカトロサーボ系の輪郭制御, 第13回計測自動制御学会九州支部学術講演会予稿集, pp. 239-242 (1994)

- [29] 中村 政俊, 鄒 俊忠, 後藤 聡, 久良 修郭: サーボ系の詳細モデルの構成と制御パラメータの決定, 第 13 回計測自動制御学会九州支部学術講演会予稿集, pp. 295-298 (1994)
- [30] 後藤 聡: 産業用ロボットアームの制御法, 第 4 回科学技術研究交流シンポジウム～環境にやさしいテクノロジー～, pp. 24-30 (1995)
- [31] S. Goto M. Nakamura N. Nanayakkara and T. Taniguchi: Automatization of Human Decision Making for Washing Time of Polluted Insulators in Substations, *Proceedings of the 34th SICE Annual Conference*, (1995 accepted)

#### D. 学内論文

- [1] S. Goto and K. Uosaki: Robust Order Selection of Autoregressive Models and Its Application, *Technology Reports of the Osaka University*, vol. 40, no. 2004, pp. 151-158 (1990)
- [2] 中村 政俊, 後藤 聡, ニシャンタ ナナヤッカーラ, 谷口 孝, 薮 千鶴男, 甲斐田 成宏, 井筒 海志: 変電所碍子急速汚損予測手法の開発 (I), 佐賀大学と民間機関等との共同研究成果報告書 *Joint Research Report*, no. 5, 35 (1994)