



Title	機械部品の計算機援用設計・加工システムに関する研究
Author(s)	山縣, 敬一
Citation	大阪大学, 1982, 博士論文
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/2624">https://hdl.handle.net/11094/2624</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

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

The University of Osaka

機械部品の計算機援用  
設計・加工システムに関する研究

昭和56年11月

山 縣 敬 一

# 目 次

序 論 .....	1
第1編 計算機援用切削加工のソフトウェア	
第1章 NCテープ作成のための簡易図形処理プロセッサ .....	3
1.1 緒 言 .....	3
1.2 APT系言語 .....	3
1.3 蓄積管端末の活用 .....	6
1.4 工具径路の定義 .....	8
1.4.1 部品形状の決定 .....	8
1.4.2 工具径路の決定 .....	9
1.5 プロセッサ開発時の誤り訂正問題 .....	13
1.5.1 誤りの事象の因果関係 .....	13
1.5.2 対話処理による誤りの追跡 .....	15
1.6 結 言 .....	16
第2章 板カム加工のNCテープ作成 .....	18
2.1 緒 言 .....	18
2.2 板カムの輪郭計算 .....	18
2.2.1 円端直動従節板カム .....	18
2.2.2 円端揺動従節板カム .....	22
2.3 設計パラメータの決定問題 .....	24
2.3.1 従節変位曲線 .....	24
2.3.2 圧力角ならびに曲率半径に関する条件 .....	27
2.4 NC指令の生成 .....	30
2.4.1 板カム加工のための数値制御の方式 .....	30
2.4.2 円弧補間の利用 .....	31
2.5 結 言 .....	32
第3章 実時間処理プログラムの構成 .....	34
3.1 緒 言 .....	34
3.2 局所の変数と非局所の変数 .....	34
3.3 擬似並行処理 .....	37
3.3.1 論理的モデル .....	37
3.3.2 t-unitによるプログラム間の動的結合 .....	40
3.3.3 同期操作 .....	42

3.4	ディスパッチャの構成	43
3.4.1	論理的モデルのスケジューリング	43
3.4.2	CPUの割り当て問題	44
3.4.3	デッドロック対策	45
3.5	結 言	46
第4章	オンラインDNCシステム	48
4.1	結 言	48
4.2	伝送制御手順	48
4.2.1	受信モード	49
4.2.2	送信モード	51
4.2.3	送受信の方向決定	51
4.3	周辺装置の制御	53
4.3.1	コンソール・タイプライタ	53
4.3.2	高速紙テープ・リーダー	54
4.3.3	磁気ディスク	56
4.4	簡易図形処理装置	57
4.5	アナログ入力インターフェース	59
4.6	オンラインDNCシステムの統合	60
4.7	結 言	63
第2編	板カム輪郭フライス切削における工作物熱変形の適応制御	
第1章	輪郭フライス切削における工作物への流入熱量の推定	65
1.1	結 言	65
1.2	理論解析	66
1.2.1	切屑生成による切削力	66
1.2.2	2次元切削モデルにおける熱の流れ	68
1.3	工具と被削材の決定による式の具体化	70
1.4	差分法によるシミュレータ	71
1.5	実験と考察	73
1.6	結 言	75
第2章	有限要素法による工作物の温度分布推定	77
2.1	結 言	77
2.2	非定常熱伝導問題	77
2.3	形状関数	78
2.4	有限要素法による温度分布の解析	81

2.5	切削モデルへの適用	83
2.5.1	移動熱源の取り扱い	83
2.5.2	工作物支持部の取り扱い	85
2.6	結 言	86
第3章 工作物熱変形推定のシミュレータ		87
3.1	結 言	87
3.2	板カムの自動要素分割	87
3.3	有限要素法による熱応力・熱変形解析	89
3.3.1	基礎方程式	89
3.3.2	有限要素法の導入	90
3.3.3	ポテンシャル・エネルギー最小の原理	92
3.4	熱変形推定シミュレータの構成	94
3.5	結 言	94
第4章 工具径路の実時間制御		96
4.1	結 言	96
4.2	適応制御実験システム	96
4.2.1	NEAC-M4 入出力システム	97
4.2.2	アナログ・データの入力	99
4.2.3	工作物の温度測定	103
4.2.4	工作物形状測定のためのギャップセンサ	106
4.2.5	リニアエンコーダ・インターフェース	108
4.3	位置決め精度に関する考察	111
4.3.1	DSサーボシステム	111
4.3.2	位置決め精度の測定	113
4.3.3	工作機械熱変形の影響	116
4.4	制 御 方 式	119
4.4.1	制御用データの構成	119
4.4.2	温度測定による熱変形推定	120
4.4.3	加工時の工具径路決定	121
4.5	実験と結果の考察	122
4.6	結 言	129
総 括		132
謝 辞		135

## 序 論

数値制御工作機械は1952年に米国で開発されて以来、電子計算機の発達と共に急速な進歩を遂げた。しかしその利用技術は周囲の使用環境の変化と共に依然として発展途上にある。当初は複雑な形状部品の加工のために開発されたものであったが、現在ではそれに止まらず、人手の省力化と密接に結びついて広く普及している。とくに1970年以降、IMS(Integrated Manufacturing System) さらには無人化工場への指向と切り離せないものとなって来ている。また一方、いろいろな規模の計算機の発達との関係も見逃すことができない。大型計算機を利用したNCテープ作成システムやDirect Numerical Control(DNC)、Computer Numerical Control(CNC)などは計算機の普及なくしては考えられなかったものである。

以上のような観点から数値制御システムを見直してみると、新たにいくつかの問題が生起される。まず第一に、NCテープ作成の省力化の問題がある。APT、EXAPT、曲面加工システムなどが開発される一方において、機能設計、工程設計との関連も含めて考えると、NCテープ作成の自動化にはまだ多くの問題が残されている。第二には、低価格で入手が容易となった小型計算機の効果的な利用法である。NC指令の自動分配を行うDNCシステムは省力化に大いに役立ち、また単なる指令の分配だけでなく、そのプロセッサの処理能力が現場で活かせる効果は大きい。またCNCシステムは今後の発展が期待できるものであるが、より柔軟な機能を持った低価格の数値制御装置の実現が期待できる。第三には小型計算機の併用もしくはCNCシステムによる適応制御の問題がある。とくに旋削においては生産性向上のための適応制御がすでに普及しつつあるが、加工精度向上のための適応制御は今後の発展題目であろう。

以上のような背景のもとに、本研究ではTSSオンラインDNCシステムをまず試作した。すなわち、大型計算機を用いたTSSの端局としてミニコンピュータを設置し、これと数値制御装置を直結してDNCシステムを構成した。APTプロセッサなどのように処理時間のかかるものはバッチ処理で処理するが、そのパート・プログラムやNCテープ・イメージの情報をファイルに保存し、エディタによって加工現場から編集できるようにした。ミニコンピュータは、端局を構成すると同時に数値制御装置と密着した形で使用され、NC指令の転送を制御したり、適応制御のコントローラとしての役割りも持っている。このように現場から必要に応じてミニコンピュータや大型計算機の処理能力を随時利用できるのがこの実験的システムの特徴であり、先に述べた三つの問題点すなわち(1)NCテープの作成、(2)小型計算機の効果的な利用法、(3)適応制御の諸問題を吟味できるようになっている。

本論文は第1編と第2編からなる。第1編ではNCテープ作成のプログラムとミニコンピュータの実時間処理のプログラムの構成法を論じ、第2編では板カム輪郭フライス切削における工作物熱変形の適応制御について論述している。NCテープ作成のプロセッサの代表的なものにAPTプロセッサがある。これは汎用性の面で極めてすぐれているが、一定の文法に従って図形定義を行い、工具動作を逐一具体的に文の形式で記述しなければならない煩わしさがある。一方、円と直線からなる部品が多いことに着目して、このような部品だけを扱うとすると、蓄積管型のグラフィック・ディスプレイ装置を用いてもっと簡便な方法でNCテープを作成することができる。この方法が第1編第1章に述べられている。また、板カムのような場合は設計計算と並行してNCテープが得られる特殊な場合であり、第1編第2章で取り扱われている。

第1編の後半、第3章と第4章ではミニコンピュータを効率よく利用するための実時間処理プログラム

の構成法について新しい試みを述べている。すなわち従来から使われて来たタスク管理の技法を用いず、各ファンクション・プログラムの独立性を重視し、プログラムの一部の差し換えが容易に実現できるようになっている。また、イベント・コントロール・ブロックとは別の概念を導入して柔軟な同期操作が記述できるように工夫されている。

数値制御工作機械で加工する場合、加工精度劣化の要因として工作機械自体の静的ならびに動的精度、工具、工作物、機械系の熱変形、工具摩耗などがあげられる。機械系の熱変形については多くの研究が行われているが、工作物熱変形に対する補正問題はフライス切削ではあまり取り上げられていない。本研究では板カム輪郭フライス切削をモデルとして、工作物の熱変形に対して工具径路を実時間で修正する実験を試みている。これが第2編の主題である。切削加工時に熱変形を測定するのは困難であるので、予め有限要素法で熱変形と温度上昇を予測し、切削加工時には工作物の温度上昇を監視して工具径路を修正するという予測修正法を採用している。第1章では加工条件に依存する流入熱量の推定を試みている。第2章と第3章は有限要素法による工作物の熱的挙動のシミュレータについて述べている。これによって切削加工時の熱変形と温度上昇を予測するのである。第4章では適応制御の実験について述べ、その有効性を明らかにしている。

# 第 1 編

## 計算機援用切削加工のソフトウェア

# 第1章 NCテープ作成のための簡易図形処理プロセッサ

## 1.1 緒言

NC工作機械の普及に伴い、NCテープ作成の省力化が重要な問題となり、NC用プログラミング言語が多数開発されて来た。<sup>1)</sup> それらは使用目的を限定して使い易さに重点をおいたものと、汎用性を目的としたものとに大別される。後者の代表的なものにAPT言語<sup>2)</sup>があり、極めて広汎な記述能力を有する。ただしこの言語を用いる場合、比較的単純な形状の部品についても煩わしい記述が要求されるので、多くのAPTサブセットが作り出される原因となっている。

本章ではこのような汎用性を目的とするものとは別に、対象を円と直線からなる2次元形状の部品に限り、蓄積管によるグラフィック・ディスプレイ端末を用いて、NCテープ作成の手順を極力簡略化したシステムについて述べる。これは計算機処理一般について云えることであるが、取り扱える対象の範囲を限定すると、比較的簡単な記述で対象を一義的に定めることができる。この場合に重要なことは限定された対象の範囲が実用上十分に意味があるかどうかという点である。この意味でここで述べるシステムでは、形状についての制約は強いが、ポケット・カッティングを可能にし、荒加工の工具径路も自動決定を行うなど、実用性にも十分配慮が払われている。

また、大型計算機のタイムシェアリング・システムに接続されたグラフィック端末を利用することにより、手続き記述言語をほとんど用いることなく、キーボードからの数値入力とクロスヘア・カーソルによる座標入力での処理を済ますことができるように工夫されている。リフレッシュメモリを用いる場合と異なり、図形処理は簡便な方法で行われる。

## 1.2 APT系言語

比較対照のために、本節ではAPT系言語の特徴ならびにAPT系言語とグラフィック・ディスプレイ装置の併用システムについて簡潔に述べる。APTの処理対象には位置決め、輪郭切削、曲面処理まで含まれるが、ここでは輪郭切削に重点をおいて述べる。APT系言語では予め図形要素を記号名をつけて定義しておき、それらで規制される面を規準として工具動作を記述して行く。図1.1に示すように、一つの工具動作の記述にパート・サーフェス、ドライブ・サーフェス、チェック・サーフェスの三つの面が必要になる。

これらの制御面はパート・サーフェスは通常平面であるが、ドライブ・サーフェス、チェック・サーフェスは平面、円筒面、一般2次元曲面などが許される。通常の工具動作では、工具はパート・サーフェスとドライブ・サーフェスに沿ってチェック・サーフェスまで動き、次は今までのチェック・サーフェスがドライブ・サーフェスとなって新しいチェック・サーフェスまで動く。APTプログラムでは予め定義された制御面の記号を用いた工具運動文により逐一工具動作を記述して行くことが必要である。簡単な例として、図1.2の形状についてのパートプログラムを図1.3に示す。

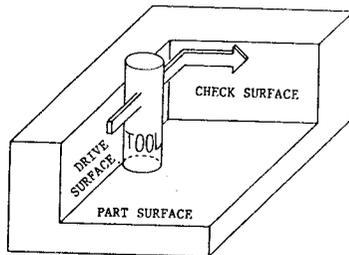


図1.1 工具動作の制御面

図1.3において(a)が図形定義であり、(b)が工具動作の記述を表している。点の定義はすべて座標値が分っているので $x$ 、 $y$ 、 $z$

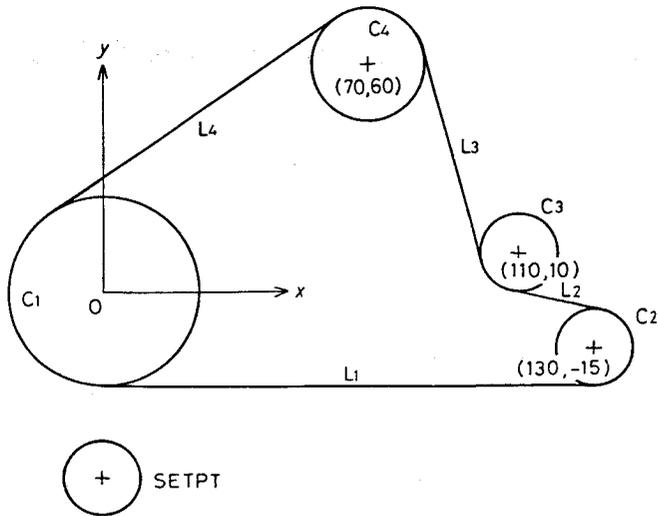


図1.2 部品形状例

```

PARTNO PART1
        MACHIN/NEDAC4200
        CLPRNT
SETPT = POINT/0,-50,50
P1    = POINT/0,0,0
P2    = POINT/130,-15,0
P3    = POINT/110,10,0
P4    = POINT/70,60,0
C1    = CIRCLE/CENTER,P1,RADIUS,25
C2    = CIRCLE/CENTER,P2,RADIUS,10
C3    = CIRCLE/CENTER,P3,RADIUS,10
C4    = CIRCLE/CENTER,P4,RADIUS,15
L1    = LINE/0,-25,0,130,-25,0
L2    = LINE/RIGHT,TANTO,C2,LEFT,TANTO,C3
L3    = LINE/LEFT,TANTO,C3,RIGHT,TANTO,C4
L4    = LINE/RIGHT,TANTO,C4,RIGHT,TANTO,C1

```

(a) 図形定義

```

CUTTER/20
OUTTOL/0.05
FROM/SETPT
RAPID
GODLTA/0,0,-50
PSIS/(PLANE/0,0,1,0)
FEDRAT/50,MMPM,MANUAL,5
INDIRV/0,10,0
GO/TO,L1
TLRGT,GORGT/L1,TANTO,C2
GOFWD/C2,PAST,L2
GOFWD/L2,TANTO,C3
GOFWD/C3,TO,L3
GOFWD/L3,TANTO,C4
GOFWD/C4,PAST,L4
GOFWD/L4,TANTO,C1
GOFWD/C1,PAST,L1
GODLTA/0,0,50
RAPID
GOTO/SETPT
STOP
END
FINI

```

(b) 工具動作

図1.3 APTプログラム例

座標値を直接指定している。円の場合もこの例では半径と中心が分っているので、簡単に記述できる。直線については、例えばL3は円C3に左から接し、円C4に右から接する直線として表している。点や直線についてもすでに定義済みの要素から、新しいものを定義して構わない。これらの定義文はAPT言語

の文法に従って記述される。すべての図形要素には、後で引用し易いようにすべて記号名（C1、L1など）がつけられている。また、工具動作で引用されるときは直線は $xy$ 平面に垂直な平面を表し、円は $xy$ 平面に垂直な円筒面を表す。

工具直径と許容公差を指定したあと、FROM文で工具の初期位置を定義している。連続切削の最初の起動文はGO文である。GO文の次から連続切削に入り、工具を図形要素の右に来るようにして、右に進みL1に沿ってC2に接する所まで進むように記述されている。その後はドライブ・サーフェスが変わるごとに、逐次GOFWD文を用いてドライブ・サーフェスとチェック・サーフェスを指定して行く。GOFWD文の代りにGORGT文やGOLFT文が使われることもある（図1.4）。TANTOは工具とチェック・サーフェスの接触規制子の一つで、工具がチェック・サーフェスに接するという条件を与えている。他にもPAST、ON、TOの指定ができる（図1.5）。

詳細は省略するが、APT言語では図形要素相互の関係や工具とサーフェスの関係が一義的に決まるように、修飾詞や接触規制子を用いてすべて文形式の記述によって定義して行く。平行を表すPARLEL、垂直を表すPERPTO、交点を表すINTOFなどが図形定義にしばしば使われる。

APT言語の解説を述べるのがここでの目的ではないが、複雑な形状や工具動作を記述するに十分な語いが用意されており、柔軟な表現が可能である。しかしながらその反面、図に書けば簡単な形でも、文法上の誤りを犯さないように面倒な文形式の記述を強いられる場合もある。

APTの機能を全面的に処理するにはかなり大型の計算機が必要になるため、そのよく使われる部分だけを抽出して小型の計算機でも処理できるようにしたサブセットが多数作られている。<sup>3)</sup>また、文の記述だけでは形状が把握しにくく、プログラムの正しさの検査が行いにくいので、グラフィック・ディスプレイ装置を併用するAPTシステムも作られている。<sup>4)~6)</sup>ただし、この場合でも図形定義や工具動作の文による記述は省略できない。グラフィック・ディスプレイ装置を用いると、プログラミングの途

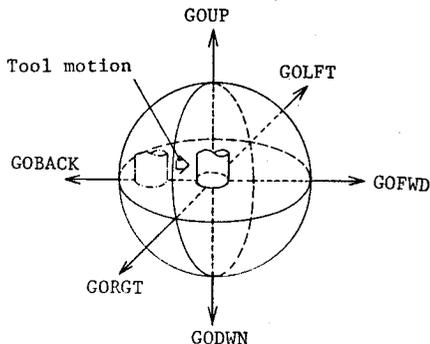


図 1.4 連続動作の方向指示

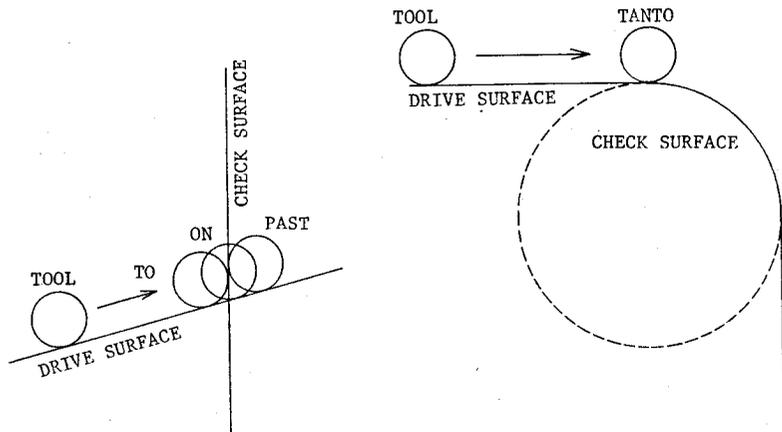


図 1.5 接触規制子

中でも図形定義や工具動作の検証ができるという利点がある。

### 1.3 蓄積管末端の活用

図 1.2 の例の場合、まず円の中心点を定義し、これをもとに円を定義し、続いて円に接する直線を定義することによって形状を定めることができる。このような図形定義にはグラフィック・ディスプレイ装置を用いて、目で確かめながら処理を進めるのが効果的である。従来より APT 系言語とは別個にグラフィック・ディスプレイを効果的に利用するシステムがいくつか作られて来た<sup>7),8)</sup> とくにライトペンの使えるシステム<sup>7)</sup>では、定義済みの図形要素を使って別の図形を定義する際、ライトペンによるピッキング機能を使うことができ大変便利である。反面、リフレッシュ・メモリを持ったシステムが必要で、大がかりな装置を必要とする。

本研究で作成したシステムでは、大型計算機のタイムシェアリング・システムに蓄積管を用いたグラフィック・ディスプレイを端局として用いており、リフレッシュ・メモリもライトペンも付属していない。したがって、すでに定義済みの図形のピッキングや管面上のメニューの選択にはクロスヘア・カーソルによる座標入力機能を利用する。

例えばすでに定義された円をピックする場合を考える。このとき管面上にその円が描かれており、その円の中心座標と半径は計算機の記憶装置に貯えられている。この状態でクロスヘア・カーソルを円周上の一点に合わせ、ピッキングの指示を与えるとその座標値が入力されるので、中心との距離を計算して登録済みの半径と比較することにより、どの円がピックされたかを知ることが出来る。直線の場合も同様であって、カーソル位置と定義済みの直線との距離によって、どの直線が指示されたかを知る。ライトペンを用いる場合との相違は、計算機内に登録された図形要素について順に調べて行かないと、ピックされたものがどれであるか分からない点である。このため、あまり複雑な形状の図形要素を許すと計算時間がかかる。現在使用中のシステムでは図形要素を円と直線に限っているが、一般 2 次曲線程度まで拡張することは可能であろう。

ここで開発したシステムにおいて許される図形要素定義法をまとめて図 1.6 に示す。ここで、座標値、距離、角度などが予め分っているものについては、正確な値をキーボードから入力する。すでに定義済みの要素を使って、例えば円に接線を引くようなときには上述のピック機能を用いる。また、円に二つの接線が引けるとき、あるいは二つの交点が存在するとき、その中の一方を選択するのにもやはりピック機能を用いる。これによって、APT では必要であった RIGHT, LEFT, XLARGE, XSMALL などの修飾語を用いなくて済ませる。

また定義法の指示そのものも管面上のメニューの選択によるので、いわゆるステートメントによる記述なしに図形要素の定義が行える。蓄積管を用いて図形処理を行う場合、画面の一部だけを消去することはできない。このため、新しい図形要素を定義すると、点滅する点がまずその図形の軌跡を動くようになっている。これを使用者は目で確かめて肯定の応答をシステムに返すと初めて管面に図形が描かれる。この機能をシステムに組み込むことにより、点滅する点による描画はやり直しができるので、誤りを十分に防ぐことができる。

以上のように図形要素を円と直線だけに限っても、これで表現される部品もかなり多いのでそれ程実用価値が損われるわけではない。逆に処理の面から見ると、このように取り扱う対象の範囲を限ってしまうと、簡便な手法が有効に使えるのである。定義された図形要素は必要なパラメータ（半径や座標値など）

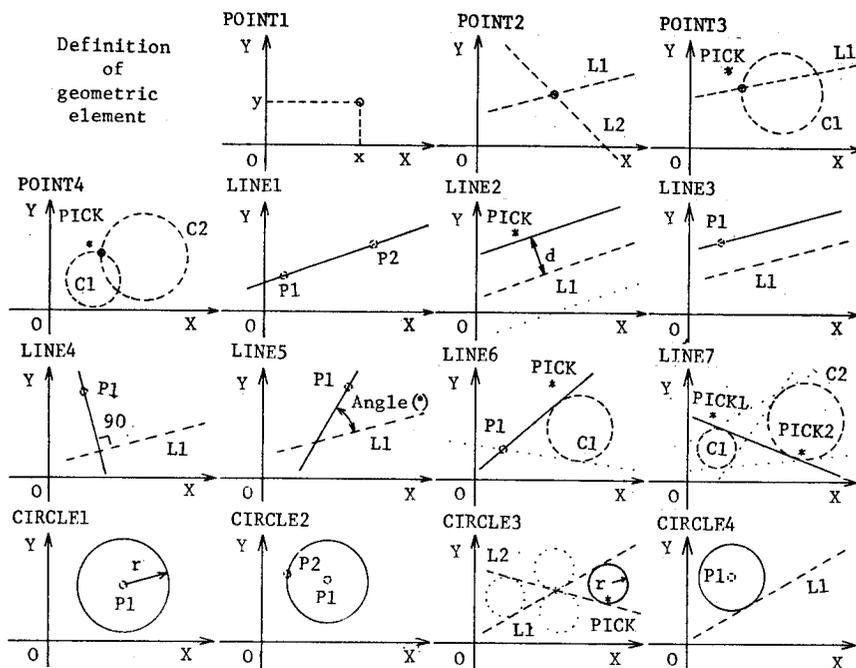


図 1.6 図形要素定義法

と共に図形要素表に登録される。この簡便な図形定義の方法はタブレット型座標入力装置とプロッタを用いても実現できる。<sup>9)</sup>この場合は下絵を書きしておく必要があり、グラフィック・ディスプレイを用いた方が使い勝手がよい。

図 1.7 に図形定義を実施した時の管面上の様子を示す。これは蓄積管のハードコピー・ユニットによる出力である。管面上右端にファンクション・エリアが取っており、クロスヘア・カーソルを動かして図形定義法の選択などが指定できるようになっている。また左端には簡単なメッセージが出力され、使用者に対して次に何を行うべきかの指示を与える。使用者が図形定義の種類を選択した後は、図形のピッキングや数値の入力をこの指示に従って進めればよい。

例えば、2直線に接する円を定義する場合を考える。2直線  $L1$ 、 $L2$  はすでに定義済みであるとする。システムからの **SELECT** の指示に従って、メニュー上の定義法 **CIRCLE 3** をカーソルで選択する。続いてシステムからの **PICK** の指示により、2直線  $L1$  と  $L2$  をカーソルでピッキングする。次にシステムからの **INPUT** の指示により半径  $r$  をキーボードから数値入力する。実際に2直線に接する円は4個存在するので、最後に所望の円の内部になる位置をピッキングする。これで一つの円が定められ、点滅するビームによって円が仮表示される。もしその円が正しければメニュー上の **DESIRED** をカーソルで選択する。以上の操作で新しい円が定義され、管面上に表示されると共に記憶装置上の図形要素表にパラメータが登録される。このような手順の繰り返しで図形定義が行われるので、**A P T** のような文形式の記述は不要である。

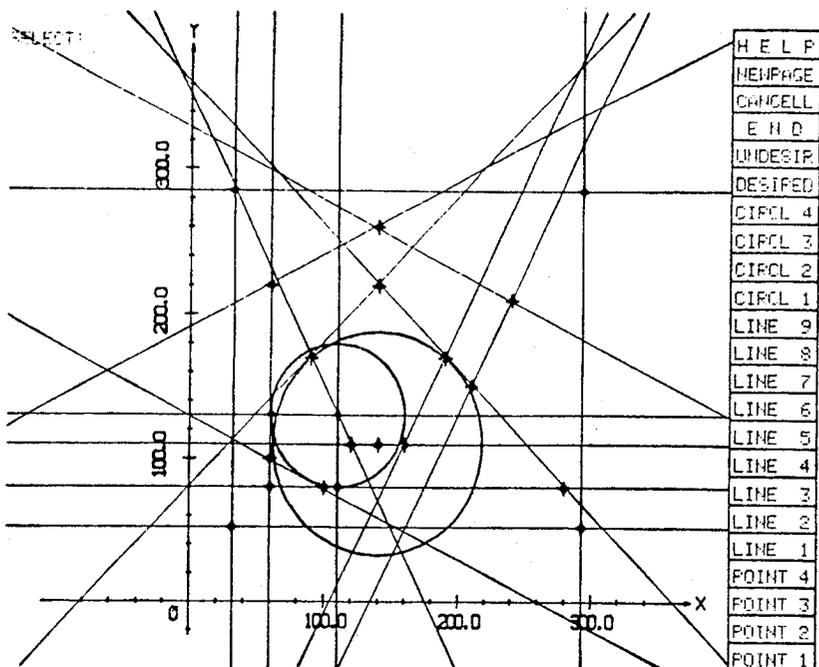


図 1.7 管面上の図形定義

## 1.4 工具径路の定義

### 1.4.1 部品形状の決定

工具径路は荒切削やポケット・カッティングも含めて総合的に決定されるので、まず必要なことは除去する部分と切残す部分との境界を形成する輪郭を決定することである。この機能は管面上に描かれた図形要素をもとにして、APTのドライブ・サーフェスに当る部分およびサーフェスが切替る点を順にピックアップすることによって遂行される。いま2次元形状だけを考えているので、一つの $xy$ 平面上でこの閉じた輪郭を決める。島を残すようなポケット・カッティングも許しているので、一つの平面上の輪郭は一つとは限らない。

ピックアップによって輪郭を定めるには、図 1.8 の左に出ているメッセージに従う。基本的には PICK START-P から始まって PICK NEXT-P と PICK EMT の繰り返しである。前者はドライブ・サーフェスの切り換わる交点または接点をピックアップすることを表し、後者はそれらの点の間をつなぐ要素 (element) をピックアップすることを表す。単に交点や接点を指して行くだけでは輪郭が一義的に決まらない場合があるので、PICK EMT の機能が用意されている。閉じた輪郭を指定するピックアップによって、ドライブ・サーフェスの連鎖が輪郭表に登録されて行く。このとき、右向きに指定して行くか左向きに指定して行くかが明瞭に区別される。この方向と TOOL RIGHT または TOOL LEFT の指示によって輪郭の内側が除去されるか外側が除去されるかが定められる。また一つのドライブ・サーフェスから別のドライブ・サーフェスへ移る点 (交点や接点) の座標値も、図形要素表のパラメータを用いて計算しておく。また、荒加工のことを考慮して素材形状も一つの輪郭として登録する。ポケットの輪郭については APT

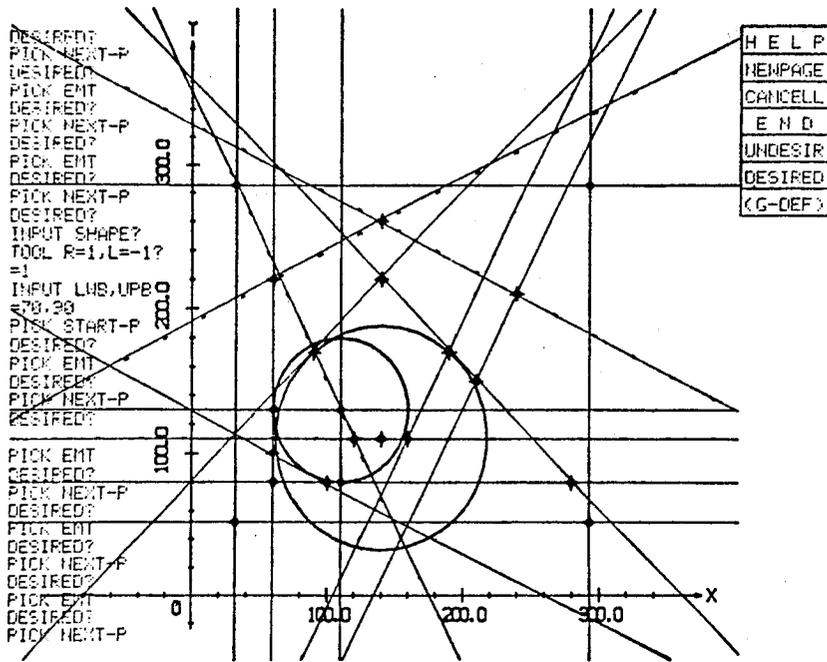


図 1.8 管面上の輪郭定義

より制限がゆるめられており、円弧を用いても構わない。<sup>10)</sup>

一つの  $x, y$  平面についてすべての輪郭が定義されると次は  $z$  座標値をかえて必要な  $x, y$  平面を定め、これについて同様にすべての輪郭を定義する。 $z$  座標値をかえることは APT でいえばパート・サーフェスを変更することに相当する。このようにしてすべての輪郭が定義されると図 1.9 に示すような表が記憶装置内に作られる。

図 1.9 において、パート・サーフェスはすべて  $z$  軸に垂直な平面であるから  $z$  座標値だけが分っていればよい。工具は一つの平面の加工が終ると、 $z$  座標値の差にあたる距離だけ  $z$  軸方向に下って次の平面の加工に移る。輪郭表とのつなぎにはポインターを用いており、輪郭の削除や追加を容易にしている。

輪郭表のドライブ・サーフェス  $g_i^{(j)}$  は工具がこれに沿って動く図形要素を表し、これもまた図形要素表へのポインタになっている。

輪郭がすべて定義されると図 1.9 の記憶形式をもとに図 1.10 のような、ワイヤフレーム状の図形表示が管面上に得られる。陰線消去は行っていない。

#### 1.4.2 工具径路の決定

荒加工の段階ではメッシュ法によって、除去する部分と切残す部分とを区別する。<sup>11)</sup> この方法では、部品形状外周の除去部分もポケット・カッティングも同様に扱われ、島を残すようなポケット・カッティングも特別な処理は不要である。素材を格子状に分割し、除去部分に 1 のフラグを立てる。

これは図 1.9 の輪郭定義からその格子が除去部分に含まれるかどうかを判定することによって行われる。格子の大きさは工具直径の 0.8 倍とし、除去部分の格子が明らかになればそれがそのまま荒加工の工具径

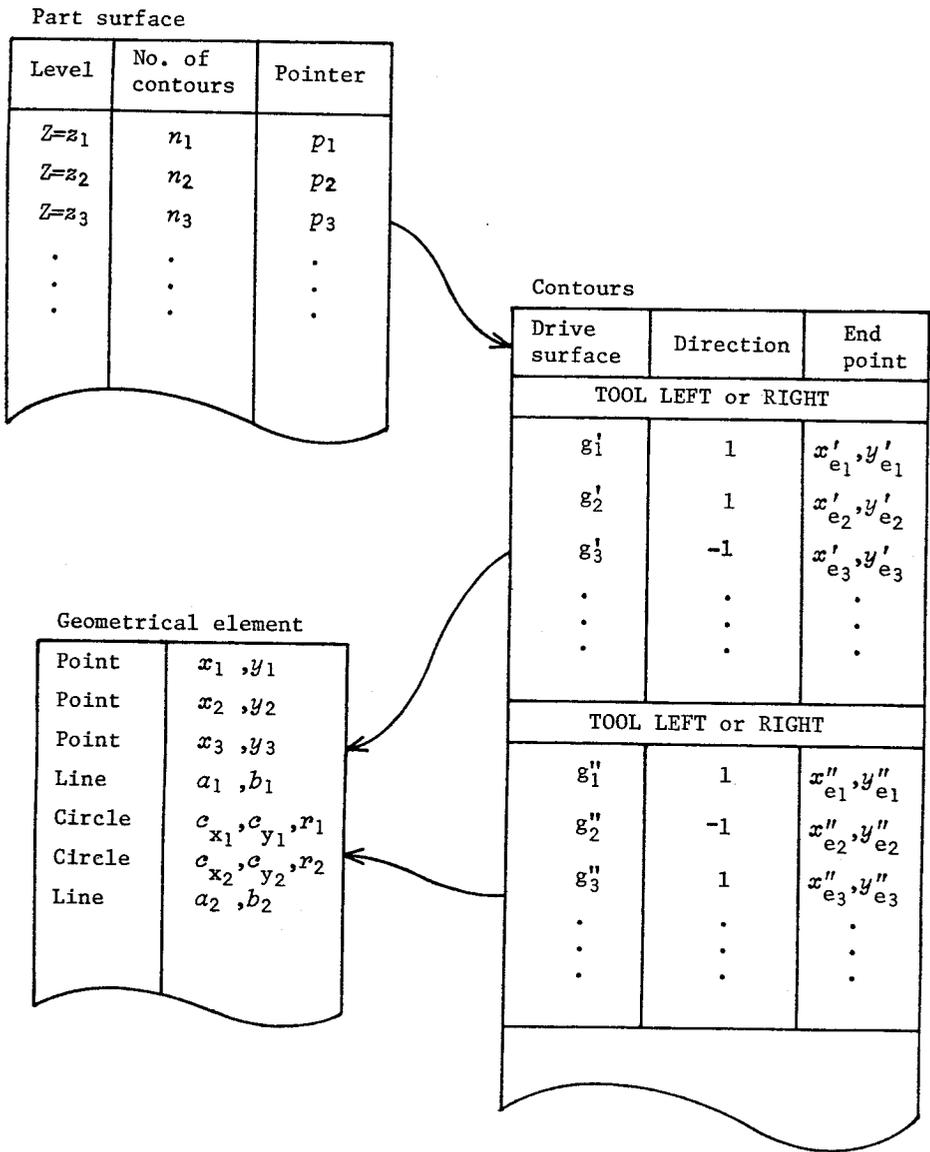


図 1.9 部品形状の記憶形式

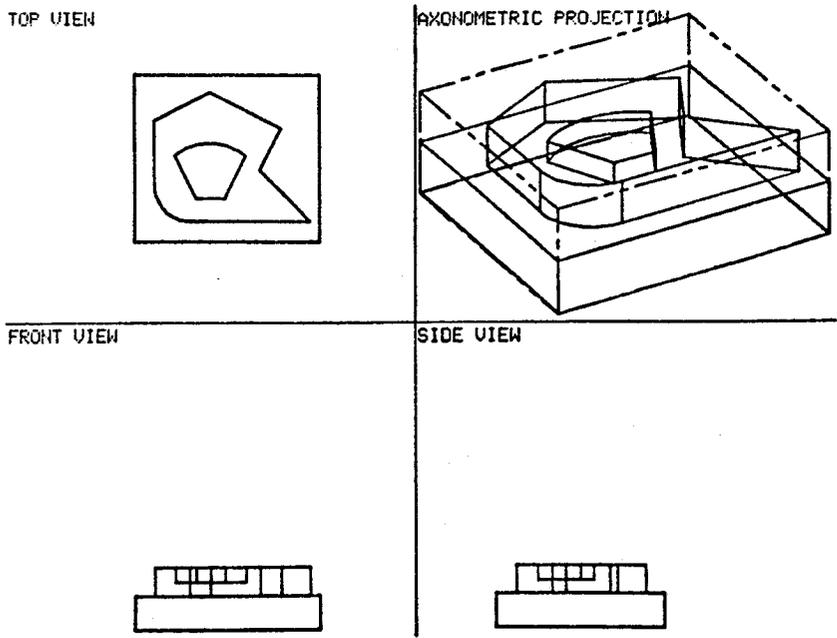


図 1.10 図形表示 (ワイヤフレーム)

路になるようにしてある (図 1.11)。このメッシュ法による荒加工の径路も、 $z$ 座標値をかえて輪郭の定義されている  $x y$ 平面それぞれについて決定する。

荒加工での工具径路が定まると、次に仕上げ加工における工具径路を求める。これには再び図 1.9 を用いてドライブ・サーフェスに沿うように工具径路を定める。この場合、荒加工のメッシュの幅を相当大き

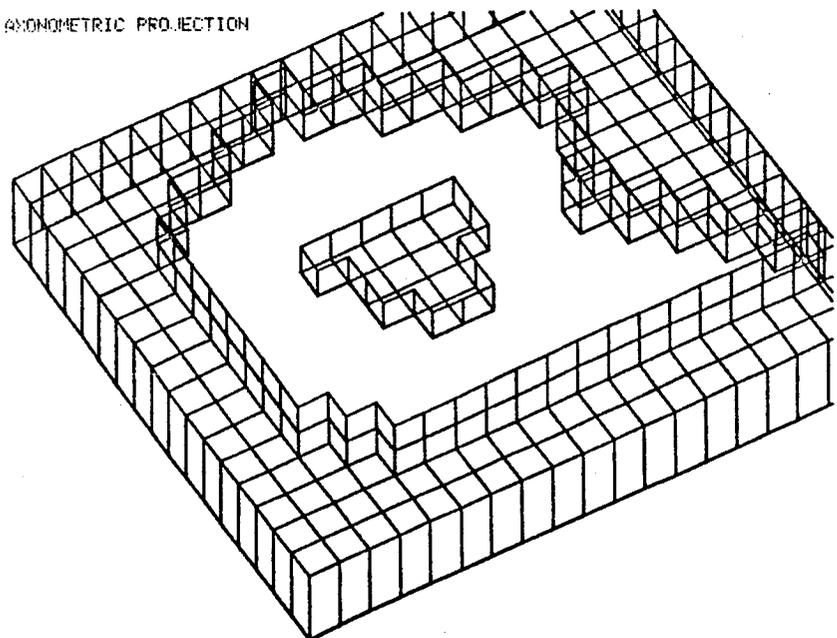


図 1.11 メッシュ法による除去部分表示

く取っているので一回だけの仕上げ切削は無理であり、5回程度の切削を行うようにする。仕上げの過程では一回の切込み深さを大きくとることは、加工精度の面で好ましくない。当然のことながら仕上げ切削においては工具径オフセットが考慮されている。ポケット・カッティングについても同様である。この場合、ドライブ・サーフェスが切替わる点の近傍では次のドライブ・サーフェスも考慮に入れることにより、連続径路を得るための円弧を挿入したり、凹部において切残しが生ずることはあっても余分の切込みが発生しないようにする。以上の処理の後、図1.12に示すような荒加工と仕上げ切削を含めた工具径路が得られる。工具径路が交錯して検証を行にくい場合、その一部を拡大してみることも可能である(図1.13)。最後にこの工具径路をもとにしてNCテープ・イメージをファイルに出力する。

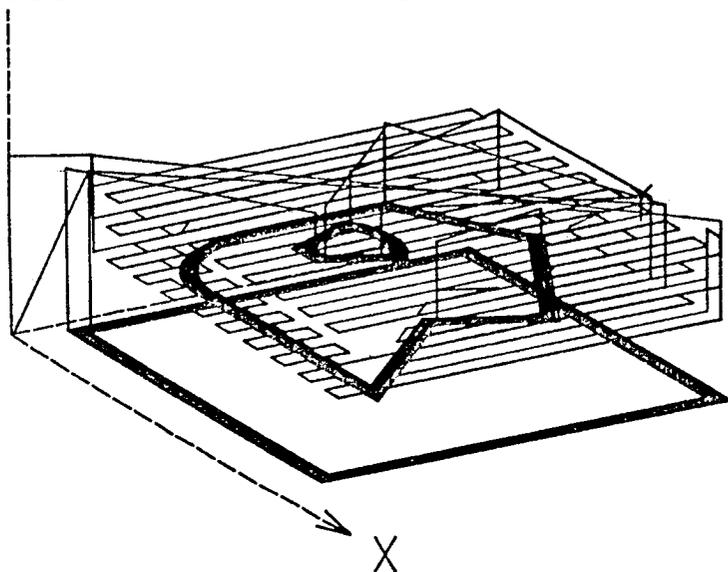


図1.12 工具径路

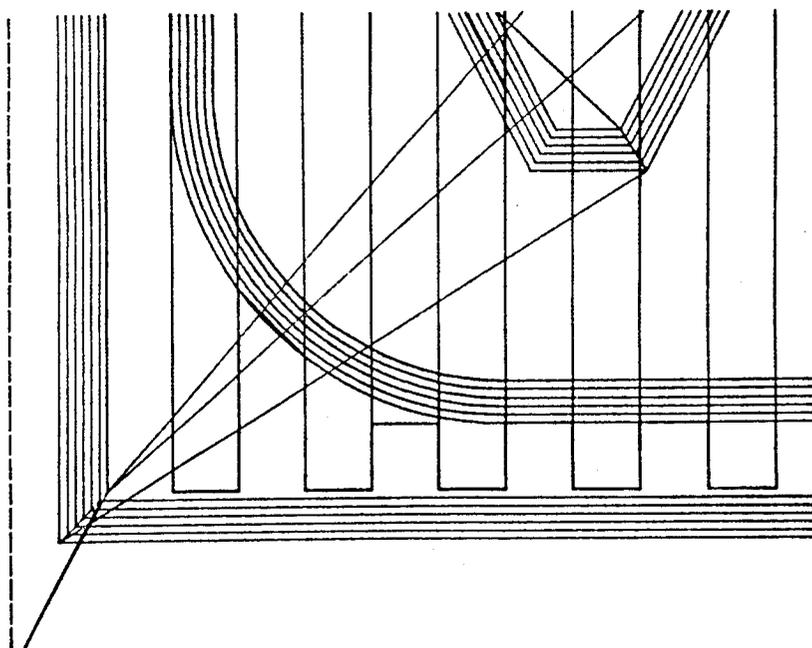


図1.13 工具径路部分拡大図

以上の処理手順をまとめると以下の四つの段階があることがわかる。

- 1) 図形要素の定義
- 2) 輪郭の定義
- 3) 工具径路の決定
- 4) NC 指令の作成

これらそれぞれの段階について、中間結果はファイルに保存されており、修正を行いたい場合には途中の段階からの修正が可能である。必ずしもすべての段階をやり直す必要はない。NC 指令もファイルに保存されているので、使用者はファイル・エディタを用いて送り速度の指定やM機能（クーラントのオン、オフなど）の指令を必要に応じて挿入することができる。

## 1.5 プロセッサ開発時の誤り訂正問題

### 1.5.1 誤りの事象の因果関係

第 1.3 節第 1.4 節で述べたシステムのプロセッサはプログラミング言語 FORTRAN を用いて作成した。すなわちパート・プログラマは TSS モードで働くこの応用プログラムをグラフィック端末を介して利用することができる。本節ではこの応用プログラムを開発した時のデバッグ問題について述べる。

FORTRAN を使って応用プログラムなどを作成する場合、文法上の形式的な誤りは翻訳処理時にエラー・メッセージとして出力されるので比較的簡単に訂正ができる。しかしながら実行時の誤り訂正には多くの労力が費され、しかもこの問題はシステムの信頼性にかかわる重要な問題である。実行時の誤り訂正が困難であるのは、本来の誤りの源が表面に現れず、表面的には別の誤りの状況が把握されることによる。そしてその両者の間には中間的な誤りの事象が介入していることが多く、表面的な誤りの状況からその中間的な事象をたどって誤りの源を見出すのが容易ではないのである。

いま FORTRAN 一般について中間的な誤りの事象を含めてあらゆる誤りの事象と誤りの原因追跡に使われる事象（たとえば " 引数に配列名を用いている " など）の全体を有限事象集合  $V_e = \{ v_1, v_2, \dots, v_n \}$  として扱う。これらをコード番号とメッセージの形式でファイルに格納しておき、キー・ワードによる検索が可能であるようにしておく。

つぎにこの事象集合に因果関係を導入する。ある一つの事象  $v_i$  についてその事象が出現する原因となり得る事象（これは一つとは限らない）すべてに  $v_i$  から有向辺をつける。この因果関係は FORTRAN 一般についていえば決定論的にまた一義的に決まるものではないので、可能性のあるものすべてに有向辺をつけ、これに確率論的の重みをつける。一つの事象から出ている有向辺の確率の和が 1 になるものとし、導入した関係を 2 項関係と仮定すると、上の手続きによって一重マルコフ連鎖と推移確率行列が得られる。これらもファイルに格納しておいて、必要に応じてその一部を主記憶装置に取り出せるようにしておく。

なお、一つの事象に注目するとき、これが原因となって現われる別の事象も一つとは限らないから、一般には一つの事象に入ってくる有向辺も複数個になり得る。また、これらの事象の中で誤りの源となり得るものは、それ以上に原因となる事象は見出せないことになるので、これには重み 1 の自己サイクルを付けて吸収事象とする。これに対して他の事象への有向辺を持つ事象を過渡事象と呼ぶ。

以上により、FORTRAN プログラムにおけるいろいろな誤りの事象相互の一般的な因果関係が、確率論的モデルとして具現化されたことになる。<sup>12)</sup> 簡単なグラフの例を図 1.14 に示す。

図 1.14 においては、記憶装置の保護領域の侵犯 (memory protection violation) が表面上

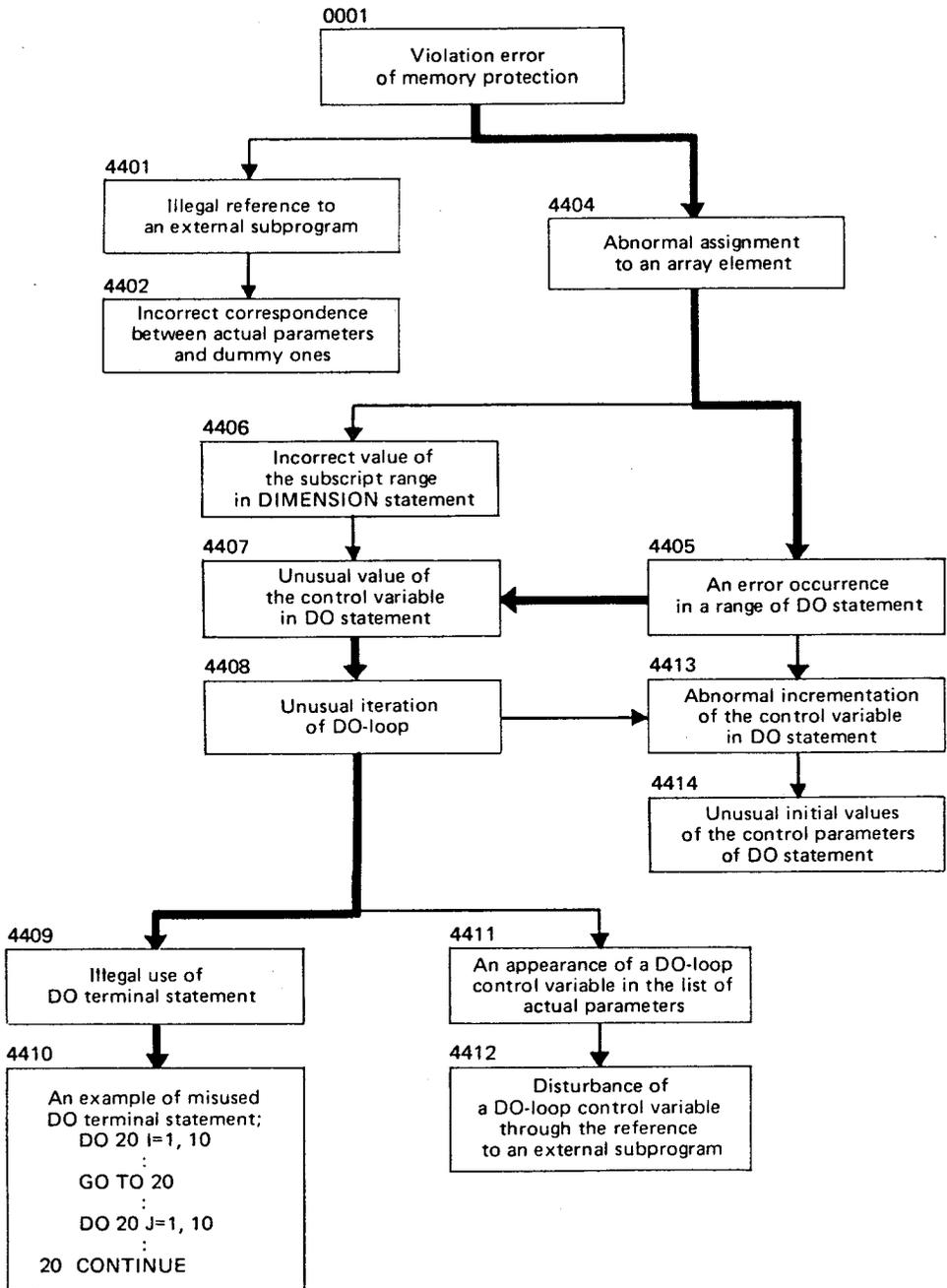


図 1.14 誤りの事象の因果関係

の誤りとして取り上げられており、その原因として、副プログラムとのパラメータの不適合、DOステートメントの制御変数の誤り、あるいはDOループの構成の誤りなどが考えられることを示している。確率的重みについては省略してある。一人のユーザがかかえている誤りは、これらの原因の一つに相当し、例えば太い矢印で示した因果関係がこれに当たっていることを示している。

### 1.5.2 対話処理による誤りの追跡

FORTRAN プログラマが現在自分の抱えている表面的な誤りの状況からキーワードを計算機に投入すると、計算機側では確率論的重みに従って、プログラムのチェック・ポイントをプログラマに提示するので、プログラマはこの助言に従って自分のプログラムの誤りを追跡して行く。いま計算機側のシステムを助言システムと呼ぶことにすると、助言システムは誤りの状況の一般的な因果関係を知識として持っていることになる。

助言システムの内部処理について具体的に述べると、まずプログラマからのキーワードの投入によって初期事象  $v_s$  を定め、この事象から有向辺に沿って到達可能な事象をデータ・ファイルから取り出す。これらについて過渡事象だけの推移確率行列  $Q$  を用いて次式を定義する。

$$H \equiv [ h_{ij} ] \equiv \sum_{k=0}^{\infty} Q^k = [ I - Q ]^{-1} \quad (1.1)$$

過渡事象だけを考慮しているので式 (1.1) は収束する。<sup>13)</sup>  $I$  は単位行列である。また過渡事象の中には強連結部分が含まれていてもよい。因果関係の順序を考えるとこの強連結部分は一まとめにして扱い、強連結部分内部での順序関係は考えない。この場合  $h_{ij}$  は  $v_i$  から出発したとして  $v_j$  を訪れる回数の期待値を与える。

つぎに、 $v_i$  から出発して  $k$  段階推移後に初めて  $v_j$  に到達する確率  $f_{ij}^{(k)}$  は次式で与えられる。<sup>13)</sup>

$$\begin{aligned} f_{ij}^{(1)} &= q_{ij}^{(1)} \\ f_{ij}^{(k)} &= q_{ij}^{(k)} - f_{ij}^{(1)} q_{jj}^{(k-1)} - f_{ij}^{(2)} q_{jj}^{(k-2)} \dots \dots \dots f_{ij}^{(k-1)} q_{jj} \\ &\quad (k \geq 2) \end{aligned} \quad (1.2)$$

ここで、 $q_{ij}^{(k)}$  は  $Q^k$  の第  $i$  行第  $j$  列目の要素を表す。式 (1.1) が収束するという条件のもとに式 (1.2) を  $k$  について加え合わせると次式が得られる。

$$f_{ij} \equiv \sum_{k=1}^{\infty} f_{ij}^{(k)} = \frac{h_{ij} - \delta_{ij}}{h_{jj}} \quad (1.3)$$

$f_{ij}$  は推移の段数を問わずに、 $v_i$  から出発していつかは  $v_j$  に到達する確率を表す。 $\delta_{ij}$  はクロネッカーのデルタである。

吸収事象については、過渡事象から吸収事象へ吸収される確率を考える。一段の推移で過渡事象から吸収事象へ移る推移確率行列を  $D$  とすると、任意の過渡事象  $v_i$  から出発して吸収事象  $v_e$  に吸収される確率は

$$[ b_{ie} ] = [ I - Q ]^{-1} D \quad (1.4)$$

で表される。

助言システムは、プログラマの指定した初期事象  $v_s$  をもとに、式 (1.3)、(1.4) を用いて到達確率の大きい事象を選び出す。また、その事象へ到達する順路も有向辺の確率の積が大きいものを選ぶ。このようにして得られた事象の系列をプログラマに提示する。プログラマはこれに基づいて抱えている問題の追跡を行う。助言システムの提示する事象系列は確率論的モデルによるものであるから、個々のプログラマの問題にそのままあてはまるとは限らない。プログラマは助言システムからの事象系列が自分の追跡過程と明らかに異なる点を見出したら、そこで助言システムに対してその旨の応答をメッセージのコード番号と共にシステムに返す。助言システムではそれによりグラフの一部を書き換えたり、別の順路で確率の大きいものを探したりする事によって別の事象系列を提示する。

以上のような対話処理のくり返しにより、助言システムの提示するメッセージはプログラムの抱えている問題に密着したものとなり、プログラムも誤りの原因追跡に有効な助言を得ることになる。この助言システムは大阪大学大型計算機センタのタイムシェアリング・システムで稼動しており、一般使用者に開放されている。<sup>14),15),16)</sup> 第 1.3 節、第 1.4 節に述べたプロセッサの開発時にもこれを利用して、十分なデバッグを行うことによって信頼性の高い応用プログラムを作成することができた。

## 1.6 結 言

本章では汎用性のある APT 言語と対比させて、簡便な図形処理による NC テープ作成のためのプロセッサについて論じた。重要な論点は、処理の対象を円と直線による 2 次元形状に限ってしまうと、簡便な方法が使えろということである。APT 言語による方法と簡易図形処理による方法は競合するものではなく、処理対象に応じて使い分ければよいのである。簡易図形処理によるシステムの特徴を以下に述べる。

- 1) 経済的な蓄積管によるグラフィック・ディスプレイ装置を効果的に利用している。
- 2) 扱形状が単純であるため言語による記述は不要であり、ディスプレイ装置の管面上の操作とキーボードだけですべて処理される。
- 3) タイムシェアリング・システムを用いることによりすべて対話型の処理で済ませる。
- 4) 形状は単純なものに限られるが、島を残すようなポケット・カッティングが可能である。
- 5) 荒加工の工具径路についてもその自動決定を試みている。

使用者の側に立ってみると、手続きあるいは工具動作を逐一記述することからの開放は大きな意味がある。しかしながら汎用性を持たせるためには言語によって逐一記述することが要求される。したがって、今後も APT 言語はその汎用性を活かして使われるであろうし、また一方では処理対象を適当に限って、それらのための専用プロセッサとして簡便なシステムが作られて行くであろう。

## 参 考 文 献

- 1) 研野和人：数値制御のソフトウェア、産業図書（1970）。
- 2) S. Hori (ed.) : APT Part Programming, IIT Research Institute, McGraw-Hill Book Company (1967).  
山岸正謙 監訳：数値制御のプログラミング、大河出版（1970）。
- 3) たとえば、  
Hajime Kishi : Low-Cost Automatic Programming and Time-Shared Software-Wired NC, Proc. of the Tenth Annual Meeting and Technical Conference, NCS (1973), pp. 206-219.
- 4) G. Greyerbiehl : CRT Graphics and Numerical Control, Proc. of the Eighth Annual Meeting and Technical Conference, NCS (1971), pp. 155-170.
- 5) L. E. Sanchez : APT-Interactive Graphics System, Proc. of the Ninth Annual Meeting and Technical Conference, NCS (1972), pp. 379-393.
- 6) J. Cremerius : APT/IGS : State of the Art in NC Graphics, Proc. of the Second IFIP/IFAC International Conference, PROLAMAT '73 (1973),

pp. 87-107.

- 7) F. Yamaguchi, S. Takashima, F. Taira and M. Kanazawa : A Conversational Numerical Control Graphics, Proc. of the Eighth Annual Meeting and Technical Conference, NCS(1971), pp.120-154.
- 8) K. J. Davies : GNC-A Graphical NC Processor, Proc. of the Second IFIP/IFAC International Conference, PROLAMAT '73(1973), pp.51-70.
- 9) 堀野 均、山県敬一、牧之内三郎 : 図形入力装置を用いた会話型プログラミングの試み、昭和49年度精機学会総合生産システム・シンポジウム前刷、(1974), pp.85-88.
- 10) H. Persson : NC Machining of Arbitrarily Shaped Pockets, CAD, Vol.10, No.3 (1978), pp.169-174.
- 11) N. Okino, Y. Kakazu and H. Kubo : TIPS-1 ; Technical Information Processing System for Computer-Aided Design, Drawing and Manufacturing, Proc. of the Second IFIP/IFAC International Conference, PROLAMAT '73(1973), pp.141-150.
- 12) Y. Isomoto and K. Yamagata : Probable Transition Searching System, Information Processing in Japan, Vol.17(1977), pp.14-18.
- 13) 北川敏男編 : マルコフ過程、共立出版(1967)。
- 14) S. Makinouchi, Y. Isomoto and K. Yamagata : Information Service System for Program Debugging, Technol. Repts. Osaka Univ., Vol.27(1977), pp.211-219.
- 15) 磯本征雄、山県敬一 : 推移確率に基づく対話型事象検索システム、情報処理、第19巻、第3号(1978), pp.241-247.
- 16) Y. Isomoto, K. Yamagata and T. Ishiketa : A Computer Aided Advising System for FORTRAN Program Debugging : ADVISOR, Information Processing 80(1980), IFIP, North-Holland Publishing Company, pp.985-990.

## 第2章 板カム加工のNCテープ作成

### 2.1 緒言

本章では第1章とはまた別の観点からNCテープの生成について論じる。一般にAPT系言語を用いてNCテープを作成する場合、部品の設計図面がすでにでき上がっているのが立て前である。もしも設計の過程から直接NCテープを作成することができれば、これは大きな省力化に結びつく。機械部品一般についてこの手法を期待することは極めて困難であるが、ある特定の部品あるいは特定の性質を持つ部品の集まりについてこれを実現することは十分可能である。

板カム(マスタカム)は設計と加工を互いに関連づけながら実現できる典型的な例である。最近のマスタカムの試作においては、従節の変位を与えてカムの輪郭を計算機を用いて決定するのが普通である。この場合、加工時の工具径路も同様の手順で計算できるので、板カムの輪郭からNC指令を生成する過程ではほとんど人手の介入を要しない。ただし、加工条件に関する指令を挿入することだけは必要である。

本章ではこのような趣旨に沿うように、FORTRANを用いて開発した応用プログラムについて述べる。与えられた従節変位からカムの輪郭を計算する手順を明らかにし、この機構にかかわる計算プログラムと従節変位曲線のライブラリを用意することによって、両者の組み合わせで標準的ないろいろのカム設計が可能であることを示す。<sup>1),2)</sup>さらに、工具径路決定のためのプログラムについてもその手順を述べ、オンラインDNCシステム(第4章で詳述)の利用により、設計から加工への一貫した処理の流れが実現可能であることを示す。

### 2.2 板カムの輪郭計算

#### 2.2.1 円端直動従節板カム

いま、カムの回転角 $\theta$ に対して、直動従節の変位 $f(\theta)$ ならびに $\theta$ に関する導関数 $f'(\theta)$ 、 $f''(\theta)$ が与えられたものとして、カムの輪郭を求める。NCテープ作成時の参考のために、従節のローラ径とは異なる半径を持つ工具で加工する場合の工具軌跡も求めておく。実際のNC指令の作成法は第2.4節で詳しく論じる。また、曲率半径と板カムを実際に回転させた時の圧力角は重要な設計パラメータであるから、これらも併せて求めておく。

以下では、解の形をできるだけ見易い形で求めるために、従節がカムのまわりをまわるものとして解析を進める。一般のリンクカム機構の解析法としては、この手法は適当でない場合もあるが、加工のための工具軌跡を求めるという観点からは、この方法が分かり易い。以下の解析で、ベクトルを複素表示によって表現する。虚数の単位として $j$ を用いる。変則的ではあるが、式を簡単にするためベクトル記号と複素表示を一つの式に混在させて書く<sup>1)</sup>。本来ならば複素表示のベクトルに別の記号を用いるべきであるが、混乱はないと思われるので変則的な表記法を用いる。

図2.1において、角 $\theta$ はカムの回転角を逆に従節の回転角におきかえたものであり、 $x y$ 座標軸はカムに固定されており、カムの回転中心に原点がある。 $P_f$ は従節の軌跡を表すベクトル、 $P_p$ はカムの輪郭を表すベクトル、 $C$ はカム中心の従節運動方向からのオフセット量を表し、その長さ $c$ は常に一定である。

図2.1より

$$C = j c e^{j\theta}, \quad L = [r_0 + f(\theta)] e^{j\theta} \quad (2.1)$$

ここで $r_0$ はカムの最小半径に対応する従節基準位置を表す。いま、従節ローラ中心軌跡を表すベクトル

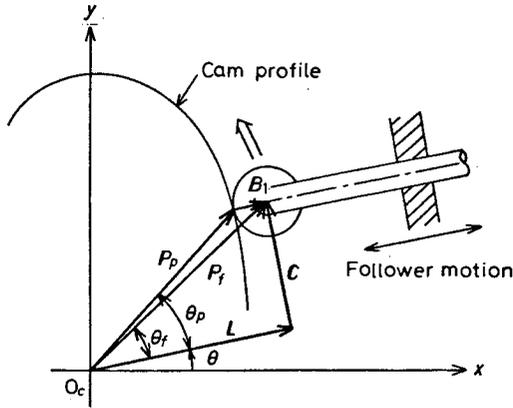


図 2.1 円端直動従節板カム

$P_f$  の大きさを  $R_f$ 、偏角を  $\theta + \theta_f$  で表すと

$$P_f = R_f e^{j(\theta + \theta_f)} \quad (2.2)$$

と書ける。

$$P_f = L + C \quad (2.3)$$

と式 (2.1) を用いて

$$R_f e^{j(\theta + \theta_f)} = [r_o + f(\theta)] e^{j\theta} + j c e^{j\theta}$$

と書けるから

$$R_f e^{j\theta_f} = [r_o + f(\theta)] + j c \quad (2.4)$$

より

$$R_f = \sqrt{[r_o + f(\theta)]^2 + c^2} \quad (2.5)$$

$$\tan \theta_f = \frac{c}{r_o + f(\theta)}$$

が得られる。従節ローラと同一半径の工具でカムを加工するときには、このパラメトリック曲線がそのまま工具軌跡を与える（動径  $R_f$ 、偏角  $\theta + \theta_f$ ）。

カムの輪郭曲線を求めるには、カムと従節ローラが接する点において、カムの輪郭と従節の軌跡が共通法線を持つ条件（包絡面条件式）が必要である。このため、従節の運動軌跡の接線の方法を求める。

式 (2.3) を  $\theta$  で微分すると

$$P_f' = L' + C' \quad (2.6)$$

上式と式 (2.1) とから

$$\begin{aligned} P_f' &= f'(\theta) e^{j\theta} + j [r_o + f(\theta)] e^{j\theta} - c e^{j\theta} \\ &= [f'(\theta) - c] e^{j\theta} + j [r_o + f(\theta)] e^{j\theta} \end{aligned} \quad (2.7)$$

従節の動点の軌跡に沿う移動長さを  $s$  で表し、

$$P_f' = s' e^{j(\theta + \phi)} \quad (2.8)$$

とおくと

$$s' e^{j(\theta+\phi)} = [f'(\theta) - c] e^{j\theta} + j[r_0 + f(\theta)] e^{j\theta}$$

が成り立ち、次式が得られる。

$$s' = \sqrt{[f'(\theta) - c]^2 + [r_0 + f(\theta)]^2} \quad (2.9)$$

$$\tan \phi = \frac{r_0 + f(\theta)}{f'(\theta) - c}$$

ここで、角度  $\phi$  は従節運動方向を基準とする接線の方向である。したがって、法線の方向を  $(\theta + \phi)$  で表すと、 $\psi = \phi - \pi/2$  を用いて

$$\tan \psi = \frac{c - f'(\theta)}{r_0 + f(\theta)} \quad (2.10)$$

である。

図 2.1 において、接点と従節ローラ中心を結ぶベクトル  $B_1$  は次式

$$B_1 = r_r e^{j(\theta+\phi)} \quad (2.11)$$

で表される。ただし、 $r_r$  は従節ローラの半径である。

さて、

$$P_p = P_f - B_1 \quad (2.12)$$

と書けることに注目し、ベクトル  $P_p$  の大きさを  $R_p$ 、偏角を  $\theta + \theta_p$  とおくと

$$R_p e^{j(\theta+\theta_p)} = [r_0 + f(\theta)] e^{j\theta} + j c e^{j\theta} - r_r e^{j(\theta+\phi)}$$

が得られ、

$$R_p e^{j\theta_p} = r_0 + f(\theta) - r_r \cos \phi + j(c - r_r \sin \phi) \quad (2.13)$$

が導かれる。

上式より、カム輪郭上の任意の点  $P_p (R_p, \theta_p)$  は  $\theta$  を媒介変数として以下のように表される。

$$R_p = \sqrt{a_p^2 + b_p^2}$$

$$\theta_p = \theta + \theta_p \quad (2.14a)$$

ただし、 $a_p = c - r_r \sin \phi$

$$b_p = r_0 + f(\theta) - r_r \cos \phi$$

$$\tan \theta_p = \frac{a_p}{b_p} \quad (2.14b)$$

従節ローラと径の異なる工具で加工する場合の工具軌跡は、切削点におけるカム輪郭の法線上に工具中心が来るという条件を用いて求める。カムの輪郭に対して工具（半径  $r_t$ ）をオフセットするベクトルを、カムに対して外向きに  $B_2$  とすれば

$$B_2 = r_t e^{j(\theta+\phi)} \quad (2.15)$$

であり、工具の中心軌跡を表すベクトル  $P_t$  は

$$P_t = P_f - B_1 + B_2 \quad (2.16)$$

で与えられる。式 (2.14) を導いたのと全く同様の手順で、工具軌跡の極座標表示  $P_t (R_t, \theta_t)$  は以下のようになる。

$$R_t = \sqrt{a_t^2 + b_t^2}$$

$$\theta_t = \theta + \theta_t \quad (2.17a)$$

ただし

$$a_t = c - (r_r - r_t) \sin \psi$$

$$b_t = r_o + f(\theta) - (r_r - r_t) \cos \psi \quad (2.17b)$$

$$\tan \theta_t = \frac{a_t}{b_t}$$

ここで  $r_r = r_t$  とおけば、従節のローラ中心軌跡を表すことになるので式(2.17)は式(2.5)と一致する。

図2.2の角度 $\alpha$ は圧力角の定義を示す。圧力角は、カムと従節との共通法線方向が従節の運動方向に対してなす角であるから、小さい程望ましい。図2.2において、法線力 $F$ の分力として従節の運動方向に直角な方向に働く力

$$F_n = F \sin \alpha \quad (2.18)$$

は従節を動かす力としては役立たない。一般に圧力角 $\alpha$ の限界値は直動従節で $30^\circ$ 、揺動従節で $45^\circ$ といわれている。直動従節カムの設計においては注意を要する条件である。

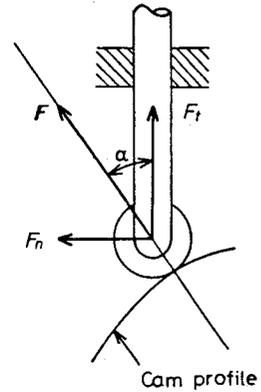


図2.2 圧力角

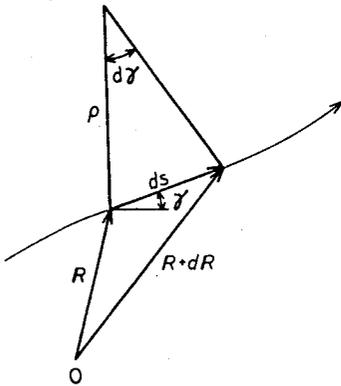


図2.3 曲率半径

本節で述べた円端直動従節の場合、法線方向を表す式(2.10)において、角度 $\psi$ は従節運動方向からの相対的な角度を表しており、そのまま圧力角を表現していることになる。

曲率半径は、図2.3に示すように、軌跡を表すベクトル $R$ の動きに対して、軌跡に沿う移動長さ $s$ の微小成分 $ds$ と接線の向きを表す角度 $\gamma$ の微小成分 $d\gamma$ との比

$$\rho = \frac{ds}{d\gamma} = \frac{ds}{d\theta} \frac{d\gamma}{d\theta} = \frac{s'}{\gamma'} \quad (2.19)$$

によって求められる。

まず、図2.1に従って従節のローラ中心軌跡の曲率半径を求める。ここでは $\gamma = \theta + \phi$ である。簡単のため

$$G(\theta) = r_o + f(\theta), \quad H(\theta) = f'(\theta) - c \quad (2.20)$$

とおくと、式(2.9)は

$$\tan \phi = \frac{G(\theta)}{H(\theta)}, \quad s' = \sqrt{[H(\theta)]^2 + [G(\theta)]^2} \quad (2.21)$$

と表される。 $\phi' = d\phi/d\theta$ を求めるため $\tan \phi$ の式を $\theta$ で微分すると、 $G'(\theta) = f'(\theta)$ 、 $H'(\theta) = f''(\theta)$ であるから

$$\phi' \cdot \sec^2 \phi = \frac{f'(\theta)H(\theta) - f''(\theta)G(\theta)}{[H(\theta)]^2} \quad (2.22)$$

が得られる。 $\sec^2 x = 1 + \tan^2 x$ を用いて整理すると

$$\phi' = \frac{f'(\theta)H(\theta) - f''(\theta)G(\theta)}{[H(\theta)]^2 + [G(\theta)]^2} \quad (2.23)$$

となる。したがって、ローラ中心軌跡の曲率半径  $\rho_f$  は

$$\rho_f = \frac{s'}{\gamma'} = \frac{s'}{1 + \phi'} = \frac{\{ [H(\theta)]^2 + [G(\theta)]^2 \}^{\frac{3}{2}}}{[H(\theta)]^2 + [G(\theta)]^2 + f'(\theta)H(\theta) - f''(\theta)G(\theta)} \quad (2.24)$$

カム輪郭の曲率半径  $\rho_p$  は、従節との接点において、輪郭曲線とローラ中心軌跡が共通法線を持つことにより

$$\rho_p = \rho_f - r_r \quad (2.25)$$

として求められる。以上の諸式は、従節の変位  $f(\theta)$  とそれらの微係数  $f'(\theta)$ 、 $f''(\theta)$  が与えられると、 $\theta$  を微小角  $\Delta\theta$  ずつかえて行くことにより、計算機で容易に計算できる。

## 2.2.2 円端揺動従節板カム

揺動従節の場合には、カムの輪郭決定に寄与するのは従節レバーの回転角である。前節の記号にならって、カムの回転角  $\theta$  に対するレバーの位置  $f(\theta)$  ならびに  $f'(\theta)$ 、 $f''(\theta)$  は分っているものとする。ここでは  $f(\theta)$  は角度を表す。式を簡単に表現するため、カムの回転中心と従節レバーの回転中心を結ぶ線  $\overline{O_c O_l}$  (図 2.4) を基準にして、従節レバーの角度  $\beta(\theta)$  を定めるものとする。従節レバーの初期位置の角度を  $\beta_0$  とすれば

$$\beta(\theta) = \beta_0 + f(\theta) \quad (2.26)$$

図 2.4 は座標軸をカムに固定し、カムを回転される代りに従節レバー中心  $O_l$  を反時計まわりに回転させた状態を示している。図から明らかなように

$$C = c e^{j\theta}$$

$$B_2 = l e^{j(\theta + \beta)} \quad (2.27)$$

と書ける。ただし、 $c$  はカムの回転中心と従節レバーの回転中心との距離であり、 $l$  は従節レバーの長さであって、共に定数である。従節ローラの中心軌跡はこれらを用いて

$$P_f = C + B_2 \quad (2.28)$$

$$= c e^{j\theta} + l e^{j(\theta + \beta)}$$

と表される。いま、ベクトル  $P_f$  の大きさを  $R_f$ 、偏角を  $\theta + \theta_f$  とおくと

$$R_f e^{j\theta_f} = c + l e^{j\beta} \quad (2.29)$$

上式の右辺の実数部と虚数部に注目すれば、従節ローラ中心軌跡上の点  $P_f (R_f, \theta_f)$  は  $\theta$  を媒介変数として

$$R_f = \sqrt{a_f^2 + b_f^2}$$

$$\theta_f = \theta + \theta_f \quad (2.30a)$$

のように極座標表示される。

ただし、 $a_f = l \sin \beta$

$$b_f = c + l \cos \beta \quad (2.30b)$$

$$\tan \theta_f = \frac{a_f}{b_f}$$

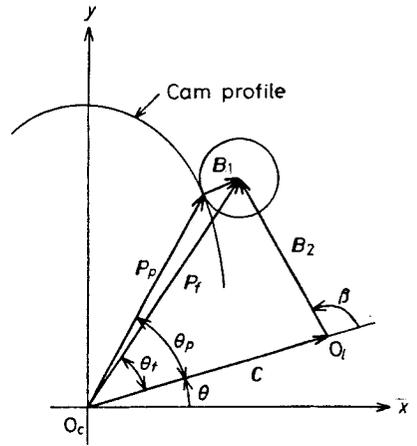


図 2.4 円端揺動従節板カム

カムの輪郭を求めるには色絡面条件式が必要である。このために接線の方向を求める。式(2.28)を $\theta$ で微分すると

$$\begin{aligned} P'_f &\equiv s' e^{j(\theta+\phi)} \\ &= j c e^{j\theta} + j l [1 + f'(\theta)] e^{j(\theta+\beta)} \end{aligned} \quad (2.31)$$

ここで $\beta'(\theta) = f'(\theta)$ を用いている。上式の両辺を $e^{j\theta}$ で割ると

$$\begin{aligned} s' &= \sqrt{[G(\theta)]^2 + [H(\theta)]^2} \\ \tan \phi &= \frac{G(\theta)}{H(\theta)} \end{aligned} \quad (2.32a)$$

が得られる。ただし

$$\begin{aligned} G(\theta) &= c + l [1 + f'(\theta)] \cos \beta(\theta) \\ H(\theta) &= -l [1 + f'(\theta)] \sin \beta(\theta) \end{aligned} \quad (2.32b)$$

法線ベクトルの偏角を $\theta + \phi$ とすると、図2.4においてカムが時計まわりに回転するとき、 $\phi = \phi - \pi/2$ であるから、

$$\tan \phi = \frac{l [1 + f'(\theta)] \sin \beta(\theta)}{c + l [1 + f'(\theta)] \cos \beta(\theta)} \quad (2.33)$$

で表される。ここで $\phi$ 、 $\psi$ 共に $O_c$ と $O_l$ を結ぶ直線からの角度である。

カムの輪郭曲線は、従節ローラとの接触点において、従節ローラの中心軌跡と共通法線を持つことから

$$\begin{aligned} P_p &= P_f - B_1 \\ &= c e^{j\theta} + l e^{j(\theta+\beta)} - r_r e^{j(\theta+\psi)} \end{aligned} \quad (2.34)$$

が成り立つ。 $r_r$ は従節ローラの半径である。ベクトル $P_p$ の大きさを $R_p$ 、偏角を $\theta + \theta_p$ とすると上式より

$$R_p e^{j\theta_p} = c + l e^{j\beta} - r_r e^{j\psi} \quad (2.35)$$

カムの輪郭曲線上の点 $P_p$  ( $R_p, \theta_p$ ) は $\theta$ を媒介変数とする極座標表示で次式のようになる。

$$\begin{aligned} R_p &= \sqrt{a_p^2 + b_p^2} \\ \theta_p &= \theta + \theta_p \end{aligned} \quad (2.36a)$$

ただし、 $a_p = l \sin \beta(\theta) - r_r \sin \psi$

$$b_p = c + l \cos \beta(\theta) - r_r \cos \psi \quad (2.36b)$$

$$\tan \theta_p = \frac{a_p}{b_p}$$

角度 $\psi$ は式(2.33)で与えられる。

半径 $r_t$ の工具で加工する場合の工具軌跡は、切削点においてカム輪郭曲線と共通法線を持つことから、第2.2.1節で導いた方法と同様に、式(2.36)における $r_r$ を $(r_r - r_t)$ で置きかえて得られる。

次に、式(2.32)を用いて従節のローラ中心軌跡の曲率半径 $\rho_f$ を求める。解析法は第2.2.1節で式(2.24)を導いたのと同様である。すなわち、式(2.32)の $G(\theta)$ 、 $H(\theta)$ を用いて

$$\rho_f = \frac{s'}{1 + \phi'} = \frac{\{[H(\theta)]^2 + [G(\theta)]^2\}^{\frac{3}{2}}}{[H(\theta)]^2 + [G(\theta)]^2 + G'(\theta)H(\theta) - G(\theta)H'(\theta)} \quad (2.37)$$

ここで、 $G'(\theta)$ と $H'(\theta)$ は次式で与えられる。

$$\begin{aligned} G'(\theta) &= l f''(\theta) \cos \beta - l [1 + f'(\theta)] f'(\theta) \sin \beta \\ H'(\theta) &= -l f''(\theta) \sin \beta - l [1 + f'(\theta)] f'(\theta) \cos \beta \end{aligned} \quad (2.38)$$

揺動従節の場合は、直動従節の場合と異なり、 $G'(\theta)$ 、 $H'(\theta)$ に $f'(\theta)$ 、 $f''(\theta)$ の他に角度 $\beta(\theta)$ を含むため、これらを式(2.37)に代入しても簡単な形にはならない。計算機を用いて評価することを考慮すると、式(2.37)は一般的に成り立つ式であるから、式(2.37)と式(2.38)は別個の関数またはサブルーチンとして用意しておく方が使い易い。

カムの輪郭曲線の曲率半径 $\rho_p$ は、従節のローラ中心軌跡と輪郭曲線がローラ接点において共通法線を持つことがこの場合にもいえるので、

$$\rho_p = \rho_f - r_r \quad (2.39)$$

によって与えられる。

圧力角 $\alpha$ は、ローラとカムの接点における共通法線の方向と従節レバーの直角方向(従節の運動方向)との差によって与えられる。図2.4の場合、従節を固定してカムが時計方向に回転すると考えると、これは向い側回転になっている。この場合、

$$\begin{aligned} \text{共通法線の方向は} & \quad \theta + \psi \\ \text{従節レバーの直角方向は} & \quad \theta + \beta - \frac{\pi}{2} \end{aligned}$$

で与えられるので、圧力角は

$$\alpha = \psi - \beta + \frac{\pi}{2} \quad (2.40)$$

として得られる。

## 2.3 設計パラメータの決定問題

### 2.3.1 従節変位曲線

第2.2節では、カムの回転角 $\theta$ に対する従節の変位 $f(\theta)$ とその導関数などが与えられたものとして、カムの輪郭の計算法を述べた。この $f(\theta)$ の形を定めるには、カムの回転角 $0^\circ \sim 360^\circ$ に対して、図2.5に例示するように、のぼり行程、くだり行程および停留からなる作動端運動線図を定める。のぼり行程とくだり行程については、カムを等速回転させたときの作動端の速度あるいは加速度を考慮した、従節変位曲線がいろいろ考案されている。<sup>1)</sup>ここでは、これらの中でしばしば使われる代表的なものをいくつか挙げる。従節変位曲線の定義に関しては、無次元化された時間 $T$ および変位 $S$ を用いる。同一形式の曲線でも、タイミング線図におけるその曲線の割り付け角や、変位のストロークの大きさはそれぞれのカム設計において具体的に定められる。変位曲線を定義する関数プログラムを、無次元化された量によって作成しておけば、割り付け角やストロークの変化は、適当なスケール処理を行うだけで済ませることができる。また、変位曲線の相互の特徴比較にも、無次元化しておく都合がよい。

カムによって動かされる従節の変位 $g$ は時間 $t$ の関数

$$g = g(t) \quad (2.41)$$

として図2.6のように与えられるものとする。このとき、無次元時間 $T$ および無次元変位 $S$ を次式で定義

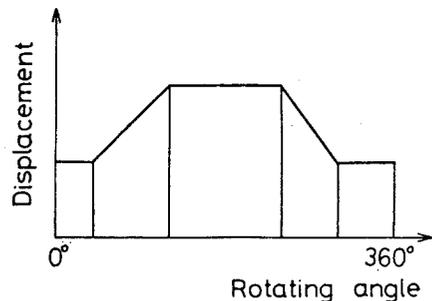


図2.5 カムのタイミング線図

する。

$$T = \frac{t}{t_h}, \quad S = \frac{g}{h} \quad (2.42)$$

ここで、 $t_h$ は最大変位に達するまでの時間であり、 $h$ はそのときの最大変位である。これらを用いると従節変位曲線は

$$S = S(T), \quad 0 \leq T \leq 1, \quad 0 \leq S \leq 1 \quad (2.43)$$

によって定義される。速度、加速度、躍動について無次元化された量として次式で定義する。

$$\text{無次元速度} \quad V = V(T) = \frac{dS}{dT}$$

$$\text{無次元加速度} \quad A = A(T) = \frac{d^2S}{dT^2} \quad (2.44)$$

$$\text{無次元躍動} \quad J = J(T) = \frac{d^3S}{dT^3}$$

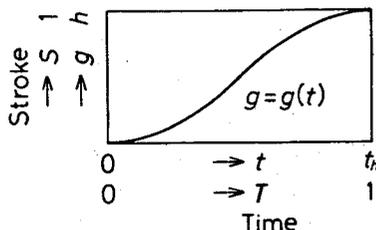


図2.6 カム曲線の無次元化表示

なお、とくにこれらの量の最大値を表すときには添字 $m$ をつけるものとする。以下で従節変位曲線の定義を与える場合、すべてのほり行程の形式で定義するが、くだけり行程ではこれらの時間軸を逆にして適用する。また、時期軸とカムの回転角の対応は、カム軸が等速で回転するものとし、時間間隔 $t_h$ に対する割り付け角を $\theta_h$ とすると、

$$\frac{\theta}{\theta_h} = \frac{t}{t_h} = T \quad (2.45)$$

である。

(1) サイクロイド曲線

サイクロイド曲線

$$S = T - \frac{1}{2\pi} \sin 2\pi T \quad (2.46)$$

は加速度曲線が1周期の正弦曲線で表される。

$$A = A_m \sin 2\pi T \quad (2.47)$$

$$A_m = 2\pi$$

この曲線は高速、軽荷重に適するが、 $A_m$ や $V_m$ がやや大きいという欠点を持つ。しかしその数学的性質が簡明であるため、一種の基本曲線として用いられている。

(2) 変形正弦曲線

変形正弦曲線は、重荷重に適用できるようにするため、速度曲線の $V_m$ の値が低くなるように考案された曲線である。このため、三つの区間に分けてその中央部の加速度曲線が周期の長い正弦曲線になるように工夫されている。加速度曲線の図に示した $T_a$

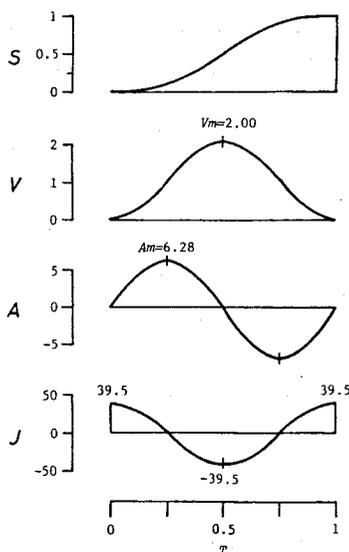


図2.7 サイクロイド曲線

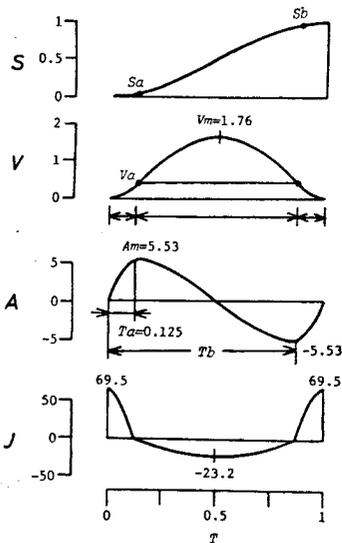


図 2.8 変形正弦曲線

の値は標準として  $\frac{1}{8}$  が取られる。とくに  $T_a = \frac{1}{4}$  のときは サイクロイド曲線となる。

曲線の式：

$$0 \leq T \leq T_a, \quad S = \frac{2T_a A_m}{\pi} T - \frac{4T_a^2 A_m}{\pi^2} \sin \frac{\pi T}{2T_a}$$

$$T_a < T \leq 1 - T_a, \quad S = \frac{(1 - 2T_a)^2 A_m}{\pi^2} \left\{ 1 - \cos \frac{\pi(T - T_a)}{1 - 2T_a} \right\} + V_a(T - T_a) + S_a$$

$$1 - T_a < T \leq 1, \quad S = \frac{4T_a^2 A_m}{\pi^2} \left\{ \cos \frac{\pi(T - 1 + T_a)}{2T_a} - 1 \right\} + V_a(T - 1 + T_a) + S_b$$

定数の式：

$$T_a = \frac{1}{8}$$

$$A_m = \frac{1}{\frac{2T_a}{\pi} + \frac{2 - 8T_a}{\pi^2}}$$

$$V_a = \frac{2T_a A_m}{\pi} \quad (2.49)$$

$$S_a = \frac{2T_a^2 A_m}{\pi} - \frac{4T_a^2 A_m}{\pi^2}$$

$$S_b = 1 - S_a$$

この曲線はサイクロイド曲線にくらべて、 $V_m$ 、 $A_m$  ともに低くおさえられており、適用範囲の広い曲線である。

(3) 変形等速度曲線

この曲線は、等速度部分が必要な場合か、とくに  $V_m$  を小さくしたい場合に使用される。緩衝曲線付きの等速度曲線と呼ぶべきものであり、 $A_m$  や  $J_m$  が大きいいため、高速には適さない。

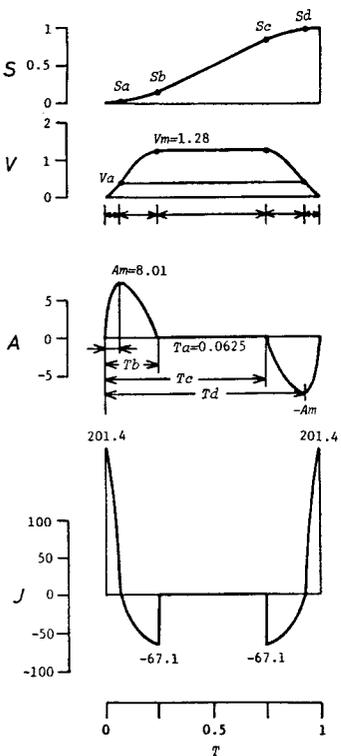


図 2.9 変形等速度曲線

曲線の式：

$$\begin{aligned}
 0 \leq T \leq T_a, \quad S &= \frac{2T_a A_m}{\pi} T - \frac{4T_a^2 A_m}{\pi^2} \sin \frac{\pi T}{2T_a} \\
 T_a < T \leq T_b, \quad S &= \frac{4(T_b - T_a)^2 A_m}{\pi^2} \left\{ 1 - \cos \frac{\pi(T - T_a)}{2(T_b - T_a)} \right\} \\
 &\quad + V_a (T - T_a) + S_a \\
 T_b < T \leq 1 - T_b, \quad S &= V_m (T - T_b) + S_b \\
 1 - T_b < T \leq 1 - T_a, \quad S &= \frac{4(T_b - T_a)^2 A_m}{\pi^2} \sin \frac{\pi(T - 1 + T_b)}{2(T_b - T_a)} \\
 &\quad - \frac{2(T_b - T_a) A_m}{\pi} (T - 1 + T_b) \\
 &\quad + V_m (T - 1 + T_b) + S_c \\
 1 - T_a < T \leq 1, \quad S &= \frac{4T_a^2 A_m}{\pi^2} \left\{ \cos \frac{\pi(T - 1 + T_a)}{2T_a} - 1 \right\} \\
 &\quad + V_a (T - 1 + T_a) + S_d
 \end{aligned} \tag{2.50}$$

定数の式：

$$\begin{aligned}
 T_a &= \frac{1}{16}, \quad T_b = \frac{1}{4} \\
 A_m &= \frac{1}{\frac{2}{\pi} \left\{ \left( 2 - \frac{8}{\pi} \right) T_a T_b + \left( \frac{4}{\pi} - 2 \right) T_b^2 + T_b \right\}} \\
 V_a &= \frac{2T_a A_m}{\pi}, \quad V_m = \frac{2T_b A_m}{\pi} \\
 S_a &= \frac{2T_a^2 A_m}{\pi} - \frac{4T_a^2 A_m}{\pi^2} \\
 S_b &= \frac{4(T_b - T_a)^2 A_m}{\pi^2} + V_a (T_b - T_a) + S_a \\
 S_c &= 1 - S_b, \quad S_d = 1 - S_a
 \end{aligned} \tag{2.51}$$

代表的な3種類のカム従節変位曲線を示したが、他にも多くの変位曲線が考案されている。<sup>1)</sup> 計算機で処理する場合、これら標準的な関数は関数副プログラムとして準備しておく。必要があれば使用者が自由に変位曲線をFORTRAN副プログラムとして作成することも可能である。カム・タイミング線図に合わせて、のほり行程とくだけり行程それぞれにこれらの曲線を割りつけ、時間軸(割り付け角)とストロークのスケールングを行えば、カム従節変位の全体を定める $f(\theta)$ が決まることになる。 $f(\theta)$ がカムの全回転角について決めれば、これを用いて第2.2節で求めた方法によって、カムの輪郭が計算される。このようにカムの輪郭計算を計算機で行う場合、基本になる副プログラムを用意しておけば、それらの組み合わせによって、個々のカムの輪郭計算が具体的に出来るという特徴がある。

### 2.3.2 圧力角ならびに曲率半径に関する条件

圧力角と曲率半径の求め方は第2.2節で述べたが、実際のカムの設計にあたっては、最大圧力角と最小

曲率半径を求め、パラメータ変更の試行錯誤を行って制約条件を満たすようにしなければならない。本節では、これらの点について一般的考察を述べる。

簡単な例として、図 2.10 に示した直進カムについて圧力角を考察する。この場合は運動曲線がそのままカムの形状を表し、カムの勾配角  $\phi$  が圧力角  $\alpha$  に一致する。いま、カムストロークを  $x_h$ 、従節のストロークを  $y_h$  とし、無次元化量による変位曲線を  $S = S(T)$  とすると、カムの形状  $y = y(x)$  は

$$\begin{aligned} x &= x_h T \\ y &= y_h S \end{aligned} \tag{2.52}$$

と書ける。このときの圧力角  $\alpha$  すなわち勾配角  $\phi$  は

$$\tan \alpha = \tan \phi = \frac{dy}{dx} = \frac{y_h}{x_h} \cdot \frac{dS}{dT} = \frac{y_h}{x_h} V \tag{2.53}$$

で与えられる。ただし、 $V$  は無次元速度である。

この例の場合、圧力角の最大値は容易に求められて

$$\tan \alpha_m = \frac{y_h}{x_h} V_m \tag{2.54}$$

であり、圧力角が最大になる点は、運動曲線が加速側と減速側で対称ならば  $x = x_h/2$  で与えられる。

$\alpha_m$  を  $30^\circ$  以下におさえるとすれば

$$\frac{x_h}{y_h} \geq \frac{V_m}{\tan 30^\circ} = \sqrt{3} V_m \tag{2.55}$$

の関係が得られる。 $V_m$  は前節で述べたように、変位曲線によって定められる。この関係から、必要な従節ストローク  $y_h$  に対して要求されるカムの大きさ  $x_h$  が求められる。

一般の板カムの場合でも、図 2.10 と類似の図を書くことができる。ただし、横軸にはカム輪郭に沿う弧の長さを取る。この場合でも上と同様の考え方ができるが、圧力角が最大になる点は行程の中央にはならず、カムの小径側に片寄った位置になる。ただし、一つの近似として次式

$$\tan \alpha_m \doteq \frac{h}{\theta_h \cdot r_p} V_m \tag{2.56}$$

$$r_p = \frac{r_1 + r_2}{2}$$

を使うことができる。<sup>1)</sup>ここで、 $r_1$  はのほり行程の最初のカム径、 $r_2$  はのほり行程の最後のカム径である。この式では、最大圧力角の位置を  $(r_1 + r_2)/2$  で近似しており、また弧の長さも  $\theta_h \cdot r_p$  で近似している。なお、図 2.10 は先端従節を用いた図であり、ローラつきの場合の実際のカム径  $r_1, r_2$  はローラ半径に相当する分だけ小さくなる。また、揺動従節では従節の運動方向が変化するため、近似の誤差はさらに増えるが、一つの目安としてやはり式 (2.56) は有効である。式 (2.89) から、圧力角を小さく

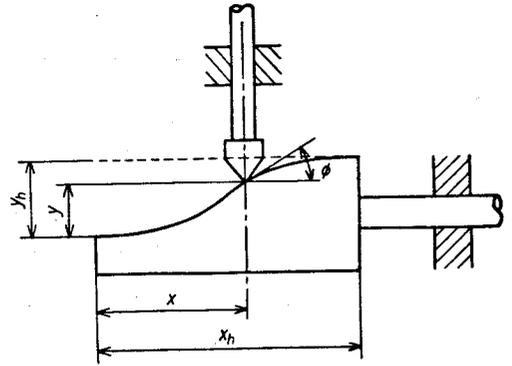


図 2.10 直動先端従節直進カム

おさえるには行程の割付角  $\theta_h$  を大きくするか、カム径を大きくすればよいことが分る。

第 2.2 節で述べたカムの輪郭計算において、カム基円半径に関する  $r_0$  または揺動の場合の基円に対するレバー角度  $\beta_0$  には自由度がある。したがって式 (2.56) などを用いて、圧力角の制約条件を満たすようにカムの大きさを決定する。当然のことながら、カム径が大きい程圧力角は小さくなる。通常、ある程度の余裕を持たせてカムの大きさを決めるが、カムを大きくすることにも制約がある。厳密さが要求されるときには、与えられたパラメータで、試行錯誤をくり返しなが最大圧力角を求める（圧力角の計算については第 2.2 節を参照）。しかし、極値を求めるのにそれ程高い計算精度が要求されないので、実際には近似式 (2.56) の近傍で直接探索法 (direct search method)<sup>3)</sup> を用いればよい。

円端従節を用いる場合、カムの形状が凸である部分で曲率半径があまり小さいことは好ましくない。この場合、カムの曲率半径を少くともローラ半径  $r_r$  以上にするのが普通である。また、凹形状の部分でも曲率半径がローラ半径  $r_r$  よりも小さければ（絶対値において）、浮上り現象を起こす。ただし、第 2.2 節で解析したように、ローラの中心軌跡を先に求め、これに基づいてカムの輪郭を求める方法では浮上り現象を起こすことはない。

また、平端従節を用いる場合には、カムの全周にわたって凸形状でなければならない。通常、曲率半径に関するこのような制約が満たされない場合には、同一の従節ストロークに対する割付け角を大きくしている。このような試行錯誤によるパラメータの決定には、時分割システムの端局を用いるのが効果的である。

上述した手順の他に、Serman<sup>4)</sup> はパラメータの修正を行いながら、制約条件を満たす最小のカムを求めるプログラムを紹介している。また、Wambold<sup>5)</sup> はカム設計のプログラムをより大きな応用プログラム・システム ICES (Integrated Civil Engineering System) に組み込んで、より使い易い形式でプログラムを整備した研究を報告している。さらに、カム設計一般については従節変位を多項式で表して、その次数や係数を数理計画法を用いて定める設計法<sup>6)</sup> や、差分を用いる設計法<sup>7)</sup> など特殊な設計法もある。本章では、一般性のある汎用のカム設計プログラムの開発を目標としているので、標準的な設計法を述べるに止め、特殊な解法には立ち入らないことにする。

図 2.12 に円端直動従節板カムの輪郭例を示す。図 2.11 がタイミング線図である。図 2.12 において、破線は工具軌跡を表す。タイミング線図ののぼり行程、くんだり行程の変位曲線として、(a)ではサイクロイド曲線、(b)では変形正弦曲線、(c)では変形等速度曲線を用いた場合の例が示されている。

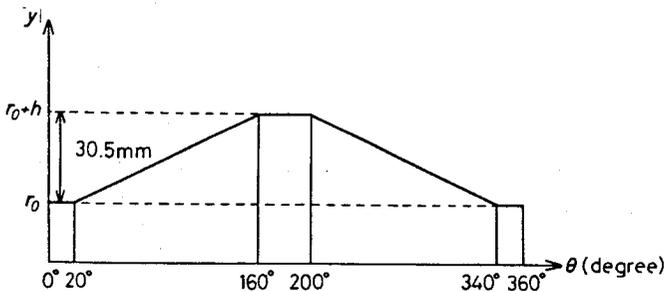
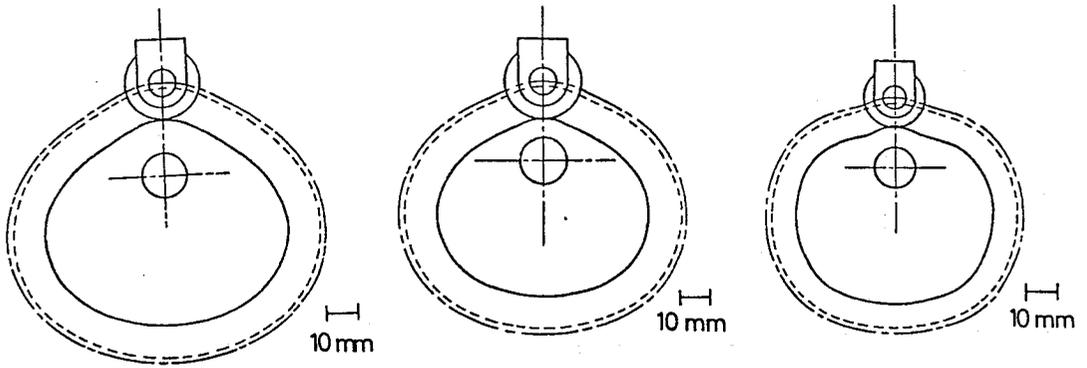


図 2.11 タイミング線図の具体例



(a) サイクロイド曲線使用  
 $(r_o=30\text{mm}, r_r=12\text{mm}, r_t=10\text{mm})$

(b) 変形正弦曲線使用  
 $(r_o=25\text{mm}, r_r=12\text{mm}, r_t=10\text{mm})$

(c) 変形等速度曲線使用  
 $(r_o=22\text{mm}, r_r=10\text{mm}, r_t=8\text{mm})$

図 2.12 板カム輪郭例

## 2.4 NC 指令の生成

### 2.4.1 板カム加工のための数値制御の方式

NC 指令の形式は NC 工作機械に依存して定められる。ところで、板カム加工の場合、被削材を回転テーブルに取り付け、工具を直進一軸（X 軸）の運動軸に取りつけて、工具に従節変位の動きを与えれば板カムの輪郭切削が可能であるように考えられる。この考え方は原理的に妥当なものであるが、これを厳密に実現するためには、極めて特殊な専用の制御装置が必要である。この原理を忠実に実行して滑らかな送り速度を与えるためには、回転テーブルの軸と工具の直進運動軸の 2 軸を同時に制御しなければならない。すなわち、カムの回転角と従節変位の関係に基づいて、回転軸と直進軸にパルス分配を行わなければならない。このような特殊なパルス分配機能も CNC (Computer NC) 装置を用いれば実現不可能ではないが、1 パルス単位の分配の速度に計算時間が追いつくかどうか疑問である。実際に切削している時間よりも、被削材の表面を工具が滑っている時間の方が多いということになりかねない。

実際に板カムを加工する場合には回転テーブルを用いることが多い。しかし、この場合板カム輪郭（または工具軌跡）をいくつかの小さいブロックに区切り、一つのブロックについてはカム輪郭をアルキメデス曲線で近似する。すなわち、1 ブロックの加工については、回転テーブルは等速回転、工具についても等速運動を与えるのである。板カムの輪郭をアルキメデス曲線の連鎖で表すには、後に述べる円弧補間の場合と同様に、輪郭とアルキメデス曲線とのずれが許容公差内に入るように、細かいブロックに分割して行くのである。

直交する  $x, y$  運動軸を用いて加工する場合、直線補間を用いる方法と円弧補間を用いる方法と 2 通りの方法がある。直線

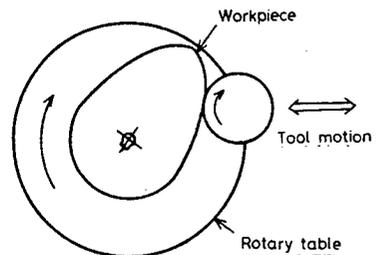


図 2.13 回転テーブルによる板カム加工

補間を用いた場合、たとえ許容公差内で最適化を図っても<sup>8)</sup>、板カムのような場合には円弧補間に比べてもとの軌跡への適合性の面で不利である。もしも円弧補間が利用できるとすればその利点は大きい。まず従節停留部のカム輪郭は円弧であり、一つのブロックとして加工できる。さらに重要なことは、円弧補間を用いると板カム輪郭の法線方向が、各ブロックの区切りでNCテープ・イメージにおいても保存されていることになり、工具径補正を用いるような場合に精度良く加工時の工具径路を定めることができるのである。

### 2.4.2 円弧補間の利用

第2.2節で述べたように、工具軌跡はカムの回転角 $\theta$ をパラメータとして表現される。そこで、適当な微小角 $\Delta\theta$ ごとに工具軌跡の座標を計算すると、工具軌跡が点列で表現される。また、板カム設計では各点での接線方向が求められるので、この接線方向を利用すると工具軌跡の2点間を二つの円弧で滑らかにつなぐことができる。

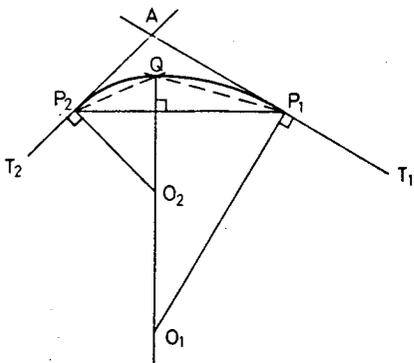


図 2.1.4 内心点による円弧の接続

この様子を図示したのが図2.1.4で、 $P_1$ と $P_2$ は工具軌跡上の点であり、 $T_1$ と $T_2$ はそれぞれの点における接線である。点 $Q$ は $\triangle P_1 P_2 A$ の内心であり、 $\angle AP_1 P_2$ の2等分線と $\angle AP_2 P_1$ の2等分線の交点である。そして、 $P_1 Q$ および $QP_2$ を円弧補間するときの円弧の中心をそれぞれ $O_1$ および $O_2$ とする。 $P_1, P_2$ の座標と $T_1, T_2$ の傾きが分かれば $O_1, O_2$ の座標が求められ二つの円弧が決まる。もちろん、これらの円弧による輪郭は正しい工具軌跡に一致しないので、許容公差の吟味が必要である。

これについての効果的な方法はないので、 $P_1$ から $P_2$ に至る工具軌跡上の何点かを取り、これとカムの回転中心とを結ぶ線が円弧と交わる点を求め、それらの点の正しい工具軌跡からのずれを調べる。少なくとも内心点近傍についてのこの検査は不可欠のものであろう。当然のことながら、 $\Delta\theta$ を小さくすると $P_1$ と $P_2$ の間隔を狭くすれば精度はよくなる。また、NC指令のデータ量を少なくするためには、許容公差の範囲内なるべく長い円弧で工具軌跡を近似することが望ましい。

円弧補間を用いる場合、工具軌跡が凸形状から凹形状へ、また凹形状から凸形状に変わる変曲点を円弧の途中に含むことは好ましくない。そこで変曲点の位置は明確に求めておいて、ここに円弧の区切りをおくようにする。変曲点を求めるには曲率半径の符号が反転する箇所を「はさみうち法」によって求める。図2.1.5は変曲点 $P_2$ における円弧の接続方法を図示したものである。工具半径が切削加工時にしか決まらないときには、カム輪郭曲線を同様の手順で円弧列として表現する。工具径路は各々の円弧について工具径オフセットを行って定める。実際に第2編で工具径路の実時間制御を行うときにこの方法を用いている。上に述べた計算手順どおりの数値計算を計算機で実行するとき、桁落ちの起り易い場合がある。例えば、

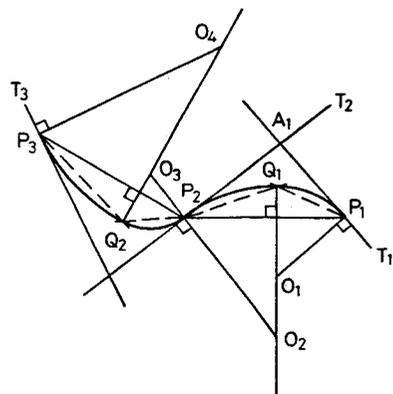


図2.1.5 変曲点における円弧の接続

図 2.14 において内心点 Q を求めるとき、カムの回転角のきざみ  $\Delta\theta$  が十分小さいと、接線  $T_1$  と  $T_2$  の角度差が小さくなる。そのため点  $P_1, P_2$  における角度の 2 等分線は平行に近くなり、交点 Q の座標が正確に求められなくなる。この場合二つの直線の偏角の差を十分な有効桁数で求めることが必要である。

第 2.2 節で接線の方法を求める際に、カムの基準線からの偏角として  $\theta + \phi$  の表現を用いた。したがって、 $T_1$  の偏角を  $(\theta_1 + \phi_1)$ 、 $T_2$  の偏角を  $(\theta_2 + \phi_2)$  とすると、 $\overrightarrow{P_1Q}$  と  $\overrightarrow{QP_2}$  の偏角の差は  $\frac{1}{2} \cdot \Delta\theta + \frac{1}{2} (\phi_2 - \phi_1)$  である。 $\Delta\theta$  は正確な値を与えることができるので、 $(\phi_2 - \phi_1)$  については与えられた従節変位（ならびにその導関数）を用いて、倍精度で計算しておくことが必要である。

NC テープの作成には、輪郭切削の指令の他に主軸回転数、送り速度などの指令が必要である。これらは輪郭計算とは独立のものであるが NC テープの作成という観点から見れば一体として処理されるべきものである。本章に述べた板カム設計計算の応用プログラムは、具体的には大阪大学大型計算機センターを利用して開発されており、ここではファイル・システムが有効に使える。輪郭切削に関する指令は検証のために一度ファイル・システムに出力され、ファイル・エディタを用いて加工条件に関する指令を付加する。したがってまた、加工条件の変更も実験室におかれた端局から容易に行える。NC テープの出力もこの端局を用いて行われるが、オンライン DNC システム（第 4 章参照）を用いれば、ファイルの内容を紙テープに出力することなく加工することも可能である。

## 2.5 結 言

マスタカムを試作するような場合、設計の過程と加工の過程が密着したシステムが望まれる。このような目的のために、板カムの設計加工システムの開発を行った。このシステムの特徴は以下のとおりである。

- 1) 板カムの輪郭計算から NC テープが直接生成されるので設計と加工を分離して考える必要がない。
- 2) 輪郭計算に必要な標準のサブルーチンと標準の従節変位曲線の関数プログラムがライブラリとして用意されており、使用者に要求される入力が必要最小限の設計諸元だけでよい。
- 3) 圧力角や曲率半径も求められるようになっており、設計パラメータの変更による試行錯誤が容易に実行できる。
- 4) NC 指令の生成には円弧補間が有効に使われているので、工具径補正などを用いる高精度加工に適している。
- 5) 大型計算機の TSS とオンライン DNC システムとはファイルを共用しているため、設計変更、加工条件の変更が容易に行える。

具体例として対象を板カムに限定したが、設計過程から切削加工に必要な NC 指令が自動的に作り出され、工具動作の手順を逐一書き下す必要がないシステムの構成が可能であることを示した。ただし、材料ファイル、工具ファイル、加工ファイルなどはまだ整備していないので、加工条件などはマニュアルで挿入しなければならない現状となっている。

板カムの設計問題については、ここでは基本的で標準的な解法のみを述べているが、特殊な設計法、制約条件や所望の従節変位の下で最適解を見出すことなど残された問題点も多い。

## 参 考 文 献

- 1) 牧野 洋 : 自動機械機構学、日刊工業新聞社、(1976)。
- 2) H.A.Rothbart : CAMS-Design, Dynamics and Accuracy, John Wiley & Sons, Inc.,(1956)。
- 3) J.Kowalik and M.R.Osborne : Methods for unconstrained Optimization Problems,American Elsevier Publishing Company, Inc.,(1968)。
- 4) C.F.Sermon and A.Liniecki : Search for Optimum Solution of a Single Disk Cam Mechanism with an Oscillating Roller Follower, ASME,paper (72-Mech-61) presented at the Mechanism Conference,(1972), pp.1-11.
- 5) J.C.Wambold and C.Rutigliano : Optimization of a Disk Cam Using a Problem-Oriented Language, ASME,paper (73-DET-59) presented at the Design Engineering Technical Conference,(1973), pp.1-10.
- 6) N.T.Thomopoulos and T.W.Knowles : Use of Linear Programming for Cam Design,Int. J.Mach. Tool Des. Res., Vol.15 (1975), pp.257-265.
- 7) F.Y.Chen : A Refined Algorithm for Finite Difference Synthesis of Cam Profiles, Mechanism and Machine Theory, Vol.7 (1972), pp.453-460.
- 8) S.Makinouchi, M.Okamoto and K.Yamagata : Optimal Curve Fitting for NC Machining by Dynamic Programming,Tech.Pepts. Osaka Univ., Vol.26 (1976), pp.255-266.
- 9) 研野和人 : 自動設計法、コロナ社、(1971)。

## 第3章 実時間処理プログラムの構成

### 3.1 緒 言

実際の加工工程において、計算機による制御や監視を行うためには、実時間処理プログラムが必要である。実時間処理プログラムには通常の計算処理のプログラムと異なった性質が要求される。本章では必要に応じて両者の場合を比較対照しながら実時間処理プログラムの構成法を論じる。

実時間処理の本質的な点は、計算機の外部で発生した事象に伴って制御プログラムが起動され、再び外部に応答を返すまでの時間に強い制約があることである。また、計算機処理を必要とする事象が逐次的に発生するとは限らず、一つの事象に対する処理途中において、より緊急を要する事象が発生するかも知れない。したがって、計算機内部では優先度付きの擬似並行処理が不可欠である。また、二つ以上のファンクション・プログラムの中で同期をとったり、互に情報を交換しながら処理を進める形式 (cooperation) も要求される。

制御用コンピュータとしてここではミニコンピュータを考えるものとする。実験用システムにおいてはミニコンピュータを実際に使用しており、また一般的にもミニコンピュータが制御用に使われることが極めて多い。この点を考慮して、大型計算機で一般的に使われているタスク管理とイベント管理の手法をここでは採用しない。上に述べた実時間処理に要求される機能を“t-unitのストリーム”という新しい概念を導入して遂行する。このことによって柔軟な処理形式が簡潔に実現できるのである。<sup>1),2)</sup> また、広範な使用環境に適応できるように、一部のプログラム・モジュールの差し換えが容易に行える。<sup>3)</sup>

### 3.2 局所の変数と非局所の変数

比較対照のため、簡単なFORTRANプログラムの場合についてまず述べる。FORTRANのプログラム

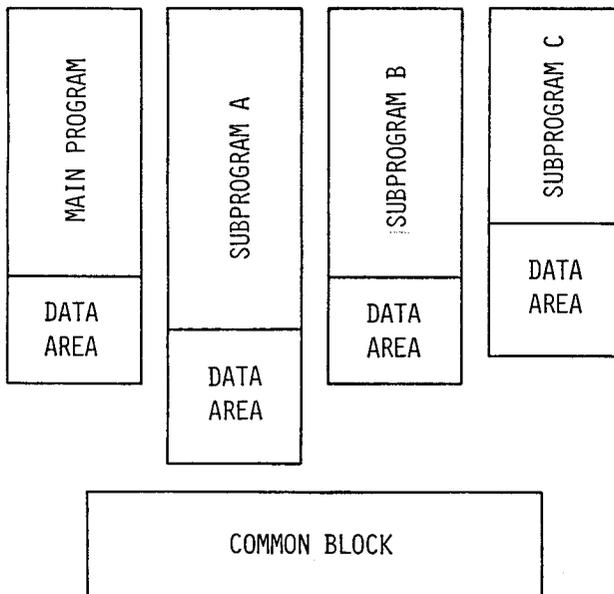


図 3.1 FORTRANの変数領域

は通常一つの主プログラムと複数個の副プログラムで構成され、各プログラムはそれぞれの内部だけで使用される局所の変数を持っている。一つの副プログラムの局所の変数領域をそのまま他の副プログラムが参照することはできないので、必要があれば引数を用いてデータの受渡しを行う。これに対して、非局所の変数領域としての共通ブロックを定義することができる。この領域は主プログラム、副プログラムを問わずどのプログラムからでも共通に参照できる。以上のようにFORTRANでは局所の変数と非局所の変数の定義が明瞭で簡潔である(図3.1)。<sup>4)</sup>

もう少し複雑な場合としてALGOL 60<sup>5)</sup>の場合を考える。ALGOLにはブロック構造の特徴があり、図3.2に示すように入れ子の形式で手続き(procedure)の宣言ができる。

ここでbeginとendで囲まれた部分が各手続きの本体である。ただし引数に関係のある記述は省略してある。また実際にはプログラムの全体もbeginとendで囲まれたブロックである。例えば手続きBに注目すると、変数yはBに局所的な変数であるが、その外側のブロックで定義された変数xは非局所的な変数であってこの値も手続きBで参照することができる。ただし、手続きBの内側のブロックの変数uやvは手続きBでは参照できない。同様に手続きCでは変数uが局所変数であり、xとyが非局所変数として参照できる。変数vは参照できない。一般にブロックの外側から内側の変数は参照できないのである。

以上は手続きの宣言の形式について述べたが、次に局所の変数や非局所の変数が実行時にどのように扱われるかを述べる。

ALGOLの手続きはその呼出しの文(または呼出しを含む文)によってその手続きが実行される。ところでALGOLでは再帰的呼出しが許されており、手続きの呼出しをくり返すうちにまだ実行途中の手続きが再び呼出されることが起る。この場合、同一の手続きについて2組の局所変数領域が作られることになる。したがって、一般的にいってすべての手続きについて、局所変数領域が実行時には動的に複数個存在し得ると考えておかなければならない。

上述したように、手続きCの実行中に手続きAやBの変数を非局所変数として参照できる(図3.2参照)。このとき手続きAやBの変数領域も動的に扱われることに注意を要する。このように実行時に変数領域の位置がかわる場合は、ベース・レジスタにその先頭アドレスを入れ、個々の変数はそれからの相対アドレスで表す。問題は各ベース・レジスタの内容を矛盾のないように管理することである。図3.2の手続きCでは入れ子の深さが3であるから3個のベース・レジスタを必要とする。一つの手続きについて複数個の変数領域がある場合、現在の処理に必要なアドレスだけがベース・レジスタに格納される。したがって、手続きの呼出しや元のプログラムへの復帰の際に、必要があればベース・レジスタの内容の変更を行う。以下にその手順を述べる。<sup>6)</sup>

手続きの呼出しにも一定の規則があつて<sup>5)</sup>、例えば図3.2の手続きCを呼び出せるのは手続きBの内部に限られる。この手続き呼出しのとき、手続きAとBに対応するベース・レジスタの内容は不変でよい。一般的に入れ子の深さiの手続きPの実行中に、入れ子の深さjの手続きQを呼び出すものとする。この

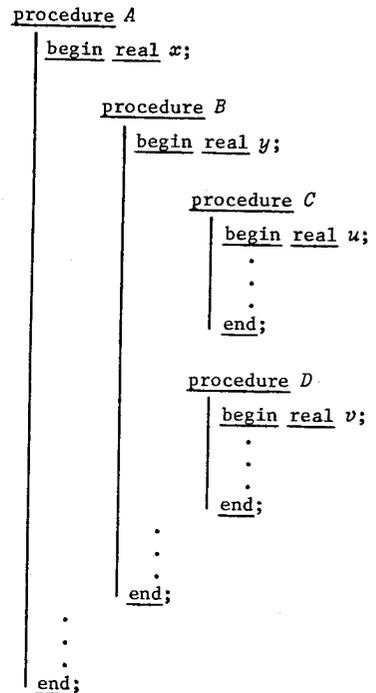


図3.2 ALGOL 60における  
手続きの宣言

とき  $j \leq i + 1$  の制約条件がある。また手続き  $P$  を実行中に入れ子の深さに対応してベース・レジスタに付与されていたアドレス(変数領域へのポインター)の組  $(r_1, r_2, \dots, r_i)$  の中で  $(r_1, r_2, \dots, r_{j-1})$  は不変でよい。ここで新しく手続き  $Q$  のための変数領域を用意し、そのアドレス  $r'_j$  を  $j$  番目のベース・レジスタに入れて  $(r_1, r_2, \dots, r'_j)$  とする。これで手続き  $Q$  の実行が可能となる。ここで手続き  $Q$  の変数領域に  $r_{j-1}$  を保存しておき、ベース・レジスタの内容が変わっても、後で  $r'_j \rightarrow r_{j-1} \rightarrow \dots \rightarrow r_1$  と順に復元できるようにチェーンを構成しておく。

手続きの実行終了後  $(r_1, r_2, \dots, r_i)$  を復元できるためには、 $i$  と  $r_i$  を保存しておけばよい。実行終了後、上に述べたチェーンをたどって復元することができる。変数領域をスタックの形式で順に確保して行くためには、ある手続きの変数領域を確保した後、次のアドレスを手続き(ブロック)ごとに保存しておく都合がよい。<sup>7)</sup> ALGOL では go to 文によって手続きから外のブロックに出ることが許されているが、このとき抜け出したブロックについては変数領域は不要になり、次の空き領域へのポインターについて操作が必要になる。この点各手続き(ブロック)ごとに次の空き領域のアドレスが記録されていると、ポインター操作を省略しても次の手続き呼出しでどこに変数領域をとるかが常に明らかになっているのである。なお、go to 文で手続きから出る場合は、ベース・レジスタの内容は不変でよい。

以上のように ALGOL 60 では、手続きとその局所の変数領域は静的に 1 対 1 の対応がつかず、実行処理中に動的割付けを行ってそれをポインターやチェーンの操作で管理して行く必要がある(図 3.3)。

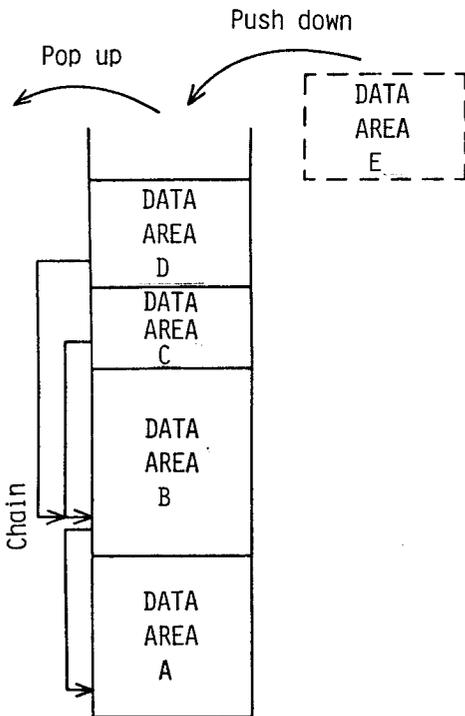


図 3.3 ALGOL 60 における変数領域管理

図 3.3 には変数領域割り当てのプッシュ・ダウン・スタックが示してある。新しいブロックに処理の流れが入るときにはスタックの最上部に変数領域が割り当てられる(Push down)。また、ブロックから処理の流れが出るときには、スタックの最上部から領域が解放される(Pop up)。ブロック構造を持つプログラミング言語では、この後入れ先出しの変数領域管理法は基本的な手法である。

実時間処理の制御プログラムでも、各ファンクション・プログラムについて局所の変数領域を一つ用意するだけでは十分でなく、やはり動的な領域割付けが必要になることがある。これを再入可能形式(reentrant)と呼び、例えば次のような場合が挙げられる。いま、あるファンクション・プログラム  $F$  が計算機内部で実行中であつたとする。このとき外部に緊急度の高い事象が発生して、それまでの処理を中断し、その事象に対応する処理に入ったとする。そこでもしも後で始められた処理が、ファンクション・プログラム  $F$  を使うとすれば、それまで実行中であつた  $F$  の中間結果すなわち局所の変数領域を保存しておいて、新しく変数領域

を用意しなければならない。

一般的にこのような場合を扱うとすれば、実時間処理においても実行時における動的割付けが必要にな

る。しかしながら、小型の計算機を制御用に用いる場合はベース・レジスタの機能などが強力でなく、再入可能形式のプログラムを作ると非常に効率が悪くなることもある。

実時間処理ではしばしばデータをキュー（queue）の形式で取り扱う。いま代表的な例として、いろいろなメッセージに対する処理を行う場合を考える。このとき、前述したように、同一のファンクション・プログラムの処理を受けるメッセージが複数個重複して存在することはあり得る。これらを図 3.4 に示すようなメッセージ・キューとして処理する。

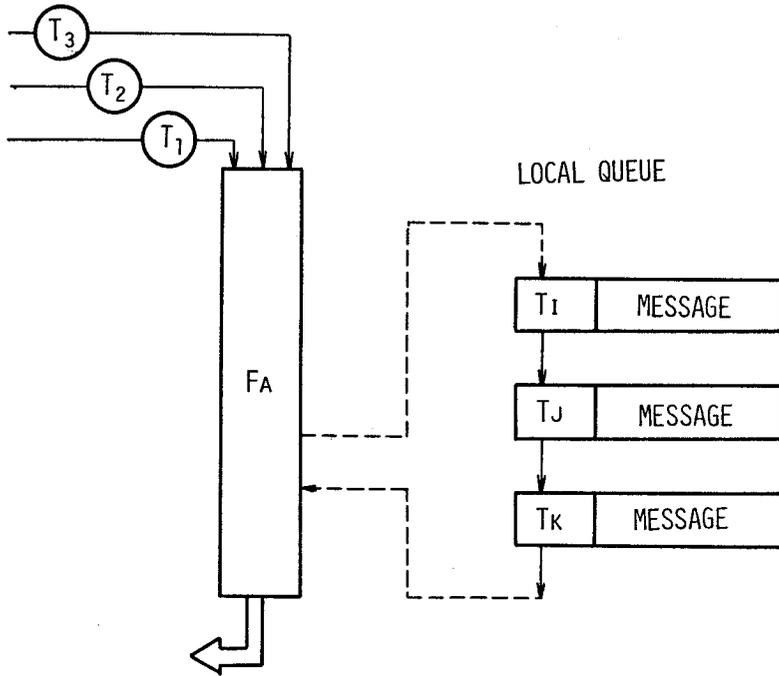


図 3.4 メッセージ・キュー

各ファンクション・プログラムは再入可能形式として作成される代りにメッセージ・キューを処理するように作成される。緊急度の高いメッセージはキューを操作して先に処理される。この方法では一つのメッセージに対する処理は中断されないが、処理の順序によって優先処理を実現する。再入可能形式に比べて優先処理の厳密さは欠けるが、プログラムの作り易さから再入可能形式は禁示して、キュー操作を行うようにすることが多い。

実時間処理のもう一つの特徴として、各ファンクション・プログラムの局所の変数よりもここに述べた広義のメッセージの持つ役割割りが重要である。実際の所、実時間処理の流れを把握するのに、これらメッセージの動きに注目すると分かり易い。この点、内部論理に従う計算処理で、プロセッサの動きが中心となるのと本質的に異なる面がある。

### 3.3 擬似並行処理

#### 3.3.1 論理的モデル

実時間処理の計算機では、外部に発生した事象に伴い割込みによってファンクション・プログラムが起

動されるので、たとえハードウェアのプロセッサ (CPU) が一つであっても、見掛け上複数個のプログラムが走行状態にある。また、事象相互の間に関連がある場合にはプログラム相互の間で信号の交換を行いながら処理を進めたり、必要に応じて同期を取ることも要求される。これらの要求を一括して処理する一般的方法としてタスク管理、イベント管理の手法が採られている。<sup>8)</sup>

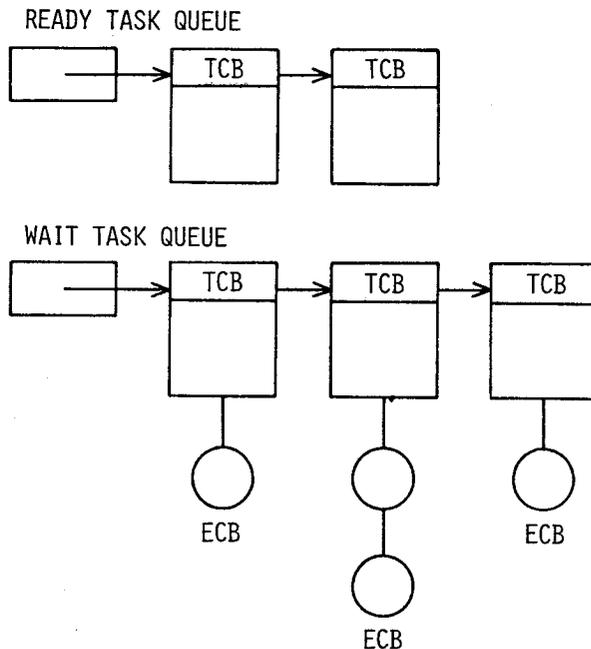
タスク管理を行う方法では、プログラムの実行による仕事をタスクという概念で統一的に扱い、プロセッサが一つであってもタスクは複数個存在して構わないものとする。起動状態にあるタスクはさらに次の三つの状態を持つ。

- (i) 実行状態 (CPUによる実行状態)
- (ii) レディ状態 (CPU待ちの状態)
- (iii) 待ち状態 (何らかのイベント待ちの状態)

これらの状態はタスク・コントロール・ブロックによって管理され、レディ状態のタスクはキューを作ってCPUの割当てを待つ。待ち状態からレディ状態への移行はイベント管理によって行われる。

イベント管理ではイベント・コントロール・ブロックを用意し、どのタスクがどのイベントを待っているかが明らかになるようにタスク・コントロール・ブロックとイベント・コントロール・ブロックをチェーン状に結合しておく。そして何らかのイベントが発生するとそれを待っているタスクを探し出して、レディ状態に移行させる (図 3.5)。

タスク、イベントという概念を用いる管理法は、一般的で柔軟性に富むが、小型の制御用計算機では繁



TCB : Task Control Block  
 ECB : Event Control Block

図 3.5 タスク管理の待ち行列

雑で実現しにくい。また別の手法として、メッセージ・スイッチングを主として行う場合には、トランザクションのキューだけに注目する方法もあるが、この方法ではプログラム間の交信 ( cooperation ) が実現しにくいという欠点がある。ここでは中間的な手法として、タスク管理よりは簡便で、プログラム間の交信も容易に行えるようなモデルを導入する。

いま、計算機の内部、外部を通じて発生するいろいろなイベントと、それらに結びついた処理要求のプログラム間での遷移に注目し、これを“ t - unit ” ( transition unit ) という概念でモデル化する。t - unit は 10~16 バイトのメモリを占有し、コントロール・レジスタあるいはメール・ボックスとしての役割を持つが、CPUとも間接的に結びついている。ここでは、タスク管理の概念は使用せず、ファンクション・プログラムの中を t - unit が通り抜けるストリーム ( stream ) によって、データ処理を定義する ( 図 3.6 )。このように抽象的なモデルを設定するのは、処理形態の柔軟性と、実際の CPU の割り当てを簡潔に行うことの二面性を両立させるためである。

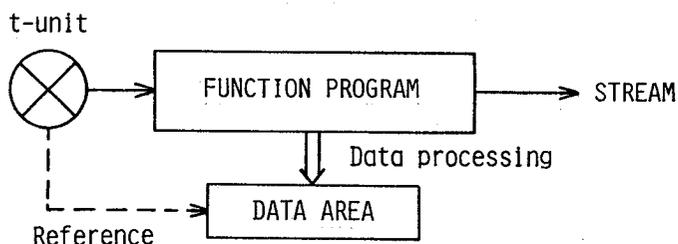


図 3.6 t - unit のストリーム

一つのファンクション・プログラムに注目するとき、一般的には複数個の t - unit が出入りすることが許される。このことは、複数個のイベント情報を受け取り、論理判断を行った後でデータ処理に入る ( または再開する ) 場合、あるいは、ファンクション・プログラムの内部にバッファ・キューを抱え込んでいるときなどに効果を発揮する。プログラムが厳密な意味ではリエントラント形式になっていなくても、t - unit のストリームは同時に複数個存在してかまわないのである。

結局、システム全体がイベント処理指向型で構成され、ファンクション・プログラム相互の擬似並行処理やコオペレーションも、ストリームの分流と合流、そして t - unit によるイベント情報の交信によって遂行される。ユーザー側はこれの使い方如何で、平易な形式の逐次処理のプログラムから、バッファ・キューを含む複雑なプログラムまで、その作成法に自由度が与えられ、モニター側は煩雑な同期操作の一部をユーザー側にまかせることによって、それ自身は簡潔になる。マクロ命令の実行によって、計算機内部では何らかの処理要求、またはイベントの通知を伴った t - unit が常時複数個動きまわることになる。この様子を図 3.7 に示す。モニターの核としてディスパッチャがあるが、ユーザー側から見ると、これは、t - unit の分配機能を持つもので、CPU の割り当ては暗に行われるに過ぎない。この中には t - unit の待ち行列があって、基本的なスケジューリング機能を持ち合わせている。

また、小型計算機によるリアルタイム処理では、割り込み処理の占める重みが大いなので、割り込み処理プログラムを一般のファンクション・プログラムと同様に扱い、自由に差し換えができるようにする。このため、インタラプト・アナライザを独立させて、これをもう一つのディスパッチャとして扱う。ここには t - unit のキューはないが、かわりに周辺装置のインターフェースがあって、これが処理要求を発生し、イベント情報やデータの入出力もこれを介して行われる。逆の見方をすれば、t - unit はこれと

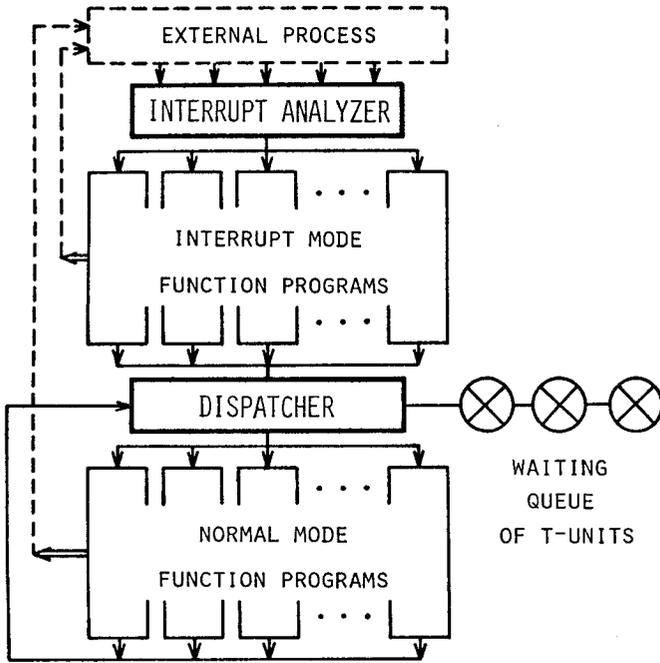


図 3.7 リアルタイム処理の論理モデル

求コード、7~10は発信者に返す応答とそのもどり場所を指す。これらに先頭の3バイトを加えた10個のバイトはモニターが関与する部分で、それ以降の領域はファンクション・プログラムごとに適宜使用される。メッセージをこの直後に続けることにすれば、バイト番号1、2のポインター・エリアを用いてメッセージ・キューを構成することが可能である。なお、第3バイト目にはマクロ命令に応じたコードが書き込まれる。

リアルタイム処理の特徴から、ファンクション・プログラム相互の結合関係は、実行前には決まっていない。実際に何らかの処理要求が発生した時点で、t-unitによってデータ領域に対するポインターが渡され、共通の参照領域が作られる。

t-unitのプログラム間の遷移は、すべて

等価なプログラム間のインターフェースの役割りを果たす。

このように考えると、図3.7に示したモデルでは、計算機の外部と内部、またハードウェアとソフトウェアの接合が滑らかである。たとえば、一つのファンクション・プログラムの機能をマイクロ・プロセッサで実現して、これをハードウェア・インターフェースに移行させるようなシステム変更が自然な形で処理される。

### 3.3.2 t-unitによるプログラム間の動的結合

すでに述べたように、t-unitとして物理的にはメモリを利用する。その内容の具体例を図3.8に示す。バイト番号<sup>注)</sup>4~6は受信者の宛名と要

BYTE NO.

1,2  
3,4  
5,6  
7,8  
9,10  
11,12  
13,14  
15,16

1,2			POINTER AREA FOR CHAINING			
3,4		T-UNIT STATUS	P <sub>1</sub>	PROGRAM NO.		
5,6						ORDER CODE TO RECEIVER
7,8		RESPONSE CODE	P <sub>2</sub>	PROGRAM NO.		
9,10	I	RETURN POINT TO SENDER				
11,12						(DATA AREA START ADDRESS)
13,14						(DATA AREA END ADDRESS)
15,16						(ADDRESS COUNTER OF DATA AREA)

P<sub>1</sub>, P<sub>2</sub> : priority for CPU assignment (2 bits)  
I : return inhibit flag (1 bit)

図 3.8 NEAC-M4におけるt-unit

注) NEAC-M4は8ビットを1語とし、4語までの可変語長が扱えるワード・マシンであるが、ここでは8ビットを1バイトと書く。

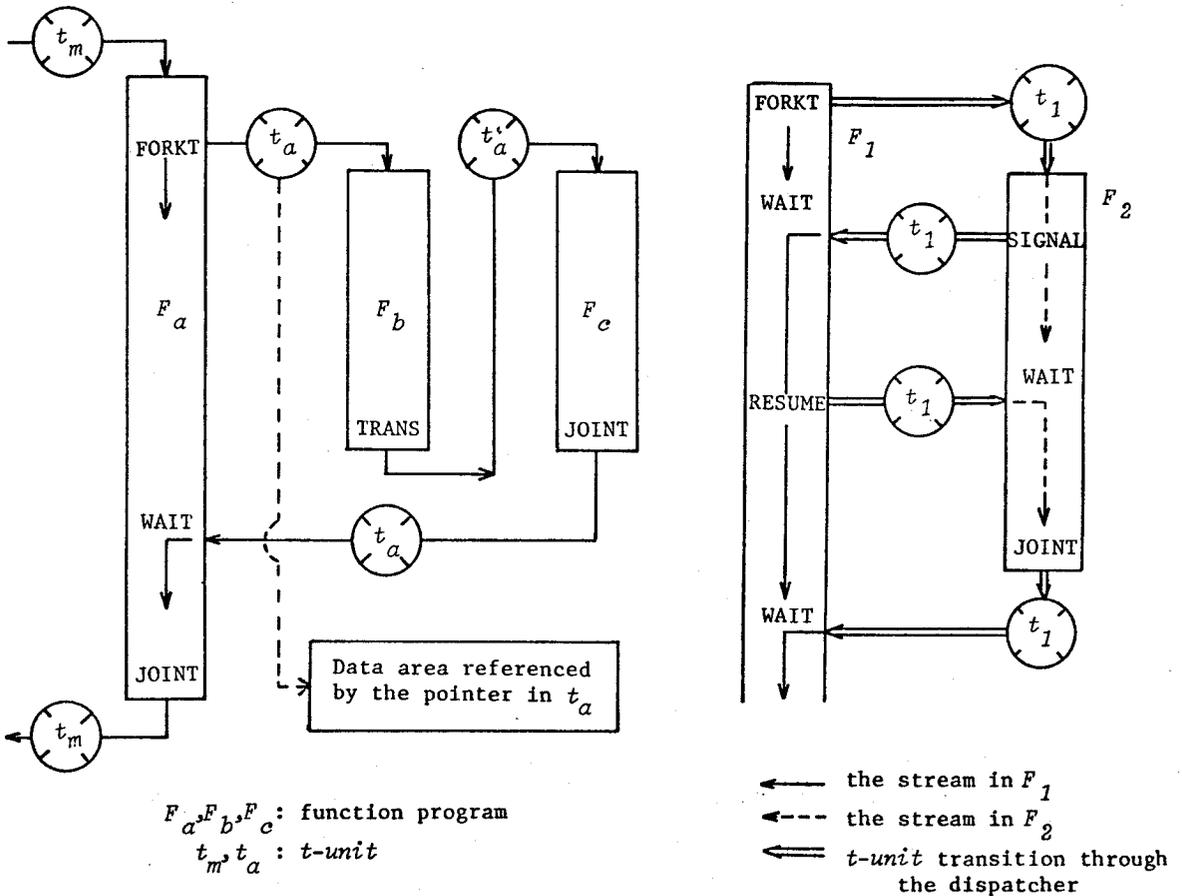
マクロ命令により、ディスパッチャを経由して実行されるが、これらマクロ命令の基盤になっているのは fork-join系である。ただし、Conwayの提唱<sup>9)</sup>したものと多少性質が異なる。まず、分流と合流の適用対象はプロセッサの流れではなくて、t-unitのストリームである。さらに、それぞれのストリームの間に従属関係があるわけではないが、t-unitとそれに伴うデータ領域の提供者は常に明瞭になっており、t-unitは原則としてその発信者にもどつてとのストリームに合流する。したがって、プログラム間の関係は semi-coroutine<sup>10)</sup>である。ファンクション・プログラム相互の結合が動的に行われるため、データ領域の提供者を常に明確にしておかないと、デッドロックを回避する手段が講じられなくなるからである。

以下の4種のマクロ命令

$\text{FORKT}(t_a, F_b)$  ,     $\text{TRANS}(t_a, F_c)$   
 $\text{JOINT}(t_a)$  ,     $\text{WAIT}$

について、その使用例を図3.9に示す。

一つのt-unit  $t_m$  の  $F_a$  におけるストリームから、FORKTによって新しく  $t_a$  のストリームが作られ、 $F_b$  を通り抜けた後TRANSマクロで  $F_c$  に移り、JOINTでもとのストリームに合流する。こ



(a) fork-join 機構

(b) t-unitによるコオペレーション

図3.9 修正型 fork-join機構

ここで fork - join 系を用いることの特徴は、新しく作られた  $t_a$  のストリームが、 $F_b$  に受け入れられるかどうかを見極めることなしに、もとの  $t_m$  のストリームが継続することである。マクロ命令実行時に、 $t_a$  はディスパッチャを経由するので、それが  $F_b$  に受け入れられる時点の判定はディスパッチャに管理されている。JOINT マクロについても同様で、このマクロ命令の実行によって  $F_c$  内の  $t_a$  のストリームは終了するが、それがもとのストリームに合流する時点はディスパッチャに依存している。これにより、プログラム間の占有関係が作り出されないので、デッドロックの回避に役立つ。

複数の t-unit を用意しておいて、FORKT マクロを続けて実行すれば、複数のストリームを作り出せる。この場合、合流点をそれぞれ別々に設定できるように、リターン・ポイントを t-unit 内に書き込めるようになっている (図 3.8)。なお、ディスパッチャのスケジューリング機能に関連して、t-unit は 2 ビットの優先度が指定できる。これは後に述べるように、CPU 割り当ての優先度を示すが、プログラム番号とは別に t-unit のそれぞれに指定できるので、同一のプログラムでも要求の種類に応じて異なった優先順位で実行することが可能である。

WAIT マクロは、何らかのイベント待ち (より具体的には t-unit 待ち) のとき、CPU をひとまず手放すために使われる。原則として、複数の t-unit を順不同で受け取れるようにするため、特定のパラメータを指定するようにはなっていない。

### 3.3.3 同期操作

もともと t-unit のモデルを導入した目的は、リアルタイム処理特有の論理的判断を伴う同期操作を、ユーザ側で記述し易くすることであった。したがって、モニターとして conditional critical region<sup>11)</sup> の一般形を処理する機能を持たず、t-unit 分配のスケジューリングに関連する操作だけを行うので、モニターの核は簡潔な構成になっている。以上の点を含めて同期操作を要約すると、つぎの 3 種類にまとめられる。

- (1) ディスパッチャ内に、プログラム番号対応のエントリー・ポイント・テーブルがあり、ここにセマフォ変数が用意されているので、これを用いて t-unit 流入の同期をとる。
- (2) すでに相互の共有エリアを持っているプログラムの間では、このエリアにセマフォ変数を用意することにより、コオペレーションを遂行する。必要ならば割込み禁止命令を併用する。
- (3) プログラムの入口部分、ならびにストリームの合流点において、一つ以上のイベント情報を受け取って論理判断を行う。

これらの中で、コオペレーションを実行する(2)の場合、相手側が WAIT マクロを実行した状態では、何らかの形で起動をかけることが必要である。この目的のために、RESUME マクロと SIGNAL マクロが用意されている。この両者は機能的には同じであって、プログラム結合時に使われた t-unit の遷移をもう一度引き起こし、wake up の機能を果たす。ただし、semi-coroutine の関係において、親から子には RESUME マクロを、子から親には SIGNAL マクロを使用する。

つぎに、複数の t-unit をも扱う(3)の場合、より具体的には、一つの t-unit の流入についてその都度 CPU が流れ込んで来るので、論理的条件が満たされなければ、WAIT マクロを実行して他のイベントを待つ。また特に必要ならば、一つのファンクション・プログラムで、複数のエントリー・ポイントを持つものを書いてよい。この場合、見掛け上は別のプログラムとして扱われる。

ディスパッチャにおけるセマフォ変数の扱いは、スケジューリング機能に関連するので、次節以降に述べる。さらに複雑な同期操作をシステムで共通に使いたい場合には、上の手段を用いて、同期をとるた

めのファンクション・プログラムを独立に作り、これを登録することによって機能の追加を図る。

### 3.4 ディスパッチャの構成

ディスパッチャには二つの役目がある。一つは論理的モデルにおいて、マクロ命令の実行に伴う  $t$ -unit の分配機能であり、もう一つは物理的な CPU の割り当て機能である。

#### 3.4.1 論理的モデルのスケジューリング

FORKT, TRANS, RESUME の 3 種のマクロ命令では、 $t$ -unit はエン트리・ポイントに送り込まれる。これらのマクロ命令は、原則として独立な並列動作の基盤の上で使用されるものであり、相手側の状況に無関係に実行されるのが立て前である。これに対して、システム全体が混乱に陥ることのないように、ディスパッチャ内に図 3.10 に示すようなエン트리・ポイント・テーブルを持っている。 $t$ -unit に書き込まれたプログラム番号が  $n$  に対応しており、実際の実行開始アドレスは、ここに一括して登録されている。したがって、プログラム差し換え時のアドレス変更、さらには RESUME マクロに対する処理再開アドレスの動的変更が可能である。

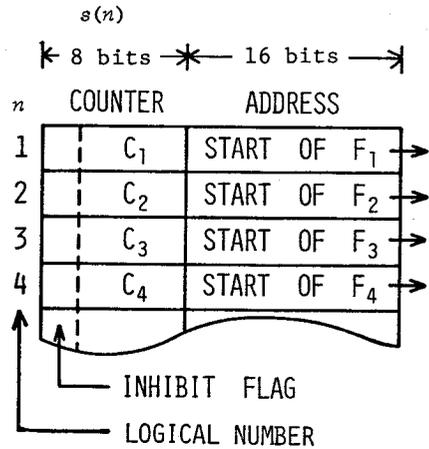


図 3.10 ディスパッチャのエン트리・ポイント・テーブル

このテーブル内には、各エン트리・ポイントごとに一種のセマフォ変数の役割を持つカウンター  $s(n)$  があって、ディスパッチャとそれぞれのファンクション・プログラムの間で共通に参照され、また値の変更が行われる。ただし、Dijkstra の P/V 操作とは多少性質が異なり、符号ビットとカウンターが独立に扱われる。カウンターは、受け入れ可能な  $t$ -unit の数を表し、符号ビットは一時的に  $t$ -unit の流入を禁止するエントリ・インヒビット・フラグとして使われる。 $s(n) \leq 0$  ならば、対応する  $t$ -unit はディスパッチャ内の待ち行列につながる。逐次的にしか使えないデバイスの制御プログラムでは、カウンターは 1 または 0 の値を取り、これによって要求を一つずつ受け入れる。なお、RESUME マクロでは、すでに一度  $t$ -unit の流入が行われた後の操作であり、カウンターは無視されてインヒビット・フラグだけが有効である。

JOINT, SIGNAL マクロによって発せられた  $t$ -unit は、それ自身の中に書き込まれたリターン・ポイントに送り込まれる。ここにはリターン・インヒビット・フラグがあり(図 3.8 参照)、一時的なストリーム合流の抑止効果を持つ。

ディスパッチャ内の待ち行列を作る場合、各エン트리・ポイントに対応してキューを作れば、メッセージ・キュー<sup>11)</sup>に近い形式となる。この手法では、各プログラムに対応した  $t$ -unit を取り出し易い利点はあるが、CPU 割り当てのための待ち行列を別に用意しなければならない。

ところで、ここで考えているモニターの構成法では、論理的条件を伴う同期操作をユーザ側にまかせているので、これに関連する  $t$ -unit はファンクション・プログラム内部に留まっているものが多く、ディスパッチャに入って来るのは真にプログラム間での処理要求が発生した場合に限られる。したがって、一時点でディスパッチャ内に存在する  $t$ -unit の数はそれ程多くはないと考えられる。

この特徴を考慮して、待ち行列はCPU割り当ての面から、t-unitの優先順位別に作る。この場合一つのキューに性質の異なるt-unitが繋がれることになるが、その反面、それまで実行中であった優先順位より高い順位にあるキューだけを調べればよいという利点がある。この点は、CPU割り当てのスケジューリングと関連するので、次節で詳しく述べる。

### 3.4.2 CPUの割り当て問題

一般的には、ディスパッチャの中に複数個のt-unitがまわって来るが、これらをそれぞれの中に書き込まれた優先度別にふるいわけると同じ優先度を持つものが二つ以上あれば、優先順位ごとにキューを作る。リアルタイム処理では優先処理は重要であり、CPUのpreemptionも行つて優先度の高いものを先に処理する。また、一方、同一優先順位に属するものについては、セマフォ変数の通過条件を満たす限り、要求発生順に処理するものとする。そこで、一つの優先順位に対応するキューを一種のコマンド列とみなし、これらについては逐次処理で良いと決めてしまえば、一つの優先順位ごとに一つの仮想プロセッサを割り当てることにして、ステート・ワードも順位の数だけ用意すれば十分であろう。

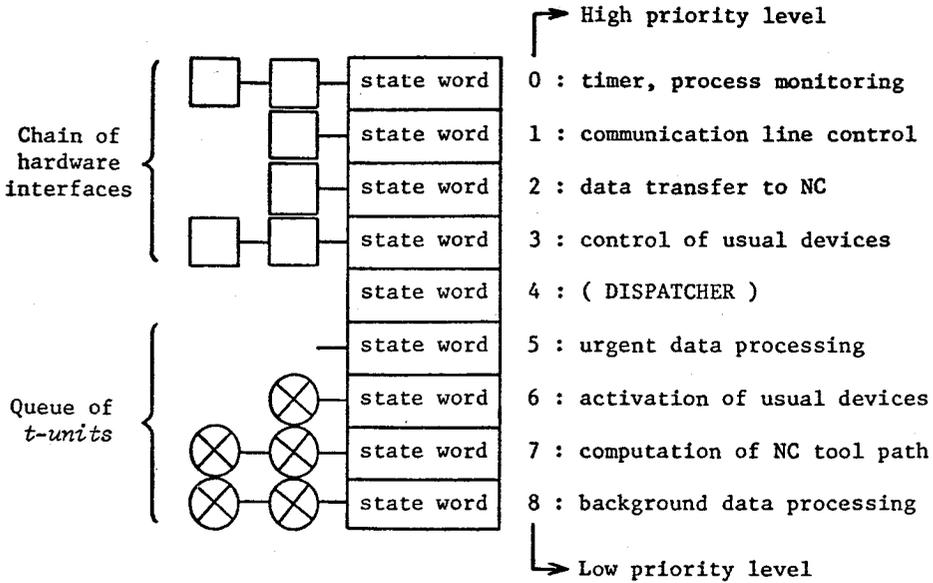
実行中の優先順位は最後にCPUを割り当てたt-unitによって決まり、その後のマクロ命令の実行によって処理の順位が切り換わるときには、レジスタの待避復旧はディスパッチャにより自動的に行われる。ただし、WAITマクロについては、同一の順位で次のt-unitの処理に移るのが原則であるから、ユーザ側で処理の再開に必要なレジスタの内容を保存しておく。これは一つの制約であるが、モニターのオーバーヘッドを増大させないための妥協策である。

ディスパッチャのスケジューリング機能の中で、優先処理の機能と逐次処理の機能を明確にしておく、ユーザ側ではt-unitの優先度のつけ方如何によってこれら二つの機能を使い分けることができる。一つのファンクション・プログラムについて、複数個のt-unitが流入する場合や、ストリームの合流点で複数個の応答を受け取るときに、優先度が同じ順位に統一されていれば、WAITマクロを実行しない限りt-unitの流入は起こらない。逆に優先度を変えておいて、エントリー・ポイントやリターン・ポイントを複数個設けておけば、一つのプログラムの中でも追い越し処理や割込み型の同期操作が可能である。さらにインヒビット・フラグの併用を考慮すれば、プログラムの処理形態について、ユーザ側に大きな選択の自由度が与えられることになる。

t-unitの中にステート・ワードを持つ手法は、タスク管理法に近いもので理論的には興味がある。しかし、モデルの設定に柔軟性があるため、実際の処理がモニター側もユーザ側も煩雑になるので、この手法は敢えて採用しない。また、t-unitのストリームとLauesenのアクティビティ<sup>13)</sup>には類似点があるが、アクティビティは一つのモニター・プロセスの枠内で使われているのに対して、t-unitはいわゆるプロセスの枠を越えて、ユーザにより同期操作一般、ファンクション・プログラム間の動的結合、優先処理の積極的利用など、汎用性のある道具として用いられる。システム全体の構成法という見地から、両者は性質を異にしているのである。

図3.11には、多重割込み4レベルとノーマル・モードの優先順位が4レベルの場合を示してある。ディスパッチャは原則として割込み禁止で働くが、これにも優先順位を一つ与えておき、t-unitのキューについてセマフォ変数の判定を行う場合、一時的に割込みを受けつける。割込みモードにはt-unitのキューはないが、必要に応じて、ノーマル・モードのキューにt-unitをつなぎ込むことが可能である。

実際にプロセッサが複数個ある場合にも、それらを別々の優先順位にわりふるものとすれば、このモデ



Level 0 - 3 : interrupt mode ( CPU assignment is executed by the interrupt analyzer )

Level 5 - 8 : normal mode ( CPU assignment is executed by the dispatcher )

図3.11 CPUディスパッチングのスケジューリング機構

ルはそのまま使えると思われる。ただし、非可分の test and set 命令ならびに相互の割込みのかけ方が明確にされなければならない。

一つの優先順位に属する t-unit の数に制約はなく、プログラムそのものはもともと優先順位とは無関係である。CPU の preemption を実施する優先順位の数、デバイスその他の物理的資源が無尽蔵にあるわけではないので、必要最小限に止める方が望ましい。順位数を多くすると、過大負荷の状態に陥る危険性があるので、待ち状態が発生する場合には、早い時点で待たせる方がシステムの効率は良くなる。

### 3.4.3 デッドロック対策

それぞれ独立に作られたプログラムに対して、組み合わせや差し換えに自由度を与えると、デッドロックに陥る危険性も高くなるが、小型計算機でのモニターでは、それを動的に検出して防止することは期待できない。したがって、デッドロックの回避は、最終的にはユーザ側の責任である。

ただ、semi-coroutine に対する fork-join 系では、デッドロックに陥る危険性は少ない。一般にデバイスなどの管理も、その制御プログラムのエントリー・ポイントにおけるセマフォ変数で処理されるが、第 3.3.2 節でも述べたように、FORKT, JOINT マクロでは、ファンクション・プログラム間の占有関係はできないので、処理途中でコオペレーションを実行しない限り、デッドロックの原因を作り出さない。

たとえば、予め 1 レコード分のデータを用意しておいて、ディスク装置の転送要求が出されたような場

合、ディスク転送処理は要求を出した側の状態と無関係に実行される。また、転送終了時には JOINT マクロによって、終了通知を返すことになるが、その通知が要求元へ帰ることを確かめずに次の要求を受けつける。したがって、このような転送要求の列は有限時間内に終了する。

ただし、共有エリアを用いてコオペレーションを実行するときには、暗に占有関係ができる。また、メモリ・スペースについても、モニターで一括管理する手法はとっていないので、それを管理しているファンクション・プログラムを常に明確にすることによって、ユーザが安全な手段を講じられるようにしておく。

### 3.5 結 言

広い意味でのモニターには、標準のデバイス制御プログラムや回線制御プログラムなども含まれるが、これらはユーザの作成するファンクション・プログラムと全く同様に扱われる。したがって、本文で述べたインタラプト・アナライザとディスパッチャを核として、これに標準のサービス・プログラムを必要に応じてシステムに登録し、ユーザのアプリケーション・プログラムを追加すれば、使用環境に応じたシステム構成ができ上がる。

ファンクション・プログラム相互の結合は動的に行われ、マクロ命令によるディスパッチャ呼び出しのためのアドレスも、固定したアドレス・テーブルを用意すれば、個々のファンクション・プログラムの登録は独立に行われるので、システム・ジェネレーションは不要である。プログラム番号を決めておいて、ディスパッチャ(またはインタラプト・アナライザ)内のエントリー・ポイント・テーブルに実行開始アドレスを設定すればよい。以上より、本章で導入した手法は以下の特徴を持つ。

- 1) 変数領域の取り扱い方として、管理プログラムにメッセージ・キューはないが、ユーザが t-unit を利用してメッセージ・キューを操作することができる。
- 2) プログラム間の同期操作や相互の情報交換が容易に行え、柔軟性のあるプログラムが記述できる。
- 3) プログラム間の結合が動的に行われるので、プログラム・モジュールの差し控えが容易であり、使用環境への適応性が大である。
- 4) 管理プログラムが簡潔であり、多くのメモリ・スペースをユーザに開放できる。
- 5) t-unit のキューの操作中にも割込みを受けつけるので、割込禁止による不感時間帯が短い。

### 参 考 文 献

- 1) 山根敬一、牧之内三郎：ミニコンピュータにおけるイベント処理指向型のリアルタイム・モニター、情報処理、Vol.18, No.6 (1977), pp.526-533.
- 2) K.Yamagata and S.Makinouchi : A Real-Time Monitor for Minicomputer Using Event Processing Model, Information Processing in Japan, Vol.17(1977), pp.144-148.
- 3) S.Makinouchi and K.Yamagata : A Modular Approach to the Control Program Construction in Real-Time Minicomputer Systems, Proc. of the 7th IFAC World Congress (1978), pp.747-754.

- 4) 電子計算機プログラム用言語FORTRAN, JIS規格 C-6201.
- 5) P.Naur(ed.): Revised Report on the Algorithmic Language ALGOL 60, Computer Journal, Vol.5(1963), pp.349-367.
- 6) 萩原 宏、渡辺勝正、山根敬一: Recursive Procedure の処理に関する諸問題、情報処理学会大会予稿集(1965), pp.49-50.
- 7) 野崎昭弘: Recursive Procedure の解析、情報処理、Vol.6, No1(1965), pp.12-20.
- 8) 藤井 純、鈴木伸夫: オペレーティング・システム、産業図書(1970).
- 9) M.E.Conway: A Multiprocessor System Design, Proc. AFIPS FJCC, Vol.24(1963), pp.139-146.
- 10) O.-J.Dahl and C.A.R.Hoare: Hierarchical Program Structures in Structured Programming, Academic Press(1972), pp.175-220.
- 11) P.B.Hansen: Operating System Principles, Prentice-Hall(1973), p.366.
- 12) E.W.Dijkstra: The Structure of the "THE"-Multiprogramming System, Comm.ACM, Vol.11, No5(1968), pp.341-346.
- 13) S.Lauesen: A Large Semaphore Based Operating System, Comm. ACM, Vol.18, No7(1975), pp.377-389.

## 第4章 オンラインDNCシステム

### 4.1 緒言

ミニコンピュータの普及により、あらゆる規模の計算機が数値制御装置の周辺でいろいろな目的に使われるようになって来た。たとえば、NCテープ作成の省力化、工具径路の検査、群管理システムの構築、適応制御システムの実現などが挙げられる。

本章で述べる実験的DNCシステムは、大型計算機センターのTSS(Time Sharing System)の端局としてミニコンピュータNEAC-M4をオンラインで使用し、さらにNEAC-M4と数値制御装置(NEDAC-4200)を接続している。<sup>1)</sup>したがって、数値制御指令をNEAC-M4あるいはTSSの大型計算機から数値制御装置に転送することができる。このような計算機の階層構造を利用して、機械加工工場で有効に稼動するDNCシステムを実現するための基礎資料を得ようとするものである。この場合、ミニコンピュータが重要な働きをすることになり、<sup>2)</sup>第3章で述べたプログラム構成法が有用になる。

ミニコンピュータの記憶容量は小さいので、一部のプログラムを差し換えることによっていろいろな機能が遂行できるようにしなければならない。すなわち、端局を使ってTSSで図形処理を行う場合、通常の端局としてAPT, FORTRAN, EDITORなどを使用する場合、さらにはCNCシステムを実現する場合など、必要なファンクション・プログラムだけを主記憶装置にロードして、果たすべき機能の多様性に備えるのである。以下本文でオンラインDNCシステムに必要な周辺装置と、その基本的な制御プログラムの組立てについて述べる。本章では第3章に引き続き"t-unitのストリーム"という概念を用いる。

### 4.2 伝送制御手順

大型計算機センターのTSSとミニコンピュータNEAC-M4のデータ通信は電話回線を通して行われる。ここで述べる実験用システムの通信方式の規格は以下のとおりである。

通信方式	全二重通信
通信速度	1200ビット/秒
伝送符号	10単位符号
情報符号	NEACコード(制御符号はISOコード)
同期方式	調歩同期方式
誤り検出	ブロック・チェック(偶数水平パリティ)
	レター・チェック(偶数パリティ)

NEAC-M4には変復調装置(モデム)とモデム制御部が接続され、M4本体とモデム制御部との間の入出力は8ビットの文字単位で行われる。モデム制御部では、送信の場合、この8ビットにスタート・ビット、ストップ・ビットを付加し、パリティ・ビットをセットして並列変換を行って伝送符号とする。受信の場合には逆の動作によって伝送符号から8ビットが取り出され、偶数パリティの検査も行われる。パリティ検査の結果はセンス命令(SEN)で検出できる。NEAC-M4側のモデムとセンター側のモデムの間は伝送符号が直列に転送される。これらの処理のため、モデム制御部には送信側、受信側それぞれに1文字分のバッファ・レジスタが設けてあって、これらが空いた時点または文字が格納された時点を割込信号あるいはセンス命令で検知できるようになっており、これを用いてNEAC-M4と大型計算機とのデータの入出力の同期がとられる。

モデム制御部への指令と状態検出のために、NEAC-M4では表4.1に示すようなファンクション命令とセンス命令が用意されている。<sup>3)</sup>

以上のように1文字単位のデータの送受信はハードウェアの機能に依存しているが、1ブロックのメッセージの送受信の管理はソフトウェアによって遂行される。この場合、簡単な端末ではこれを単なる入出力デバイスとみなして、例えば出力では機械的に連続してメッセージを送るに過ぎない。他方、ここで述べ

ているようにコンピュータを端局として置く場合には、1ブロックのメッセージの送受信ごとに相手局の状態確認を制御コードを用いて行う。この状態確認のための双方の約束事を伝送制御手順と呼び、その目的は以下のように要約される。

表 4.1 モデム制御部指令コードと状態検出コード

コード	機能	
BTI 0	1文字入力	
BTO 0	1文字出力	
FUN	0	リセット コントローラ
	2	受信モード セット
	4	送信モード セット
	6	割込制御
SEN	0	データ通信可
	2	受信バッファ・レジスタ レディ
	3	受信エラー
	4	送信バッファ・レジスタ レディ
	5	送信エラー
7	割込検知	

- (i) 遠隔地にある端末の受信準備ができていないにも拘らず、一方的にセンターからデータを送信してそのデータが失われるのを避ける。
- (ii) 回線上やデバイスのエラーが発生しても、混乱の起らないように回復手段を講じておく。
- (iii) 端局側の都合で、メッセージの受け取りを遅ら

せたいとき、そのタイミングを遅らせて完全なメッセージの受け取りを可能にする。

- (iv) 伝送を中断したい時、両方の局でこれを知って、メッセージが脱落するのを防ぐ。
- (v) メッセージの伝送が中絶した時、マシン・トラブルによるものか否かを常に検知しておく。

相手局の状態確認を行って、伝送制御手順を遂行するために特別な伝送制御符号が使われる。これを表4.2にまとめておく。

表 4.2 伝送制御符号

以下の節で伝送制御手順(N6040手順)を要約して述べる。

制御符号	主な機能
DLE・ENQ	相手局の呼び出し
DLE・DC1	マシン・エラーの通知
DLE・DC2	ストップ・ボタン押下の通知
DLE・DC3	リスタート・ボタン押下の通知
ACK	1ブロック正常受信の応答
NAK	1ブロック異常受信の応答
ENQ・ENQ	相手局の応答促進
EOT・EOT	伝送中断の通知

#### 4.2.1 受信モード

会話は端局のスタート・ボタンを押すことによって始まり、端局からセンター(TSS)に伝送制御符号DLE・ENQが送られる。そして、センターから端局にDLE・ENQを返送し、端局からセンターへ再びACKを送る。この手続きによって双方の状態の確認が行われる。受信モードはセン

ターからDLE・ENQ・UA・UAを端局に送ることによって開始され、出力デバイスが指定される。ここにUAはユニット・アドレスを表し、各デバイスの識別コードである。このとき端局側で出力デバイスの準備ができていれば、端局はACKをセンターに返す。これによってメッセージが端局に送られる。受信モードでは、端局側の主記憶装置に2ブロック分のバッファ・エリアを用意しておき、メッセージの受信と周辺装置への出力を並行して行う。すなわち、端局では少くとも一つのバッファが空いていることを確認して、ACKを返送し、次のメッセージを受けとる。このようにすれば、出力デバイスの動作速度が伝送速度よりも遅いために情報が正確に受信できなくなることはない。

メッセージ1ブロックの長さは最大144字で、その先頭にはSTX、終りにはETXの制御コードがつけられ、可変長さのメッセージの始めと終りが明瞭に識別できるようになっている。

メッセージを受信するたびごとにACKを返すという手順が受信モードの標準であるが、パリティ・エラーなどの異常が検出された場合はACKの代わりにNAKを返送する。この場合はセンターから前と同じメッセージが送られて来るので、エラー回復が可能である。ただし、4回以上NAKを返送してもエラーが検出されるときは、回線上のトラブルと見なして一度回線断とする。しかしながら、実験用システムの使用経験ではこのようなことは起っていない。

受信モードが継続する場合であっても、出力デバイスが変る場合、あるいは受信モードから送信モードに切替る時には再びセンターからDLE・ENQ・UA・UAが送られる。このように入出力デバイスの指定は常にセンター側が行う。

メッセージの出力中に伝送を一時的に切断したい場合がある。このときは端局付属のストップ・ボタンを押す。このとき端局は、メッセージの一部が失われることのないように、バッファの内容をすべて出力した時点でACKの代わりにDLE・DC2を返送する。これによってセンターは一時的にデータの伝送を中断

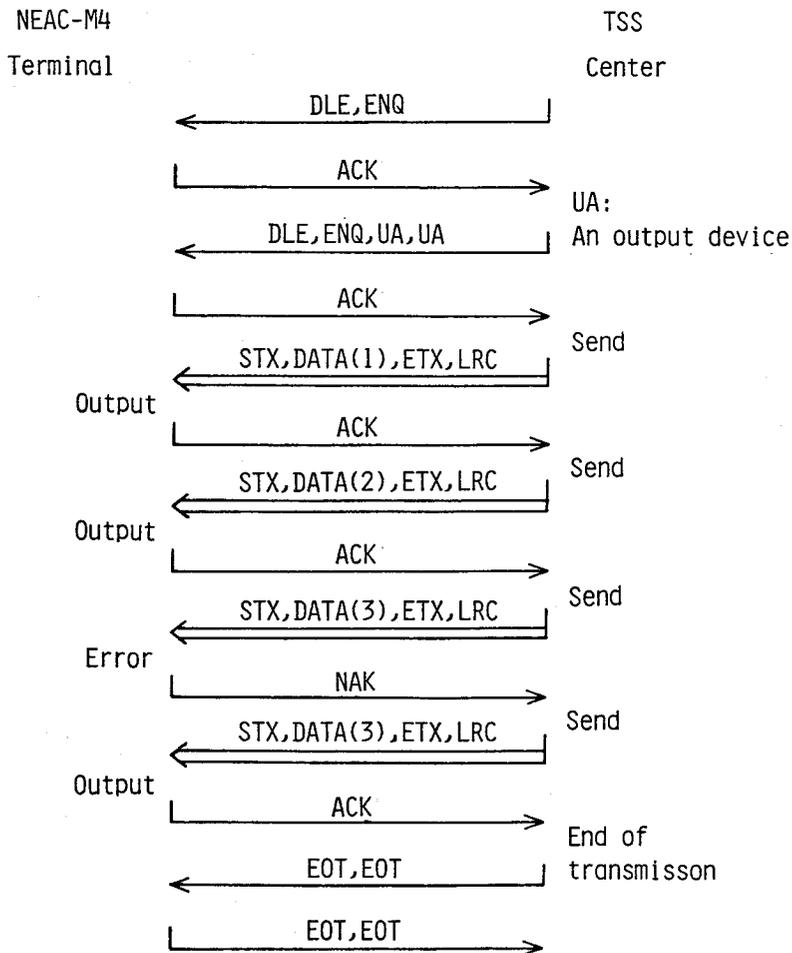


図 4.1 受信モードにおける標準伝送制御手順

する。

また、センター側が内部処理に時間がかかってデータ伝送が途切れる場合、センター側は制御符号 EOT・EOTを送るので、端局も EOT・EOTを返送して待ち状態に入る。したがって、この場合と回線上のトラブルで伝送が中絶した場合とは明瞭に区別される。

受信モードにおける標準伝送制御手順を図 4.1 に示す。

#### 4.2.2 送信モード

受信モードと送信モードが切換える場合には、センターは DLE・ENQを端末に必ず送る。端局はメッセージ・バッファの状態など切換えの準備ができたときに ACKを返送する。送信モードのときは、次にセンターから DLE・ENQ・UA・UAが送られて来て、UA で入力デバイスが指定される。このように、送信モードのときもデバイスの指定はセンターが行う。

端局では STX・UAを直ちに返送し、入力デバイスからの入力動作に入りメッセージの送信を開始する。各入力デバイスにはブロックの切れ目を定義する文字が定められており(タイプライタでは Line Feed)、この文字を入力するとメッセージの終りを示す制御符号 ETB・LRCが送信される。入力送信の場合も、少くとも1ブロック分のバッファ・エリアが準備され、入力デバイスの速度と伝送速度との調節が図られる。センターでメッセージが正常に受信されると ACKが返って来るので、次のメッセージを送信すればよい。センターでエラーが検出された時は、NAKが返って来るので同じデータを再送する。したがって、センターから ACKを受け取るまではバッファ・エリアの内容は保存しておかなければならない。NAKを4回受け取ったら Send Alarm Lampを点灯して一まず回線断とする。

メッセージ送信後 800 ミリ秒の時間監視を行って、センターからの ACK, NAKの応答を待つ。この間に応答がない場合は ENQ・ENQをセンターに送り応答を促す。これも4回行って無応答のときは Send Alarm Lampを点灯して一まず回線断とする。このように常時時間監視を行って、伝送が途切れるのを検査しているのである。

送信中入力デバイスが切換わる時には、初めの手順にもどってセンターから DLE・ENQと DLE・ENQ・UA・UAが送られて来て新しいデバイスが指定される。伝送の中断は受信モードと同様であって、端局の方で一時的に中断したいときは、DLE・DC2を送り、センターの方で入力要求が中絶したときは EOT・EOTの交換により待ち状態に入る。送信モードにおける標準伝送制御手順を図 4.2 に示す。

#### 4.2.3 送受信の方向決定

端局との送受信の方向はセンター側の NEACシステム 900 の処理によって定められるが、この処理はユーザの使用するコマンドによって決定される。例えば、LISTコマンド実行中には、ファイルの内容を機械的に端局に出力するだけでその途中に入力処理は介入しないから受信モードが継続する。伝送制御手順としては途中で ACKの返送が実際に行われているが、これは端局にいるユーザには分らない。端局コンピュータの内部処理として遂行されるにすぎない。これと逆の場合が、高速紙テープ・リーダーからパーマネント・ファイルにデータ(またはプログラム)を登録する READコマンドの場合で、この場合は送信モードが継続する。

一般に端局を用いて会話処理を行うときは受信モードと送信モードが交互に現れる。例えばビルド・モードで AUTOX コマンドを用いてテキスト入力を行う場合、行番号を受信してはテキスト入力をくり返すので送信モードと受信モードが交互に現れる。また、エディタを Verify モードで使用中のときも、編集コマンドを入力してはその結果を出力するので送信受信が交互に行われる。このような場合にも、伝送制御

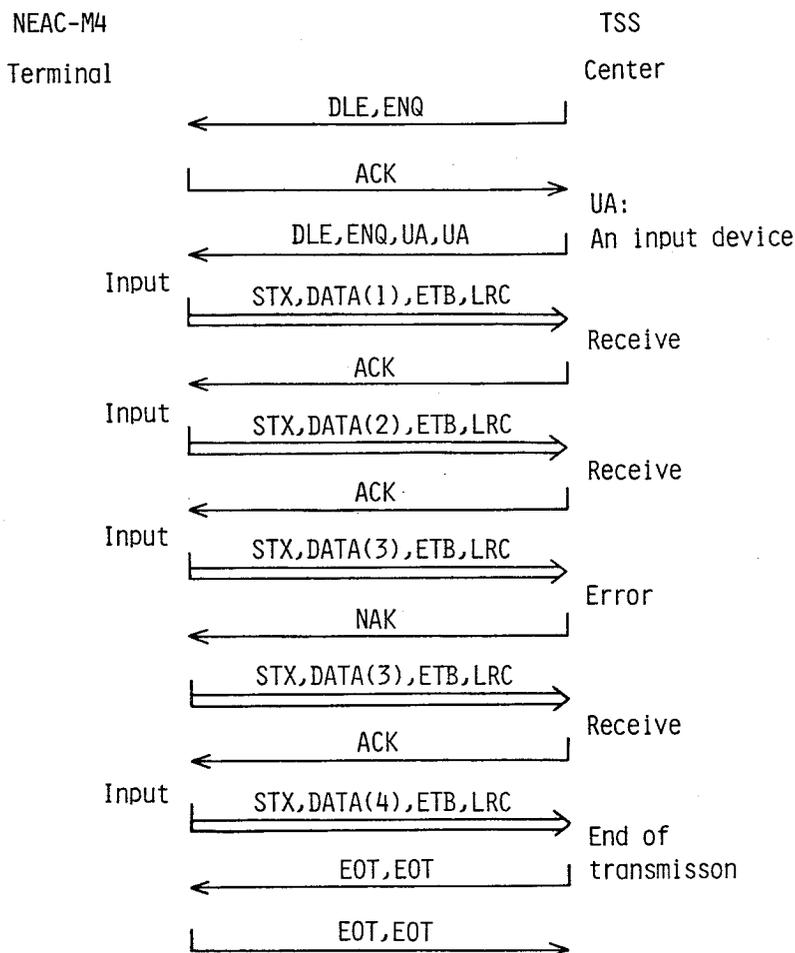


図 4.2 送信モードにおける標準伝送制御手順

手順としてはたえず `DLE · ENQ` によるモードの切換えが行われているがこれはユーザには見えない。

別の使い方として、ユーザの作成した FORTRAN のプログラムを実行する場合を考える。このときには端局に対する `WRITE` 文の実行で受信モードになり、`READ` 文の実行で送信モードになる。計算実行中であって、入出力文が中断するときは `EOT · EOT` 交換による伝送中断状態にある。したがって FORTRAN のプログラム作成時に `READ` 文と `WRITE` 文の組み合わせによって、中間結果を見てから次のパラメータを入力し、計算処理を次の段階へ進めるような使い方が可能である。

端局にいるユーザは伝送制御手順の受信モード、送信モードの区別を必ずしも知る必要はない。端局に入力要求が出るときには入力促進の文字 "`*`" あるいは "`=`" などが印字されるのでこの時に入力を実行すればよい。実際には、端局に `Send Lamp`、`Receive Lamp` が用意されているので、実行中のモードが分かるようになっている。これらのランプの点滅も伝送制御手順の中に包含されているのである。

以上を要約すると、通信回線を通してデータを伝送する場合、1文字ごとの伝送はハードウェアで実行される。メッセージの1ブロックごとに伝送制御手順が実行されるがこれはあくまでも端局コンピュータの内部処理であって、ユーザには見えない。ユーザに見えるのは制御符号を除いた真のメッセージの入出

力のみである。

### 4.3 周辺装置の制御

#### 4.3.1 コンソール・タイプライター

実験用システムに設置したタイプライターの各部仕様は以下の通りである。

印字方式	タイプホイール式
印字速度	10字/秒
印字文字数	64種
印字数	74字/行
印字紙巾	216mm
印字間隔	10字/25.4mm
コード	8単位 ASCII

また、紙テープの読み取り機構、さん孔機構が付属しており、読取り速度、さん孔速度は共に10字/秒である。NEAC-M4における制御命令を表4.3に示す。

これらの命令を使えば1文字ごとの入出力は実行可能であるが、使用上の便宜のため1ブロックのメッセージの入出力を行う制御プログラムを用意しておく。出力の場合は、1ブロック分のメッセージを主記憶上のデータエリアに格納しておいて出力要求を出す。入力の場合は1ブロック分のメッセージが入り得るバッファ・エリア（空き領域）を用意して入力要求を出す。入出力要求の出し方は第3章で述べた `t-unit` とマクロ命令を利用する。`t-unit` のオーダ・コードに入出力の区別やブロック区切りのコードを指定し、またデータ・エリアのアドレスも書き込んでおく。入出力要求は `FORKT` マクロによる。1文字ごとの入出力はタイプライター制御プログラムで実行されるので放置すればよい。1ブロックの入出力完了後 `JOINT` マクロでその旨の通知が主プログラムに対して行われる。図4.3は `t-unit` を使う最も基本的な場合である。

タイプライター制御プログラムで1文字ごとの入出力を行うのに割込みを利用する。出力の場合と紙テープ読取りの場合は100m secごとに割込みが発生し、キーボード入力の際は打鍵したときに割込みが発生するので、このタイミングに従って文字単位の処理を行う。ブロックの区切りのコードは通常 `Line Feed` が用いられる。入力の場合、ブロックの区切りのコードが検出されなくても、用意されたバッファ・エリアが一杯になれば入力完了とみなす。これ以上の入力処理を行って他のデータ領域やプログラム領域を破壊するのを避けるためである。

リアルタイム処理では何らかのエラー状態が発生したときに簡潔なエラー・リカバリの機能が備わっていることが望ましい。このための簡単な処理プログラムをタイプライター制御プログラムに含ませてある。エラーが検出された状態でタイプライターにエラー・コードを印字し、ブザーを鳴動させる。この場合、オペレータはタイプライターから次の文字のいずれかを入力することができる。

表4.3 タイプライター制御命令

コード	機能
BTI 01	1文字入力
BTO 01	1文字出力
21	制御部リセット
61	読込み割込み禁止解除
FUN 01	書込み割込み禁止解除
C1	紙テープ1文字読取り
41	紙テープ連続読取り
SEN 21	出力バッファ・レディをセンス
SEN 41	入力バッファ・レディをセンス

- R ( Restart )
- S ( Skip )
- E ( End )
- D ( Dump )
- M ( Manual in )

各文字の解釈はエラーを検出したプログラムにまかされているが標準的な解釈は以下の通りである。

R : もう一度同じ処理を実行する。スイッチの入れ忘れのような簡単な操作ミスはこれで回復できる。

S : エラーの起った状態をとばして、処理を先に進める。

E : 処理の続行が不可能なときは強制的に処理を打ち切る。

D : エラーが起った時のデータをタイプライタに印字して原因を調べる。

M : エラーが検出されたデータの代りに別のデータをタイプライタから入力する。

実際にはエラー・リカバリはむつかしい問題であって各処理プログラムごとに考慮しておかなければならない。ここでは基本的に標準的なものとして差し当り上の5種類を考えておくのである。

エラー・リカバリの場合も、各処理プログラムとタイプライタ制御プログラムの情報の交換が t-unit を介して行われる点は、通常の入出力処理と変わりはない。1 台のタイプライタが二つの役割を担っているのである。通常の入出力処理とエラー・リカバリの要求が重なって発生した場合は、エラー・リカバリの方が優先処理される。

### 4.3.2 高速紙テープ・リーダー

実験用システムに設置された紙テープ・リーダーの仕様は以下の通りである。

読取り速度	300字/秒
読取り方式	光電式二重読取り
テープ走行速度	30インチ/秒
紙テープ規格	国際8単位さん孔テープ

また、NEAC-M4には表4.4に示す命令が用意されている。

高速紙テープ・リーダーの場合もタイプライタと同様に1ブロック読み込みの制御プログラムを用意する。連続読取りモードで紙テープを走行させ、1文字ごとの主記憶装置への格納は割込み処理によって行う。データ処理に時間がかかる場合は、ブロックの切れ目で停止して次の入力要求を待つ。ブロックの区切りには通常 Line Feed の符号を用いるが他の符号を指定することもできる。各処理プログラムとこの制御プログラムとの情報交換は t-unit を介して行う。符号系は原則として ASCII コードであるが、バ

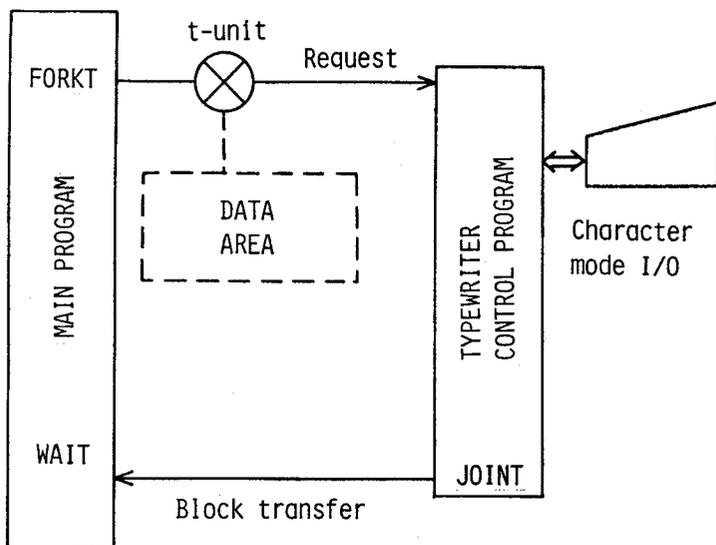


図 4.3 タイプライタ入出力要求

表 4.4 紙テープ・リーダー制御命令

コード	機能
FUN	0 制御部リセット
	2 1文字読取り
	3 連続読取り開始
	4 連続読取り停止
	6 割込み禁止
	7 割込み禁止解除
	SEN
2 バッファ・レディをセンス	
3 リード・エラーをセンス	
4 奇数パリティ・エラーをセンス	
5 偶数パリティ・エラーをセンス	
7 割込み要求をセンス	
BTI 0	1文字レジスタに移送

イナリイ・テープを読む機能もある。連続読取りモードであるので、入力終了時には停止指令を出す必要があるが、t-unitのオーダ・コードの指定でこれを実行する。パリティ・チェックもt-unitのオーダ・コードで指定のあった時だけ実施する。

いま、高速紙テープ・リーダーからデータを読み取り、計算処理を行って結果をタイプライタに印字する場合を考える。この場合、主プログラムの中で入力プロセス、計算プロセス、出力プロセスの3者を並行して動作させ、時間の損失を防ぐことが望ましい。そのために、入力側と出力側共にダブル・バッファを用意し、一方で入出力を行っている間に他方のバッファで計算処理を遂行するようにする。この様子を図4.4に示す。

図4.4で、リード・バッファは交代方式で使う。

すなわち、バッファ1を計算プロセスに、バッファ2を入力プロセスに用い、それが済むとバッファ1と2を入れ換えてバッファ2を計算プロセスに、バッファ1を入力プロセスに用いる。この場合二つのバッファは共に入力プロセスに属する。

これに対して出力側は、計算プロセスと出力プロセスの両方にバッファを持たせ、実際に出力を実行するときには二つのバッファの間でデータの写しを取ってから物理的な出力を遂行する。これは計算結果を

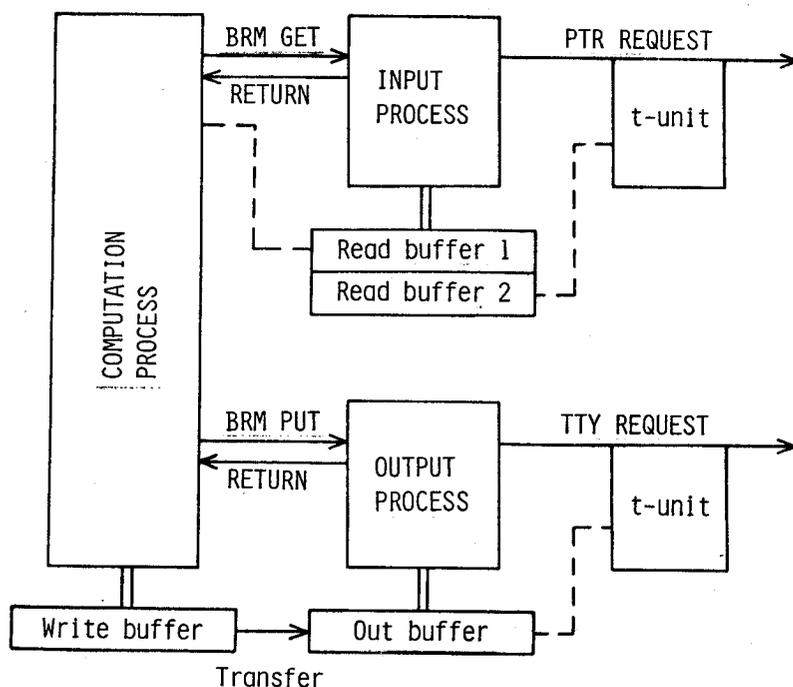


図 4.4 入出力プロセス

格納する場所 ( Write - Buffer ) が複数個ある場合を考慮したためである。ライト・バッファからアウト・バッファに写しを取った後は計算プロセスと出力プロセスは再び並行動作に入る。

入力プロセスと出力プロセスは主プログラム中のクローズド・サブルーチンの形式を取っており、NEAC-M4のBRM命令<sup>3)</sup>で呼び出すようになっている。例えば入力側について、計算プロセスからデータの要求があっても、実際の入力動作が完了していなければ入力プロセスの中で入力動作完了を待つ。このようなタイミングの調節は第3章で述べたディスパッチャを利用して行われる。出力側も同様に待ち時間の調節は出力プロセス内部で行われるので、計算プロセスではこの点を考慮する必要はなく、次のデータが必要になった所でBRM GET命令を実行し、計算結果が出た所でBRM PUT命令を実行すればよい。待ち時間が介入するときにはBRM命令に対するリターンが遅れるだけである。

以上のように時間調節を行う入力プロセスと出力プロセスをクローズド・サブルーチンとすることによって、計算プロセスは単純化され、入出力の遊び時間を吸収することができる。

### 4.3.3 磁気ディスク

本システムで使用する小型磁気ディスク装置は8インチの磁気円板1枚とマルチヘッド式浮動ヘッドで構成され、磁気円板は両面で2個のタイミング・トラックと64個のデータ・トラックを持っている。1トラックは16個のセクタに分割され、各セクタは64語(1語は8ビット)の記憶容量を持つ。したがって、磁気ディスク装置1台の記憶容量は64K語である。磁気円板の回転数は3600回転/分、平均待ち時間は8.3 msecである。

主記憶のコア・メモリと磁気ディスクの間のデータ転送はDMA(Direct Memory Access)チャンネル機構を介して行われる。すなわちNEAC-M4は、1文字ごとの移送を行うのではなく、DMAチャンネル機構に指令を出してブロック転送を行う。この指令にはディスク・アドレス、コアメモリ・スタート・アドレス、コアメモリ・エンド・アドレスが含まれているから、主記憶の連続した領域は一つの指令で一度に磁気ディスクへブロック転送される。転送時の最小のブロックは1セクタ(64語)である。転送の終了は割込みによってNEAC-M4本体に知らされる。

パリティ・チェックは1セクタごとに行われる。書込み動作のとき、64語のデータのうち1語が奇数ビットで構成された語の個数を数え、セクタの最後にこの個数をパリティ・ワードとして、7ビット構成でディスク装置に書き込む。読出し動作の場合も同様に、読取りデータの1セクタ中の奇数ビットで構成された語の個数を数え、これとパリティ・ワードを比較する。不一致の場合はリード・エラー・フロップがセットされ、読取りエラーとなる。

ブロック転送をハードウェアが実行するので、磁気ディスクの制御プログラムは比較的簡単であって、エラー・チェックと割込み処理がその主要な役割りである。ディスク・アドレスとコア・アドレスはt-unitに書き込んで転送要求を出すようにする。読取り、書込みの区別もt-unitのオーダ・コードで指定する。

制御プログラムではこのt-unitを受け取って、DMAチャンネル機構に物理的に指令を出す。この際装置のビジー・チェックやディスク・アドレスが妥当なものかどうかをチェックし、エラーがあればタイプライタを介してオペレータを呼び出す。転送終了時には割込み処理を行って、t-unitを転送要求を出した主プログラムに返す。

実験用システムでは、主記憶装置が12Kバイトと小さいため、いろいろなプログラムをディスク装置に入れておいて、必要なプログラムだけを主記憶装置に持って来て動作させるようになっている。

#### 4.4 簡易図形処理装置

TSSによって簡単な図形処理が行えるように、電磁誘導形図形入力装置(タブレット)とプロッターがNEAC-M4に接続されている。タブレットは交番磁束を発生するマグネチック・ペンと、グレイ・コード状にセンス・コイルをはりめぐらした入力面で構成されている。ペンを入力面にあてると、電磁誘導によりセンス・コイルに起電力が発生する。この誘導起電力は、ペンがセンス・コイルの内にあるときと外にあるときとで逆極性になるので、この極性を判別してペンの位置をコード化する。<sup>4)</sup>

タブレット制御プログラムでは、タイマーを用いて10 msecごとにサンプリング入力を行いペン先の動きを追跡する。このときノイズの混入を避けるため2度読みを行い、一致しないデータは読み捨てる。また入力されるのはグレイ・コードであるので、これを2進数に変換して記憶装置に入れる。入力ビット数はx方向、y方向共にそれぞれ8ビットである。

タブレット入力の制御プログラムの動作には1点入力モードと連続読取りモードの2種類があって、主プログラムからのt-unitのオーダ・コードにより、何れかが指定される。1点入力モードは1点の座標値だけを入力するもので、図形要素のピックアップやタブレットの一部をマルチ・キーボードの代用とするときのキーの位置の検出に用いられる。これに対して連続読取りモードでは、ペンの動きを逐一追跡し曲線を一連の点列の座標値として入力するものである。

なお、TSSにデータを送る時には、2進数のままでは通信制御符号との混同が起るので、10進数(ASCIIコード)に変換してからメッセージとして転送する。記憶容量の小さいミニコンピュータでは図形処理が困難なため、多くの場合データはTSSに転送されて処理される。<sup>5)</sup>

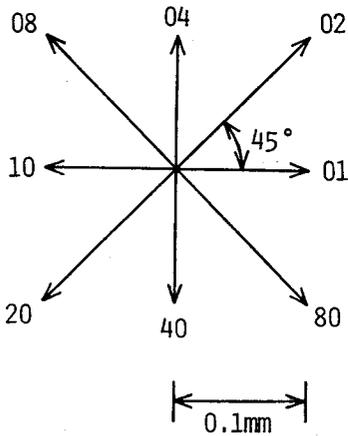


図4.5 方向指定コード (16進法)

出力命令BTOを1回実行すると、プロッタは0.1mmを1ステップとしてペン先を動かす。移動方向はアキュムレータにセットされたコードにより図4.5に示す8方向の中の一つが選択される。プロット速度は400ステップ/秒である。ペンのアップ、ダウンはFUN命令によって実行される。BTO命令あるいはFUN命令の実行後、ペンの移動が完了すると割込み信号が発生するので、制御プログラムではこの機能を利用して次の動作に移る。

プロッターの制御プログラムに要求される最も基本的な機能は、2点を結ぶ直線近似を求めこれに沿ってペンを移動させること(直線補間)である。主プログラムから直線補間のプロット動作を要求するときのt-unit

を図4.6のように定める。

図4.6で( $p, q$ )が現在のペンの位置から見た直線の端点の座標値で、ペンを現在位置からその点まで直線補間で移動させる。座標値( $p, q$ )は0.1mmを単位とする整数で指定する。ペン先の状態はオーダ・コードにより以下の3種が指定できる。

オーダ・コード=1のとき ペン・ダウンでプロット

POINTER AREA FOR CHAINING	
t-unit STATUS	PROGRAM NO.
ORDER CODE TO RECEIVER	
RESPONSE CODE	PROGRAM NO.
I	RETURN POINT TO SENDER
$p$ (binary 16 bits)	
$q$ (binary 16 bits)	
WORK AREA	

図4.6 直線補間要求のt-unit

オーダ・コード=2のとき ペン・アップでプロット

オーダ・コード=3のとき 破線でプロット

直線補間の計算手順を以下に述べる。いま点  $P_e(p, q)$  の  $x$  軸からの偏角  $\theta$  が  $0^\circ \leq \theta \leq 45^\circ$  の場合を考えると、

$$p \geq 0, \quad q \geq 0, \quad p \geq q \quad (4.1)$$

が成り立つ。ペン先は図4.5に示した8方向にしか移動できないため、 $x$  軸方向に  $p$  ステップ動き、その中の  $q$  ステップだけが偏角  $45^\circ$  の方向に動く。移動方向の指令コードは  $x$  軸方向のときは16進数"01"、偏角  $45^\circ$  の方向のときは16進数"02"である。指令コードの選択は直線からの距離が短いような格子点(0.1 mmきざみ)を取るように二者択一を行う。

図4.7において、任意の点  $P(x, y)$  から直線  $OP_e$  に下した垂線の長さ  $h$  は

$$h = \frac{py - qx}{\sqrt{p^2 + q^2}} \quad (4.2)$$

である。ここで符号にも意味を持たせて、点  $P$  が直線  $OP_e$  の上方にずれているときは正、下方にずれているときは負である。二者択一を行う際、式(4.2)の分母は共通であるので、分子だけを取り上げ

$$k = py - qx \quad (4.3)$$

によって直線からのずれを評価するものとする。

ペン位置が原点にある初期状態では  $k = 0$  である。いまペン先が第  $i$  ステップまで進んだ位置での  $k$  の値を  $k_i$  とすると、第  $(i+1)$  ステップ目においては

$$\begin{aligned} x \text{ 軸方向に動くとき} \quad k_{i+1}^{(1)} &= k_i + (-q) \\ \text{偏角 } 45^\circ \text{ 方向に動くとき} \quad k_{i+1}^{(2)} &= k_i + (p - q) \end{aligned} \quad (4.4)$$

となる ( $x$  あるいは  $y$  は単位ステップだけ動く)。  $k_{i+1}^{(1)}$  と  $k_{i+1}^{(2)}$  の絶対値をとって大小を比較し、小さい方のペンの動きを採用する。これに従い式(4.4)の  $k_{i+1}$  も何れか一方に定められる。終点位置  $P_e(p, q)$  に至るまでこの手順を反復して直線補間を行う。

ここに述べた方法では、式(4.2)あるいは式(4.3)を直接計算する必要がなく、 $k$  の各ステップごとの増分値  $\Delta k$  として  $-q$  または  $(p - q)$  を加え込んで行くだけで二者択一の評価ができる。ハードウェアで乗除算機構を持たないミニコンピュータでは有効な方法である。直線の偏角が  $45^\circ$  以下の場合について述べたが、一般の偏角の場合も計算手順は同様であって、二者択一で定められる方向指定のコードが変わるだけである。

プロッター制御プログラムでは上述の計算を実行し、割込みを利用してペン動作の制御を行い、以下のエラー状態を検出する。

- スイッチ・オフ
- 右スケール・オーバー
- 左スケール・オーバー
- ペーパー・エンド

エラーが検出されると、タイプライタにメッセージを印字してオペレータに注意をうながす。この基本

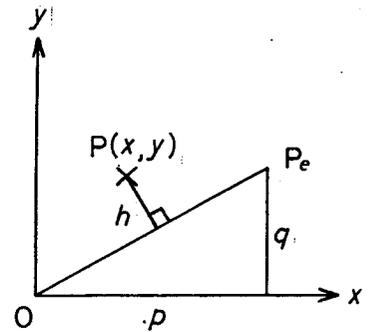


図4.7 直線への距離

の制御プログラムを利用して、プロッターは図形出力、NC 工具軌跡の検証、計測データのプロットなどに用いられる。また、タブレットと共に用いて、第1章で述べた単純形状の部品のNCテープ作成を、蓄積管を用いるのと同様の方法で実行できる。

#### 4.5 アナログ入力インターフェース

多くの計測器機の出力はアナログ量であるので、これを計算機に読み込むためにはA-D変換器を必要とする。このため、NEAC-M4のアナログ入力インターフェースを作製した。そのブロック図を図4.8に示す。用途が多岐に及ぶことを考慮して、汎用性のある構成になっている。入力端子は16チャンネルあり、その中8チャンネルについてはmV入力用に電圧増幅1000倍の増幅器が取り付けられており、他の8チャンネルにはピーク・ホールド回路が装着できるようになっている。データを読み込むときには、アナログ・マルチプレクサが働いて16チャンネルの中の1チャンネルが選択される。

A-D変換器の入力レンジは-10V~+10Vであり、出力は符号を含めて10ビット(負の時は2の補数)である。A-D変換器の入力では入力レンジを一杯に使うことが望ましいので、アナログ・マルチプレクサの後ろに可変利得の増幅器が設けられている。ピーク・ホールド回路のモード設定、マルチプレクサのチャンネル選択、可変利得のゲイン選択は何れもFUN命令を使うことによって、プログラムで制御できる。制御プログラムで使用される命令を表4.5に示す。

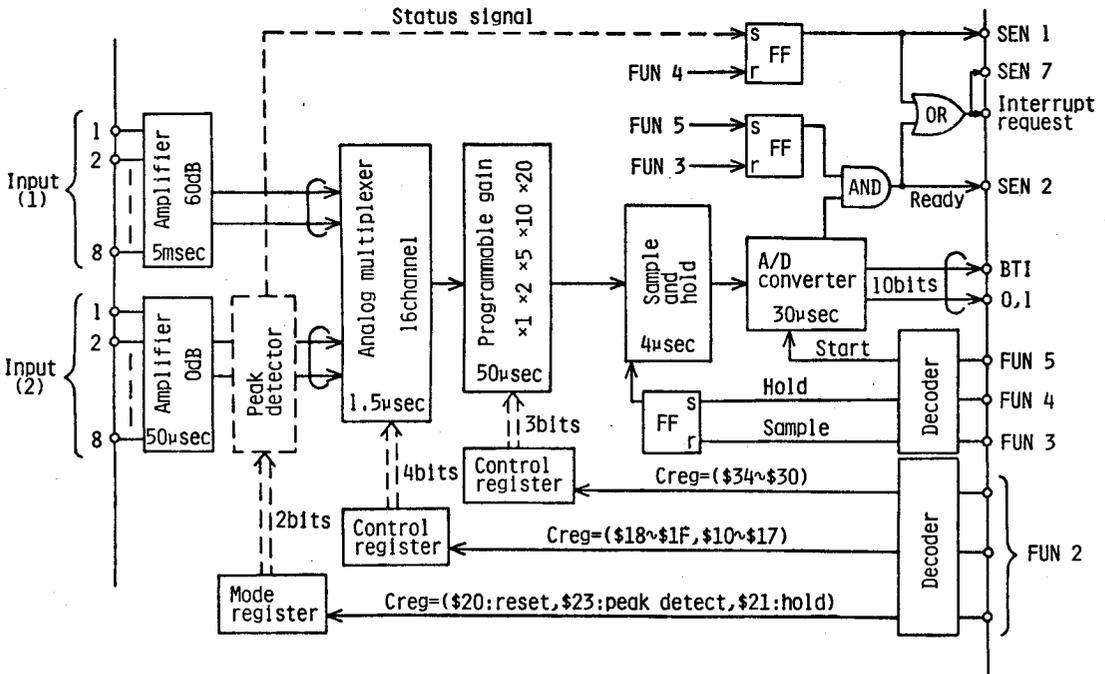


図 4.8 アナログ入力ブロック図

表 4.5 アナログ入力制御命令

コード	機能
BTI	0 A-D変換器下位8ビット入力
	1 A-D変換器上位2ビット入力
FUN	0 制御部リセット
	1 割込要求リセット
	2 コントロール・レジスタ設定
	3 サンプル・モード設定
	4 ホールド・モード設定
	5 A-D変換開始
	6 割込み禁止
7 割込み禁止解除	
SEN	1 ピーク検出
	2 A-D変換終了入力レディ
	7 割込み要求検知

表中 FUN2 命令によって、アキュムレータに格納されたコードがコントロール・レジスタに移され、上述のチャンネル選択などが実行される。A-D変換器を動作させるには FUN3, FUN4, FUN5 の命令をこの順に続けて実行すればよい。変換時間は  $30\mu\text{sec}$  であるので、この時間待ちは通常割込みを使用せず、SEN2の命令を反復実行しセンス・ループで待つことになる。サンプリング・インターバルはタイマーを用いてプログラムで制御する。タイマーを用いてサンプリングの時刻を知るのにタイマーからの割込みを利用する。1回のサンプリングで複数個のチャンネルから入力したい場合があるが、自動的にチャンネルを走査する機能はインターフェースには含まれていないので、これもプログラムによって遂行する。

制御プログラムの基本のサブルーチンとして、チャンネルと可変利得の倍率を指定すると一点入力を行うものが用意されている。さらに、これを基にしてタイマーのサンプリング・インターバルとデータの格納場所を指定すると、サブルーチンをくり返し使用し、データを連続して入力するプログラムが作られている。後者のプログラムは、比較的速い現象をとらえるため、データ入力と演算を独立して行うのに適している。低速の現象で1点を入力しては演算を実行する場合には、前者のサブルーチンがあれば充分である。回路についての詳しい説明は第2編で述べる。

#### 4.6 オンライン DNC システムの統合

数値制御装置 (NC 装置) にはオフライン・モードとオンライン・モードを切り換えるロータリ・スイッチが設けられており、後者のモードでは NC 指令をミニコンピュータ NEAC-M4 から転送できるようになっている。NC 装置がデータ要求状態になると、割込み信号が発生するので、この時刻をとらえて NEAC-M4 主記憶上の NC 指令を転送する。この割込み処理を実行するために、NC 装置についても他の周辺装置と同様に基本の制御プログラムを用意した。制御プログラムと主プログラムとのパラメータの授受は t-unit を介して行っている。

この制御プログラムでは、単に NC 装置にデータを転送するだけでなく、NC 指令をタイプライタに印字する機能を備えている。これによってオペレータは NC 指令の検査を行うことができる。ただし、この機能はセンス・スイッチによって割愛することができ、タイプライタ印字による時間遅れが加工の過程に影響するときにはこの機能を省略する。また逆に NC 指令の検査だけをオペレータが実施したい時には、タイプライタへの印字のみを実行し NC 装置への転送を省略することもできる。

割込み処理について特徴的なことは、NC 指令転送後、工具が動いている間は割込み信号が出ないので、この間に次の NC 指令を準備することができる。したがって、原則として NC 指令の転送が終わった所で t-unit を主プログラムに返す。ただし、とくに主プログラムの方で割込み発生の特定点を知りたいときには、次の割込要求が出るのを待ってから t-unit を返す。両者の機能の使い分けは、t-unit のオーダ・コードで指定することによって遂行される。

NEAC-M4のデータ処理能力を活かして、NC指令をNEAC-M4内部で生成する場合もある。<sup>6)</sup>  
 NEAC-M4とNC装置を一体として扱い、適応制御機能を持たせて加工の過程と並行してNC指令を修正して行くような場合はこの代表的な例である。NC指令コードの一覧表を表4.6に示す。

表 4.6 NEDAC 4200 指令コード一覧表

G 機 能	G00	位置決め、直線切削	独立1軸選択、工具径補正無効	
	G01	直線補間	始点からみた終点座標X、Yと円弧中心I、Jを $\mu\text{m}$ 単位で指定	
	G02	円弧補間(時計方向)	但し最後の桁は0または5	
	G03	" " (反時計方向)		
	G04	ドゥエル(G04X10000)	X指定でミリ秒単位、この間停留	
	G05	送りの一時停止	START釦で送り再開	
	G17	XY平面選択	例えばG18のとき、Xの指令でZ軸が動き、Yの指令でX軸が動く。省略したときはXY平面	
	G18	ZX " "		
	G19	YZ " "		
	G 機 能	G40	工具半径補正キャンセル	X軸、Y軸の順に工具が動く 補正量はNC装置のパネルで設定
G41		工具半径補正(左側)		
G42		" " (右側)		
G43		" " (ベクトル指定)		
M 機 能	M00	プログラム ストップ	START釦で再スタート オプション指定はNC内部スイッチ 紙テープ最後の指令	
	M01	オプション ストップ		
	M02	エンド オブ プログラム		
	M03	スピンドル ON	} 手動操作でも可能 (スピンドルは時計方向回転)	
	M05	" OFF		
	M08	クーラント ON		
	M09	" OFF		
	M13	スピンドル、クーラント共にON		
	M 機 能	M51	スピンドル回転数 M51:100 M52:200	
		↳	(rpm) M53:400 M54:520	
M56		(手動操作も可) M55:1040 M56:2080		
F 機 能	F0	送り速度(mm/min) F0:2400		
	}	F1:100~1000 F2:50~500		
		F3:40~400 F4:20~200		
		F5:10~100 F6:5~50		
	F9	F7:4~50, F8:2~20, F9:1~10		

これまで述べて来た通信回線や周辺装置を接続したシステム構成図を図4.9に示す。このシステムでは、大型計算機、ミニコンピュータ、NC装置ならびにその他の周辺装置を有機的に結合することにより、加工の工具径路を求めることから、加工の適応制御、工作物の精度検査に至るまでを首尾一貫して遂行することを目的としている。APTによるパート・プログラムの処理やFORTRANによるカム輪郭の計算などはミニコンピュータで処理できないので、大型計算機のTSSによって処理され、計算結果はファイルに格納される。パート・プログラムそのものもファイルに格納されているので、これについての修正や加工条件の変更などがNEAC-M4の端局から容易に実行できる。TSSでNCテープ・イメージが作られたときは、これをオンラインでNC装置に転送して加工を実施することも可能であるし、紙テープに出力

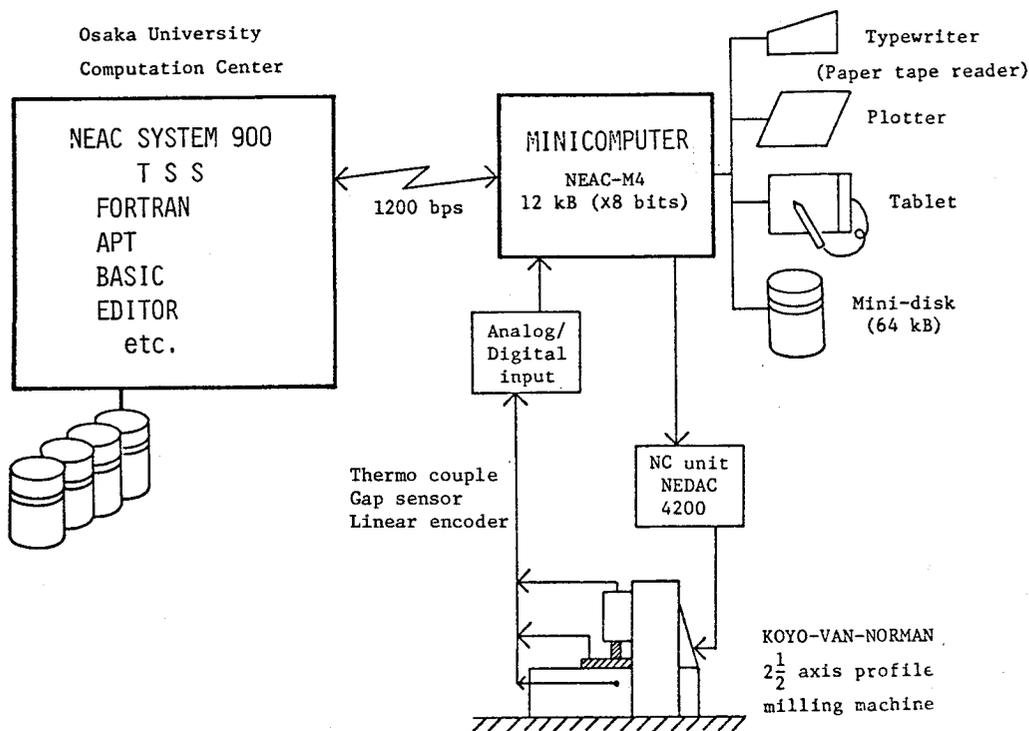


図 4.9 システム構成図

することも可能である。また、プロッターを用いて工具軌跡を検査しながらNCテープ・イメージを修正する手段もある。

先に述べたように適応制御機能を遂行したり、リニア・エンコーダを用いてテーブルやヘッドの位置を監視しながら加工を行うときにはNEAC-M4とNC装置はCNC (Computer Numerical Control) システムとして動作する。これらについて第2編の応用例でくわしく述べる。以上のように、NEAC-M4は大型計算機の処理能力や大容量のファイルが何時でも使える状況の下で、加工過程に密着した制御機能を受け持つ。したがって、多種多様な制御プログラムを働かせることになるが、第3章で述べたプログラム構成法を用いることによって、これが容易に実現できる。各ファンクション・プログラムは独立にアSEMBルされ、相互のパラメータの受け渡しは実行時にt-unitによって行われる。システム・ジェネレーションは不要であるから、ファンクション・プログラムのライブラリを作っておいて、その都度必要なものだけを主記憶装置に持って来ればよい。一部のプログラムの差し換えが容易に実行できるという利点を活用する。基本的なプログラム・モジュールが主記憶装置に占める大きさを表4.7に示す。

表4.7に示したモニターは、第3章で述べたt-unitの流れを制御する管理プログラムで、ディスパッチャ、キュー管理、割込解析処理の機能ならびに管理テーブルを含む。算術演算プログラムは、CNC装置として工具径路の計算に要求される高精度演算を実行するために作成したもので、有効桁数約10進13桁の浮動小数点演算プログラムである。この中にはsine関数、cosine関数なども含まれる。これら三角関数の計算には級数展開を用いており、これらの係数は文献7)に依っている。アナログ入力インターフェースやリニア・エンコーダからのデジタル入力は使い方が多様であるため、応用プログラムの中のサブルーチンとして制御プログラムに組み込む。したがって、独立したモジュールになっていない。

表 4.7 プログラム・モジュールの大きさ

モジュール名	大きさ (バイト)
モニター	1.5 KB
タイプライタ制御	600 B
紙テープ・リーダー制御	600 B
ディスク装置制御	260 B
DNC転送制御	400 B
タブレット制御	1 KB
プロッタ制御	1 KB
通信回線制御	2 KB
算術演算プログラム	3 KB

ため、表 4.7 には掲げていない。

#### 4.7 結 言

本章では第 3 章で述べたソフトウェアの構成法に基づいて、ミニコンピュータ NEAC-M4 における制御プログラムの組み立て方について具体的に述べた。これから明らかなように、NEAC-M4 はオンライン DNC システムのコントロール・センターとしての役割を持っており、その役割は大別して二つに分けられる。

- (1) TSS 端末としての働きによって、大型計算機の処理能力と大容量ファイルを効果的に利用する。
- (2) NC 装置と一体となった CNC 装置として、実時間で NC 指令を生成し、工具径路の修正などを行う。

これらの機能を 1 台のミニコンピュータで実現することにより、部品形状の計算、NC テープの作成から加工の適応制御までの一貫したシステムが構成される。そしてそれぞれの過程においていろいろの周辺装置が有効に利用されている。本章で述べた制御プログラムを用いると緒言で述べた多くの機能が具体的に実現できることが示された。

#### 参 考 文 献

- 1) 牧之内三郎、山県敬一、他：DNC システムの試作、精機学会春季大会学術講演前刷 (1972)、pp. 421-422.
- 2) 牧之内三郎、山県敬一：TSS 端末用ミニコンピュータの制御プログラム、情報処理学会大会予稿集 (1973)、pp. 347-348.
- 3) NEAC-M4 命令、日本電機協。
- 4) 井口征士、十倉健二、桜井良文：電磁誘導形図形入力装置、電子通信学会論文誌、Vol. 56-D, No. 2 (1973)、pp. 93-98.
- 5) 堀野 均、牧之内三郎、山県敬一：図形入力装置を用いた会話型プログラミングの試み、精機学会総合生産システム・シンポジウム (1974)、pp. 85-88.

- 6) 山県敬一、牧之内三郎：DNCシステムによる複雑な形状部品の輪郭切削について、精機学会関西地方定期学術講演会前刷（1974）、pp.137-140.
- 7) 山内二郎、宇野利雄、一松 信：電子計算機のための数値計算法Ⅲ、培風館（1972）、第3章、pp.45-72.

## 第 2 編

# 板カム輪郭フライス切削における 工作物熱変形の適応制御

# 第1章 輪郭フライス切削における工作物への流入熱量の推定

## 1.1 緒言

数値制御工作機械の普及により、広く加工の自動化・省力化が進められている。その一面において、単なる省力化に止まらず、生産効率の向上や加工精度の向上に対する強い要望があり、小型電子計算機の低価格化と相まって、最適化制御や適応制御の試みがいろいろ実行されている。加工精度の向上を考えると、おろそかにできない要因の一つに機械・工具・被削材系の熱変形問題がある。<sup>1)</sup> これらの中で、工作機械の熱変形については、位置決め精度をも劣化させるものであり、多くの研究が報告されている。<sup>2)-8)</sup> しかし、切削加工における工具・被削材系の熱変形については十分な研究が行われているとはいえない。

本研究では、エンドミル側面による輪郭フライス切削における被削材の熱変形問題を取り上げる。切削対象は板カムモデルである。ここで工作機械、工具(エンドミル)、被削材それぞれの熱変形問題が異なった性質を持ち、独立した要因として扱えることを明らかにしておく。被削材の熱膨張は切削熱によるものであり、切削開始当初は切削点近傍の温度上昇はあるものの、全体の熱膨張には至らない。その後数分ないし数十分の時間の経過と共に全体の熱膨張を起し、その量は次第に増加する。したがって、工具径路の補正を行わない場合、輪郭切削の継続と共に過切込みを生じ、形状精度劣化の一つの要因となっている。これに対し、工具として直径20mm程度のエンドミルを考えると、工具自体の熱容量はさ程大きくないため、切削当初から熱膨張を起し、実質的な工具径に変化をもたらす。この影響は寸法精度劣化の一つの要因である。また、工作機械の熱変形はその影響が数時間の時間の経過と共に現れるので、一つの輪郭を切削する数十分の間隔での相対的な変化は無視できると考えてよい。工作機械の熱変形については、第4章で具体的な数値を示す。

被削材の熱変形があまり取り上げられていないのは、切削油を用いて仕上げ過程を入念に行えば、小さく抑え込む事が可能であることによるが、生産効率の向上と加工精度の向上の両面を考えると避けて通れない問題である。また輪郭切削では被削材の仕上げ形状が変化に富んでおり、その一つについて有限要素法を用いることなど、計算機使用料のコストが高くつくおそれはある。しかしながら、マスタカムを試作するときなどには設計計算も計算機を用いて行われるので、その一部として熱変形解析を行うとすれば、それ程大きな負荷ではないと考えられる。

被削材の熱変形を考えると、流入熱量をどのように見積るかが重要な問題である。輪郭フライス切削において切削加工時に、切削点から工作物へ流入する熱量を把握することができれば、有限要素法などの熱変形解析により、工作物周上の点の熱による変位を求めることができる。ところで、2次元切削の場合は工作物への流入熱量が理論的に解析されているが、<sup>9)</sup> フライス切削のような3次元切削では理論的解析が困難である。しかしながら、フライス切削における切屑生成のための切削力については理論的解析が可能である。この点を考慮し、工具、工作物、切屑への熱の流れについては2次元切削モデルを適用し、これに必要な切削抵抗についてはフライス切削の切削力を用いて工作物流入熱量の式を導く。

ここでは板カムの輪郭切削を具体的に考えているので、フライス切削としてはエンドミル側面による上向き切削を考える。したがって、フライスのモデルは円筒フライスである。

理論式を導いた後、何種類かの加工条件の組み合わせにより実験値との比較を行う。ただし、加工実験を行ったとしても、工作物への流入熱量を実測することは極めて困難である。この間隙を埋めるため、差分法による熱伝導解析のプログラムを作成する。加工実験の際工作物のいくつかの点で温度を測定するこ

とはできるので、差分法のシミュレータにより試行錯誤的に流入熱量をかえながら実測温度との一致点を探し、流入熱量の推定を行うのである。

以上述べたように、切削モデルによる理論解析、差分法によるシミュレーション、加工実験による温度測定、これらを統合して工作物への流入熱量を把握しようとするのが本章の目的である。

## 1.2 理論解析

### 1.2.1 切屑生成による切削力

まず3次元モデルのフライス切削における、切屑生成にかかわる切削抵抗を求める。<sup>10)</sup> 図1.1において、1刃あたりの送りを  $S_z$ 、切削厚さを  $h$  とすると近似的に次式が成り立つ。<sup>11)</sup>

$$h \doteq S_z \sin \varphi \quad (1.1)$$

いま上向き切削を考えると、1刃あたりについての切屑生成仕事量を  $A_z$  (単位切削巾当たり) とすると、

$$A_z = \int_0^{\varphi_t} p_h \cdot h \cdot r \cdot d\varphi = \int_0^{h_t} p_h \cdot h \cdot r \frac{dh}{S_z \cos \varphi} \quad (1.2)$$

が成り立つ。ここに、 $r$  はフライスの半径、 $p_h$  は単位切削断面あたりでの切削力(比切削力)であるが、単なる定数ではなく切削厚さに関する。また、最大切削厚さ  $h_t$  は図1.1より

$$h_t \doteq S_z \sin \varphi_t = \frac{S_z}{r} \sqrt{r^2 - (r-t)^2} = 2 S_z \sqrt{\frac{t}{D} \left(1 - \frac{t}{D}\right)} \quad (1.3)$$

ただし、 $D$  はフライスの直径 ( $2r$ ) である。

式(1.2)の  $A_z$  は仕事量であるから熱量への換算も可能であるが、切屑、工作物、工具などへの熱の配分率が不明のため、この熱量を算出しても当面の目的には役立たない。そこでこれをもとに円周方向の

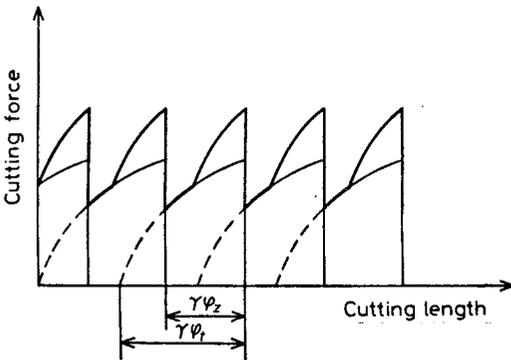


図1.2 切削力の周期的変動

平均切削抵抗を求める。フライス切削力の周期的変動(多刃切削の場合)を図1.2に示す。図1.2では一つの刃の切削が終わらないうちに、次の刃の切削が開始されている様子を示しており、初めの刃の切削力の上に次の刃の切削力を上のせた図になっている。ここに示された変動の周期  $r\varphi_z$  を用いれば、平均切削抵抗は単位幅あたり

$$P_t = \frac{A_z}{r\varphi_z} = \frac{1}{S_z \varphi_z} \int_0^{h_t} p_h \cdot h \cdot \frac{dh}{\cos \varphi} \quad (1.4)$$

である。

ところで、 $p_h$  は切削厚さ  $h$  に依存するが2次元切削における値から、基準切削力  $C_s$  とあるべき数  $\epsilon$  を用いて

$$p_h = C_s h^{-\epsilon} \quad (1.5)$$

とおくことができる。また式(1.1)より

$$\frac{1}{\cos \varphi} = \left\{ 1 - \left(\frac{h}{S_z}\right)^2 \right\}^{-\frac{1}{2}} = 1 + \frac{1}{2} \left(\frac{h}{S_z}\right)^2 + \frac{3}{8} \left(\frac{h}{S_z}\right)^4 + \dots \quad (1.6)$$

が成り立つ。したがって、式(1.4)の $h$ に関する積分が可能となる。また、フライスの刃数を $z$ 、送り速度を $f$ 、回転数を $n$ とすると

$$\varphi_z = 2\pi/z \quad (1.7)$$

$$S_z = \frac{f}{nz} \quad (1.8)$$

である。積分の近似計算に

$$\frac{h}{S_z} \leq 1, \quad \varepsilon \ll 1 \quad (1.9)$$

$$\frac{h_t^2}{S_z^2} \div \sin^2 \varphi_t = 4 \frac{t}{D} \left(1 - \frac{t}{D}\right) \quad (1.10)$$

を用いると、平均切削抵抗の基礎方程式として次式が得られる。<sup>10)</sup>

$$P_t \div \frac{2}{2-\varepsilon} \cdot \frac{f}{\pi n} \cdot \frac{t}{D} \cdot C_s h_t^{-\varepsilon} \quad (1.11)$$

式(1.11)をもとに、切削幅を $w$ として、エンドミル側面による円筒フライス削りにおける切削力を求める。式(1.3)を再び用いて、次式が成り立つ。

$$C_s h_t^{-\varepsilon} = \left\{ \frac{4t}{D} \left(1 - \frac{t}{D}\right) \right\}^{-\frac{\varepsilon}{2}} C_s S_z^{-\varepsilon} \quad (1.12)$$

いま、簡単のため切削厚さ $S_z$ における比切削力(=  $C_s S_z^{-\varepsilon}$ )を $p_s(S_z)$ で表すものとする。円筒フライス削り、とくに仕上げ過程では

$$\frac{t}{D} \ll 1, \quad \frac{\varepsilon}{2} \ll 1 \quad (1.13)$$

が成り立つので、結局、式(1.11)は切削幅 $w$ について

$$P_{tw} = \frac{2^{1-\varepsilon}}{2-\varepsilon} \cdot \frac{wf}{\pi n} \cdot \left(\frac{t}{D}\right)^{1-\frac{\varepsilon}{2}} p_s(S_z) \quad (1.14)$$

と書きかえられる。切削速度 $V_t$ を用いると、

$$V_t = \pi D n \quad (1.15)$$

であるから、

$$P_{tw} = \frac{2^{1-\varepsilon}}{2-\varepsilon} \cdot \frac{wft}{V_t} \left(\frac{D}{t}\right)^{\frac{\varepsilon}{2}} p_s(S_z) \quad (1.16)$$

と表される。<sup>11)</sup>

ここで記号の意味をもう一度明らかにして単位系を定める。

- $P_{tw}$  : 切屑生成にかかわる平均切削力 kgf
- $w$  : 切削幅 mm
- $f$  : フライスの送り速度 mm/min
- $V_t$  : 切削速度 ( $V_t = \pi D n$ ) mm/min
- $t$  : 切込み深さ mm
- $D$  : フライスの直径 mm
- $\varepsilon$  : 比切削力における切削厚さにかかるべき数
- $p_s(S_z)$  : 切削厚さ $S_z$ における比切削力 kgf/mm<sup>2</sup>

比切削力とべき数  $\epsilon$  については、切削幅がフライスの幅より小さい場合には旋盤削りの値を利用する<sup>10), 11)</sup>すなわち、

$$\begin{aligned} p_s(S_z) &= k_\delta \cdot k_\kappa \cdot p_{s0} \\ &= k_\delta \cdot k_\kappa \cdot C \cdot \sigma_B^{0.5} \cdot S_z^{-0.2} \end{aligned} \quad (1.17)$$

$$\epsilon = 0.2$$

で表され、 $k_\delta$ 、 $k_\kappa$  は刃先形状と材質にかかわる係数であり、 $\sigma_B$  は被削材の引張り強さ ( $\text{kgf}/\text{mm}^2$ ) である。文献 10) には、 $k_\delta$ 、 $k_\kappa$ 、 $p_{s0}$  の具体的な値が表の形式で与えられている。

### 1.2.2 2次元切削モデルにおける熱の流れ

第 1.2.1 節で述べた切屑生成のための切削力の大部分が熱に転化する。この熱の流れを解析するのにフライス削りの切削モデルでは極めて困難であるので、図 1.3 に示すようにこれを 2次元切削モデルにおきかえて考察する。切屑生成のための切削力は 2次元切削モデルでは、主にせん断面におけるせん断エネルギーとすくい面における摩擦エネルギーとして消費され熱に転化される。フライス切削では切削厚さが刃先の回転位置によって異なり、切削抵抗もそれに応じて変化する。また、刃数の少ないときには断続切削となる。このため発生熱量の大きさは周期的に変動するが、エンドミル側面による輪郭切削において工作物全体の熱的挙動を論ずる場合には、平均化された熱量が一樣に発生すると考えても支障はないであろう。ここでは熱の流れを考慮するのに図 1.3 (b) の 2次元モデルを用い、入力エネルギーとなる切削抵抗は第 1.2.1 節で述べた平均化された切削抵抗を用いる。

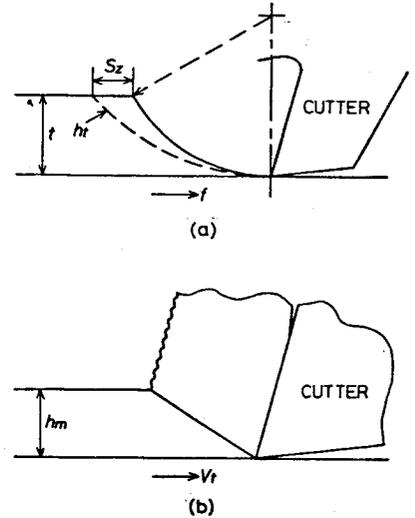


図 1.3 フライス切削モデルの 2次元化

第 1.2.1 節と重複するが、図 1.3 で  $S_z$  は一刃あたりの送りであり、送り速度、主軸回転数、フライスの刃数をそれぞれ  $f \text{ mm}/\text{min}$ 、 $n \text{ rpm}$ 、 $z$  とすると、

$$S_z = \frac{f}{nz} \quad (1.18)$$

と表され (mm)、また最大切削厚さ  $h_t$  は次式で与えられる。

$$h_t = 2 S_z \sqrt{\frac{t}{D} \left( 1 - \frac{t}{D} \right)} \quad (1.19)$$

ここで  $t$  は切込み深さ、 $D$  はフライスの直径である。2次元モデルにおける切込み深さ  $h_m$  はフライス切削における平均切削厚さを取るものとし、

$$h_m = \frac{1}{2} h_t = S_z \sqrt{\frac{t}{D} \left( 1 - \frac{t}{D} \right)} \doteq \frac{f}{nz} \sqrt{\frac{t}{D}} \quad (1.20)$$

とおくものとする。

図 1.4 に 2次元切削モデルにおける速度関係を示す。流れ型切屑生成機構には三つの速度成分、切削速度  $v_t$ 、切屑速度  $v_c$ 、せん断速度  $v_s$  があり、連続の条件よりこれらは閉じた三角形を形成する。

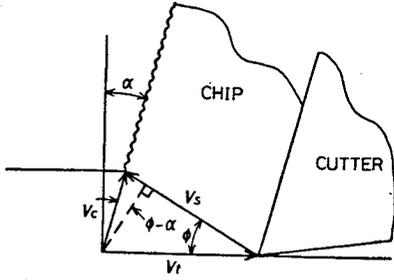


図 1.4 2次元切削の速度関係

図に示した幾何学的関係より次式が成り立つ。

$$V_c = \frac{\sin \phi}{\cos(\phi - \alpha)} V_t \quad (1.21)$$

$$V_s = \frac{\cos \alpha}{\cos(\phi - \alpha)} V_t$$

また、図 1.5 に切削力の平衡系を示す。この図において(1)水平分力  $F_h$  と垂直分力  $F_v$  の合力(またはせん断力  $F_s$  とせん断面垂直力  $F_n$  の合力)と、(2)すくい面摩擦力  $F$  とすくい

面垂直力  $F_m$  の合力がつり合っていないなければならない。

この関係から

$$F_s = F_h \frac{\cos(\phi + \psi - \alpha)}{\cos(\phi - \alpha)} \quad (1.22)$$

$$F = F_h \frac{\sin \phi}{\cos(\phi - \alpha)}$$

が成り立つ。ここで Lee-Shaffer の切削方程式

$$\phi = \frac{\pi}{4} - \psi + \alpha \quad (1.23)$$

が成り立つものとして、 $\psi$  を消去すると、

$$F_s = F_h \frac{1}{\cos \phi + \sin \phi} \quad (1.24)$$

$$F = F_h \frac{\cos(\phi - \alpha) - \sin(\phi - \alpha)}{\cos \phi + \sin \phi}$$

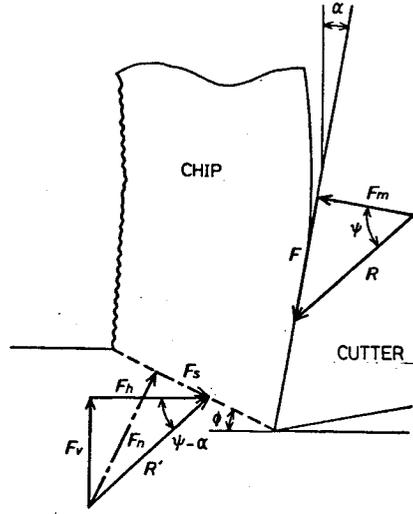


図 1.5 切削力の平衡系

を得る。式(1.21)と式(1.24)を用いて、せん断面

における単位時間あたりのせん断エネルギー  $U_s$  と、すくい面における単位時間あたりの摩擦エネルギー  $U_f$  は、切削方向分力  $F_h$  と切削速度  $V_t$  を用いて次のように表される。

$$U_s = F_s V_s = F_h V_t \frac{\cos \alpha}{\cos(\phi - \alpha)} \cdot \frac{1}{\cos \phi + \sin \phi} \quad (1.25)$$

$$U_f = F V_c = F_h V_t \frac{\sin \phi}{\cos(\phi - \alpha)} \cdot \frac{\cos(\phi - \alpha) - \sin(\phi - \alpha)}{\cos \phi + \sin \phi}$$

ここで  $F_h$  に対して、第 1.2.1 節で求めた  $P_{tw}$  (式(1.16))を用い、切削過程で消費されるエネルギーはすべて熱に変わるとすると、単位時間あたりにせん断面に発生する熱量  $Q_s$  とすくい面で発生する熱量  $Q_f$  は次式で与えられる。

$$Q_s = \frac{P_{tw} V_t}{J} \cdot \frac{\cos \alpha}{\cos(\phi - \alpha)} \cdot \frac{1}{\cos \phi + \sin \phi} \quad (1.26)$$

$$Q_f = \frac{P_{tw} V_t}{J} \cdot \frac{\sin \phi}{\cos(\phi - \alpha)} \cdot \frac{\cos(\phi - \alpha) - \sin(\phi - \alpha)}{\cos \phi + \sin \phi}$$

ただし、 $J$  は熱の仕事当量であり、 $V_t = \pi D n$  である。

これらの発生熱量は図 1.6 に示すように、切屑、工具、工作物に流入する。 $R_1, R_2$  はそれらの流入熱

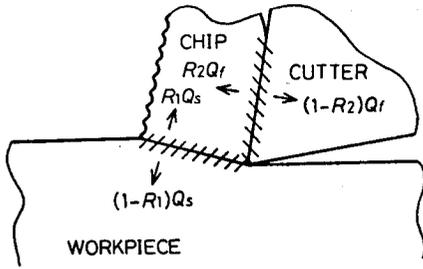


図 1.6 発生熱量の流れ

で与えられる。

### 1.3 工具と被削材の決定による式の具体化

直径 20 mm のエンドミル側面によるフライス切削で被削材 S35C を切削するものとし、これまで述べて来た理論解析から具体的に式を導く。ただし、加工条件として、切削速度  $V_t$  mm/min, 送り速度  $f$  mm/min, 切込み深さ  $t$  mm, 切削幅  $w$  mm ならびに主軸回転数  $n$  rpm などはパラメータとして残しておく。表 1.1, 表 1.2 に工具ならびに被削材に関する定数を示す。

表 1.1 工具に関する定数

材 質	高速度鋼 SKH3
ねじれ角	$30^\circ$
すくい角	$\alpha = 12^\circ$
くさび角	$\delta = 70^\circ$
送り角	$\beta = 8^\circ$
直 径	$D = 20$ mm
刃 数	2

表 1.2 被削材に関する定数

材 質	S35C
密 度	$\rho = 7.843 \times 10^{-6}$ kg/mm <sup>3</sup>
比 熱	$C = 111.6$ cal/(kg·°C)
熱伝導率	$\lambda = 0.738$ cal/(mm·min·°C)
ヤング率	$E = 2.1 \times 10^4$ kgf/mm <sup>2</sup>
ポアソン比	$\nu = 0.3$
線膨張係数	$\alpha = 1.12 \times 10^{-5}$ °C <sup>-1</sup>
引張り強さ	$\sigma_B = 56.5$ kgf/mm <sup>2</sup>
ブリネル硬さ	$H_B = 159$

切屑生成に関わる切削力  $P_{tw}$  は式 (1.16)

ならびに式 (1.17) によって与えられる。

ここで、式 (1.17) の各係数は文献 10) に与えられた表より

$$\begin{aligned} k_\delta &= 0.92 \\ k_\alpha &= 1.00 \\ C &= 26.9 \end{aligned} \quad (1.29)$$

とおく。とくに  $C$  は被削材が炭素鋼の場合について、 $p_{s0}$  (式 (1.17)) の表から逆算して求めたものである。 $\sigma_B$  は表 2 に与えられており、 $S_z = \pi D f / V_t z$  である。以上の値を用いて、結局次式が得られる。

$$P_{tw} = 143.1 z^{0.2} D^{-0.1} w t^{0.9} f^{0.8} V_t^{-0.8} \quad (1.30)$$

これを用いて 2 次元切削モデルにおけるせん断面での発生熱量を式 (1.26) の  $Q_s$  として求めるためには、せん断角  $\phi$  が明らかにされなければならない。ここでは被削材が鋼材であること、すくい角が  $10^\circ$  ないし  $15^\circ$  の範囲にあること、また切削速度が  $15$  m/min ないし  $50$  m/min の範囲にあることを考慮し、多数の実測データを掲げた文献 13) より、 $\phi = 27^\circ$  とおく。厳密にはせん断角は切削速度の影響を受けるがここでは一定と仮定する。仕事量の熱への変換に  $1 \text{ kgf} \cdot \text{mm} = 2.343 \times 10^{-3} \text{ cal}$  を用いると、式

量の割合を示す。 $R_1, R_2$  については M.C. Shaw の解析法<sup>12)</sup> によって求められており、たとえば

$$R_1 = \frac{1}{1 + 1.328 \sqrt{\frac{K_1 \gamma}{V_t t_1}}} \quad (1.27)$$

である。ここに  $K_1$  は温度伝導率、 $\gamma (= \cot \phi + \tan(\phi - \alpha))$  はせん断ひずみ、 $V_t$  は切削速度、 $t_1$  は切削厚さで、フライス切削モデルでは式 (1.20) の  $h_m$  を用いる。

結局、工作物への流入熱量は

$$Q_w = (1 - R_1) Q_s \quad (1.28)$$

(1.26) の  $Q_s$  は

$$Q_s = 0.252 z^{0.2} D^{-0.1} w t^{0.9} f^{0.8} v_t^{0.2} \quad (1.31)$$

となる。

この発生熱量の中、工作物に流入する熱量の割合は式(1.27)の  $R_1$  を用いて  $(1 - R_1)$  で与えられる。ここで、せん断ひずみ  $\gamma$  は  $\alpha = 12^\circ$  ,  $\phi = 27^\circ$  を用いて

$$\gamma = 2.231 \quad (1.32)$$

であり、また  $K_1 = \lambda / \rho C$  である。平均切削厚さ  $t_1$  mm は、式(1.20)より

$$t_1 = \frac{\pi D f}{z v_t} \sqrt{\frac{t}{D}} \quad (1.33)$$

を用いる。これらの値を式(1.27)に代入すると

$$(1 - R_1) = \frac{1}{1 + 0.0308 z^{-0.5} D^{0.25} t^{0.25} f^{0.5}} \quad (1.34)$$

となる。式(1.31)、式(1.34)を用いて、式(1.28)の  $Q_w$  は

$$Q_w = \frac{0.252 z^{0.2} D^{-0.1} w t^{0.9} f^{0.8} v_t^{0.2}}{1 + 0.0308 z^{-0.5} D^{0.25} t^{0.25} f^{0.5}} \quad (1.35)$$

で与えられる。

#### 1.4 差分法によるシミュレータ

加工実験を行ったとき、温度は実際に測定できるが、流入熱量の測定は極めて困難である。このため、切削加工時に外周から熱量が流入したときの工作物の温度分布を求めるシミュレーション・プログラムを差分法を用いて作成する。これを用いれば、測定点での温度が一致するように試行錯誤的に流入熱量をかえながら計算することによって、温度測定から流入熱量を逆に推定することが可能になる。

ここでは応用例として、板カムの輪郭切削を想定しているので、基礎実験は円板を用いて行った。ただし、工作物はバイスによって支持されているので、この支持部からの熱の流出が無視できないため、解析モデルとしては工作物と共に十分な熱容量を持った物体をあわせて考え、現実モデルに忠実なシミュレーションを行う。解析モデルを図1.7に示す。支持部以下の物体は、解析を容易にするため円柱状とし、底面の温度は室温に保たれており、その他の表面では空気への熱伝達があるものとする。また、工作物の支持部境界はバイスで締めつけてあるにすぎず、ここでの熱伝導には真実接触面積などに依存する大きな熱抵抗が存在するので、これも考慮に入れなければならない。なお、加工はエンドミル側面による輪郭切削を考えているので、熱源は図1.7の工作物円板部の外周を一定速度で動く移動熱源であり、任意の時刻での円板部の温度分布を求めるのがシミュレータの目的である。

工作物の分割図を図1.8に示す。半径方向、角度方向、 $z$ 方向それぞれに等分割とするが、一つの要素の温度をその中央の点で代表させるため、外界空気への熱伝達がある部分については、半径方向、 $z$ 方向の厚さを一般の要素の  $\frac{1}{2}$  に取ってある。また、半径方向については中心部も  $\Delta r / 2$  としてある。分割

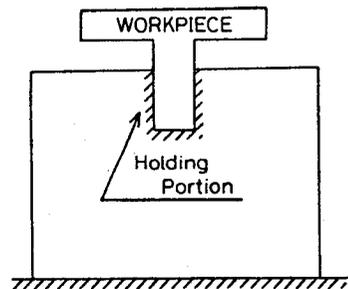


図1.7 差分法による解析モデル

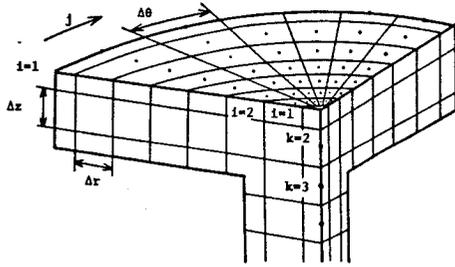


図 1.8 工作物分割図

された各要素の物理量を表すのに添字をつけるものとし、半径方向に添字  $i$ 、角度方向に添字  $j$ 、 $z$  方向に添字  $k$  を用いるものとする。一つの要素ごとに、各方向について要素の中央にある点を格子点とし、要素の温度をこの点で代表させる。

添字  $(i, j, k)$  を持つ要素について熱の流れを考え、 $\Delta \tau$  時間の中に温度  $T_{i, j, k}$  が  $T_{i, j, k}'$  に上昇したものとすると、差分式は以下の形式に書ける。ここでは簡単のため、いま考えている要素についての

添字  $(i, j, k)$  の代りに略して添字 0 をつけて書く。

$$\begin{aligned} \frac{C_0(T_0' - T_0)}{\Delta \tau} &= K_0^{(i+1)}(T_{i+1, j, k} - T_0) + K_0^{(i-1)}(T_{i-1, j, k} - T_0) \\ &+ K_0^{(j-1)}(T_{i, j-1, k} - T_0) + K_0^{(j+1)}(T_{i, j+1, k} - T_0) \\ &+ K_0^{(k-1)}(T_{i, j, k-1} - T_0) + K_0^{(k+1)}(T_{i, j, k+1} - T_0) \\ &- H_0(T_0 - T_b) + q_0 \end{aligned} \quad (1.36)$$

ここで、 $C_0$  は考えている要素の熱容量、係数  $K_0$  は単位時間、単位温度差あたりについて隣接する要素から熱伝導によって流入して来る熱量、係数  $H_0$  は空気への熱伝達によって流出する熱量 ( $T_b$  は外界温度)、 $q_0$  は熱源からの流入熱量を表す。ただし、空気との熱伝達がない要素では  $H_0 = 0$ 、熱源のない領域では  $q_0 = 0$ 、また  $K_0$  の中で隣接要素のない部分については零とおくものとする。

$C_0$  ならびに係数  $K_0$  は要素の形状によって異なるが、物体内部の要素について具体的に示せば、熱伝導率を  $\lambda$  として近似的に以下ようになる。

$$\begin{aligned} C_0 &= i (\Delta r)^2 \Delta \theta \Delta z c \rho \\ K_0^{(i-1)} &= \lambda \left( i - \frac{1}{2} \right) \Delta \theta \Delta z, \quad K_0^{(i+1)} = \lambda \left( i + \frac{1}{2} \right) \Delta \theta \Delta z \\ K_0^{(j-1)} &= K_0^{(j+1)} = \frac{\lambda \Delta z}{i \Delta \theta} \\ K_0^{(k-1)} &= K_0^{(k+1)} = \frac{\lambda i (\Delta r)^2 \Delta \theta}{\Delta z} \end{aligned} \quad (1.37)$$

式  $C_0$  に含まれる  $c$  と  $\rho$  は物質の比熱と密度を表す。係数  $K_0$  は格子点間の距離に反比例し、熱の流れに直交する要素間の接触面積に比例することから導かれる。空気への熱伝達項  $H_0$  については、外界空気との接触面積を  $dS$  とすると、 $H_0 = kdS$  で与えられる。 $k$  は空気と物質の間の熱伝達率である。熱源が二つの格子点の間にあるときは、熱量を格子点までの距離によって比例配分し、両者に熱の流入があるものとして取り扱う。

工作物支持部の境界面の一部はバイスに固定されており、ここでの熱伝導に関しては見かけ上の接触面積に比べて真実接触面積は十分小さいと考えられるので、この境界面では式 (1.36) を修正して、境界面に対応する係数  $K_0$  に未定定数  $\beta$  を乗じておく。これは真実接触面積と見かけ上の接触面積との比ならびに接触抵抗に対応するものである。これに対する具体的な値は、実測温度と計算値のつき合わせを行う際

に、円板部の周辺に近い温度と支持部に近い中心部の温度の両方において、実測値と計算値が合致するように定める。

材料定数は、工作物の部分については表 1.2 (第 1.3 節) の内容を用い、円柱部については鋳鉄のパスに対応させて次の表 1.3 の数値を用いた。また、空気への熱伝達率は室内無風状態の値

$$k = 3.0 \times 10^{-9} \text{ kcal}/(\text{mm}^2 \cdot \text{sec} \cdot ^\circ\text{C})$$

を用いた。<sup>2)</sup>

表 1.3 円柱部の物理定数

密度	$\rho = 7.27 \times 10^{-6} \text{ kg}/\text{mm}^3$
比熱	$c = 0.10 \text{ kcal}/(\text{kg} \cdot ^\circ\text{C})$
熱伝導率	$\lambda = 0.75 \times 10^{-3} \text{ kcal}/(\text{mm} \cdot \text{min} \cdot ^\circ\text{C})$

### 1.5 実験と考察

実験には半径 70mm 厚さ 10 mm の円板 (S35C) を用いる。加工条件をかえて切削実験を行い、図 1.9 に示す  $P_1, P_2, P_3, P_4$  の 4 点に熱電対を埋め込んで自動平衡記録計で温度測定を行う。切削は円板のま

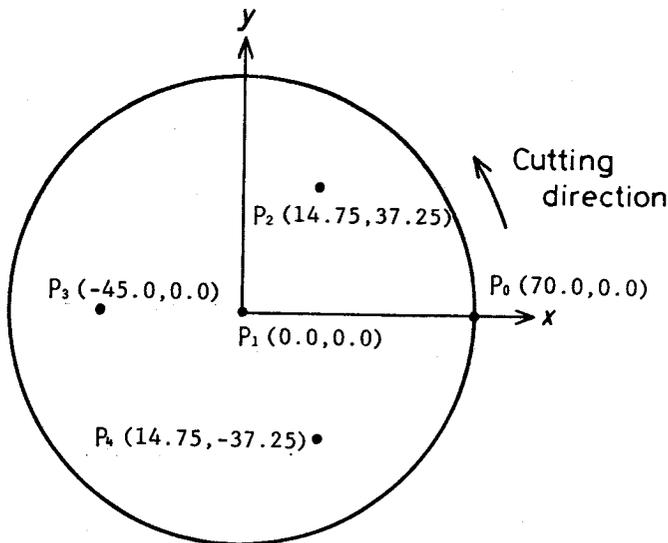


図 1.9 温度測定点

わりを一周して終了とする。上向き乾切削で、切込み深さ、切削速度、送り速度をパラメータとして変更する(工具、被削材質については表 1.1、表 1.2 を参照)。このときの 4 点の温度変化の例を図 1.10 に示す。

前述したように、流入熱量の測定は困難であるから、実験では温度測定のみを行う。一方、前節で述べた差分法のプログラムによるシミュレータを利用し、試行錯誤で流入熱量を与えて温度分布を計算し、実測温度との一致点を探す。基本的には流入熱量と温度上昇は比例関係にあるので、ある程度の近似値が把握できれば後は内挿計算で一致させることができる。なお、この場合に支持部境界面に未知のパラメータ  $\beta$  が挿入されているので、これについても試行錯誤により、中心点の温度変化と、外周に近い点での温度

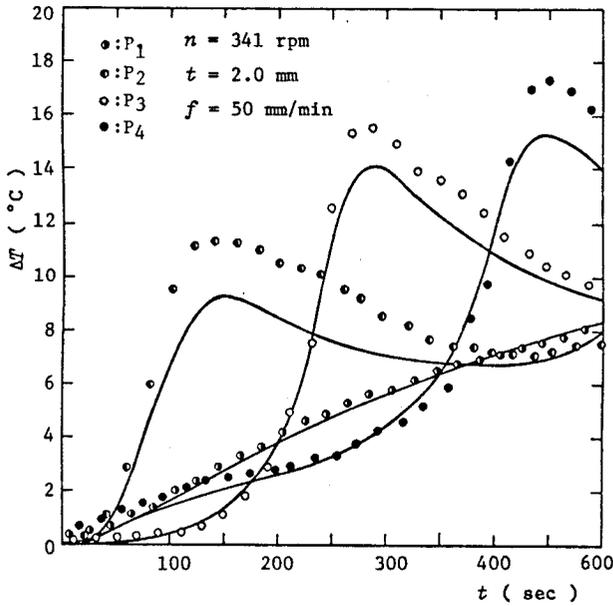
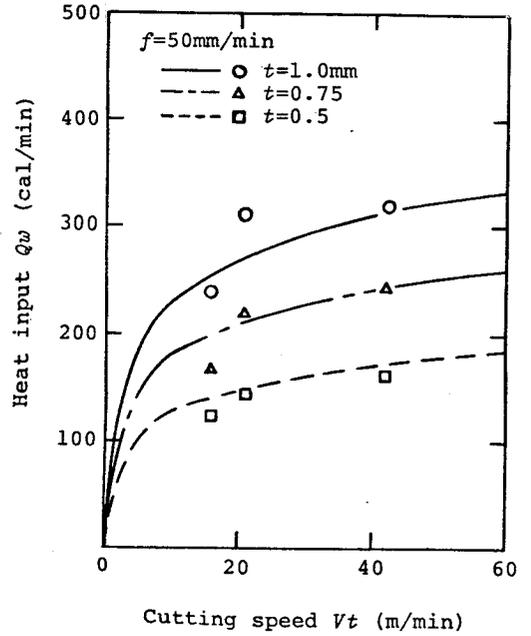
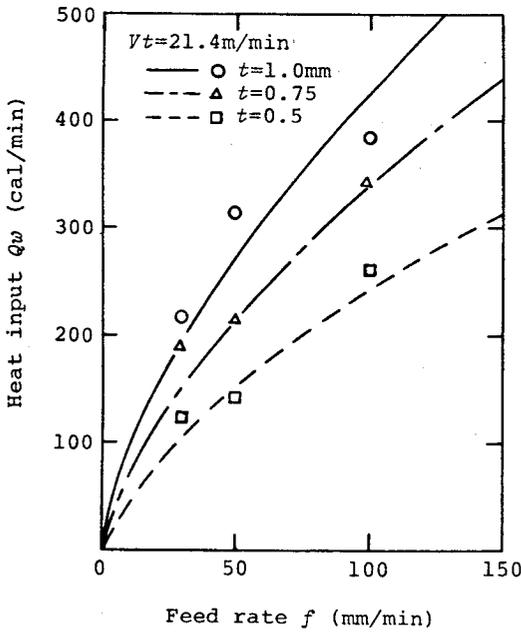


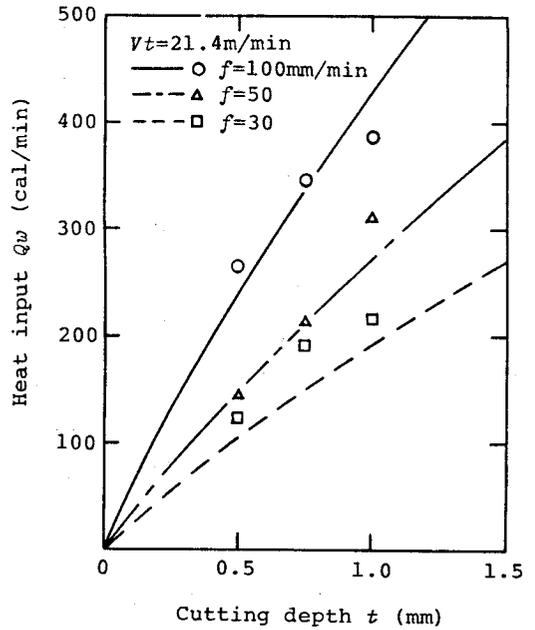
図1.1.0 測定点における温度変化の例



(a) 切削速度と流入熱量



(b) 送り速度と流入熱量



(c) 切込み深さと流入熱量

図 1.1.1 工作物への流入熱量

変化が共に実測値と計算値で合致するように定める。おおむね $\beta$ は0.04~0.16の値を取る。

第1.3節で求めた理論からの流入熱量の予測値と、上に述べた手順による実測温度からの推定値を合わせてプロットした図を図1.1.1に示す。実線、一点鎖線、破線によるものが理論からの予測値である。

図1.1.1の中で(a)は切削速度、(b)は送り速度、(c)は切込み深さとの関係をそれぞれ表している。理論から

の予測値と実測温度からの推定値で傾向は一致しているが、個々のデータについては差のあるものもある。このことは実験を行う際、刃先状態や実験室内の雰囲気など不確定な要素が介入することによるばらつきを表していると考えられる。

実際の所、同一の加工条件であっても工作物への流入熱量は一回の切削ごとに異なるであろう。したがって、工作物の熱変形を考慮して切削し、加工精度の向上を図ろうとするならば、切削加工時の工作物温度測定が不可欠であるといえる。ただ、適応制御を行うにあたって、事前に工作物への流入熱量がある程度分っている方が望ましいので、図 1.11 の曲線すなわち第 1.3 節の式 (1.35) をこの目的に使用する。

なお、厳密には理論解析の際、逃げ面での摩擦仕事を考慮すべきである。しかしながら、図 1.11 で見る限り摩擦仕事からの流入熱量を考慮しなくても実測温度からの推定値に傾向は一致しており、摩擦仕事からの流入熱量は切屑生成仕事からの流入熱量に比べて十分小さいと考えられる。予測式として式 (1.35) を用いることで十分であろう。

## 1.6 結 言

輪郭フライス切削の加工時における工作物への流入熱量は、工具の刃先状態あるいは加工時の雰囲気などの不確定要素が介入するため、事前に正確な値を知ることは困難である。しかしある程度の予測が可能ならば、予測と修正による適応制御が実施できる。実際の加工時に温度測定を行うことは可能であるが、熱変形を直接測定することは困難なので、予測修正法の方がよりよい精度が期待できる。以上の目的に沿って、工作物への流入熱量を予測する式の導出を試みた。

解析にあたってはいくつかの仮定やモデル化を行っている。たとえば

- 1) Lee-Shaffer の切削方程式が成り立つと仮定している。
- 2) 切削力の平衡系を 2 次元切削モデルで利用している。
- 3) 熱の流れについては M.C. Shaw の解析結果を利用している。

などである。厳密な解析を行うにはこれらの点もそれぞれ吟味が必要であろう。ただ、解析そのものが本章の目的ではないので、近似的に流入熱量の予測ができればそれでよいとしている。その意味で、導出した式によって実測温度からの推定値をある程度まで把握できるので、予測式としての効果は十分にあると考えられる。

## 参 考 文 献

- 1) 西脇信彦：機械・工具・被削材系の熱の流れについて、日本機械学会誌、Vol.81, No.721 (1978)、pp.1296-1301.
- 2) 奥島啓式、垣野義昭、菊池敏彦：工作機械の熱変形に関する研究（第 2 報）— 内部熱源による非定常熱変形 —、精密機械、Vol.38, No.7 (1972)、pp.565-571.
- 3) K.Okushima, Y.Kakino and A.Higashimoto: Compensation of Thermal Displacement by Coordinate System Correction, Annals of the C.I.R.P., Vol.24 (1975)、pp.327-331.
- 4) 竹内芳美、大久保信行、佐田登志夫：工作機械の熱変形制御（第 2 報）— 変動熱源に対する制御 — 精密機械、Vol.41, No.11 (1975)、pp.1084-1089.

- 5) T.Sata, Y.Takeucki and N.Okubo : Improvement of Working Accuracy of a Machining Center by Computer Control Compensation, Proceedings of the 17th International M.T.D.R. Conference(1976), pp.93-99.
- 6) 竹内芳美、大久保信行、佐田登志夫：温度分布測定による工作機械構造の実時間熱変形制御、精密機械、Vol.42、No11(1976)、pp.1043-1048.
- 7) 佐藤和信、竹内芳美、大久保信行、佐田登志夫：有限要素法を利用した機械構造内の熱源強さの同定法とその応用、精密機械、Vol.44、No12(1978)、pp.1452-1456.
- 8) 竹内芳美、佐田登志夫、佐藤 真、鈴木博一：計算機補正による数値制御工作機械の加工精度の向上（第2報）－環境温度変化に対する誤差補正－、精密機械、Vol.46、No12(1980)、pp.1532-1536.
- 9) 臼井英治：切削・研削加工学（上）、第1.2節、共立出版（1971）.
- 10) 益子正巳、阿武芳朗、森本公生 編：フライス削りとフライスの設計、第1.2節、大河出版（1972）.
- 11) 益子正巳：フライス削りの切削抵抗、精密機械、30巻、1号（1964）、pp.60-71.
- 12) E.G.Loewen and M.C.Shaw : On the Analysis of Cutting-Tool Temperatures, Trans. ASME, Vol.76, Feb.(1954), pp.217-231.
- 13) D.M.Eggleston, R.Herzog and E.G.Thomsen : Observations on the Angle Relationships in Metal Cutting, Trans. ASME, Journal of Engineering for Industry, Aug.(1959), pp.263-279.

## 第2章 有限要素法による工作物の温度分布推定

### 2.1 緒言

本章では、一般の板カムの加工を想定して、切削熱による工作物の温度分布を推定する方法について述べる。解法は有限要素法によるものであって、この計算機プログラムと、次章の熱変形解析のプログラムを合わせて、切削加工時の工作物の熱的挙動をシミュレートするシステムを構成しようとするのがここでの意図である。温度分布を求めることの必要性は、一つは熱変形解析に必要であること、もう一つは実際の加工過程において、切削中に温度測定を行い、シミュレーションによる予測値とのつき合わせをするためである。熱伝導解析だけならば第1章で述べた差分法でもよいが、ここでは同じモデルで熱変形解析も行うことと、板カム一般の加工を取り扱うため、同一のプログラムで任意形状を取り扱うことのできる有限要素法を用いる。

温度分布の解析において、以下の仮定をおく。

- 1) 材料の熱伝導特性は等方性である。
- 2) 物体への単位時間あたりの流入熱量は一定である。
- 3) 熱源の移動速度は一定である。

ここで扱う熱源は切削熱であり、たとえば2枚刃のエンドミルによるフライス切削では、切削点近傍に注目すれば熱の流入は断続的である。しかしながら、工作物全体の熱変形を調べるには、平均化された連続熱源が移動すると考えても支障はないであろう。また実際の加工にあつては、切削点での発熱量ならびにその移動速度(工具の送り速度)は、微視的に見れば時々刻々に変化していると考えられるが、以下の有限要素法ではこれらを一定とみなして取り扱う。

なお、切削油を用いることは考慮されていない。實際上切削油を用いることによって、熱変形量を小さくおさえることはできるが、皆無にすることは困難である。本研究ではCNC装置により、工具径路の修正を行って熱変形誤差を取り除くことを目標としており、切削過程のモデルは単純化して扱われている。

### 2.2 非定常熱伝導問題

一般に、物体内の任意の点 $P(x, y, z)$ の温度 $T$ が時間 $t$ と座標の関数 $T(t, x, y, z)$ で与えられるとすると、非定常熱伝導の微分方程式は次式で与えられる。

$$\rho c \frac{\partial T}{\partial t} = \lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + Q \quad (2.1)$$

ここに、

$\rho$  : 物体の比重量

$c$  : 物体の比熱

$\lambda$  : 物体の熱伝導率

$Q$  : 単位時間、単位体積あたりに発生する熱量(物体内部)

物体表面 $S$ における境界条件は、温度が $S$ の一部 $S_f$ で指定されているときは固定境界条件

$$T = T_f(x, y, z) \quad (2.2)$$

を与える。また、 $S$ の一部 $S_h$ で外部流体への熱伝達がある場合は、自由境界条件として次式を与える。

$$\lambda \left( \frac{\partial T}{\partial x} \nu_x + \frac{\partial T}{\partial y} \nu_y + \frac{\partial T}{\partial z} \nu_z \right) + h (T - T_b) - q = 0 \quad (2.3)$$

ここに、

$\nu_i$  : 境界面上における外向き法線の方向余弦

$h$  : 物体表面と周囲の相対熱伝達率

$T_b$  : 物体周囲の流体温度

$q$  : 物体表面の熱流束

である。また初期条件として、時刻  $t = 0$  における温度分布

$$T = T_0(x, y, z) \quad (2.4)$$

が与えられているものとする。

板カム加工のモデルについては、物体内部の発熱は存在しないので式(2.1)の $Q$ は零としてよい。切削熱は物体表面から流入して来るので、これは式(2.3)の $q$ で扱う。また周囲への熱伝達は原則として室内温度を持つ空気との間で行われる。切削油を用いるときは、この状況がかわってくるが、ここではこの問題は考えない。

以上の境界条件の下での、式(2.1)の汎関数はEulerの理論により次式で与えられる。<sup>1)</sup>

$$\begin{aligned} \chi = \int_V \left[ \frac{1}{2} \lambda \left\{ \left( \frac{\partial T}{\partial x} \right)^2 + \left( \frac{\partial T}{\partial y} \right)^2 + \left( \frac{\partial T}{\partial z} \right)^2 \right\} + \rho c \frac{\partial T}{\partial t} T \right] dV \\ + \int_{S_h} \left\{ \left( \frac{1}{2} h T^2 - h T_b T \right) - q T \right\} dS \end{aligned} \quad (2.5)$$

ここで、第1項は物体の体積全体の積分であり、第2項は表面 $S_h$ 上での積分である。また、非定常項 $\partial T / \partial t$ についてはある時刻においては位置のみの関数と考え、温度 $T$ の変分を受けないものとしている。式(2.1)を式(2.2)、(2.3)の境界条件の下で解くことは、汎関数 $\chi$ が極値を持つことと等価である。<sup>2)</sup>

$$\delta \chi = \frac{\partial \chi}{\partial T} \delta T = 0 \quad (2.6)$$

## 2.3 形状関数

考えている物体を有限個の要素に分割し、その代表的な要素 $e$ について考える。熱伝導解析は3次元問題として解く必要があるので、要素として四面体要素を採用する(図2.1)。

いま、要素 $e$ 内の温度 $T$ を次式で表すものとする。

$$T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \quad (2.7)$$

ただし、形状関数を論ずるにあたっては、ある特定の時刻における温度分布を考えるので、 $T$ は位置のみの関数としている。4個の節点 $i, j, m, p$ についても式(2.7)が成り立つとすると、

$$\begin{aligned} T_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 z_i \\ T_j &= \alpha_1 + \alpha_2 x_j + \alpha_3 y_j + \alpha_4 z_j \\ T_m &= \alpha_1 + \alpha_2 x_m + \alpha_3 y_m + \alpha_4 z_m \\ T_p &= \alpha_1 + \alpha_2 x_p + \alpha_3 y_p + \alpha_4 z_p \end{aligned} \quad (2.8)$$

が成り立つ。これを $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ に関する連立1次方程式

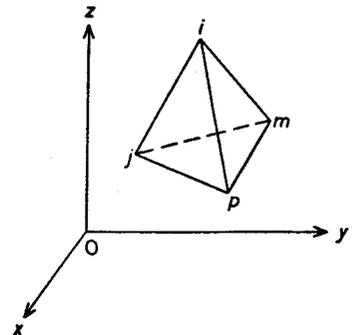


図 2.1 四面体要素

とみなして解き、その結果を式(2.7)に代入し、節点温度 $T_i, T_j, T_m, T_p$ についてまとめると、式(2.7)は

$$T = N_i(x, y, z)T_i + N_j(x, y, z)T_j + N_m(x, y, z)T_m + N_p(x, y, z)T_p \quad (2.9)$$

の形式に整えられる。式(2.9)は、節点での温度を用いて要素内の任意の点 $P(x, y, z)$ の温度を近似的に与えている。

具体的に $N_i$ などを求めると、次のようになる。

$$N_i = (a_i + b_i x + c_i y + d_i z) / 6V$$

$$6V = \begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_m & y_m & z_m \\ 1 & x_p & y_p & z_p \end{vmatrix} \quad (2.10a)$$

ここで $V$ は四面体の体積を表し、 $a_i, b_i, c_i, d_i$ は

$$a_i = \begin{vmatrix} x_j & y_j & z_j \\ x_m & y_m & z_m \\ x_p & y_p & z_p \end{vmatrix}, \quad b_i = - \begin{vmatrix} 1 & y_j & z_j \\ 1 & y_m & z_m \\ 1 & y_p & z_p \end{vmatrix} \quad (2.10b)$$

$$c_i = - \begin{vmatrix} x_j & 1 & z_j \\ x_m & 1 & z_m \\ x_p & 1 & z_p \end{vmatrix}, \quad d_i = - \begin{vmatrix} x_j & y_j & 1 \\ x_m & y_m & 1 \\ x_p & y_p & 1 \end{vmatrix}$$

である。 $N_j, N_m, N_p$ も、添字を $i, j, m, p$ の順序に循環して式(2.10a), (2.10b)から得られる。ただし四面体要素の節点の番号をつける順序は一定でなければならない。ここでは図2.1において、最後の節点 $p$ から見て他の3節点を反時計方向に $i, j, m$ の順序で番号つけることとする。

式(2.9)をベクトル表示すると、次式により節点の座標と温度から、要素内の温度を近似的に定めることができる。

$$T = \{N\}^T \{T\}^e = [N_i \ N_j \ N_m \ N_p] \begin{Bmatrix} T_i \\ T_j \\ T_m \\ T_p \end{Bmatrix} \quad (2.11)$$

ここで $\{N\}$ を形状関数と呼ぶ。

またこの $\{N\}$ を式(2.10)の形式でもう一度表現すると、

$$\begin{Bmatrix} N_i \\ N_j \\ N_m \\ N_p \end{Bmatrix} = \frac{1}{6V} \begin{bmatrix} a_i & b_i & c_i & d_i \\ a_j & b_j & c_j & d_j \\ a_m & b_m & c_m & d_m \\ a_p & b_p & c_p & d_p \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} \quad (2.12)$$

となり、 $\{N\}$ は一つの座標系を構成していると考えられる。いま、全く別の観点から、要素 $e$ の内部の点 $P(x, y, z)$ を節点の座標を用いて次のように表すものとする。

$$\begin{aligned}
 x &= L_i x_i + L_j x_j + L_m x_m + L_p x_p \\
 y &= L_i y_i + L_j y_j + L_m y_m + L_p y_p \\
 z &= L_i z_i + L_j z_j + L_m z_m + L_p z_p \\
 1 &= L_i + L_j + L_m + L_p
 \end{aligned}
 \tag{2.13}$$

ここで導入された座標 ( $L_i, L_j, L_m, L_p$ ) は体積座標と呼ばれる。たとえば  $L_p$  は要素内の点  $P$  が面  $i, j, m$  と作る小四面体と要素全体との体積比を表している。このような物理的意味から体積座標と呼ばれるのである。式 (2.13) を  $L_i, L_j, L_m, L_p$  について解くと、それぞれ  $N_i, N_j, N_m, N_p$  に一致することが示される。<sup>1)</sup> 四面体の四つの節点だけを採る一次要素で、要素内の温度を式 (2.7) で定義する場合には、体積座標そのものが形状関数を与えることになる。その他の一般の場合には、体積座標の多項式が用いられる。<sup>3)</sup>

ここで、体積座標を導入した理由は、次の積分公式が有効に使えるからである。<sup>1), 4)</sup>

$$\iiint L_i^a L_j^b L_m^c L_p^d x dx dy dz = \frac{a! b! c! d!}{(a+b+c+d+3)!} \cdot 6V
 \tag{2.14}$$

これまで3次元問題のための四面体要素について述べてきたが、2次元問題のための三角形要素についても全く同様に導けるので結果だけをまとめて述べる。

三角形要素についても、三つの節点だけを採る一次要素で考える。すなわち、要素内の点  $P$  の温度  $T$  を次式で表す。混乱はないと思われるので、四面体の場合と同じ記号を用いる。

$$T = \alpha_1 + \alpha_2 x + \alpha_3 y
 \tag{2.15}$$

これを節点の座標と温度を用いて

$$T = \{N\}^T \{T\}^e = [N_i \ N_j \ N_m] \begin{Bmatrix} T_i \\ T_j \\ T_m \end{Bmatrix}
 \tag{2.16}$$

の形式に書くことができる。 $N_i, N_j, N_m$  は

$$\begin{Bmatrix} N_i \\ N_j \\ N_m \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} a_i & b_i & c_i \\ a_j & b_j & c_j \\ a_m & b_m & c_m \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix}
 \tag{2.17}$$

ここに、 $A$  は三角形  $ijm$  の面積で

$$2A = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix}
 \tag{2.18}$$

であり、 $a_i, b_i, c_i$  などは

$$\begin{aligned}
 a_i &= x_j y_m - x_m y_j \\
 b_i &= y_j - y_m \\
 c_i &= x_m - x_j
 \end{aligned}
 \tag{2.19}$$

と表される。他の係数は  $i, j, m$  の添字を循環的におきかえて得られる。

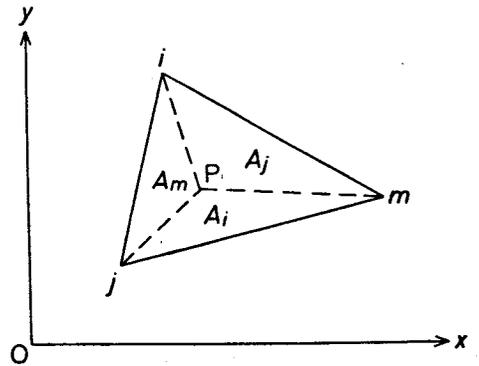


図 2.2 三角形要素と面積座標

この場合も、 $N_i, N_j, N_m$ は面積座標を表すもので、

$$N_i = \frac{A_i}{A}, \quad N_j = \frac{A_j}{A}, \quad N_m = \frac{A_m}{A} \quad (2.20)$$

となる(図2.2)。したがって、次の積分公式が有効に使える。<sup>1), 4)</sup>

$$\iint N_i^a N_j^b N_m^c dx dy = \frac{a! b! c!}{(a+b+c+2)!} \cdot 2A \quad (2.21)$$

## 2.4 有限要素法による温度分布の解析

式(2.5)で示された汎関数の極値を求めるためには、 $\chi$ の微係数に対する各要素の寄与をすべて求め、その和を零とすればよい。関数 $\chi$ の中で一つの要素 $e$ に関連している値を $\chi^e$ と表し、積分を一つの要素について行うものとすれば、式(2.5)を微分することにより、要素 $e$ の節点 $i$ に関する寄与は、

$$\begin{aligned} \frac{\partial \chi^e}{\partial T_i} = & \int_v \left\{ \lambda \frac{\partial T}{\partial x} \cdot \frac{\partial}{\partial T_i} \left( \frac{\partial T}{\partial x} \right) + \lambda \frac{\partial T}{\partial y} \frac{\partial}{\partial T_i} \left( \frac{\partial T}{\partial y} \right) \right. \\ & \left. + \lambda \frac{\partial T}{\partial z} \frac{\partial}{\partial T_i} \left( \frac{\partial T}{\partial z} \right) + \rho c \dot{T} \frac{\partial T}{\partial T_i} \right\} dv \\ & + \int_s \left\{ hT - hT_b - q \right\} \frac{\partial T}{\partial T_i} ds \end{aligned} \quad (2.22)$$

で表される。ただし、 $\dot{T} = \partial T / \partial t$ である。ここで式(2.11)の $T$ を用いると

$$\begin{aligned} \frac{\partial T}{\partial T_i} &= N_i \\ \frac{\partial T}{\partial x} &= \frac{\partial N_i}{\partial x} T_i + \frac{\partial N_j}{\partial x} T_j + \frac{\partial N_m}{\partial x} T_m + \frac{\partial N_p}{\partial x} T_p \\ \frac{\partial}{\partial T_i} \left( \frac{\partial T}{\partial x} \right) &= \frac{\partial N_i}{\partial x} \end{aligned} \quad (2.23)$$

などとなるので、これらを用いて式(2.22)を整理する。

簡単のためベクトル表示を用いると

$$\begin{aligned} \frac{\partial \chi^e}{\partial T_i} = & \lambda \int_v \left\{ \left[ \frac{\partial N_i}{\partial x} \quad \frac{\partial N_j}{\partial x} \quad \frac{\partial N_m}{\partial x} \quad \frac{\partial N_p}{\partial x} \right] \{T\}^e \frac{\partial N_i}{\partial x} \right. \\ & + \left[ \frac{\partial N_i}{\partial y} \quad \frac{\partial N_j}{\partial y} \quad \frac{\partial N_m}{\partial y} \quad \frac{\partial N_p}{\partial y} \right] \{T\}^e \frac{\partial N_i}{\partial y} \\ & \left. + \left[ \frac{\partial N_i}{\partial z} \quad \frac{\partial N_j}{\partial z} \quad \frac{\partial N_m}{\partial z} \quad \frac{\partial N_p}{\partial z} \right] \{T\}^e \frac{\partial N_i}{\partial z} \right\} dv \\ & + \rho c \int_v [N_i \quad N_j \quad N_m \quad N_p] \{\dot{T}\}^e N_i dv \\ & + h \int_s [N_i \quad N_j \quad N_m \quad N_p] \{T\}^e N_i ds \\ & - \int_s (hT_b + q) N_i ds \end{aligned} \quad (2.24)$$

要素 $e$ の他の節点 $j, m, p$ についても同様の式が導ける。これらをまとめて、次式のようにベクトル

表示で表す。

$$\left\{ \frac{\partial \chi}{\partial T} \right\}^e = \begin{Bmatrix} \frac{\partial \chi^e}{\partial T_i} \\ \frac{\partial \chi^e}{\partial T_j} \\ \frac{\partial \chi^e}{\partial T_m} \\ \frac{\partial \chi^e}{\partial T_p} \end{Bmatrix} \quad (2.25)$$

以上の結果をまとめると、要素  $e$  について

$$\left\{ \frac{\partial \chi}{\partial T} \right\}^e = [H]^e \{T\}^e + [P]^e \dot{\{T\}}^e + [C]^e \{T\}^e - \{Q\}^e \quad (2.26)$$

と表すことができる。マトリックス  $[H]^e$ ,  $[P]^e$ ,  $[C]^e$  の右肩に付した  $e$  は、要素  $e$  だけを考えていることを表す。したがって、各マトリックスは4個の節点に対応する行と列を持つ  $4 \times 4$  のマトリックスである。ベクトル  $\{T\}^e$  は四つの節点における時間に関する温度勾配を要素として持つ。第  $i$  節点に対応する行を取りあげ、またマトリックスについては第  $j$  節点に対応する列も選んで、要素を例示すると以下のようになる。

$$H_{ij}^e = \int_v \lambda \left( \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right) dv$$

$$P_{ij}^e = \int_v \rho c N_i N_j dv \quad (2.27)$$

$$C_{ij}^e = \int_s h N_i N_j ds$$

$$Q_i^e = \int_s (h T_b + q) N_i ds$$

積分は何れも要素  $e$  について行う。とくに  $s$  に関する積分は要素が境界を含まないとき、零になると決めておく。式(2.10)で定義された  $N_i$  ならびに同様の形式で求められる  $N_j$ ,  $N_m$ ,  $N_p$  を用いると、式(2.27)の積分は式(2.14), (2.21)を用いて以下のように具体的に求められる。<sup>5)</sup> ただし、ここでは、節点  $i$ ,  $j$ ,  $m$  で構成される面が物体の境界面になっているものとする。

$$[H]^e = \frac{\lambda}{36V} \begin{bmatrix} b_i b_i + c_i c_i + d_i d_i & b_j b_i + c_j c_i + d_j d_i & b_m b_i + c_m c_i + d_m d_i & b_p b_i + c_p c_i + d_p d_i \\ & b_j b_j + c_j c_j + d_j d_j & b_m b_j + c_m c_j + d_m d_j & b_p b_j + c_p c_j + d_p d_j \\ & & b_m b_m + c_m c_m + d_m d_m & b_p b_m + c_p c_m + d_p d_m \\ & & & b_p b_p + c_p c_p + d_p d_p \end{bmatrix}$$

対 称

$$(2.28)$$

$$[P]^e = \frac{\rho c V}{20} \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \quad (2.29)$$

$$[C]^e = \frac{hA}{12} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.30)$$

$$[Q]^e = \frac{(hT_b + q)A}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{Bmatrix} \quad (2.31)$$

ここに、 $A$ は境界面  $ijm$ の面積である。

これまで一つの要素  $e$  についてのみ述べて来たが、同様の手順をすべての要素について考えるものとし、物体全体の節点の数だけの行と列を持つマトリックス（またはベクトル）を構成する。一般に一つの節点は複数個の要素から影響を受けるが、その場合はそれぞれの要素からの寄与を加え合わせてマトリックス要素の値とする。以上のことから、すべての節点に関する  $\chi$  の微係数が求められたことになり、式(2.26)より次式

$$\left\{ \frac{\partial \chi}{\partial T} \right\} = ([H] + [C])\{T\} + [P]\{\dot{T}\} - \{Q\} \quad (2.32)$$

が得られる。汎関数が極値を持つ条件より、

$$([H] + [C])\{T\} + [P]\{\dot{T}\} = \{Q\} \quad (2.33)$$

となり、式(2.31)の  $q$  に境界面からの流入熱量を与えて  $\{T\}$  および  $\{\dot{T}\}$  についてこの方程式を解けば、すべての節点の温度が求められる。各マトリックスの意味は、 $[H]$ が熱抵抗マトリックス、 $[C]$ は外部への熱伝達にかかわるマトリックス、そして  $[P]$ は各部の熱容量を表すマトリックスであって、ベクトル  $\{Q\}$  は境界節点からの熱の流入を表すものである。

非定常項の扱いは、Wilson - Cloughの方法による。<sup>1)</sup>いま  $\Delta t$  の時間刻みの間に  $\{\dot{T}\}$  の値が時間と共に直線的に変化するものと仮定する。そうすると、時刻  $t$  における  $\{T\}$  の値は

$$\{T\}_t = \{T\}_{t-\Delta t} + \frac{1}{2} (\{\dot{T}\}_{t-\Delta t} + \{\dot{T}\}_t) \Delta t \quad (2.34)$$

または、これを書き直して

$$\{\dot{T}\}_t = -\{\dot{T}\}_{t-\Delta t} + \frac{2}{\Delta t} (\{T\}_t - \{T\}_{t-\Delta t}) \quad (2.35)$$

を得る。この式を式(2.33)の  $\{\dot{T}\}$  に代入すると、次の漸化式が得られる。

$$([H] + [C]) + \frac{2}{\Delta t} [P] \{T\}_t = [P] \left( -\frac{2}{\Delta t} \{T\}_{t-\Delta t} + \{\dot{T}\}_{t-\Delta t} \right) + \{Q\}_t \quad (2.36)$$

この関係式により、時刻  $t - \Delta t$  における  $\{T\}$  と  $\{\dot{T}\}$  が分っていれば、連立一次方程式を解いて  $\{T\}_t$  を求めることができる。初期値として  $\{T\}_{t=0} = \{T_0\}$  は与えられているものとし、 $\{\dot{T}\}_{t=0}$  は  $\{T_0\}$  を式(2.33)に代入して求める。

## 2.5 切削モデルへの適用

### 2.5.1 移動熱源の取り扱い

緒言において述べたように、ここでは図2.3に示すような板カムの輪郭切削をモデルとして考えている。

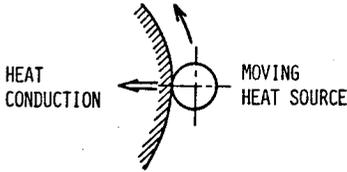
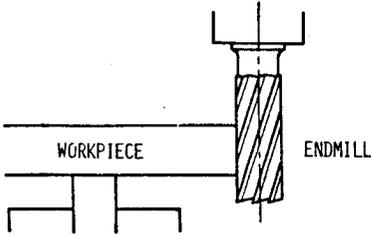


図 2.3 板カム切削モデル

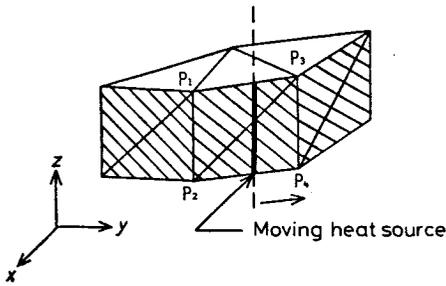


図 2.5 移動線熱源のモデル

熱源は工具の位置における切削熱であり、工作物の上下面および周上においては空気への熱伝達がある。要素分割は  $x y$  平面上で三角形要素に分割するが、 $z$  軸方向の厚さはあまり厚くないものと考えて  $z$  軸方向の分割は行わない。ただし、有限要素法では 3 次元問題として扱うので、 $x y$  平面上の一つの三角形要素について、これを三角柱とみなし、図 2.4 に示すようにこれを三つの四面体に分割して考える。

熱源は移動線熱源として扱われるので、以下にその要領を述べる。図 2.5 において斜線を施した面が板カムの外周を表している。熱源

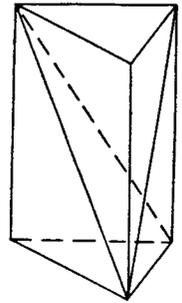


図 2.4 四面体要素への分割

の移動速度は加工条件として指定される工具の送り速度  $f \text{ mm/min}$  である。

Wilson - Clough の方法で非定常

問題を扱う場合、任意の時刻における熱源の位置は必ずしも節点の位置に一致するとは限らない。他方有限要素法では、流入熱量は節点に対して与えなければならない(式(2.36)のベクトル  $\{Q\}_t$ )。このためたとえば図 2.5 のような位置に熱源があるとき、その流入熱量を  $P_1 P_2$  側と  $P_3 P_4$  側に対して、距離によって比例配分する。すなわち、熱源から工作物への流入熱量を  $Q_w$  とするとき

$$\begin{aligned}
 P_1 P_2 \text{ 側} &: Q_1 = Q_w \left( 1 - \frac{f}{l} t_1 \right) \\
 P_3 P_4 \text{ 側} &: Q_2 = Q_w \frac{f}{l} t_1
 \end{aligned}
 \tag{2.37}$$

によって等価節点熱源におきかえる。ここに  $l$  は節点  $P_1 P_3$  間の距離を表し、 $t_1$  は熱源が節点  $P_1$  を通過した後の経過時間を表す。工作物上面 ( $P_1, P_3$ ) と下面 ( $P_2, P_4$ ) への分割は、均等配分で問題はないであろう。式(2.36)のベクトル  $\{Q\}_t$  の所定の節点に対応する要素にこれらの値を代入して、逐次計算を進めればよい。

なお、工作物形状が円柱であるような場合は、有限要素法によらなくても、移動熱源問題を解析的に解くことができる。<sup>6)</sup> ただし、一般の板カムのような任意形状の物体を対象とする場合には有限要素法が有効である。

## 2.5.2 工作物支持部の取り扱い

図 2.6 に示すように、工作物はバイスで工作機械のテーブルに固定されており、この支持部と切削点以外では空気への熱伝達が行われるようになっている。ところで、支持部からの熱の流出をどのように取り扱うかが大きな問題である。一つの考え方は、工作機械の熱容量が十分大きいものと考えて支持部の表面に室温固定の条件を与える方法である。ただし実際には支持部の表面では異なった材質間での熱伝導が行われており、この考え方は熱の流出を大きく見積りすぎると思われる。

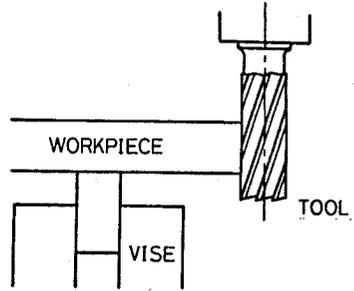


図 2.6 工作物支持部

そこで一つの時間きざみ  $\Delta t$  については固定境界条件を与えるものとし、時刻が進むに従ってその境界温度を修正して行くモデルを導入する。図 2.7 に示すように工作物支持部を 12 個の三角柱で表し、そのおのおのを 3 個の四面体に分割する。この角柱の

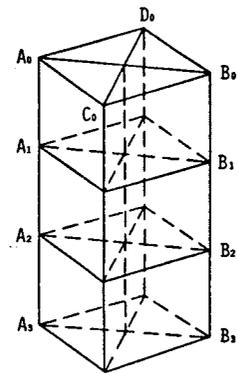


図 2.7 工作物支持部モデル

上面 ( $A_0, B_0$  など) は板カム工作物の下面と同一平面上にあり、板カムと支持部を一体として有限要素法で処理する。図 2.7 において、 $A_i, B_i$  ( $i = 1, 2, 3$ ) が固定温度を与える点 (バイスとの接触面に対応) であり、その他の節点については空気への熱伝達があるものとする。

節点  $A_i, B_i$  の温度は時刻の経過に伴って上昇すると考えられるので、以下のような方法で温度を指定する。いま、時刻が  $(k-1)$  ステップ経過したときの  $A_0, B_0$  における温度がそれぞれ  $T(A_0)^{(k-1)}, T(B_0)^{(k-1)}$  として求めたとすると、次の時刻  $k$  ステップでの各点の温度を、境界条件として次式で与える。

$$\begin{aligned} T(A_i)^{(k)} &= \beta_i T(A_0)^{(k-1)}, \\ T(B_i)^{(k)} &= \beta_i T(B_0)^{(k-1)}, \quad i = 1, 2, 3 \end{aligned} \quad (2.38)$$

$\beta_i$  については、実験による支持部底辺の温度測定と節点間距離を考慮して

$$\beta_1 = 0.8, \beta_2 = 0.6, \beta_3 = 0.4 \quad (2.39)$$

として与える。 $\dot{T}(A_i)^{(k)}, \dot{T}(B_i)^{(k)}$  については上の値を式 (2.33) に代入して求める。この方法では熱の流出が、熱伝達で考えるよりは多く、室温固定の境界条件の場合より少なくなる。上面板カム部すなわち点  $A_0, B_0$  の温度上昇に伴い、支持部境界温度も時間の経過に従って上昇するからである。

ところで、このような境界条件を与えると、有限要素の節点温度がある時刻において既知のもの未知のもの両方を扱わなければならない。有限要素解析の基礎方程式 (2.33) において  $[H] + [C]$  をまとめて  $[\bar{H}]$  と書くと

$$[\bar{H}]\{T\} + [P]\{\dot{T}\} = \{Q\} \quad (2.40)$$

ここで、節点番号を適当につけかえ、既知量には添字  $k$  を、未知量には添字  $u$  をつけるものとする、式 (2.40) は次のように書き換えられる。

$$\begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} \\ \bar{H}_{21} & \bar{H}_{22} \end{bmatrix} \begin{Bmatrix} T_u \\ T_k \end{Bmatrix} + \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{Bmatrix} \dot{T}_u \\ \dot{T}_k \end{Bmatrix} = \begin{Bmatrix} Q_k \\ Q_u \end{Bmatrix} \quad (2.41)$$

未知の温度を求めるためには、上式から導かれる次式

$$[\bar{H}_{11}]\{T_u\} + [P_{11}]\{\dot{T}_u\} = \{Q_k\} - [H_{12}]\{T_k\} - [P_{12}]\{\dot{T}_k\} \quad (2.42)$$

に対して、各時間のステップごとに、式(2.38)による計算値を $\{T_k\}$ 、 $\{\dot{T}_k\}$ に与えて、Wilson - Cloughの方法を適用すればよい。

## 2.6 結 言

プログラミング言語 FORTRANを用いて、輪郭フライス切削における工作物温度分布を予測・推定するプログラムを作成した。解析法は“場の問題”における有限要素法である。このプログラムは一般の板カムの加工に適用できるようになっている。以下にその特徴を示す。

- 1) 工具の送り速度  $f$  を入力として与えると、熱源移動速度を  $f$  とする非定常、移動熱源問題として処理する。
- 2) 工作物全体の温度上昇は比較的ゆるやかであるので、要素として4個の節点だけを考える四面体要素を用いて計算の簡略化が計られている。
- 3) 上下面での熱伝達ならびに工作物支持部の熱流束も考慮に入れてある。

第1章では温度分布の推定に差分法を用いたが、モデルが円板に限られていたためその取り扱いが容易であった。本章で述べたプログラムは次章に述べる熱変形解析のプログラムに組み込まれて使用される。また、対象物は一般形状の板カムであるから、本章での有限要素法の採用は当を得たものといえる。ただし、有限要素法は差分法に比べて計算時間がかかるので、温度分布の解析を差分法で行い熱変形解析だけに有限要素法を用いることは処理時間を短縮するのに有効である。そのためには差分法で一般形状を容易に取り扱えるようにすること、ならびに温度分布のデータを熱変形解析に引き継ぐための形式などに工夫が必要になる。これらは今後の課題である。

## 参 考 文 献

- 1) O.C.Zienkiewicz : The Finite Element Method in Engineering Science, McGraw - Hill Publishing Company Ltd.(1971).  
吉識雅夫、山田嘉昭 監訳：基礎工学におけるマトリックス有限要素法、培風館(1975)。
- 2) 林 毅、村 外志夫：変分法、コロナ社(1966)。
- 3) 藤野 勉：熱伝導と熱応力、培風館(1972)。
- 4) 山田嘉昭 編：マトリックス法の応用、東京大学出版会(1972)。
- 5) 久保元 勇、湯川政次：機関部品の熱的強度解析、クボタ技報、第1巻第1号(1976)、pp.58-70。
- 6) N.R.Des Ruisseaux and R.D.Zerkle : Temperature in Semi-Infinite and Cylindrical Bodies Subjected to Moving Heat Sources and Surface Cooling, Trans. ASME, Journal of Heat Transfer, August(1970), pp.456-464。

## 第3章 工作物熱変形推定のシミュレータ

### 3.1 緒言

本章においても加工対象として板カムを考えるものとする。本編第1章で切削中に工作物へ流入する熱量を、加工条件をパラメータとする推定式として導いた。また第2章では、切削中における工作物の温度分布を予測するプログラムについて述べた。したがって、熱応力熱変形を解析するプログラムを付加すれば、熱変形推定のシミュレータができ上ることになる。本章では、これまでに作成されたいくつかのプログラムを統合して、シミュレータとして完結したものを作成する方法について述べる。

本章で述べる熱変形の解析にも、また第2章における温度分布の予測にも有限要素法を用いているので、与えられた工作物（ここでは板カム）の要素分割を考えなければならない。ところで板カムの形状そのものも、第1編第2章で述べたように、計算機プログラムの実行によって決定される。したがって当然有限要素への分割も、人手を介さずに計算機内部で実行することが望ましい。この点も本章の主題の一つである。

なお熱応力・熱変形解析にあたり、以下の仮定をおく。

- 1) 物体は弾性体である。
- 2) 物体の変位と歪は微小である。
- 3) 材質は弾性変形に対して等方性を有する。
- 4) 弾性定数（ヤング率、ポアソン比、線膨張係数）などの温度依存性はない。

切削加工中の工作物の熱変形解析については、厳密には非定常問題であるが、これを準定常問題として扱う。すなわち、第2章で述べたようにある短い時間きざみ $\Delta t$ ごとに温度分布を求めるが、そのときの温度分布のみに注目して応力変位解析を行う。記述の都合上、第2章と第3章に分けて述べるが、本来は熱伝導問題と熱応力解析は一体として処理されるべきものであり、実際のプログラムもそのように構成されている。

### 3.2 板カムの自動要素分割

熱変形解析に先立って、工作物の要素分割法について述べる。これは本来、前章の熱伝導問題を解析する前に板カムの輪郭が定まった時点で実行されるものである。温度分布の予測とそれに基づく応力解析の、いずれについても同一の要素分割を利用する。

ここでは板カムモデルについて、まず $x-y$ 平面において三角形要素に分割する。分割手法の基本は文献1)による。通常は物体形状の特徴を見て、設計者が大きくいくつかの領域に分ける。各々の領域についてはプログラムにより格子状に分割し、その一つの小矩形に一本の対角線を入れて三角形要素を作る。

ところで、板カム輪郭決定の際、のぼり行程、くだけり行程、停留などの区分に従って従節変位曲線を与える（第1編第2章参照）。そこでこの区分を自動要素分割にも利用する。すなわち、これらの行程ごとに大きくいくつかの領域に分割する。この領域は多くの場合扇形になる。ただし、停留などの部分について扇形の弧の部分が半径方向の長さにくらべて非常に小さいときは、それに隣接する扇形と一体にして扱う。扇形を格子に分割するには、図3.1に示すように $\xi-\eta$ 平面上の正方形における格子をもとの $x-y$ 座標上の扇形に写像するのである。写像に使われる形状関数は以下のものを使用する。<sup>2)</sup>

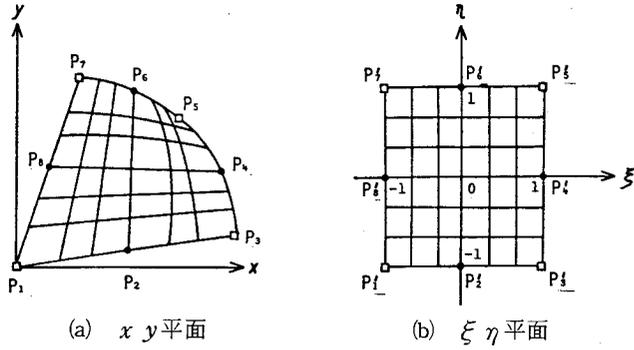


図 3.1 格子分割例

$$\begin{aligned}
 P_a^V &= \sum_{j=1}^8 F_j(\xi, \eta) \cdot P_j^V \\
 &\equiv \frac{1}{4} \xi \eta (1-\xi)(1-\eta) P_1^V + \frac{1}{4} (1-\xi^2)(1-\eta)^2 P_2^V \\
 &\quad - \frac{1}{4} \xi \eta (1+\xi)(1-\eta) P_3^V + \frac{1}{4} (1+\xi)^2 (1-\eta^2) P_4^V \\
 &\quad + \frac{1}{4} \xi \eta (1+\xi)(1+\eta) P_5^V + \frac{1}{4} (1-\xi^2)(1+\eta)^2 P_6^V \\
 &\quad - \frac{1}{4} \xi \eta (1-\xi)(1+\eta) P_7^V + \frac{1}{4} (1-\xi)^2 (1-\eta^2) P_8^V
 \end{aligned} \tag{3.1}$$

ここに、 $P_j^V$  は  $x y$  座標系における点  $P_j$  での状態量で、 $P_a^V$  は  $\xi \eta$  座標系から写像された点の  $x y$  座標系での状態量である。状態量はここでは座標値そのものを表す。

板カム全体の温度分布や熱変形を調べる場合、各要素の大きさはおよそ均等であることが望ましく、極端に小さい要素や大きい要素が混在することは好ましくない。ところで図 3.1 のように  $\xi \eta$  座標系で各辺での分割数を同じにした場合には、これを  $x y$  座標系に写したときもとの扇形状の如何によっては、要素の大きさにむらができることになる。したがって、均一な分割を求めるのには、各辺の分割数を等しくするのではなく、 $x y$  座標系での格子間隔がなるべく均一になるように分割数を定めるべきである。ただし、このようにすると長い辺（または弧）と短い辺（または弧）の分割数が異なることになり、図 3.1 の  $\xi \eta$  座標系の正方形のように格子を作ることができなくなる。この点について、林<sup>2)</sup> は分割数を合わせるために虚点と呼ばれるものを導入し、この虚点を使って格子を作った後で、虚点まわりの点をいくつか間引いて、小さい矩形に分割する手法を考案している。ここでもこの手法を用いることとし、図 3.2 にこのような場合の様子を示す。

まず、格子分割の粗さをどの程度にするかを定めるため、板カム基本円の半径をいくつに分割するかを設計者が入力する。要素分割の粗密の程度は意識的に調整したい場合があるので、これだけは設計者の意図図を入れ、その他の処理は計算機によることとする。分割数の決め方は、それぞれ  $x y$  座標系の辺または弧において、格子間隔の幅が基本円の分割になるべく一致するように等間隔分割とする。ただし、この点は特に厳密さを要しない。図 3.2 に示すように、 $\xi \eta$  座標系で分割数が一致しないときは、図中  $\Delta$  印で示すように虚点をおいて格子を作る。その上で斜線部分を一つの矩形として扱うようにする。この部分は密

になりすぎている部分であるから、このような処理によって均一化を図るのである。

格子状の小さい四角形に対しては、虚点を間引いていない場合は、一本の対角線を図 3.3 (a) のように入れる。このとき 2 本の対角線の中で短いものを選ぶ。虚点を間引いて得られる四角形では、たとえば図 3.3 (b) のように一つの四角形から複数の三角形が作り出される。<sup>2)</sup> 以上の手順により  $x-y$  平面上の物体はすべて三角形要素に分割されるので、節点番号をつけて有限要素法のマトリックスを構成する。図 3.4 に板カムの要素分割例を示す。

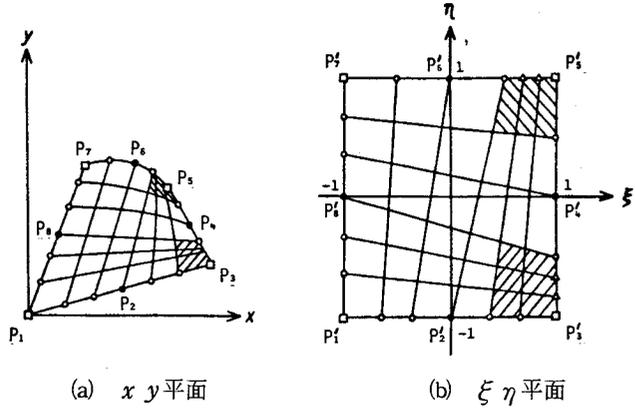


図 3.2 要素分割の改良

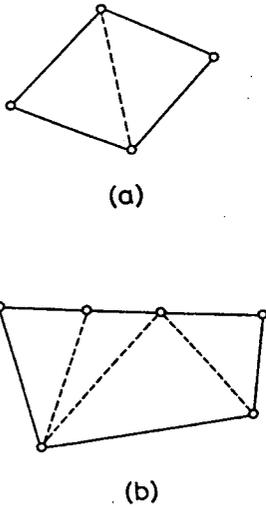


図 3.3 三角形要素

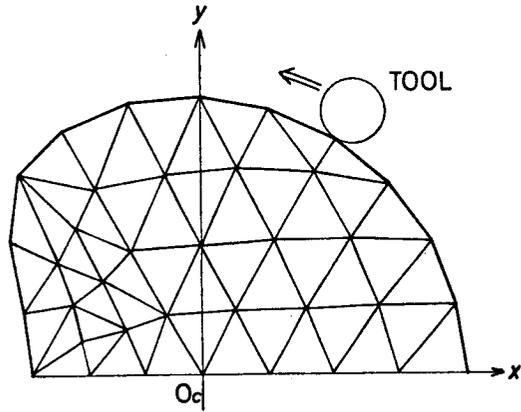


図 3.4 板カムの要素分割例

### 3.3 有限要素法による熱応力・熱変形解析

#### 3.3.1 基礎方程式

ある物体についてすべての外力を除いたとき、一様温度  $T_0$  で応力がない状態を基準状態とする。加工される工作物についていえば、室温  $T_0$  (たとえば  $20^\circ\text{C}$ ) に放置された被削材の応力のない状態を基準状態とする。いまこの物体が温度分布  $T(x, y, z)$  になったときの状態を考えるものとする、基礎方程式は以下のようなになる。<sup>3)</sup>

(i) 構成方程式 (一般化された Hooke の法則)

$$\epsilon_x = \frac{1}{E} [\sigma_x - \nu(\sigma_y + \sigma_z)] + \alpha(T - T_0)$$

$$\epsilon_y = \frac{1}{E} [\sigma_y - \nu(\sigma_z + \sigma_x)] + \alpha(T - T_0) \quad (3.2)$$

$$\epsilon_z = \frac{1}{E} [\sigma_z - \nu(\sigma_x + \sigma_y)] + \alpha(T - T_0)$$

$$\gamma_{xy} = \frac{1}{G} \tau_{xy}, \quad \gamma_{yz} = \frac{1}{G} \tau_{yz}, \quad \gamma_{zx} = \frac{1}{G} \tau_{zx}$$

ここで、

- $\epsilon$  : 各軸方向のひずみ、 $\gamma$  : せん断ひずみ  
 $\sigma$  : 各軸方向の垂直応力、 $\tau$  : せん断応力  
 $E$  : ヤング率、 $G$  : せん断弾性係数  
 $\nu$  : ポアソン比、 $\alpha$  : 線膨張係数

である。

(ii) 応力の平衡方程式

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + X &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + Y &= 0 \\ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + Z &= 0 \end{aligned} \quad (3.3)$$

ここで  $X, Y, Z$  は物体力である。

(iii) ひずみと変位の関係式

$$\begin{aligned} \epsilon_x &= \frac{\partial u}{\partial x}, & \epsilon_y &= \frac{\partial v}{\partial y}, & \epsilon_z &= \frac{\partial w}{\partial z} \\ \gamma_{xy} &= \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}, & \gamma_{yz} &= \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}, & \gamma_{zx} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{aligned} \quad (3.4)$$

ただし  $u, v, w$  はそれぞれ  $x, y, z$  方向の変位である。

(iv) 境界条件

(a) 固定境界条件 ( $S_u$  上で変位が与えられる)

$$u = u_0, \quad v = v_0, \quad w = w_0 \quad (3.5)$$

(b) 自由境界条件 ( $S_\sigma$  上で表面力  $\bar{X}, \bar{Y}, \bar{Z}$  が与えられる)

$$\begin{aligned} \bar{X} &= \sigma_x \nu_x + \tau_{xy} \nu_y + \tau_{xz} \nu_z \\ \bar{Y} &= \tau_{yx} \nu_x + \sigma_y \nu_y + \tau_{yz} \nu_z \\ \bar{Z} &= \tau_{zx} \nu_x + \tau_{zy} \nu_y + \sigma_z \nu_z \end{aligned} \quad (3.6)$$

ここで  $\nu_x, \nu_y, \nu_z$  は物体表面の外向き法線の方向余弦である。

### 3.3.2 有限要素法の導入

ここでは、加工モデルが板状であるから板カムの熱応力解析を平面応力問題として扱うこととする。物体を三角形要素に分割し、図 3.5 に示すような代表的な要素  $e$  に注目する。一つの節点は二つの変位成分  $u_i, v_i$  を持ち、要素  $e$  の 3 節点では以下のように 6 個の成分を持つ。

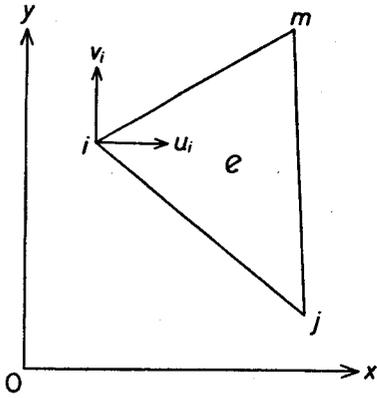


図 3.5 三角形要素

$$\{\delta_i\} = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}, \quad \{\delta\}^e = \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_m \end{Bmatrix} \quad (3.7)$$

要素内の変位を次式で与える。

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y \\ v &= \alpha_4 + \alpha_5 x + \alpha_6 y \end{aligned} \quad (3.8)$$

6 個の定数  $\alpha_i$  は、 $(x, y)$  に節点座標を代入して得られる 2 組の 3 元連立方程式を解くことにより、節点変位によって表され、次式が得られる。

$$\begin{aligned} u &= \frac{1}{2A} \{ (a_i + b_i x + c_i y) u_i + (a_j + b_j x + c_j y) u_j \\ &\quad + (a_m + b_m x + c_m y) u_m \} \end{aligned}$$

$$\begin{aligned} v &= \frac{1}{2A} \{ (a_i + b_i x + c_i y) v_i + (a_j + b_j x + c_j y) v_j \\ &\quad + (a_m + b_m x + c_m y) v_m \} \end{aligned} \quad (3.9)$$

ここで、 $A$  は三角形  $ijm$  の面積

$$2A = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} \quad (3.10)$$

を表し、 $a_i, b_i, c_i$  は

$$\begin{aligned} a_i &= x_j y_m - x_m y_j \\ b_i &= y_j - y_m \\ c_i &= x_m - x_j \end{aligned} \quad (3.11)$$

である。他の係数  $a_j, b_j, \dots, c_m$  は  $i, j, m$  の添字を循環的におき換えて得られる。なお、変位の式 (3.9) をマトリックス表示で表せば

$$\{f\} = \begin{Bmatrix} u \\ v \end{Bmatrix} = [IN'_i \quad IN'_j \quad IN'_m] \{\delta\}^e \quad (3.12)$$

ここに、 $I$  は  $2 \times 2$  の単位マトリックスであり、 $N'_i$  などは

$$\begin{aligned} N'_i &= \frac{1}{2A} (a_i + b_i x + c_i y) \\ N'_j &= \frac{1}{2A} (a_j + b_j x + c_j y) \\ N'_m &= \frac{1}{2A} (a_m + b_m x + c_m y) \end{aligned} \quad (3.13)$$

である。<sup>4)</sup>

変位とひずみの関係式は、基礎方程式 (3.4) に式 (3.9) を適用して得られる。ただし、ここでは平面応力問題として扱う。マトリックス  $[B]$  を用いて表せば

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [B]\{\delta\}^e \quad (3.14)$$

ここで[B]は

$$[B] = \frac{1}{2A} \begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix} \quad (3.15)$$

となる。この場合マトリックス[B]は要素内の位置に無関係であり、要素内の全域でひずみは一定になる。このことは変位関数として一次式(3.8)を用いたことの帰結である。

弾性応力とひずみの関係は、基礎方程式(3.2)について平面応力問題であることを考慮して次式で表される。

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D] \left( \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - \{\alpha\} \Delta T \right) \quad (3.16)$$

ここで、

$$\{\alpha\}^T = \{\alpha, \alpha, 0\} \quad (3.17)$$

$$\Delta T = T - T_0 \quad (3.18)$$

$$[D] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \quad (3.19)$$

であって、各記号の意味は式(3.2)におけるものと同様で、 $\alpha$ は線膨張係数、 $T$ は物体の温度分布、 $E$ はヤング率、 $\nu$ はポアソン比である。また、等方性材質であるから $G = E/2(1+\nu)$ を用いている。

以上で要素 $e$ の性質が明らかにされており、節点変位を用いてひずみが表され、ひずみ(ならびに熱ひずみ)から弾性応力が求められる。

### 3.3.3 ポテンシャル・エネルギー最小の原理

証明は省略するが、ポテンシャル・エネルギー最小の原理<sup>5)</sup>によれば、与えられた境界条件の下で平衡方程式とHookeの法則をとともに満足する変位場は、ポテンシャル・エネルギーが停留値(極値)をとるような変位場であり、またこのとき、ポテンシャル・エネルギーは最小値をとる。有限個の要素に分割された物体に対しても、この原理を適用することにより、平衡方程式を満たす変位場が近似的に求められる。節点変位をパラメータとして扱い、物体全体のポテンシャル・エネルギー $\Phi$ を最小にするためにはつぎのような一連の方程式を求める必要がある。

$$\frac{\partial (\Phi)}{\partial \{\delta_i\}} = 0 \quad (3.20)$$

要素内の単位体積あたりのひずみエネルギーの変化率は、変化するのはひずみだけであるから、ある要素 $e$ の節点 $i$ について

$$\begin{aligned}
\frac{\partial(\{\sigma\}^T\{\epsilon\})}{\partial\{\delta_i\}} &= \left\{ \begin{array}{l} \frac{\partial\epsilon_x}{\partial u_i}\sigma_x + \frac{\partial\epsilon_y}{\partial u_i}\sigma_y + \frac{\partial\gamma_{xy}}{\partial u_i}\tau_{xy} \\ \frac{\partial\epsilon_x}{\partial v_i}\sigma_x + \frac{\partial\epsilon_y}{\partial v_i}\sigma_y + \frac{\partial\gamma_{xy}}{\partial v_i}\tau_{xy} \end{array} \right\} \\
&= \begin{bmatrix} \frac{\partial\epsilon_x}{\partial u_i} & \frac{\partial\epsilon_y}{\partial u_i} & \frac{\partial\gamma_{xy}}{\partial u_i} \\ \frac{\partial\epsilon_x}{\partial v_i} & \frac{\partial\epsilon_y}{\partial v_i} & \frac{\partial\gamma_{xy}}{\partial v_i} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \\
&= \frac{\partial\{\epsilon\}}{\partial\{\delta_i\}} \{\sigma\}
\end{aligned} \tag{3.21}$$

で与えられる。ここで式(2.7)を考慮した上で、式(3.14)を用いると、

$$\begin{aligned}
\frac{\partial\{\epsilon\}}{\partial\{\delta_i\}} &\equiv \begin{bmatrix} \frac{\partial\epsilon_x}{\partial u_i} & \frac{\partial\epsilon_y}{\partial u_i} & \frac{\partial\gamma_{xy}}{\partial u_i} \\ \frac{\partial\epsilon_x}{\partial v_i} & \frac{\partial\epsilon_y}{\partial v_i} & \frac{\partial\gamma_{xy}}{\partial v_i} \end{bmatrix} \\
&= \begin{bmatrix} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} [B]^T \\
&= \frac{\partial\{\delta\}^e}{\partial\{\delta_i\}} [B]^T
\end{aligned} \tag{3.22}$$

を得る。<sup>4)</sup>

式(3.21)に式(3.22)ならびに式(3.16)を代入し、式(3.14)を用いると

$$\frac{\partial(\{\sigma\}^T\{\epsilon\})}{\partial\{\delta_i\}} = \frac{\partial\{\delta\}^e}{\partial\{\delta_i\}} ([B]^T[D][B]\{\delta\}^e - [B]^T[D]\{\alpha\}\Delta T) \tag{3.23}$$

を得る。いま、物体力を無視するものとし、熱ひずみだけを考えることにすれば、要素 $e$ の全体について体積積分して

$$\frac{\partial(\theta)^e}{\partial\{\delta_i\}} = \frac{\partial\{\delta\}^e}{\partial\{\delta_i\}} \left[ \left( \int [B]^T[D][B]dv \right) \{\delta\}^e - \int [B]^T[D]\{\alpha\}\Delta T dv \right] \tag{3.24}$$

を得る。ここで、 $[B]$ は節点の座標値によって定まり、 $[D]$ は材質のヤング率とポアソン比より定められる。要素内の座標を含まないので積分は容易である。 $\Delta T$ は基準状態からの温度上昇を表し、第2.5節で述べた方法によって求められる。ただし、温度分布は三次元問題として解析しているが、ここでの熱変形解析は平面問題として扱っているため、上下面の平均をとって、各節点(三角形要素)の温度とする。ここで熱ひずみについても要素内では一定とみなす。

要素 $e$ の他の節点 $j$ および $m$ に関する微分もまとめて書くと、要素 $e$ について次式が成立する。<sup>4)</sup>

$$\frac{\partial(\vartheta)^e}{\partial\{\delta\}^e} = \left\{ \begin{array}{c} \frac{\partial\vartheta^e}{\partial\{\delta_i\}} \\ \frac{\partial\vartheta^e}{\partial\{\delta_j\}} \\ \frac{\partial\vartheta^e}{\partial\{\delta_m\}} \end{array} \right\} = [k]^e \{\delta\}^e - \{F\}^e \quad (3.25)$$

すべての要素、すべての節点について式(3.25)と同様な式を求め、物体全体のマトリックスを組み立て、ひずみエネルギーの変化率の総和を求めてこれを零とおくと、次の剛性方程式を得る。

$$[K]\{\delta\} = \{F\} \quad (3.26)$$

ここで $[K]$ は剛性マトリックスと呼ばれ、 $\{\delta\}$ はすべての節点の $x$ 、 $y$ 方向の変位を成分として持つ。これを解いて、すべての節点変位 $\{\delta\}$ が求められる。なお、拘束条件として、工作物(板カム)の回転中心で工作物をバイスに固定するものとし、この点での変位は零とする。

### 3.4 熱変形推定シミュレータの構成

有限要素法で実際の計算に用いられる式は、温度分布を求めるための式(2.37)と、変位を求めるための式(3.26)が基本となる。これらの式に現れる各種マトリックスの中で $[H]$ 、 $[C]$ 、 $[P]$ 、 $[K]$ は要素節点の位置と物理定数により定まるもので、時刻 $t$ には依存しない。これらのマトリックスを予め組み立てておくのが、シミュレータの前半部分である。

シミュレータの後半部では、時間の経過と共に熱源位置が移動しベクトル $\{Q\}$ がかわる。また、温度分布の変化と共に式(3.26)の $\{F\}$ が変化する。これらの値は時間のきざみ $\Delta t$ ごとに求めて行く。以上の計算により任意の時刻におけるすべての節点の温度と変位が得られるが、これらの中で実際の加工時に必要になるのは、加工時の温度計測点の温度と、切削点近傍の節点変位だけでよいので、これらを抽出してファイルに保存する。有限要素法にかかわる計算はミニコンピュータでは無理であり、大型計算機を用いて実行する。

ここでは一般の板カムを扱えるようにしているが、有限要素法を用いて温度分布と熱変形を解析しているが、円板モデルの場合には理論的解析が可能である。<sup>6)~8)</sup> 切削加工時の熱変形量を実験的に把握することは困難であるから、プログラム検証のため、有限要素法を用いて円板モデルの解析を行い、理論解析の結果とつき合わせて妥当性を確かめた。計算時間はACOSシリーズNEACシステム900を用いて、30節点で時間のきざみ150ステップの簡単なモデルでCPUタイム1分をこえる。一般の板カムでは50~60節点となるため、数分の使用時間を要する。

### 3.5 結 言

切削加工時の工作物熱変形を予測するためのシミュレータを有限要素法により作成した。その特徴として以下の点が挙げられる。

- 1) 板カム一般についての自動要素分割手法が採り入れられており、輪郭決定後人手を介さず有限要素法の適用が可能になっている。
- 2) 有限要素法という汎用性のある手法を用いているので、板カムに限らず、他の形状の工作物についても、また研磨や研削などにも熱源の与え方を変えるだけで適用できると考えられる。

3) 大型計算機のファイル・システム(TSSとも共用できる)を土台にして、板カム輪郭計算から熱変形解析まで、一貫したデータ処理を実現している。

計算時間がかかることを別にすれば、有限要素法は極めて適用範囲が広い。理論的解析法は円板モデルで上下面断熱といった強い制約条件の下でしか使えない。また差分法では自動要素分割が統一的に扱にくく、どの部分が境界になるかによってプログラムの一部訂正が要求される。また、差分法では熱応力解析が困難であるから、自由膨張だけを考慮して熱変形量を求めることになる。簡単な円板モデルでの比較計算では応力解析を行うのに比べて±5%程度の誤差が算出されている。それ程厳密さを要求しない場合には自由膨張だけを考えるのも一つの手段であろう。

マスタカムを試作するような場合には、設計段階でも試行錯誤がくり返されるので、ここで述べた有限要素法の適用もそれ程大きな負荷とはならないと思われる。

### 参 考 文 献

- 1) O.C.Zienkiewicz and D.V.Phillips : An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by 'Isoparametric' Co-Ordinates, International Journal for Numerical Methods in Engineering, Vol.3(1971), pp.519-528.
- 2) 林 泰道 : 有限要素法のための自動要素分割手法, 三菱重工技報, Vol.9, No5 (1972), pp.649-655.
- 3) 竹内洋一郎 : 熱応力, 日新出版(1971).
- 4) O.C.Zienkiewicz and Y.K.Cheung : The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Company Ltd.(1967).
- 5) Y.C.Fung : Foundations of Solid Mechanics, Prentice-Hall, Inc.(1965).  
大橋義夫, 村上澄男, 神谷紀生 共訳 : 固体の力学/理論, 培風館(1970).
- 6) T.R.Hsu : Thermal Shock on a Finite Disk Due to an Instantaneous Point Heat Source, Trans. ASME, Journal of Applied Mechanics, March (1969), pp.113-120.
- 7) S.Makinouchi, K.Yamagata, E.Ohmura and M.Harada : Thermal Deformation of a Disk due to a Moving Heat Source; Technol. Repts. Osaka Univ., Vol.28(1978), pp.253-260.
- 8) 大村悦二, 牧之内三郎, 山県敬一 : フライス切削における円板の熱的挙動の理論的解析, 日本機械学会論文集(C編), 45巻, 399号(1979), pp.1305-1314.

## 第4章 工具径路の実時間制御

### 4.1 緒言

最近の数値制御工作機械では、ミニコンピュータを用いてDNC(Direct Numerical Control)あるいはCNC(Computer Numerical Control)システムを構成して、適応制御による最適化を行い、生産性の向上または生産費用の最小化などを試みる例が次第に増えて来ている。<sup>1)</sup> これらの中には旋削加工に関する適応制御の報告<sup>2)~6)</sup>、フライス切削に関する適応制御の報告<sup>7)~12)</sup>などがある。しかしながら、これらはいずれも生産性向上のための加工条件の最適化(Technological Adaptive Control)を計っており、加工精度については、制約条件の設定の中で精度の劣化を来さないように考慮されている程度で、積極的には論じていない。すなわち、寸法精度、形状精度などの精度向上を目的とした制御については抽象的・概念的なものに止まっている。<sup>13),14)</sup> ただし、旋削の場合については、工具・工作物の熱変形を補正して寸法精度の向上を図った例<sup>15)</sup>がある。

本研究では板カム輪郭切削における工作物熱変形に対する適応制御を取り上げる。<sup>16)</sup> 第1章から第3章までに述べたように加工条件に依存する工作物熱変形の推定が可能となった。したがって、板カム輪郭に基づきながら、その熱変形を考慮して、熱変形量だけ工具を輪郭の外側に逃がすNC指令を作成すれば、工作物熱変形にかかわる形状誤差の減少が期待できる。しかしながら、有限要素法の計算においては、加工中の流入熱量を一定とみなしており、その値もあくまで推定値であるにすぎない。実際の加工の過程においては送り速度や切込み深さの変動があり、工具刃先の摩耗状態によっても発熱量は変化するはずである。さらには空気の対流による熱伝達の様子もたえず変化していると考えなければならない。

以上の点を考慮して本研究では単に予測値に頼るだけでなく、切削加工中にも工作物の温度上昇を測定し、予測値とつぎ合わせて熱変形量を推定する方式を採用する。したがって、工具径路の決定も切削と並行して行う。このような予測と修正による適応制御の有効性を吟味するのが本章の目的である。

第1章でも述べたように、工作物熱変形の補正は形状精度の劣化要因を取り除くためである。ここで工具に関わる要因も考えておかねばならない。エンドミル側面による輪郭切削においては、工具のたわみと熱変形が精度劣化の要因となる。とくに工具のたわみについては、刃先丸みに大きく依存することが報告されており、<sup>17)</sup>やはり形状精度の劣化をもたらす。文献17)には、テストピースをある時間間隔で切削し、その切削面から実効工具径を測定する方法が提案されており、工具熱変形の補正もこの中に含まれると思われるので、工具に関わる問題はこれで解決できると考えられる。

### 4.2 適応制御実験システム

本研究の適応制御実験には、第1編第4章で述べたオンラインDNCシステムを使用する。ただし、ここで述べる実験ではミニコンピュータとNC装置は一体となって動作するので、オンラインCNCシステムと考えるのが妥当である。第1編第2章で述べた板カム輪郭計算は、大型計算機のタイムシェアリング・システムを用いて、設計者が会話処理で実行する。その後、本編の第2章、第3章で述べた切削加工時の工作物の熱的挙動のシミュレーションについては、有限要素法を用いているので大型計算機のバッチ処理で処理する。その結果得られた制御用データは大型計算機のファイルに保存しておく。CNCコンピュータは大型計算機の端局となっているので、必要に応じて何時でもこのデータをCNCコンピュータに取り込むことができる。ただし、このデータはNC指令の形式にはなっていない。

板カムの輪郭フライス切削において、工作物の熱変形を考慮して工具径路を定めるには、加工中における工作物の温度測定が必要であり、また、加工終了時の形状測定も行わなければならない。このためCNCシステムに計測機能を付加する必要がある。

#### 4.2.1 NEAC-M4入出力システム

ミニコンピュータNEAC-M4のタイプライタ、紙テープ・リーダなどの標準的な周辺装置については、メーカーからインターフェースが提供されるので、データの入出力を実行する場合、入出力命令の使い方さえ分っておればよい。しかしながら、特殊な計測器などをNEAC-M4に接続するためには、入出力システムの構成を把握してインターフェースを設計する必要がある。入出力システムの構成を図4.1に示す。<sup>18)</sup>

入出力チャンネルにはDMA(Direct Memory Access)チャンネルと標準入出力チャンネルの2種類がある。第1編第4.3.3節で述べたように主記憶と磁気ディスク間的高速転送にDMAチャンネルを用いている。DMAチャンネルは、それ自身アドレス・カウンタと転送のタイミングを司る能力を持っており、データ転送の領域についての情報を標準入出力チャンネルから受け取ると、それ以後の実際のデータ転送を中央処理装置(CPU)の動作とは独立に実行する。磁気ディスクの場合、アクセス・タイムを別にすれば、62.5KBytes/sの転送速度を実現している。

標準入出力チャンネルは中央処理部での入出力関係の命令を実行することによって制御される。入出力関係の命令には次の4種がある。

FUN命令：周辺装置の制御命令

SEN命令：周辺装置の状態検出

BTI命令：データ8ビットの入力命令

BTO命令：データ8ビットの出力命令

これらの命令は図4.2に示すように、同一の形式を持つ2バイト命令である。また、図4.3は入出力

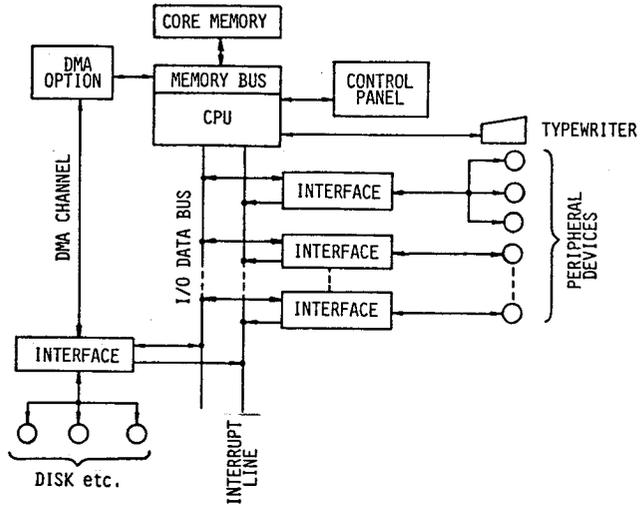


図4.1 NEAC-M4入出力システム

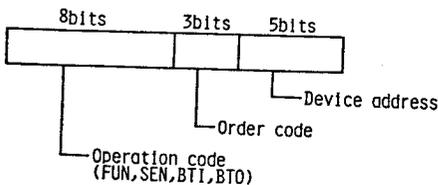


図4.2 入出力命令の形式

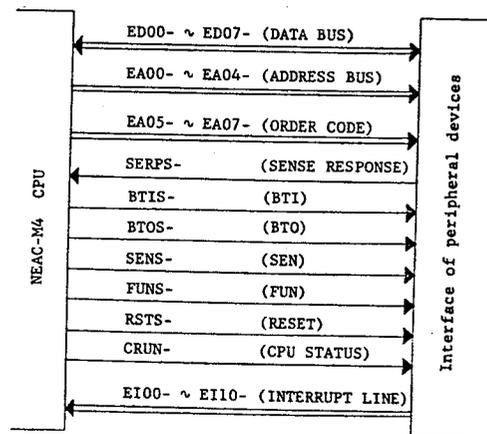


図4.3 NEAC-M4入出力チャンネル

チャンネルの構成を示す。

図 4.3 において、データ・バスはアキュムレータの下位 8 ビットと周辺装置との間でデータの転送を行うための 8 ビットの双方向性バスである。アドレス・バス (5 ビット) は命令で指定したデバイス・アドレスがそのままこのラインに現れ、各周辺装置ごとに定められたアドレスと整合が行われた後、その一致した装置によって入出力チャンネルが占有される。オーダ・コード (3 ビット) も命令に指定されたものそのままが周辺制御部に送られ、周辺制御部でデコードされる。そして、一種類の命令について 8 種類の機能を持たせることができるようになっている。例えば、SEN 命令では、周辺装置 (あるいは制御部) の 8 種類の状態の検出が可能である。このオーダ・コードの使い方は各周辺装置ごとに定められている (第 1 編第 4 章参照)。SERPS- から CRUN- までの制御ラインはセンス応答と実行された命令の種類、中央処理部の状態を伝えるための信号制御ラインである。

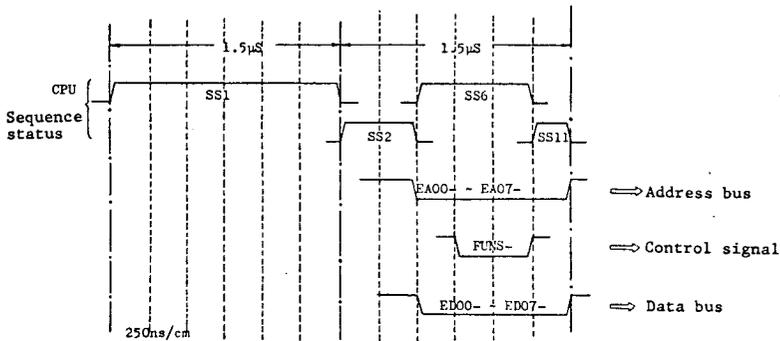


図 4.4 FUN 命令タイミング図

各信号線のタイミングを見るため、一例として FUN 命令のタイミング図を図 4.4 に示す。横方向一目盛が 250 ns を表す。FUN 命令の実行時間は 3 μs である。FUN 命令の実行を示す FUNS- の制御信号は 500 ns のパルス (負論理) であり、これを内を含むようにアドレス・バスとデータ・バスに 1 μs のパルスが出る。オーダ・コードはアドレス・バスに含まれており、データ・バスにはアキュムレータ下位 8 ビットのデータが乗る。したがって、FUN 命令は周辺装置の制御命令であるが、BTO 命令と同様にコントロール・パラメータを周辺装置に転送することができる。この機能は、後に述べるアナログ入力インターフェースで効果的に用いている。また、オーダ・コードやデータ・バスの 8 ビットの情報は通常周辺制御部でラッチされた上で、いろいろな制御に用いられる。

NEAC-M4 には割り込ラインが 11 本ある。本実験システムでは、これらを図 4.5 に示すように使い分けられている。図 4.5 で上部右端に示した 4 個のアドレス (16 進数) は、割り込み発生時にそれまで実行していたプログラムを中断し、割り込ラインに応じた固定アドレスの命令を実行することを意味する。

割り込ラインの割り当て方は、#0 ~ #2 に処理優先度が高く、実時間処理の時間的制約の強いものを接続する。#3 ~ #10 には標準的な周辺装置を接続する。この割り当て方は割り込み要求に対するマスクのセット、リセットに関係している。本実験システムでは多重割り込みを可能にしているので、例えば NC 装置の割り込み処理中では通信回線からの割り込み (MODEM Interface) は受けつけるが、#3 ~ #10 の標準デバイスについてはマスクをリセットして割り込み要求を受けつけないようにする。このようなマスクの管理は MIN 命令で実行される。#3 ~ #10 の割り込みラインについては、割り込み発生時の固定アドレ

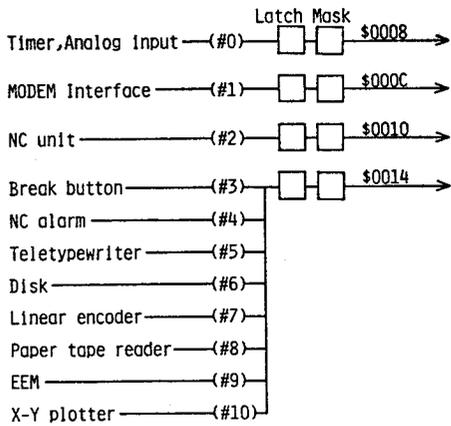


図 4.5 割込ライン

スは一つしかないが、CIS命令(Copy Interrupt Status)によって、8本の割込みラインの状態をアキュムレータに取り込むことができ、割込み解析プログラム(Interrupt Analyzer)によって管理される。優先順位もライン番号に無関係にプログラムによって定められる。

以上NEAC-M4の入出力システムは、もともと制御用計算機として設計されたものだけに、比較的使い易い構成を取っている。

#### 4.2.2 アナログ・データの入力

ここで述べる実験では、工作物の温度測定がアナログ入力であり、また、形状測定に用いるギャップ・センサのピーク値がアナログ量である。アナログ入力インターフェースのブロック図は、第1編第4章に示されている。ここでは応用に即してもう少しくわしく述べる。アナログ・デジタル変換回路の基板の配線図を図4.6に示す。図面において、左寄りに16チャンネルのアナログ・マルチプレクサ(AD7506J)があり、中央部には抵抗分割による可変利得がある。右寄りにあるのがサンプルホールド回路とA-D変換器(ADC-10Z)で、その出力は10ビットである。図面には出ていないが、アナログ・マルチプレ

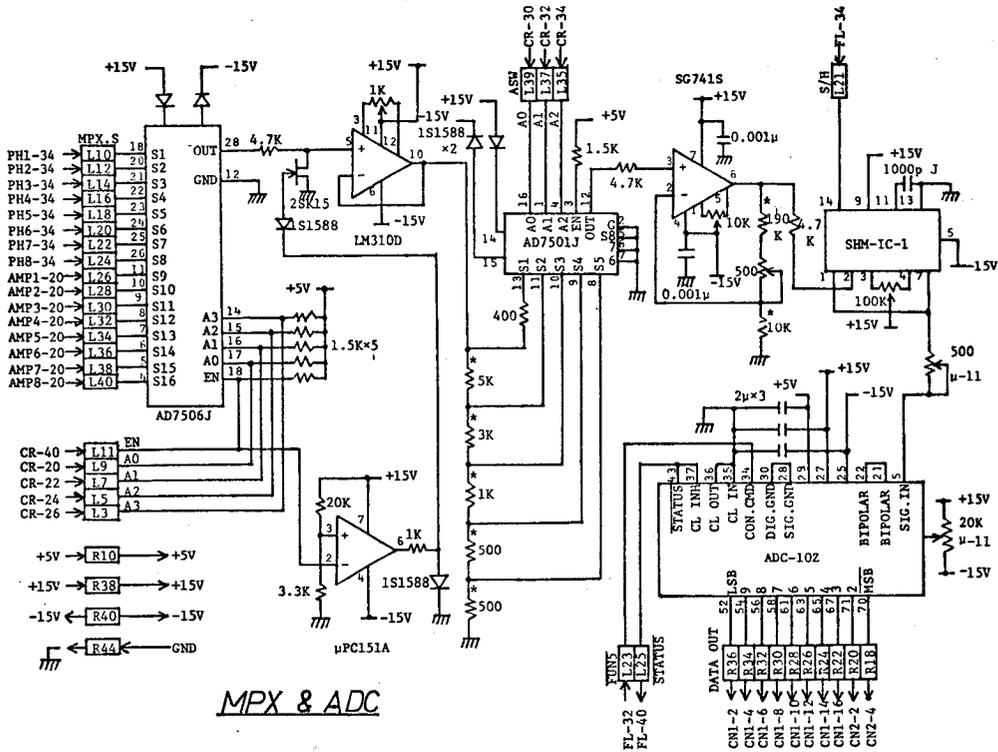


図 4.6 A-D変換回路

クサの前段には各チャンネルごとに1枚の基板が用意されており、必要に応じて、熱電対用増幅器、ピーク・ディテクタあるいはバッファ・アンプが装着できるようになっている。

汎用性を持たせるため、入力チャンネルの選択、可変利得の利得設定、ピーク・ディテクタのモード設定はミニコンピュータNEAC-M4の命令で実行できるようになっている。このためのコントロール・レジスタを図4.7に示す。

2D、2Cと書かれた4個のフリップ・フロップで入力側のアナログ・マルチプレクサにより一つの入力チャンネルを選択する。同様に3D、3Cと書かれた3個のフリップ・フロップで可変利得が定まり、2Bと書かれた2個のフリップ・フロップでピーク・ディテクタのモード設定が定められる。図面下方の1Cと書かれたデコーダで、どのコントロール・レジスタに値をセットするかが定められる。

コントロール・レジスタの設定に

はFUN 2 命令を用いる。この命令を実行すると、アキュムレータの下位8ビットがコントロール・パラメータとしてデータ・バスを通過してコントロール・レジスタに送られる。予めアキュムレータにプリセットしておくコードを図4.8に示す。PHはピーク・ディテクタを、MPXはアナログ・マルチプレクサを、そしてPGAは可変利得をそれぞれ表す。一つのコントロール・レジスタの設定にFUN 2 命令を1回づつ実行しなければならないが、レジスタの内容は変更の必要性が生じない限り、設定値が保持される。

図4.9は図4.6のサンプル・ホールド回路とA-D変換器の動作を制御するものである。動作の指令はどれもFUN命令によるもので、デコーダ(1C)によりオーダ・コードがデコードされ、図に示すようにFUN 3, FUN 4, FUN 5の制御信号が取り出される。まず、FUN 3では二つのフリップ・フロップをリセットし、これによってサンプル・ホールド回路をサンプリング・モードにする(2C(下))と共に、A-D変換終了のステータス信号のマスクを閉じる(3B)。次にFUN 4により、サンプル・ホールド回路のモード設定をホールドに切替える(2C(下))と共に、ピーク・ディテクタのピーク検

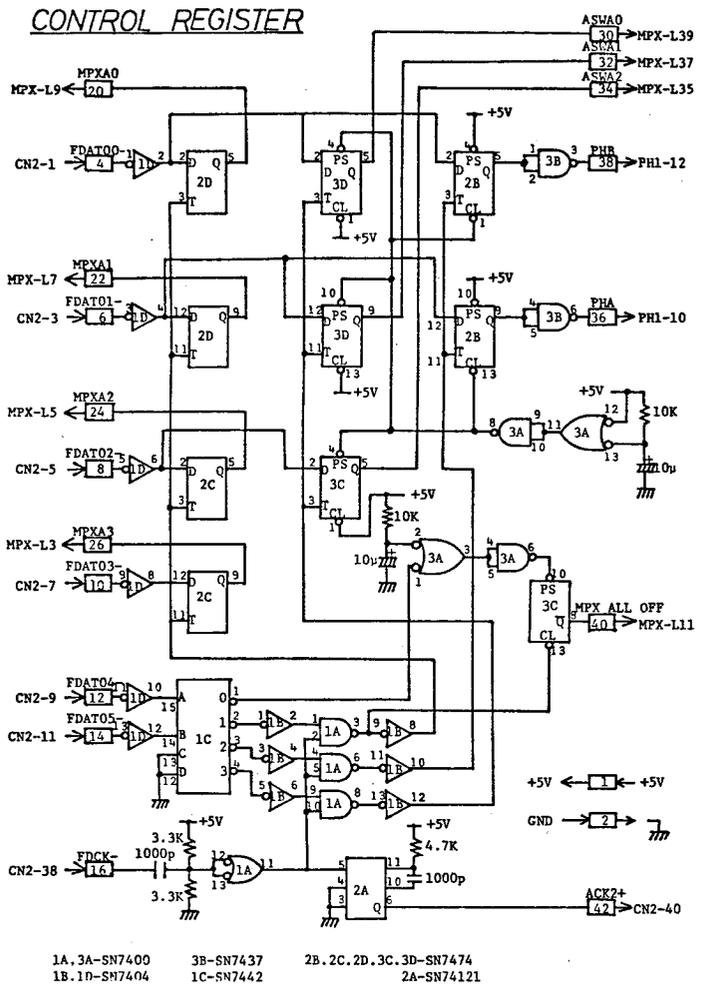


図4.7 コントロール・レジスタ

	7	6	5	4	3	2	1	0
DATA BUS								

MPX ALL OFF	0	0	0	0	*	*	*	*
PH	0	0	1	0	*	*	*	*
MPX	0	0	0	1	*	*	*	*
PGA	0	0	1	1	*	*	*	*

\*\*\*\*\*: CONTROL CODE

PH	: 0000 - RESET	MPX	: 0000 - (2)-1, 1000 - (1)-1
	0001 - HOLD		0001 - (2)-2, 1001 - (1)-2
	0011 - DETECT		0010 - (2)-3, 1010 - (1)-3
			0011 - (2)-4, 1011 - (1)-4
PGA	: 0000 - *20		0100 - (2)-5, 1100 - (1)-5
	0001 - *10		0101 - (2)-6, 1101 - (1)-6
	0010 - *4		0110 - (2)-7, 1110 - (1)-7
	0011 - *2		0111 - (2)-8, 1111 - (1)-8
	0100 - *1		

図 4.8 プリセット・コード

出ステータスをリセットする(2C(上))。最後に FUN 5 により A-D 変換器にスタート・パルスを送り、変換終了ステータス信号のマスクを開ける(3B)。以上の操作により、アナログ・デジタル変換が終了すると、そのステータス信号により NEAC-M4 本体に割込みがかり、また、SEN 2 命令で検知できるようになる(変換時間 30 μsec)。A-D変換器の出力は 10ビットであるので、やはりオーダ・コードを利用して BTI 0 と BTI 1 の2回に分けてアキュムレータに取り込む。

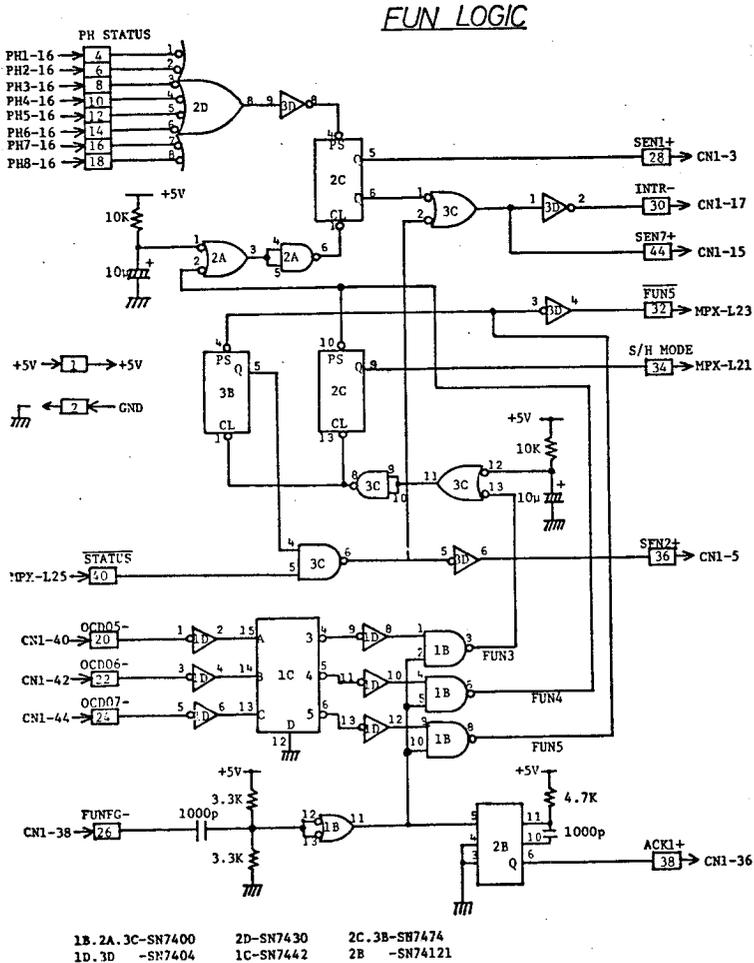


図 4.9 A-D変換制御回路

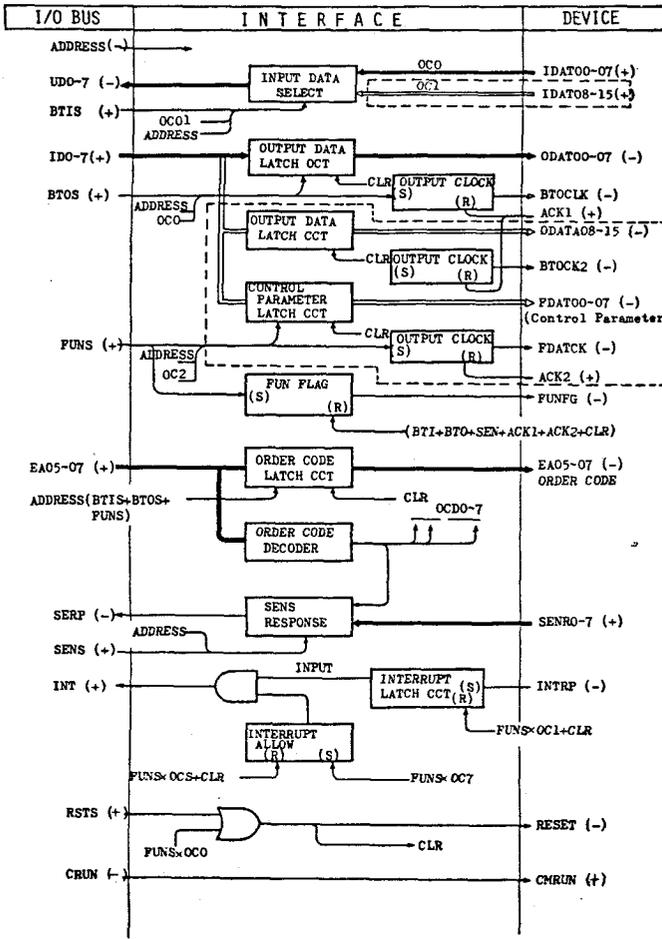


図 4.10 汎用インターフェース (ML 4891)

デジタル側のケーブルについても、これが長くなる場合はドライバー回路とレシーバー回路を設け、ツイストペア線で配線することが望ましい(図 4.11)。

これまで述べて来たアナログ入力インターフェースは、NEAC-M 4 の汎用インターフェース (ML4891, 図 4.10) を介して一つの周辺装置として本体の標準入出力チャンネルに接続される。<sup>19)</sup> 図 4.10 において、太い線がデータの流れを表し、細い線はゲート回路やフリップ・フロップを制御する信号線である。アナログ入力の場合出力側のラインは、コントロール・パラメータとオーダ・コードだけを用いる。A-D変換後の入力は IDAT 00-15 のラインを用いて行われる。一般に計測の行われる場所はコンピュータに近い所とは限らないので、アナログ入力機器は一つのシャーシの中に組み込み、汎用インターフェース ML 4891 (コンピュータ本体内部に実装) と約 10 m のケーブルで接続している。これは雑音対策として、アナログ量のケーブルを引きまわすよりは、デジタル側のケーブルを延長する方が安全であることによる。なお、デ

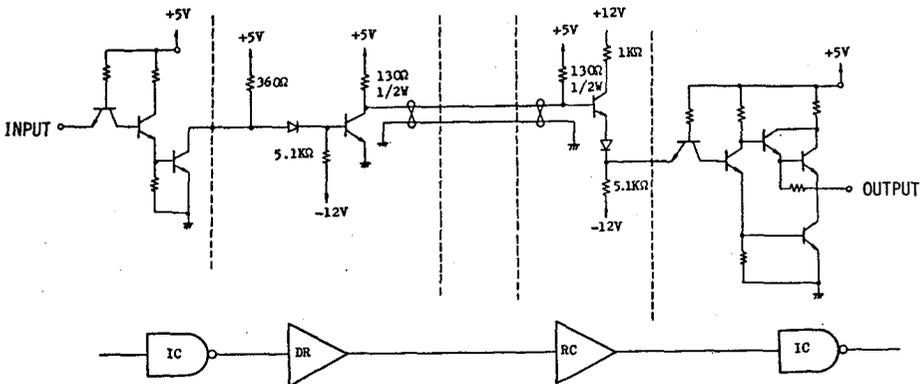


図 4.11 ドライバー回路とレシーバー回路

本研究で作製したアナログ入力インターフェースは、これまで述べて来た事で明らかのように、汎用性を重視してすべてプログラム制御で処理しており、アナログ入力インターフェースの中にはタイマーも自動スキャン機能を持っていない。したがって高速現象の計測には不向きである。単にアナログ・データを入力してメモリに格納するだけの簡単なプログラムでも1点の入力に100.5 $\mu$ sかかる。また、NEAC-M4本体の持っているタイマーを利用する場合においても、インターバルの最小設定単位は1msであり、高速現象の把握のためには別途ウェーブメモリ等を使用することを考えなければならない。

### 4.2.3 工作物の温度測定

切削加工時の工作物の温度測定には、銅・コンスタンタン熱電対（以下ではC C熱電対と書く）を用いる。本研究では、板カムの加工を取り上げるので、カムの回転中心の位置で深さ5mmに埋め込む（被削材の厚さ10mm）。熱電対を用いるとき、基準接点の温度をどのように定めるかが問題になる。氷点式は精度は高いが、加工という作業と合わせて考えると、その都度アイス・ボックスを用意することは好ましくない。そのため、本実験システムでは補償式を採用するものとし、図4.1.2に示す冷接点補償器を作成した。

この方式では基準接点は室温にさらされており（図中Rの点）、この点の温度をIC温度センサーで測定する。この温度は絶えず変

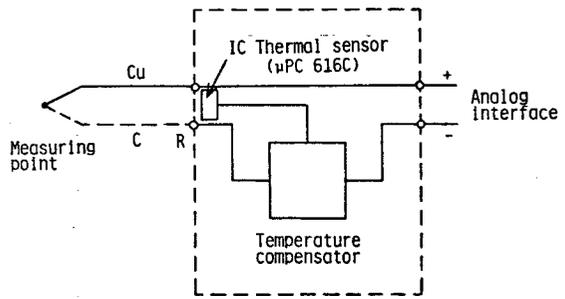


図 4.1.2 冷接点補償器

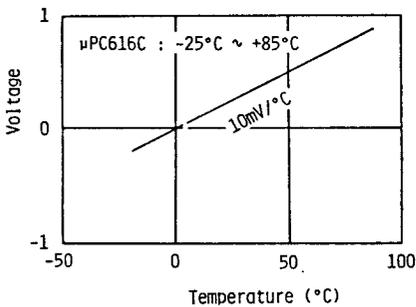


図 4.1.3 IC温度センサーの特性

化していると考えなければならないが、温度補償部により、予め定められた基準温度との温度差に対応する電圧をC C熱電対の出力電圧に加えて、基準温度から見た測温点の温度に対する電圧を得ている。IC温度センサーには $\mu$ PC616C（日本電気）を用いる。<sup>20</sup>このセンサーは測定温度範囲は $-25^{\circ}\text{C} \sim +85^{\circ}\text{C}$ と狭いが、図4.1.3に示すように10mV/ $^{\circ}\text{C}$ と高感度であり、雰囲気温度測定には適している。温度補償部の回路を図4.1.4に示す。スイッチが二つ用意されており、SW1は基準温度の切換えスイッチであり、0側にした場合、SW2は電圧計メータの表示を切換えるもので、A側では0 $^{\circ}\text{C}$ を基準としたIC温度センサーの検出温度に対応する電圧を、B側ではIC温度センサーの検出温度と設定された基準温度との差に対応する電圧をそれぞれ表示する。SW2をA側にした場合、IC温度センサーの出力を増幅器A3で10倍して5Vの電圧計をふらせているので、10 $^{\circ}\text{C}$ 当り1Vの出力が得られ、0 $^{\circ}\text{C} \sim 50^{\circ}\text{C}$ の範囲で室温温度計として使用できる。

冷接点補償器の調整ならびに校正は以下の手順で行う。

#### (1) IC温度センサーの零点調整

魔法瓶の中に水と氷を入れたものを用意する。IC温度センサーの基板からの結線を延長し、水のかからないようにカバーをしてセンサーを魔法瓶につける。この状態でセンサーの出力（ICの2、

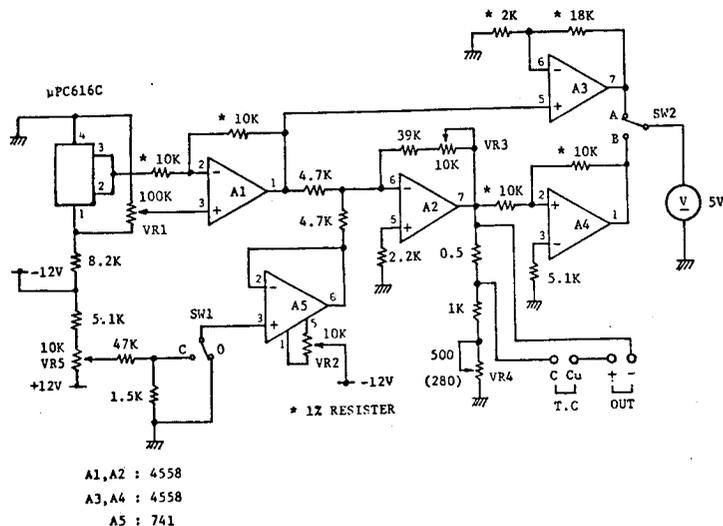


図 4.1 4 温度補償部の回路

3番ピンとグラウンド間)が0VになるようにVR1を調整する。氷点の温度が0℃であることを基準温度計で確認しながら行う。

(2) 基準温度0℃の設定

IC温度センサーを氷点につけた状態で、SW1を0側にし、増幅器A5の出力電圧が0VになるようにVR2を調整する。

(3) 温度補償回路の感度調整

魔法瓶に適当な温度のお湯を入れ、これにセンサーと基準温度計をつける。SW1は0側にしておく。増幅器A2の出力が、基準温度計の読みに対して10mV/℃の感度で得られるようにVR3を調整する。例えば50℃に対しては500mVの出力が得られるようにする。この状態で、SW2をA側にしてもB側にしても電圧計の表示値は変わらない。IC温度センサーは0℃で出力0Vであり、温度補償部(A5, A2)の基準温度も0℃に設定されているからである。

(4) 熱電対起電力対応の温度係数設定

IC温度センサーの出力は10mV/℃であり、熱電対の感度は低いので、アッテネータで温度係数を換算して、CC熱電対の起電力に対応する電圧で補償する必要がある。手順は(3)と同様に魔法瓶の中のお湯にセンサーと基準温度計をつける。CC熱電対の規準熱起電力表<sup>21)</sup>を用意し、基準温度計の読みに対する熱起電力が、増幅器A2の出力段0.5Ωの抵抗の両端に現れるようにVR4を調節する。例えば20℃に対しては0.789mV(0℃基準)となるようにする。

(5) 0℃以外の基準温度設定

SW1をC側、SW2をB側にする。魔法瓶に設定したい温度のお湯を用意し、これにセンサーと基準温度計をつける。VR5によって補償器付属の電圧計の振れが0になるように調整する。これで基準温度の設定が出来る。実験開始時の室温を基準とする場合は魔法瓶やお湯を使う必要もない。基準接点の温度は絶えず変動していると考えなければならないが、これをIC温度センサーで検出し、その出力に対して、0℃から見た基準温度に対応する電圧が増幅器A5により加算(または減算)されるので、冷接点補償器の出力には常に測温点と基準温度の差に比例する電圧が得られる。

すなわち、I C温度センサーの温度と基準温度との差に比例する電圧で補償を行うことになる。

以上の調整は一度行えばよいのであって、使用の都度調整し直す必要はない。冷接点補償器の外観を図4.15に示す。メータのスイッチは通常A側にして室温監視に用いる。基準温度の設定に用いるVR5についてはシャーシの外側から変更できるようになっている（COMPENSATION POINT）。冷接点補償器の出力は銅導線によって前節で述べたアナログ入力インターフェースに接続される。

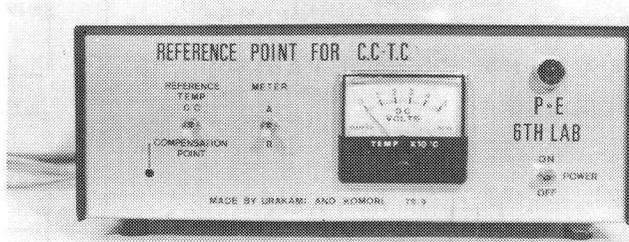
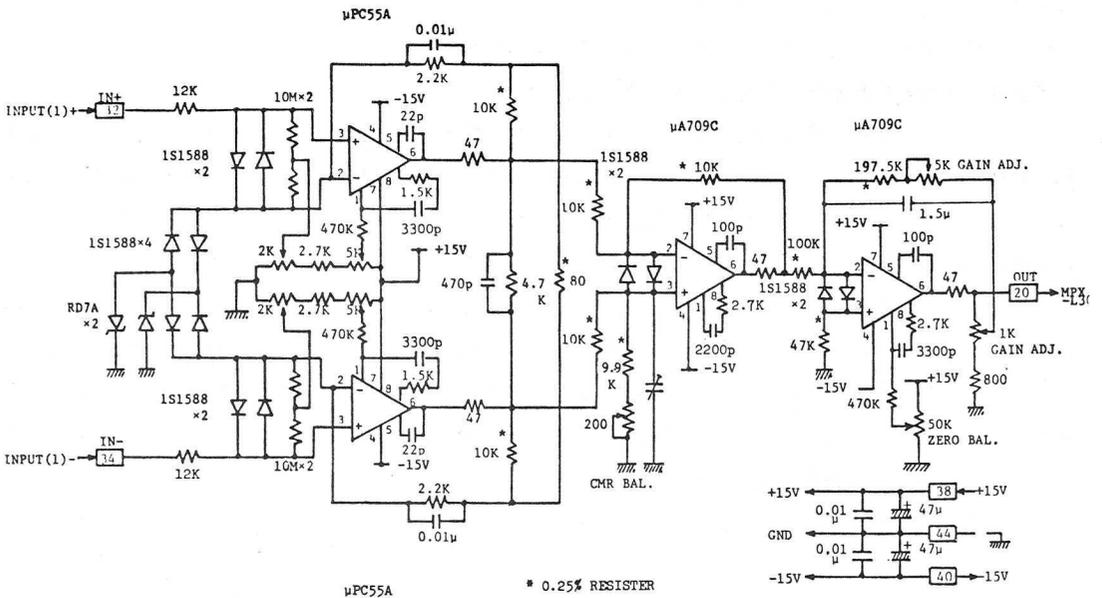


図 4.1 5 冷接点補償器外観

本研究で取り上げている工作物の温度上昇は数十°Cの範囲であり、C C熱電対の起電力が $40\mu\text{V}/\text{C}$ であることを考えあわせると、アナログ入力インターフェースの初段に高利得（電圧増幅1000倍、60dB）低雑音の直流増幅器が必要になる。具体的な回路の例を二つ、図4.16と図4.17に示す。何れも差動入力形式の計装用増幅器（instrumentation amplifier）の構成になっている。図4.16では初段で250倍、差動増幅段で1倍、終段で4倍の利得構成で1000倍の利得を得ており、同相成分除去比も60dB以上ある。また、工作物の温度変化はゆるやかであるので、終段をカットオフ周波数0.5Hzのハイカ



AMP1-8

図 4.1 6 熱電対アンプ例(1)

ットフィルターとして低雑音化を図っている。工作物の温度変化を測るのにはこれではよいが、応答速度が遅いので一般性には欠ける。図 4.17 はこれを補う意味で、デュアル FET 入力とすることにより低雑音化を図ったものである。また FET 入力では温度ドリフト特性が良好であり、入力インピーダンスも高くとることができる。この増幅器で SN 比は 60 dB 以上取れており、CC 熱電対を用いたときの雑音による誤差は約  $\pm 0.125^{\circ}\text{C}$  である。

A-D 変換器を使用する場合、ある程度周波数の高い雑音成分でも拾ってしまうので、高利得増幅器を使用するときは周囲から混入する雑音にも十分注意する必要がある。冷接点補償器からの結線には 2 芯シールド線を用い、電源部には AC ライン・ノイズ・フィルターを入れ、筐体は確実に接地する。

工作物の温度上昇が数十  $^{\circ}\text{C}$  であると考え

ると、感度の面からいえば、熱電対よりも IC 温度センサーを使う方が望ましいが、IC 温度センサーは被測定物との間にある程度の接触面積が必要であり、また、深さ 5 mm に埋め込むのが困難である。工作物温度上昇の測定のための温度センサーには一考の余地が残されている。

#### 4.2.4 工作物形状測定のためのギャップ・センサ

板カムモデル工作物の形状測定にはギャップ・センサ（電子応用協、AEC-1553T）を用いる。基本動作原理は、ピックアップに高周波コイル（455 kHz）を使用し、ピックアップ・コイル 磁界内に電導体が近づいたとき、電導体内に発生する渦電流によりピックアップ・コイルのインダクタンスが変化することを検出し、電導体とピックアップ・コイルの間隙を測定するものである。出力特性を図 4.18 に示す。ピックアップ・コイルは数種類用意されているが、当面の目的に広い測定範囲は必要としないので、分解能が高く（ $0.3\ \mu\text{m}$ ）、対象物の曲率半径の影響を避けるため、最も小さいコイル（AEC 5P05,  $5\text{mm}\phi$ ）を用いる。この場合の標準的な使い方では、図 4.18 で  $D_0 = 0.5\text{mm}$  とし、測定範囲は出力電圧 0 V を基準として  $\pm 0.5\text{mm}$  である。また、対象物の曲率半径の影響は、曲率半径 10 mm 以下で感度の低下が見られ、10 mm 以上では影響を受けない。

このセンサーを用いて形状測定を行う場合、NC 装置を用いて、ピックアップ・コイル自身を低速（100 rpm）で回転させながら、工作物の周りを理想輪郭に沿って動かす。コイルが丁度工作物の法線方向に向いたとき最小の電圧を示すので図 4.19 に示すような波形が得られる。形状測定で重要なのは、この最小値であるので、極性を反転した上でアナログ入力インターフェースに接続する。アナログ入力インターフェースでは初段の基板にピーク・ディテクタを装着したものをを用いる（図 4.20）。これにより、ピー

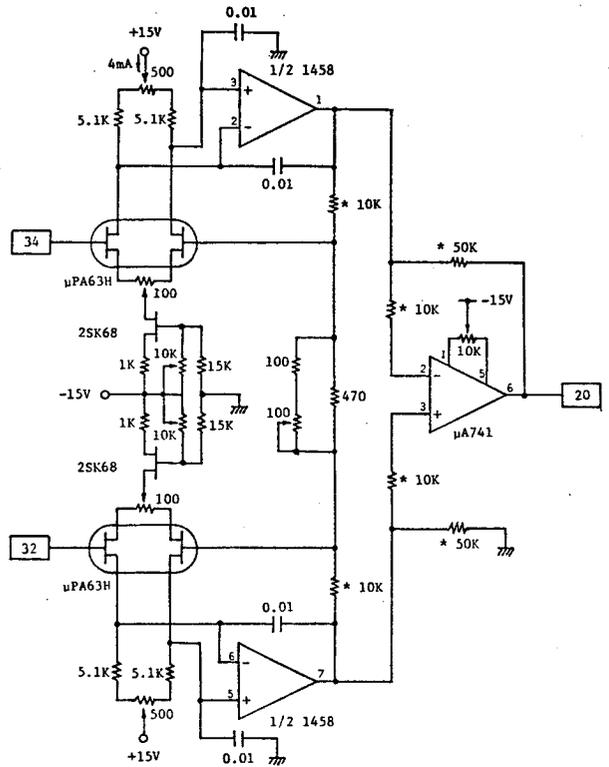


図 4.17 熱電対アンプ例(2)

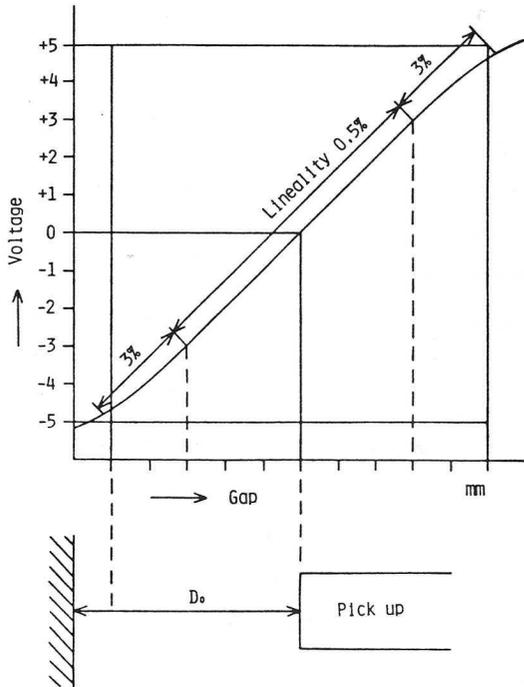
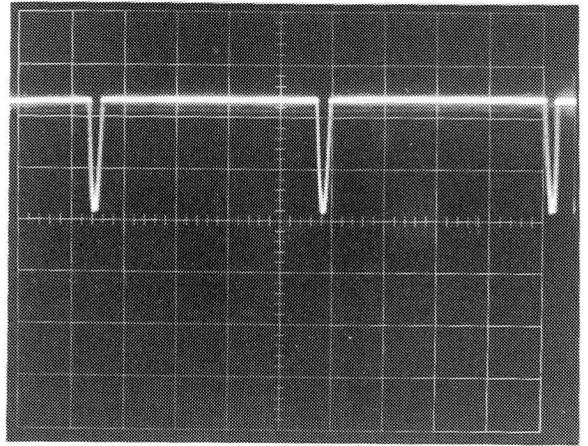


図 4.18 ギャップセンサー出力特性



Vertical : 5V/DIV  
Horizontal : 0.2s/DIV

図 4.19 形状測定時のギャップ・センサーの出力

ク値を A-D変換器を通してミニコンピュータに入力する。もしも工作物が理想形状に仕上がっていれば、どの点で測定してもこのピーク値は同じである筈であり、偏差があるとすればそれは形状誤差を示すことになる。なお、ピーク・ディテクタにはピーク・ディテク

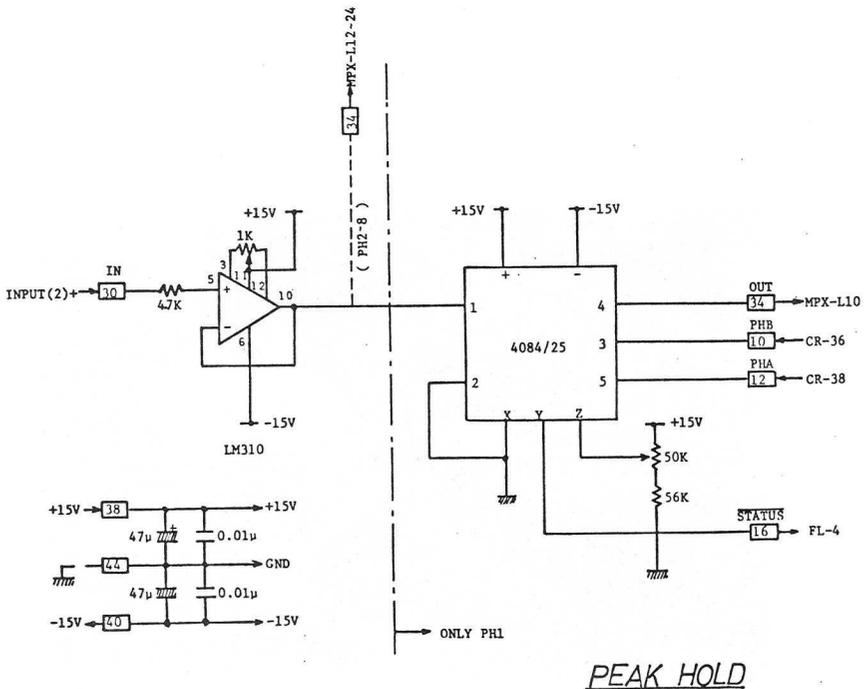


図 4.20 ピーク・ディテクタ基板

ト・モード、ホールド・モード、リセット・モードの3種のモードがあるが、これらはすでに述べたアナログ入力インターフェースの2ビットのコントロール・レジスタによって制御され、その切換はプログラム(FUN命令)によって実行される。

ギャップ・センサーは測定対象の材質によって感度が異なるため、実際の工作物S 3 5 Cを用いて校正曲線を求める必要がある。実際に形状測定を行うのと同様の状態で、センサーのピックアップ・コイルを回転させながら、NC装置を用いて10 $\mu$ m単位の送りを与えてピーク値をミニコンピュータに入力する。ただし、テーブルの位置決めは後述するようにあまり正確ではないので、次節で述べるリニア・エンコーダを用いてテーブルの正確な位置を読み取る必要がある。具体的な測定範囲はおよそ-50 $\mu$ m $\sim$ +50 $\mu$ m程度であり、この間の直線性は良好であるので、校正曲線は直線でよい。最小2乗法により係数を求める。ギャップ・センサー本体には感度ならびに零電位のレベル・シフト調節機能がついているので、10ビットのA-D変換器(入力範囲-10V $\sim$ +10V)が有効に使えるように調節する。

#### 4.25 リニア・エンコーダ・インターフェース

高精度加工を目標とする場合、テーブル(x軸)と主軸頭(y軸)の実際上の位置を正確に把握することは重要な問題である。この目的のために、光電式リニア・エンコーダ(日本光学工業)を実験システムのx軸とy軸に取りつけた。原理図を図4.21に、また、スケールをテーブルに取り付けたときの外観を図4.22に示す。

図4.21で、光源W、レンズL、インデックス・スケールS<sub>i</sub>、受光素子P1、

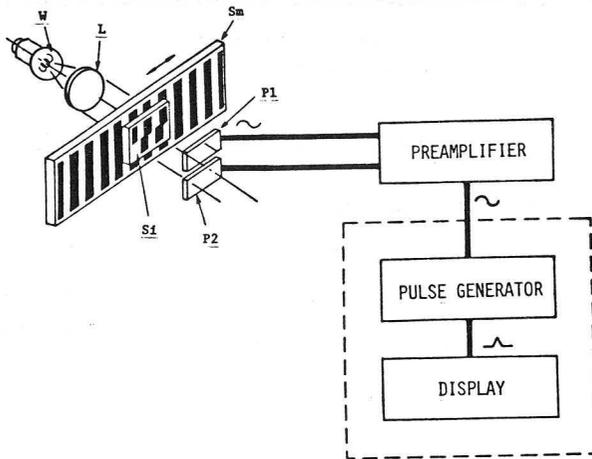


図 4.2 1 リニア・エンコーダ原理図

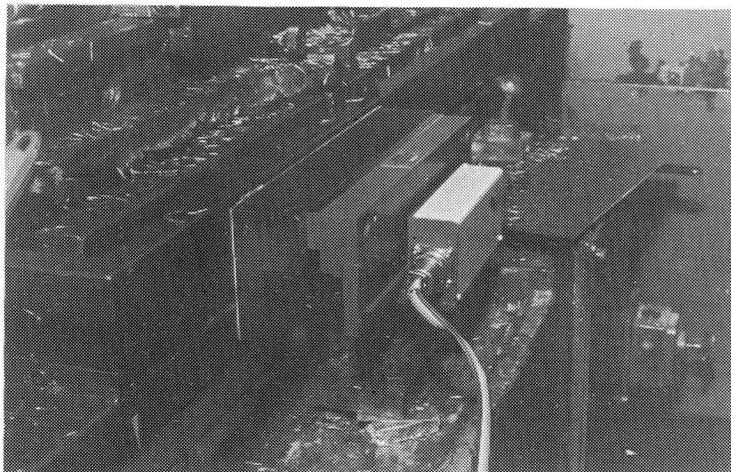


図 4.2 2 リニア・エンコーダ外観

P 2は固定されており、ガラス・スケール  $S_m$  が移動する。 $S_m, S_i$  には等間隔の格子縞があり、 $S_m$  が  $S_i$  に対して1ピッチ移動するごとに、受光素子への投射光が明暗をくり返す。これはヘッド内蔵の前置増幅器で増幅され正弦波として取り込まれる。格子縞のピッチは  $8 \mu\text{m}$  であるが、インデックス・スケール  $S_i$  には2群の格子縞があり、これによって  $S_m$  の移動方向を判別すると共に、 $90^\circ$  位相の異なる正弦波を得ている。さらに内挿によって、 $45^\circ$  位相差信号、 $135^\circ$  位相差信号を取り出し、四つの信号を用いてテーブルあるいは主軸頭の移動距離が検出される。位相の異なる四つの正弦波を矩形波に変換し、その立上りと立下りでパルスを作ると8個のパルスが生成され可逆パルス・カウンタに加え込まれる。したがって、カウンタの読みは  $1 \mu\text{m}$  単位である。本実験では  $200\text{mm}$  のガラス・スケールを用いており、その精度は  $1.6 \mu\text{m}$  である。<sup>22)</sup>

表4.1 リニア・エンコーダ・インターフェース制御命令

数値表示は10進数6桁と符号で表され、これがそのまま、外部端子に出ているので、これを用いてミニコンピュータに数値を入力することができる。ただし、10進数1桁を4ビットのBCD(Binary Coded Decimal)コードで表しているため、10進数6桁と符号で25ビット、 $x$  軸、 $y$  軸両方で50ビットの入力が必要である。したがって、アナログ入力インターフェースの所で述べた汎用インターフェース(ML4891)を用いても16ビットしか入力できないため、リニア・エンコーダ用のインターフェースを、第4.2.1節のNEAC-M4 入出力システムを参考にして作成した。命令の使い方を表4.1に示す。

標準入出力チャネルのデータ・バスは8ビット構成であるので、BTI命令1回の実行で10進数2桁を読み込むことができる。そこで表4.1

コード	機能
FUN	0 制御部リセット
	1 割込信号リセット
	3 $x$ 軸、 $y$ 軸カウンタリセット
	4 $x$ 軸カウント開始
	5 $x$ 軸カウント停止
	6 割込禁止
	7 割込禁止解除
SEN	0 $y$ 軸カウンタリセット終了検出
BTI	0 $x$ 軸2桁入力 ( $10^0, 10^1$ )
	1 " ( $10^2, 10^3$ )
	2 " ( $10^4, 10^5$ )
	3 $x$ 軸符号入力
	4 $y$ 軸2桁入力 ( $10^0, 10^1$ )
	5 " ( $10^2, 10^3$ )
	6 " ( $10^4, 10^5$ )
7 $y$ 軸符号入力	

に示すように、オーダ・コード8種類すべてを使うことによって  $x$  軸6桁、 $y$  軸6桁ならびに両者の符号をすべて入力することが可能である。配線図を図4.2.3 ( $x$  軸のみ) に示す。オーダ・コードの切換によって順次2桁ずつ読み込まれる。 $y$  軸についても同様である。

インターフェースの全体のブロック図を図4.2.4に示す。図中右端のLE.X, LE.Y はそれぞれ  $x$  軸、 $y$  軸のリニア・エンコーダ・カウンタ部を示す。FUN回路ではFUN 3 によって両方のカウンタのリセットを行う。とくに  $y$  軸のカウンタはリセットに  $14\text{ms}$  かかるため、RESET 回路でワンショット・マルチバイブレータにより  $14\text{ms}$  幅のパルスを作り出している(図4.2.5)。また、このリセットが完了したかどうかの判定をSEN命令でセンスできるようになっている。

テーブルと主軸頭の移動量の検出のためには以上の制御機能があればよい。ただし、ここでは文献17)に述べられている、工具の実効工具径をインプロセス測定で把握する手法を考慮に入れておくことが望ましい。エンドミル外周による輪郭フライス切削では、工具は常に工作物から逃げた状態で切削しているので、切残しを生じ寸法誤差の要因となる。しかもその逃げ量は刃先丸みに依存することが解析されており、

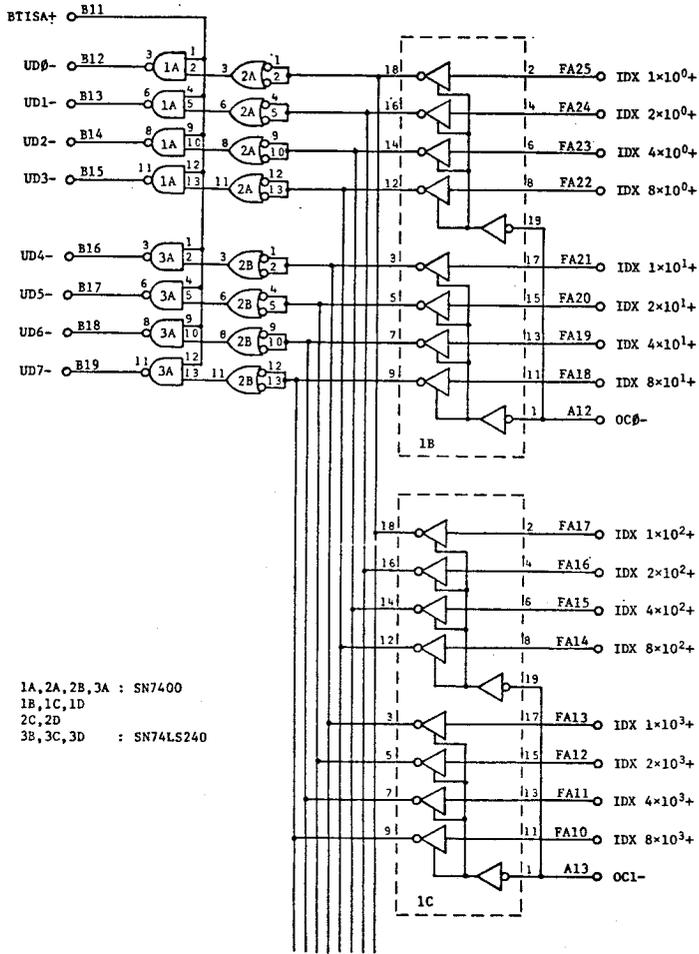


図 4.2.3 BTI 入力回路 (一部)

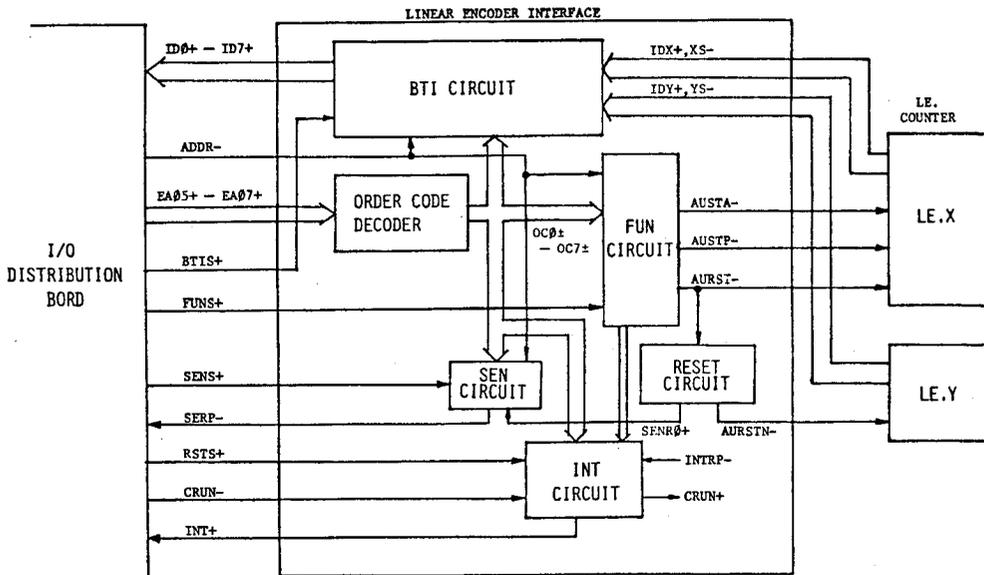


図 4.2.4 リニア・エンコーダ・インターフェース

一定時間間隔でテストピースを切削し、その切削面の位置を正確に知って、実効工具径を把握する手法が提案されている。これにリニア・エンコーダを利用する。すなわち、コンタクト・フィーラをテスト・ピースの切削面に当て、その瞬間にリニア・エンコーダのカウンターを停止させて、切削面の正確な位置を知る。このためにはリニア・エンコーダのカウンタについて、外部信号によって停止と開始が制御できなければならない。そこでx軸のカウンタ部について改造を行ってこの機能を実現している。表4.1ならびに図4.2.4において、FUN

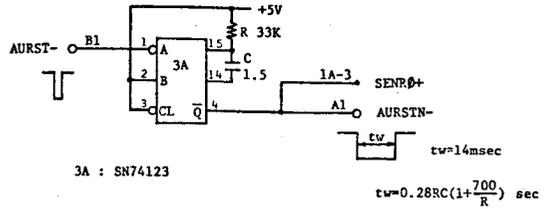


図 4.2.5 リセット回路

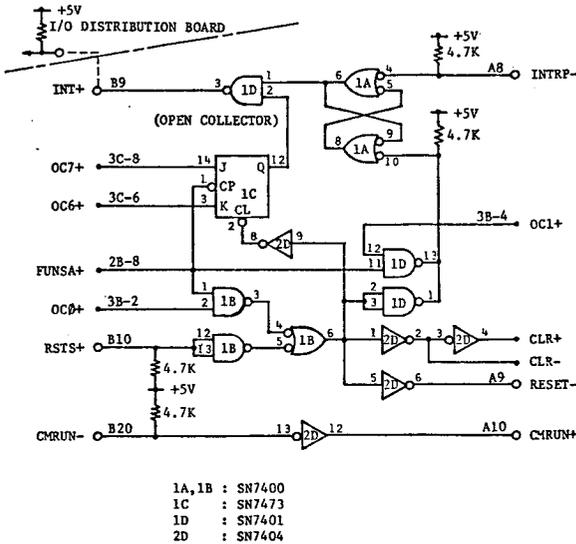


図 4.2.6 割込信号制御回路

4 でカウンタ開始、FUN 5 でカウンタ停止となるようにし、コンタクト・フィーラだけでなくプログラムでもこの制御できるようにしてある。また、コンタクト・フィーラが規準切削面に触れたとき、割込み信号が得られるようにINT回路も用意されている。

図4.2.6に割込信号の制御回路を示す。コンタクト・フィーラからの割込信号は図中右上のINTRP-に接続されラッチされる(1A)がそれがそのまま計算機本体への割込みとはならない。ゲート回路(1D)とフリップ・フロップ(1C)によりマスクがかけられており、FUN 7 命令でフリップ・フロップがセットされると始めて割込信号が計算機本体に伝えられる。また、FUN 6 命令の実行でマスクを閉じて割込み禁止となり、

また FUN 0 または1で割込み要求のラッチをリセットする。第4.2.1節で図4.5に関連して述べたように割込みラインにもマスクをかけることができるが、各周辺装置についても割込みの禁止とその解除をFUN命令で実行できるようにして、プログラムの作成を容易にしているのである。

### 4.3 位置決め精度に関する考察

#### 4.3.1 DSサーボシステム

加工精度向上のための適応制御を実施するにあたり、数値制御工作機械が本来持っているテーブルや主軸頭の位置決め精度を評価しておく必要がある。本研究の実験に用いた数値制御装置NEDAC 4200のDSサーボシステム(Dual Phase Selecting Servo, 日本電気)の原理図を図4.2.7に示す。<sup>23)</sup> 図に示すように、モータの回転位置は多相位置検出器によって検出され、位置決めループ(閉ループ)を構成しているが、それ以降の歯車列、送りネジ、テーブル位置の検出は行っていない。この準閉ループ系の特徴は機構誤差、非線形性、機械剛性がループの外に出ているので安定性がよいという利点がある。<sup>24), 25)</sup> 反面、テーブルや主軸頭の位置決め精度は機械そのものの特性に依存するため注意を要する所である。

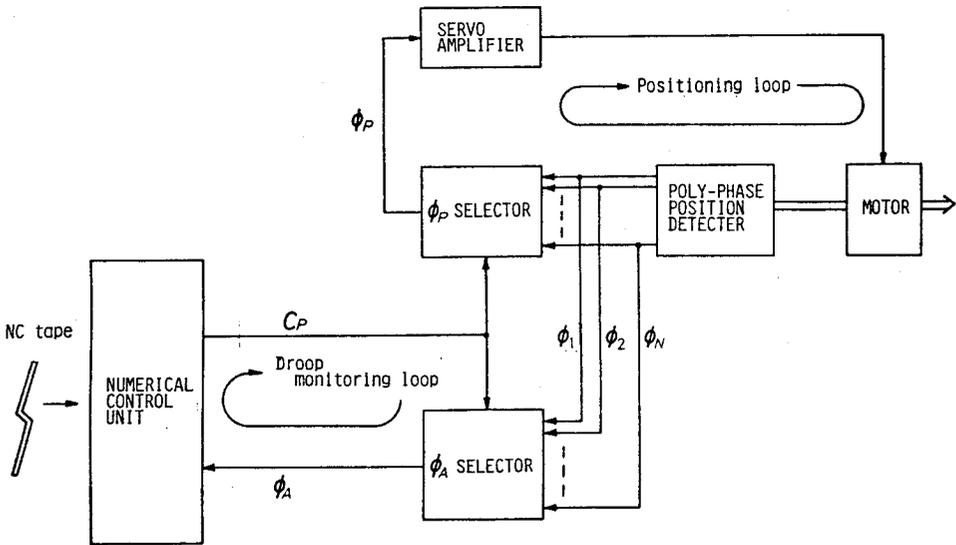


図 4.27 DSサーボシステム

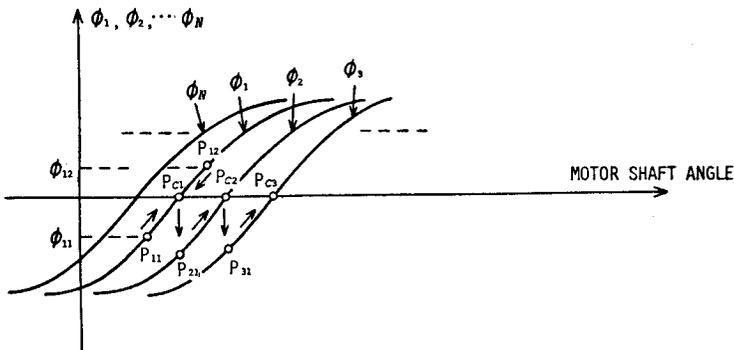


図 4.28 位置決め制御の原理

DSサーボのもう一つの特徴は、ドループ監視ループが存在することで、これによりパルスモータのような開ループ系にありがちな乱調現象を皆無にし、パルスが抜け落ちることを完全に防止している。これが Dual phase(Positioning phase, Alarm phase)といわれる所以である。

モータ軸の回転位置は多相位置検出器 (poly-phase detector) により検出される。 $\phi_P$  セレクタ (図 4.27) では、検出器の出力  $\phi_1, \phi_2, \dots, \phi_N$  の中から、指令位置信号  $C_P$  パルスを受けとるごとに、現在選ばれている出力の一つつぎの出力を選び出して位置決め相信号  $\phi_P$  とする。位置決め制御の原理を図 4.28 に示す。いま  $\phi_P = \phi_1$  であるとする。モータの回転軸の実際位置が  $\phi_P = \phi_{11}$  の値を生ずるような点  $P_{11}$  にあるとすると、その時の  $\phi_P$  の値 ( $=\phi_{11}$ ) に相当する駆動エネルギーがモータに加えられ、 $\phi_P$  の値が零になる位置  $P_{c1}$  に向かって矢印のごとくモータの軸がまわる。このことは丁度点  $P_{11}$  が  $\phi_P = \phi_1$  上を点  $P_{c1}$  に向かって移動することに対応する。モータの回転軸が  $\phi_P = \phi_{12}$  の値を生ずるような点  $P_{12}$  にあるときも、モータは逆転して  $P_{12}$  が  $\phi_P = \phi_1$  上を  $P_{c1}$  に向かって移動する。つぎに制御装置からセレクタに指令位置信号  $C_P$  パルスが一つ与えられると、 $\phi_P = \phi_2$  となる。このとき  $\phi_P$  の値は点  $P_{21}$  に対応する値になるのでモータは回転し、 $P_{21}$  は  $\phi_P = \phi_2$  上を点  $P_{c2}$  に向かって移動する。この一つのパルスによってモータ回転軸は 1.2 度回転し、テーブルあるいは主軸頭は

5  $\mu\text{m}$  移動することにする。

ところで指令位置と実際の回転軸の位置との差すなわちドループは、定常状態では問題にならないが、外乱がモータに加わり負荷が重くなったり、指令速度が急に増減したりするとドループが有効ドループ範囲を越えることがしばしば発生する。これを放置すると、ドループが大きくなるにつれて  $\phi_P$  が小さくなり、指令位置に向かって実際位置が追従する機能が弱まるため、このような事態は直ちに脱調に連なることになる。再び図 4.27 において、DSサーボシステムでは  $\phi_A$  セレクタによって、多相位置検出器の出力から監視相信号  $\phi_A$  を選択する。 $\phi_A$  は  $\phi_P$  と位相が  $90^\circ$  ずれており、これを制御装置にフィードバック

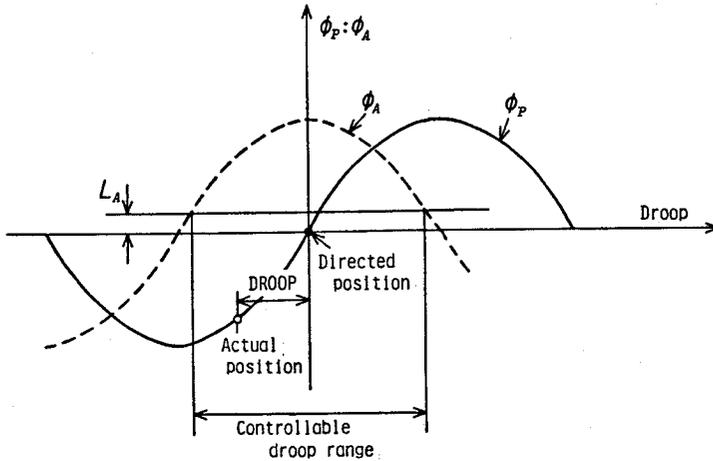


図 4.29 監視相信号  $\phi_A$

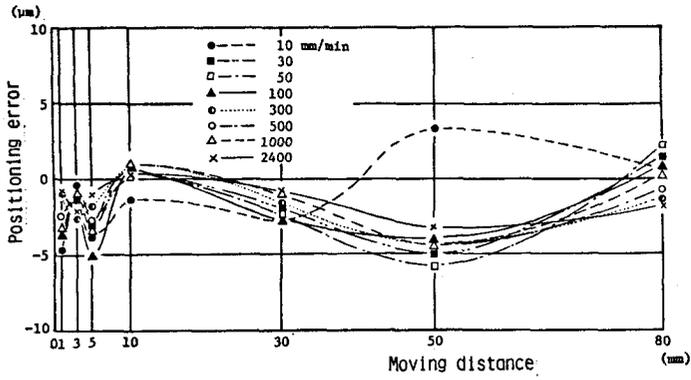
することによって、実際位置が有効ドループ範囲内にあるか否かを判定し、もし実際位置が有効ドループの範囲を越えれば、ただちにパルス発生を抑えて脱調現象を防ぐ（図 4.29）。

### 4.3.2 位置決め精度の測定

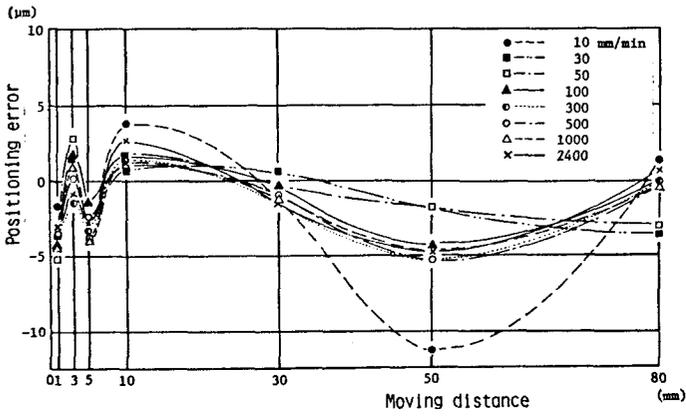
第 4.2.5 節で述べたように、本研究で用いた NC フライス盤には工作物の形状精度測定のために、X 軸（テーブル）と Y 軸（主軸頭）に光電式リニア・エンコーダが取り付けられている。これを用いると  $x y$  平面上の機械スライド位置決め精度を実測することができる。図 4.30 に実測例を示す。ここでは特定の目標位置一点を定め、移動距離と送り速度をかえて実測したものである。なおこの図では、バックラッシュの影響を除くため、各送り方向に予め 10 mm の送りを与えその上で送りの開始点を定めている。位置決め制御が 5  $\mu\text{m}$  単位に行われることを考えれば、 $\pm 5 \mu\text{m}$  の間では当然ばらつくことが考えられ、その意味からも良好な結果を示していると思われる。送り速度をかえてもあまり大きな変化は見られない。ただ、ばらつきによって  $\pm 10 \mu\text{m}$  程度の誤差が現われることもあり、この点は注意を要する。

図 4.31 は実際に直線切削を実行しながら位置決め精度を調べたものである。加工条件を表 4.2 に示す。図 4.31 を見ると位置決め誤差はおおむね  $\pm 5 \mu\text{m}$  の中に入っており、この程度の切削負荷の影響はほとんどないといえる。

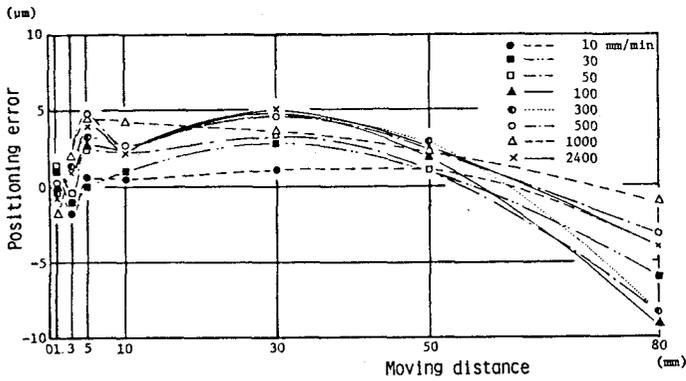
つぎに、バックラッシュも含めて位置決め精度を調べるため、特定の位置からある距離だけ往復動作をさせて、もとの位置にもどるかどうかを確認した。その結果が図 4.32 である（X 軸についてのみ示す）。ここでは往復動作の距離を横軸にとり、送り速度をパラメータとして変えている。本研究に用いた NC フ



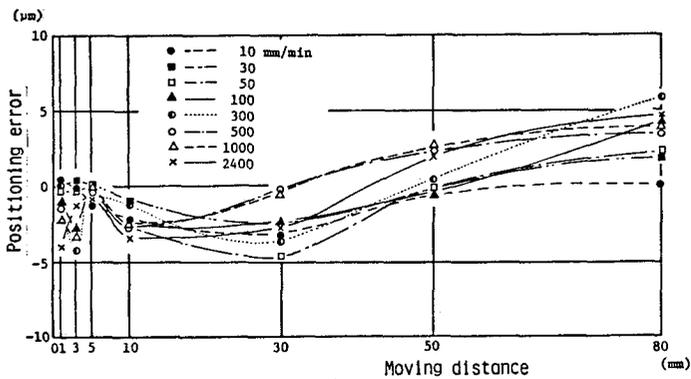
(a) X軸(+)方向、無負荷



(b) X軸(-)方向、無負荷

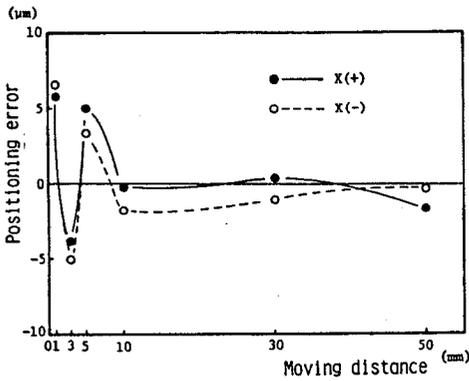


(c) Y軸(+)方向、無負荷

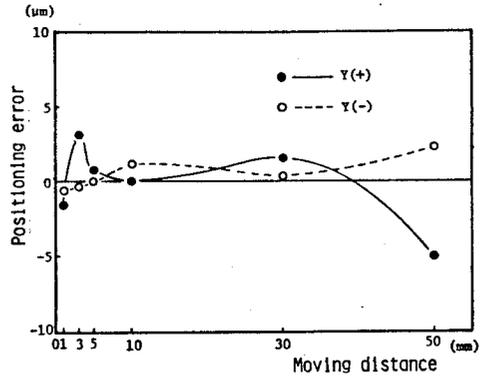


(d) Y軸(-)方向、無負荷

図 4.30 送り速度による停止位置の変化

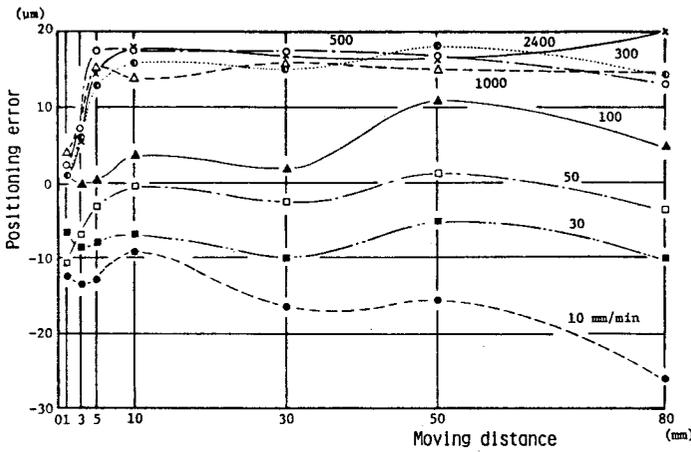


(a) X軸方向、負荷時

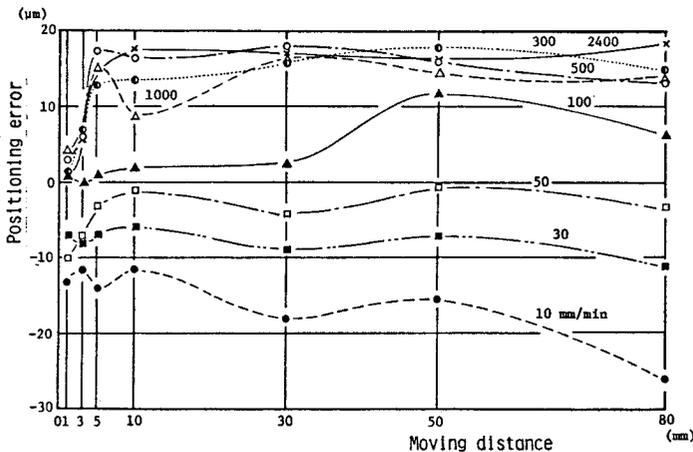


(b) Y軸方向、負荷時

図 4.3.1 切削負荷時の位置決め精度



(a) X軸(+)方向→(-)方向



(b) X軸(-)方向→(+)方向

図 4.3.2 バックラッシュを含めた位置決め精度

表 4.2 加工条件

エンドミル	SKH3, 20mmφ
被削材	S35C
送り速度	50mm/min
主軸回転数	354rpm
切込み深さ	2mm
切削幅	10mm
切削油	使用

ライズ盤のバックラッシュ補正は、各運動軸において動きの方向が逆転するとき、電氣的に数パルス挿入することによって行っている。このパルス数は手動で0から15パルス(75μm)までの値を設定できる。図4.3.2を見ると、往復運動における切削速度の影響が極めて顕著に現れている。ちなみに、バックラッシュ補正パルスの設定は通常の加工実験で使われる50mm/minの送り速度で調整してある。この図から、バックラッシュの補正パルスの調整は、実際に加工するときの送り速度

で行わないと意味がないことを示している。

位置決め精度を動的精度として扱う場合には、その精度の数値がどのくらいの確からしきで、すなわち

多数回の位置決め動作に対してその中の何回は、その表示精度に入ると考えてよいのか、といった信頼度の考え方の導入が必要である。米国のNMTBA(National Machine Tool Builders' Association)では図4.3.3に示すような位置決め誤差の発生頻度曲線

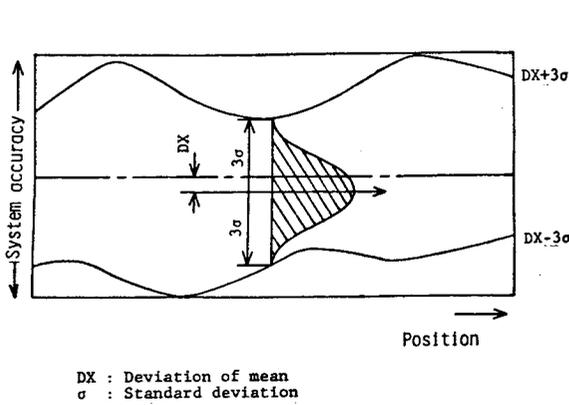


図4.3.3 システム精度

から標準偏差  $\sigma$  を求め、 $\pm 3\sigma$  によって信頼度を表現することを提唱している。<sup>26)</sup> 位置決め誤差が正規分布をすると仮定すると、くり返し位置決め動作の中、99.7%がこの領域に入ることになる。

本研究で用いている実験用システムでは、ミニコンピュータからNC指令を転送することができ、またリニア・エンコーダの表示値はデジタル・インターフェースを介してミニコンピュータに読み込むことができる。これらの点を利用して図4.3.4に示す構成で、NMTBAによる3シグマ処理を実行するプログラムを作成した。このシステムでは、予め停止位置の座標と送り速度ならびに送りの反復数を入力しておく、各停止位置への送りとドウェルを実行し、位置決め誤差を読みとってディスクに格納する。すべてのくり返しが終了すると、ディスク内のデータに対して統計処理を行い、結果をタイプライタに出力する。

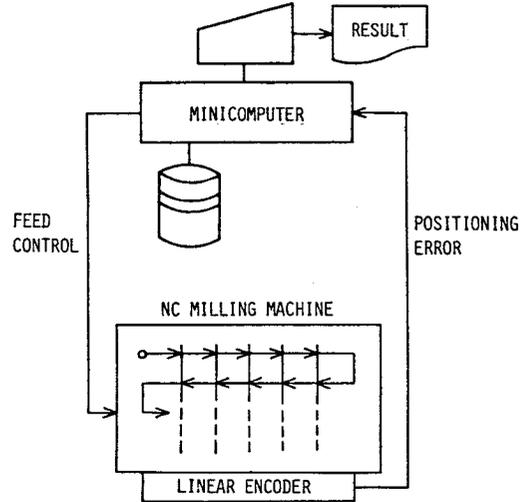


図4.3.4 位置決め精度測定システム

3シグマ処理による精度評価を、図4.3.1の場合と同様に表4.2の加工条件の下で直線切削を行いながら測定した結果を図4.3.5に示す。くり返し数は7回である。X軸とY軸とでは大差はなく、おおよそ $\pm 20\mu\text{m}$ の結果が出ている。ただし、これは $20\mu\text{m}$ の誤差が実際に出現したことを意味するものではなく、実測された標準偏差を機械的に3倍して得られたものである。本実験に用いているNC装置は、制御の単位が $5\mu\text{m}$ と粗いので、位置決め誤差の分布が正規分布になるとは考えにくい。当然 $\pm 5\mu\text{m}$ の間ではばらつきが生ずるので標準偏差は大きくなるが、それを単純に3倍するのでは、誤差を過大評価することになるとと思われる。図4.3.0あるいは図4.3.1より考えれば多くの場合が $\pm 10\mu\text{m}$ の間に入るものと思われる。確からしさを明らかにできないが、實際上 $\pm 10\mu\text{m}$ を位置決め誤差と考えてよいであろう。逆の見方をすれば、 $\pm 10\mu\text{m}$ 以下に加工精度などの誤差を抑え込むことは、本実験の装置を持ってしては困難であるといわざるを得ない。

### 4.3.3 工作機械熱変形の影響

位置決め精度を考えるのに無視できない一つの要因として、工作機械の熱変形がある(第1章の参考文献参照)。本研究で用いているのは門型立フライス盤であり、加工精度に最も影響を与えるのは、工具が取り付けられる主軸頭下端のテーブルから見た相対変位である。フライス盤はy軸に関して対称であるの

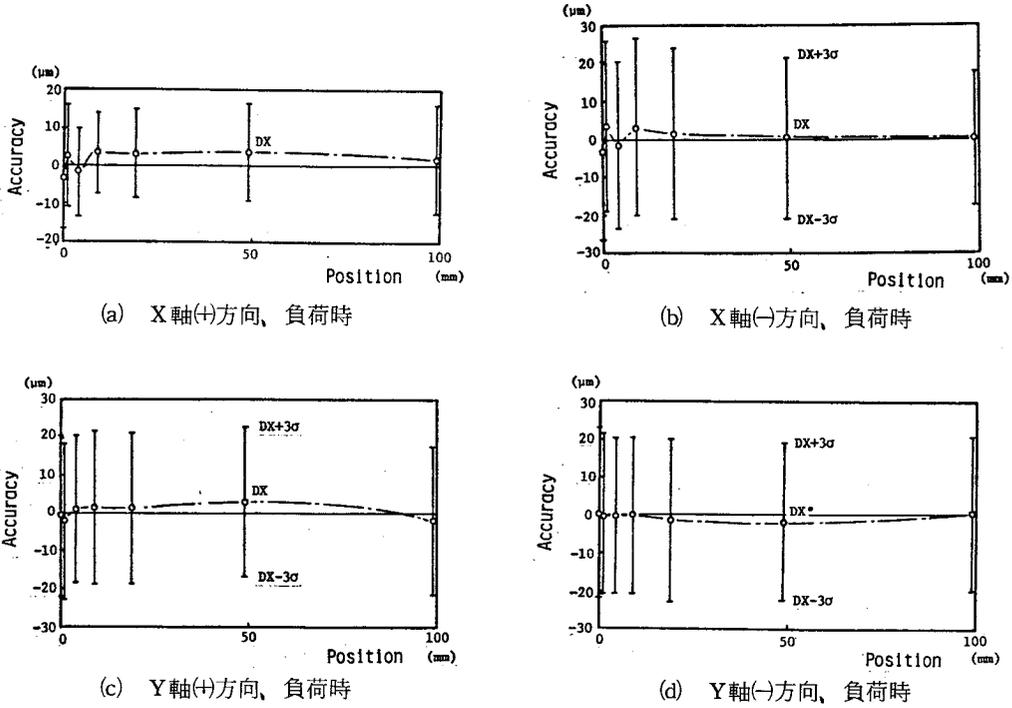


図 4.3.5 3シグマ法による精度評価

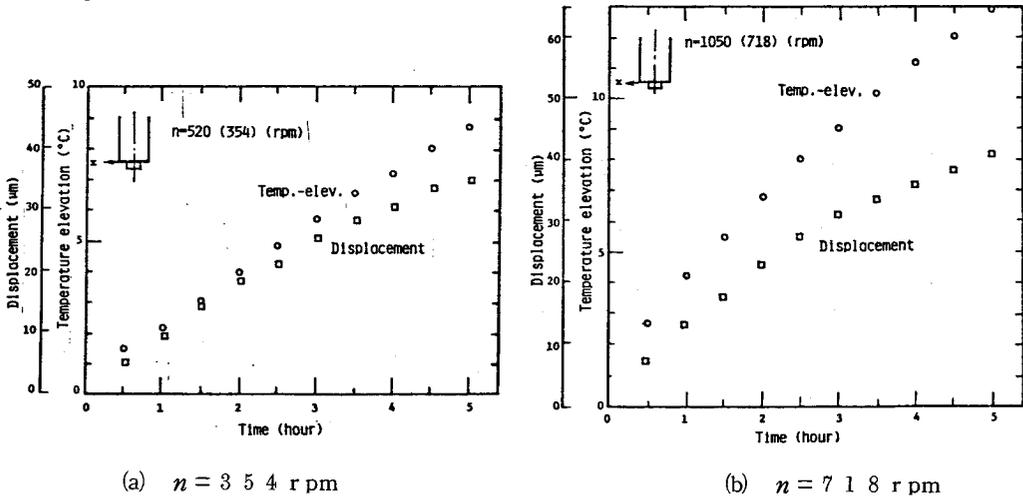


図 4.3.6 主軸部下端の温度上昇と変位

で、 $x$  軸方向の変位が問題となる。主要熱源は主軸回転による歯車列ならびに軸受けからの発熱と、油圧サーボモータの油（約  $40^\circ\text{C}$ ）からの熱の流入である。以上の点を考慮して、実際に主軸を回転させながら、テーブル上に固定されたダイヤルゲージを用いて、主軸頭下端の  $x$  軸方向の変位を 30 分間隔で測定した。結果を図 4.3.6 に示す。主軸部下端の温度上昇を併せて示してある。工作機械は熱容量が大きいので、5 時間経過後も定常状態に達していない。主軸回転数の違いによって流入熱量が異なるため、図(a), (b)で温度上昇、変位共に異なった結果が出ている。主軸回転数  $n$  が  $354 \text{ rpm}$  のときで 5 時間経過後約  $35 \mu\text{m}$ 、 $n$  が  $718 \text{ rpm}$  のときで約  $40 \mu\text{m}$  の変位が見られ、位置決め精度に大きく影響することが分る。

ただし、本研究で取り扱う実験では、板カム輪郭を一回の切削で仕上げることを目標としており、一回の輪郭切削の所要時間は10minないし20min程度であるから、この間での主軸頭下端の相対的変位は2~3 μmであり無視して差し支えない。

一つの部品の加工に数時間を要するときは、主軸頭下端の変位に対する補正が必要になるが、CNC装置を用いる場合には、主軸頭下端の変位に対応して工具径路を修正することはある程度までは可能である。ただし、熱的挙動は極めて不安定であるので適応制御の方式を採用する必要がある。実際に加工しているときには主軸頭が移動するため、その変位を直接測定することはむづかしい。そこで試みに主軸頭下端の温度変化と変位の関係を調べると、図4.37のようになる。これによれば、変位はほぼ温度上昇に比例しているのので、温度測定を行えば変位は容易に推定できる。

工作機械の温度測定について簡単に述べる。工作機械の室温からの温度上昇は高々20℃程度であり、温度センサーとしては、熱電対の冷接点補償器(第4.2.3節)に用いたIC温度センサーμPC616Cを利用すると便利である。工作物の温度測定と異なり、センサーの熱容量や接触面積などが問題にならない。ただ、工作機械には主軸モータなどが取り付けられているため、雑音対策を十分に講じる必要がある。このためIC温度センサーと一体になるように増幅器を取りつけ、見掛け上センサーの出力が100mV/℃となるようにして使用する(図4.38)。図に示すように零点調整用ボリュームがついているので、ある特定の基準温度(例えば0℃あるいは測定開始時の室温)について出力が0Vとなるように一度調節すれば、温度測定中には冷接点補償は不要である。測温部に

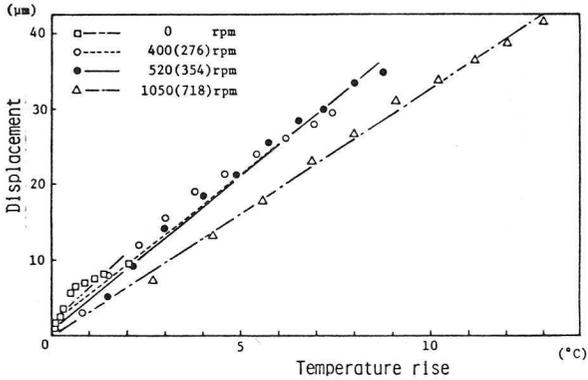
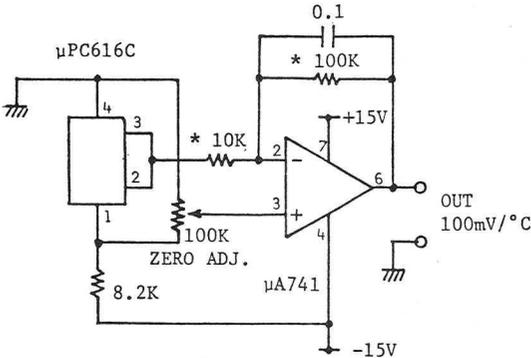
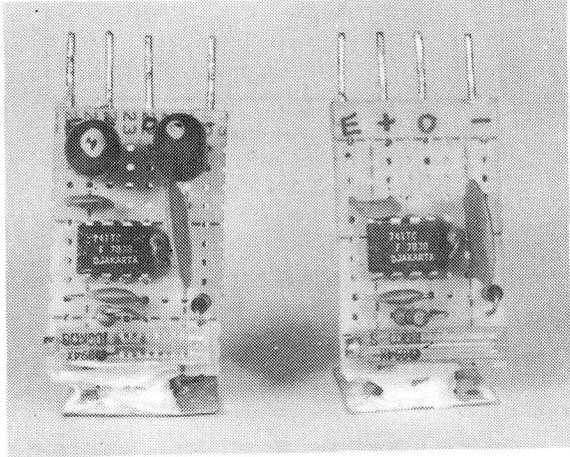


図 4.37 主軸頭下端の温度変化と変位



\* 1% RESISTOR

(a) 回路図



(b) 外観

図 4.38 IC温度センサー

は 2 cm×2 cm の銅板を使用し、IC 温度センサーのモールド・パッケージをこの銅板に接着し、銅板を工作機械の地金に接触させて温度を測る。この部分の熱伝導をよくするため、シリコングリスを塗る。

アナログ入力インターフェースを介してミニコンピュータにデータ入力を行うものとすれば、最大 16 点の同時測定が可能である。この場合、アナログ入力インターフェースの初段の増幅器基板をすべてバッファ・アンプに差しかえて、プログラマブル・ゲインの調節だけで十分な利得が得られる。図 4.39 に工作機械温度測定の例を示す。これは有限要素法による温度分布解析の結果を実測温度とつき合わせたもので、括弧内の数値が実測温度である。このように工作機械の熱的挙動を実時間で把握し、位置決め精度の向上を図ることも可能であると思われる。<sup>27)</sup>

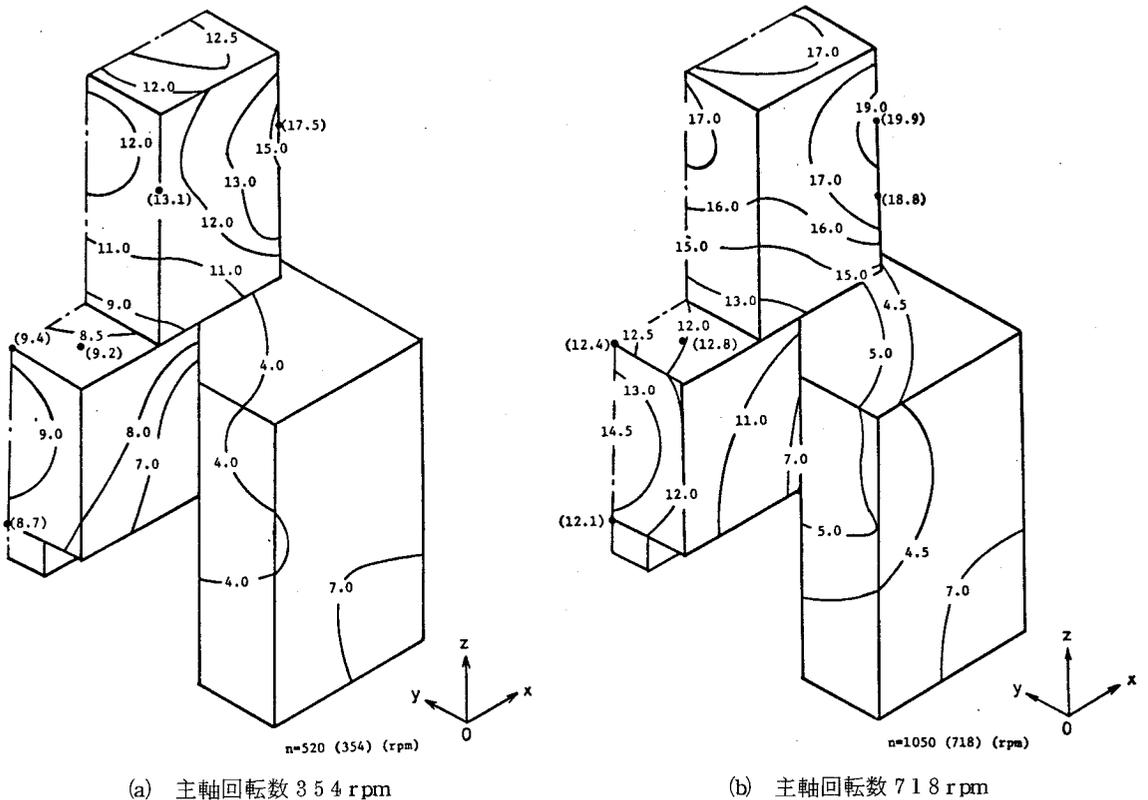


図 4.39 工作機械の温度分布 (5 時間経過後)

## 4.4 制御方式

### 4.4.1 制御用データの構成

工作物熱変形を考慮して板カム輪郭切削を行う場合、緒言において述べたように、熱の流入流出量を予め定量的に知ることはできないため、熱変形量を実時間で推定する制御方式を取る。しかも加工中に切削点の変位を直接求めることは困難であるから、測定温度から推定しなければならない。このため、制御用データには以下の 3 種類の情報を準備しておく。

- (1) 板カム輪郭を表す円弧の区切り点の座標と、その点での法線の偏角
- (2) 板カム回転中心における切削熱による温度上昇予測値

### (3) 工作物熱変形に起因する切削点の変位予測値

NC加工するため、板カム輪郭を円弧列で近似する方法を第1編第2章で述べたが、円弧の区切り点の座標とその点での法線の偏角(制御用データ(1))が分っておれば、円弧補間のNC指令は容易に作れるのである。制御用データの(2)と(3)は本編第2章と第3章で述べた有限要素法によって求められるものであり、その解析に用いられた有限要素節点に関連して得られたデータである。すなわち、工具がある輪郭外周上の有限要素節点を通過するときのカム回転中心の温度上昇と、そのとき工具の位置する節点の熱変位(x軸成分、y軸成分)の予測値が与えられている。板カム輪郭の円弧の区切り点の決定と自動要素分割プログラムとは独立に実行されるので、有限要素節点と円弧の区切り点とは必ずしも一致しない。そのため、輪郭外周上の有限要素節点についてもその位置を極座標で表し、関連する制御用データに円弧の区切り点と区別するための識別子をつけて、輪郭にかかわる円弧の区切り点のデータとませ合わせる。

ミニコンピュータに大きなメモリ容量を期待することはできないので、制御用データを逐次型ファイルの構成とし、順に一つずつデータを読み取って行くだけで制御が実施できるようにする。またこのことによって、シミュレーション時の時間のスケールと実際の切削過程での実時間のスケールを合わせることができるのである。なお、有限要素節点に関する制御用データについては、後述するように内挿演算を行う必要があるので、常に一つの先のデータが読み取れるようにしておくことが必要である。

#### 4.4.2 温度測定による熱変形推定

制御用データの中で円弧の区切り点になっている点列を $\{P_j\}$ ( $j=1, 2, 3, \dots$ )とし、輪郭外周上の有限要素節点の列を $\{M_i\}$ ( $i=1, 2, 3, \dots$ )とする。これらの点列の偏角を改めて $\theta$ で表し、有限要素節点に関するものには添字 $i$ を、円弧の区切り点に関するものには添字 $j$ をつけるものとする。円弧の区切り点の間隔は通常1mm~数mmであり、有限要素節点はこれらよりもあらく、一つの間隔の中に数個の円弧区切り点を含む。制御用データは通信回線を通して送られて来るが、工具の送り速度は通常1mm/s程度であるので、伝送速度1200bpsで十分間に合う。ただし、くり返し同じ実験をするときには紙テープに出力しておいて、ミニコンピュータの紙テープ・リーダーから読み込む。

工作物の温度変化は比較的緩慢なプロセスであるから、ミニコンピュータがある有限要素節点 $M_i$ に関するデータを読み取ったところで板カム回転中心の温度測定を行う。この時点で工具は対応する有限要素節点 $M_i$ の近傍に来ていることになる。測定温度 $\Delta T_i'$ (室温からの上昇温度)と有限要素法による予測値 $\Delta T_i$ を用いて、熱量についての補正の因子を以下のように定める。

$$k_i = \frac{\Delta T_i'}{\Delta T_i} \quad (4.1)$$

この値は次の節点 $M_{i+1}$ で温度測定を行うまで、近似的に有効であると考えられる。

また一方、点 $M_i$ における熱変位の $x$ 、 $y$ 成分の予測値を $u_i, v_i$ で表すと、加工時の温度測定に基づく点 $M_i$ の変位推定値を近似的に

$$u_i' = k_i u_i, \quad v_i' = k_i v_i \quad (4.2)$$

によって求める。加工中に実際の変位を直接測定することは困難であるから、上昇温度の予測値と実測値の比を用いて変位を間接的に推定しているのである。この比例関係がかなり広い範囲について成り立つことは、有限要素法の計算から確かめられている。<sup>28)</sup>

図4.4.0に示すように、点 $M_i$ と点 $M_{i+1}$ の間にはいくつかの円弧の区切りが存在するが、これらの点では偏角 $\theta$ について内挿を行う。例えば図4.4.0に示すように点 $P_j$ での $x$ 軸方向の推定変位を次式で計

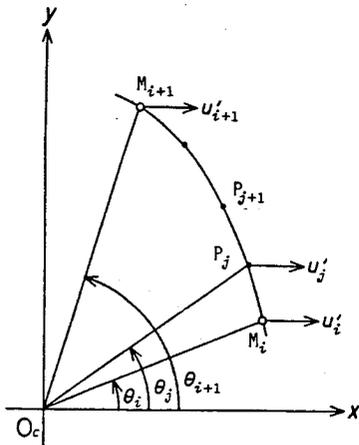


図4.4.0 輪郭上の点列と  $x$  方向の変位

修正パルスが要求される点が高々10個程度の場合、外周上の有限要素節点が30ないし50個あればこの近似式を用いても大きな誤差は生じないと考えられる。もともと  $k_i$  は温度変化そのものを表す因子ではなく、シミュレーションの結果と実際のプロセスとの温度変化の違いを見ているのであるから、シミュレーションと実時間のプロセスの条件が大きく変っているのであれば  $k_i$  の変化は小さく、それ程短いサンプリング間隔を設定する必要はないと考えられる。急激な加工条件の変化にも対処できるためには、シミュレーションの過程においても短時間間隔ごとのデータの記録を残しておく必要があるであろう。

#### 4.4.3 加工時の工具径路決定

DNCあるいはCNCシステムでは切削加工と並行して実時間でNC指令を生成することが可能である。本実験システムでも、前節で述べた推定熱変位と輪郭から求められた点の座標値に基づいて、実時間でNC指令を生成して行く。NC指令は  $x y$  座標系の円弧補間を用いる。したがって、必要に応じて極座標値を  $x y$  座標値に変換する。

熱変位を考慮して工具径路を決める場合、以下の三通りの方法が考えられる。

- (1) 最も厳密な方法は、輪郭上の点列から円弧の区切り点を構成する内心点の決定を、熱変位した点に基づいて実時間で決定して行く方法である(第1編第2.4節参照)。この方法では高速の演算処理が必要であるため、ミニコンピュータに倍精度浮動小数点演算機構が備わっていることが必要である。他方実用面から考えると、工作物熱変形の補正にここまで精度が要求されるかどうか疑問であり、もう少し簡略な方法でもよいのではないかと思われる。
- (2) 第2の方法は内心点の決定については予め計算したものをを用い、ただ各円弧の区切り点については忠実にその変位を考慮しようとするものである。本研究ではこの方法を採用するので具体的に図を用いて詳しく後述する。
- (3) 第3の方法は各円弧の区切り点において、その法線方向の変位だけを考えようとするものである。実際問題として接線方向の変位は一般的に小さく、無視できる場合が多いと考えられるので、この方法でも十分であろう。とくにこの方法の利点は、工具径路についてみれば、法線方向に逃がすだけでよいので、広い意味の工具径補正に含めて考えることができ、それだけ制御プログラムが簡単になることである。实用価値の極めて高い方法といえる。

さて、第2の方法について、変位を拡大して示した図を図4.4.1に示す。ここで、NC指令の作成には

算する。

$$u_j' = (k_i u_{i+1} - u_i') \frac{\theta_j - \theta_i}{\theta_{i+1} - \theta_i} + u_i' \quad (4.3)$$

$y$  軸方向の成分  $v_j'$  についても同様である。点  $M_{i+1}$  では再び温度測定を行うので改めて

$$u_{i+1}' = k_{i+1} u_{i+1} \quad (4.4)$$

が計算されるが、それまでの間の点  $P_{j+1}, \dots$  での推定変位量は  $u_{i+1}' = k_{i+1} u_{i+1}$  として式(4.3)で計算する。ここで熱量の補正のファクターについて、時間遅れが伴っていることを注意しておくことは必要であろう。しかしながら仕上げ加工の過程では、切削速度が切削途中で急に2倍になるといったような急激な加工条件の変化は考えなくてもよい。したがって、補

$x, y$ 座標系を用いる。また、工作物が熱変形した後も輪郭設計時に求めた法線の方法は近似的に有効であると仮定する。図において、点 $P_j$ が $P_j'$ へ、 $P_{j+1}$ が $P_{j+1}'$ へ変位したものとす。  $P_j'$  を通り  $O_j P_j$  に平行な直線を引き、  $P_{j+1}'$  を通って  $O_j P_{j+1}$  に平行な直線との交点を  $O_j'$  とする。  $O_j'$  を中心とし線分  $O_j' P_j'$  を半径とする円弧を求める。工具径路はこれより工具半径だけ半径の大きい円弧  $Q_j Q_{j+1}$  によって定められる。  $Q_{j+1}$  は  $O_j' P_{j+1}'$  の延長線と円弧の交点である。工具が  $Q_{j+1}$  に達したとき、切削点は必ずしも  $P_{j+1}'$  に一致しないので、  $5 \mu\text{m}$  以上の差があるときは、切削点を  $P_{j+1}'$  に引き込むように補正の NC 指令を作る。輪郭が十分細かく区切ってあれば、この補正は丸めの操作に吸収されるかまたは 1 パルス分で済む。なお、  $O_j$  と  $O_j'$  の差が小さく無視できる程度であれば、上に述べた第 3 の方法でよいことになる。

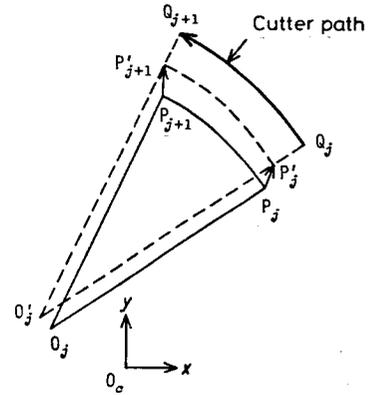


図4.4.1 熱変形に対する工具径路決定

ミニコンピュータは、以上の計算を一つの NC 指令で加工が行われている間に実行し、次の NC 指令を作成して待つようになっている。計算そのものは簡単であるが、交角の小さい 2 直線の交点を求めることが多いので、けた落ちが起こりやすく、有効けた数を十分に多く取っておくことが必要である。大型計算機による設計計算でも倍精度演算を用いて有効けた数 10 進 18 けた、ミニコンピュータでも 10 進 13 けた程度（仮数部 46 ビット）の有効けた数で演算を行っている。本実験に用いる NC 工作機械は送りの最小単位が  $5 \mu\text{m}$  であり、終点座標と円弧中心座標の NC 指令値は工具現在位置からのインクリメンタル量（最後のけたは 0 または 5）で表現される。しかしながら、NC 指令が次々に作成されて行く各段階において、丸めの累積誤差を防ぐため、計算はすべて板カム回転中心を原点とする座標系で実行し、最終的には NC 指令を作るときに丸めの操作を行って相対移動量を算出する。計算の途中で丸めの操作を入れることは一切行わない。このようにすれば、計算上の誤差は常に  $5 \mu\text{m}$  以下に抑えられ、計算誤差は累積しない。

なお、図 4.4.1 に示す工具径路実時間補正では、工具径補正も実時間で実行できる利点がある。文献 17) によれば、工具刃先の丸みに応じて工具のたわみが生じ、公称工具径による工具径路では切残しによる寸法精度の劣化を来す。しかも刃先丸みは切削過程で刻々変化して行く。文献 17) ではテストピースを削ることによって、切前面から見た実効工具径を測定し、これに基づいて工具径補正を行う方法を提唱している。このような場合、従来からある操作盤上の手動操作による工具径補正機能では不十分であり、切前面から見た実効工具径を実時間でミニコンピュータに入力して工具径路の修正を行う必要があり、CNC システムが効果的に利用できる。また、本研究では具体的に取り上げていないが、工具の熱変形問題も残されている。すなわち、工具の熱変形量の把握ができれば、工具の熱変形にもとづく工具径路の実時間補正も CNC システムで実施可能である。

#### 4.5 実験と結果の考察

適応制御実験システムを図 4.4.2 に示す。加工終了時の輪郭測定機能もこの中に含まれる。各構成要素については、第 4.2 節で述べた。ミニコンピュータの記憶容量は 12 K バイトと小さいため、補助記憶装置として 64 K バイトのミニディスクが設置されており、加工実験における中間経過のデータ保存、輪

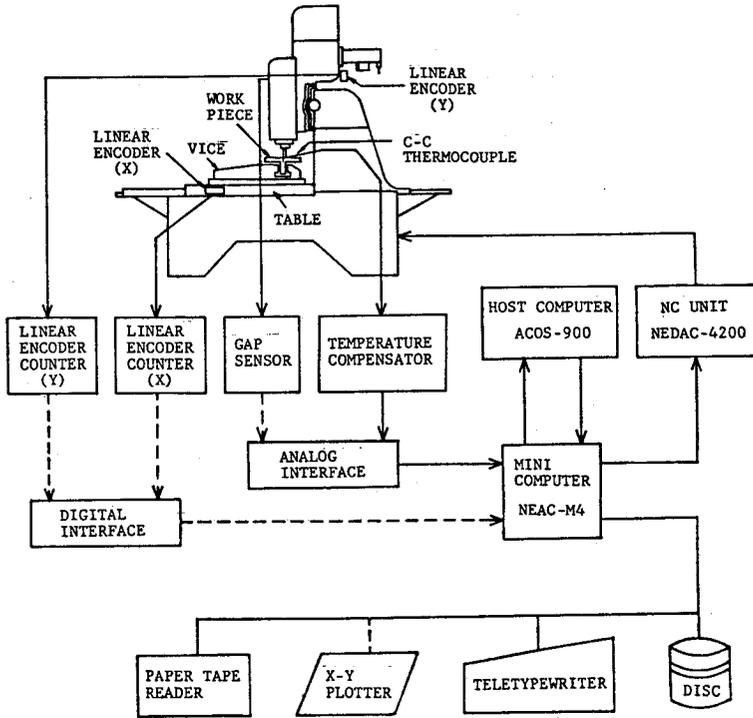


図 4.4.2 適応制御加工システム

郭計測時のデータの一時待避などに使われている。

実験に使用したNCフライス盤 (KOYO-VAN-NORMAN 2GNC) の諸元を表 4.3 に示す。加工実験において、工具は2枚刃のエンドミル (SKH3, 直径 20mm) を用いた。上向き切削で切削油は用いていない。被削材は厚さ 10mm の S35C で、加工条件は、切削速度 21m/min, 送り速度 50mm/min, 切込み深さ 2mm である。以下に3種類の形状の加工例を示すが、加工条件は同じである。加工条件を変えた実験についても結果を示すべきであるが、熱の問題は同一加工条件でも同じ過程を取らず、変動はその都度異なるので、不必要に加工条件を変えず、標準的なもの一種類を設定して実験を行った。

加工実験は室温状態にある荒加工処理をした被削材をバイスに取り付け、カム回転中心に対応する位置に銅・コンスタンタン熱電対を深さ 5mm に埋め込み、増幅器とアナログ入力インターフェースを通してミニコンピュータに接続する (図 4.4.3)。仕上げ切削は一度で済ませるので、工具が工作物まわりを一周した状態で工具を離し、工作物はバイスに取りつけたまま室温まで冷却する。これは輪郭測定をNC装置を用いて実行するためである。

カム輪郭の検証には通常カム試験機を用いるが、ここでは加工精度を直接調べたいため、加工時に工作物をバイスに取り付けた状態を動かさず、ギャップ・センサーとNC装置を用いて輪郭の検証を行う。ギャップ・センサーの使い方については第 4.2.4 節で述べた。重複する点もあるが、輪郭測定の概要を述べる。工作物が室温まで冷却されたならば、主軸に工具の代わりにギャップ・センサのピックアップ・コイルを埋め込んだ直径 19mm のプラスチックの丸棒を取りつける (図 4.4.4)。工作物は電導体であるから、ピックアップ・コイルと工作物の間隙が検出できる。図 4.4.5 に示すように主軸を低速 (100rpm) で回転させながら、NC装置の送りを利用して、仕上工作物の周りを理想形状に沿って主軸を動かす。ピッ

表 4.3 NCフライス盤諸元

		諸 元			
テーブル作業面	作 業 面 積		940×405mm		
	T 溝 の 巾		16mm		
	T 溝 の 数		4		
	T 溝 間 の 距 離		80mm		
運 動 範 囲	テ ー ブ ル 前 後 送 り		610mm		
	主 軸 頭 左 右 送 り		510mm		
	主 軸 頭 上 下 送 り		300mm		
主 軸	テ ー パ 孔		ASA#40		
	回 転 速 度 範 囲		100-2100rpm		
	速 度 変 換 数		6 段		
	自 動 送 り 込 み 量 ( Z 軸 )		1-1000mm		
テ ー ブ ル か ら 主 軸 先 端 ま で の 最 大 距 離			400mm		
テ ー ブ ル か ら 主 軸 先 端 ま で の 最 小 距 離			100mm		
機 械 寸 法	正 味 重 量		3500kg		
	巾		1950mm		
	奥 行 き		2830mm		
	高 さ		2250mm		
電 動 機	用 途	製 作 会 社	型 式	出 力	極 数
	主 軸 用	三 菱	SFE	3.7/18.5/1	2/4/8
	テ ー ブ ル 送 り 用	NEC	FM-512	油 圧 サ ー ボ 70kgcm <sup>2</sup>	

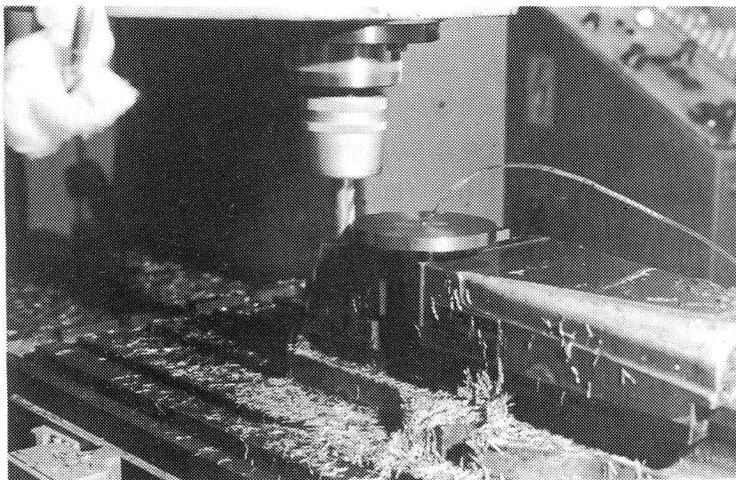


図 4.43 加 工 実 験

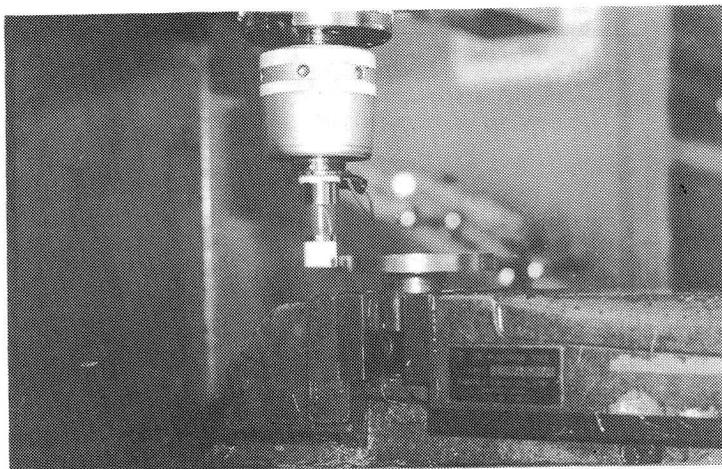


図 4.4.4 ギャップ・センサーのピックアップ部

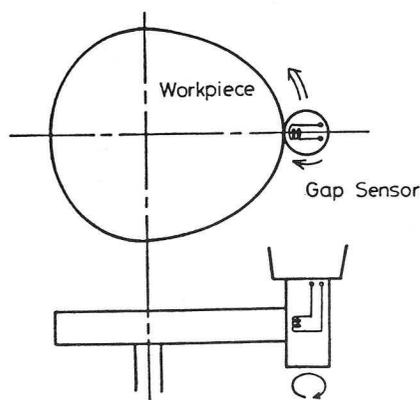


図 4.4.5 ギャップ・センサーによる輪郭測定

度ミニディスクに貯えられ、測定終了時にプロッタに出力される。

3 種類の形状について加工実験した結果を以下に示す。

(a) 真円加工の場合

図 4.4.6 に温度変化と熱変形量の予測値（有限要素法による）と温度変化の測定値を示す。半径は 53 mm である。切削開始から熱源の位置が工作物の周上 135° あたりまでは予測値と実測値は大体一致しているが、熱源の位置が工作物の周上 135° を越えると、実測値が予測値を 50% 程度上まわっている。熱変形量もそれに応じて大きくなっていると考えられる。

図 4.4.7 にギャップ・センサーによる輪郭形状測定結果を示す。矢印は切削開始点を示す。バックラッシュの影響を避けるため、切削開始点を  $x$  軸および  $y$  軸と 45° の交角を持つ点に選ぶ。工具の進入方向は接線方向である。点線は理想形状を、実線は理想形状からの偏差を 200 倍に拡大したときの工作物の形状を示しており、これらの曲線はプロッタに出力されたものである。熱膨張を無視して切削した場合、工作物の熱変形による形状誤差は、工作物が室温まで冷却したとき、切り込

クアップ・コイルはスリップリングを介して本体と接続され、間隙に対応する電圧はミニコンピュータに入力される。なお、理想形状とは板カムの設計計算によって求められた板カムの形状であり、工具半径のオフセットは 10mm とする。これにより板カム周囲に対してピックアップ・コイルが最も近接したとき、その間隙は約 0.5 mm に保たれる。ピックアップ・コイルは回転しながら移動し、丁度板カム周囲の法線方向を向いた

ときにギャップ・センサーは最小値をとるので、ピーク・ディテクタを介してこの値をミニコンピュータに入力する（第 4.2.4 節）。もしも工作物が理想形状に仕上がっていれば、どの点を取ってもこの値は同じになる筈である。厳密に言えば、この方法では輪郭を測っているのではなく、理想輪郭からの偏差を見ているのである。この計測法では寸法精度は測れないが、熱変形に起因するような形状精度の測定には適している。なお、工作機械の位置決め誤差が介入すると測定に意味がなくなるので、1  $\mu$ m 読みの光電式リニア・エンコーダを用いて実際のピックアップ・コイルの位置をミニコンピュータに入力し、これに基づいて間隙を評価する。得られた結果は一

みすぎの誤差となって現れる。図 4.4.7 (a) で切削前半に切残しの誤差があるように見えるが、これはギャップ・センサーによる輪郭測定の際、各測定点での相対的な変化しか把握できないことによる。間隙の真値が把握できないのはこの方法の欠点であり、寸法精度も出せない。どこかに基準面を作って、接触子のようなものでこの絶対的な位置を知り、間隙の真値を把握できるように、この計測法には一考の余地が残されている。

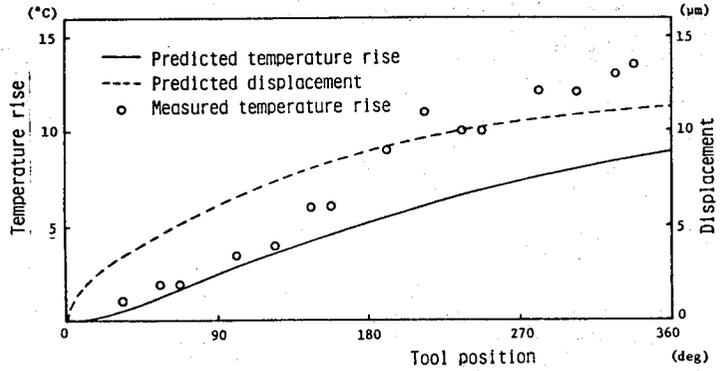


図 4.4.6 真円加工における温度変化

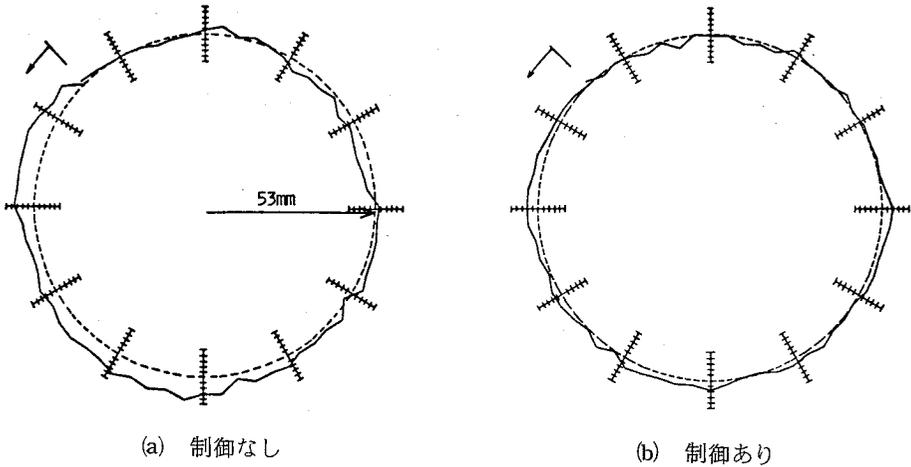


図 4.4.7 真円加工の輪郭測定結果 (1目盛 10 $\mu$ m)

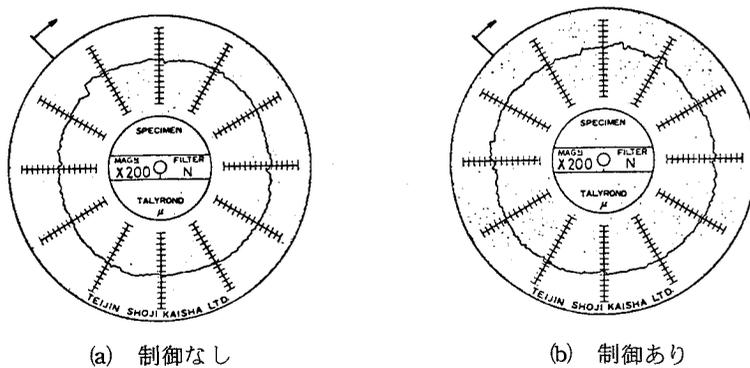


図 4.4.8 タリロンドによる形状測定 (1目盛 10 $\mu$ m)

真円加工の場合には、タリロンド真円度測定器によっても形状誤差を知ることができる。その測定結果を図 4.4.8 に示す。理想形状からの偏差はギャップ・センサーによるものの方が明瞭に現れている。また、バックラッシュについては、実験を一回行うごとにリニア・エンコーダを用いて補正しているので、その影響はほとんど現れていない。

(b) 楕円形カム加工の場合

平端直動従節の場合の板カムで、従節変位曲線は

$$f(\theta) = 10(1 - \cos 2\theta) \quad (4.5)$$

であり基円半径 44mm としている。計算による温度変化と熱変形量の予測値ならびに加工時の温度変化の測定値を図 4.4.9 に示す。やはり温度変化の測定値はやや大きめに観測されている。

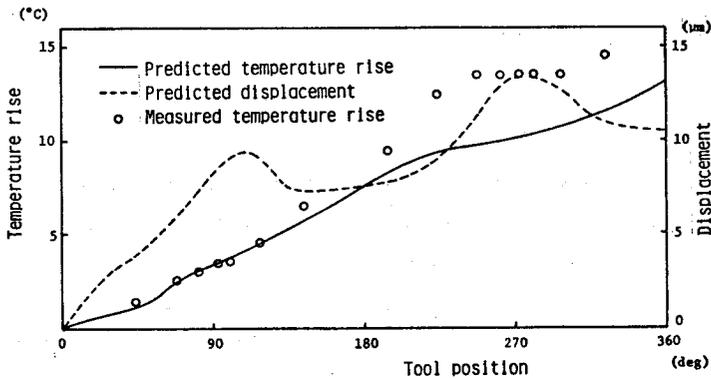


図 4.4.9 楕円形カム加工における温度変化

ギャップ・センサーによる輪郭形状測定結果を図 4.5.0 に示す。真円加工の場合と同様に計測法の欠陥が出ており制御なしの場合、切削前半部で切り残しがあるように見えるが、制御による形状精度の改善は明らかであろう。ただ、この例の場合、制御を実施した場合でもまだ 20 μm 程度の誤差が残っており、制御方式の改善が望まれる所である。

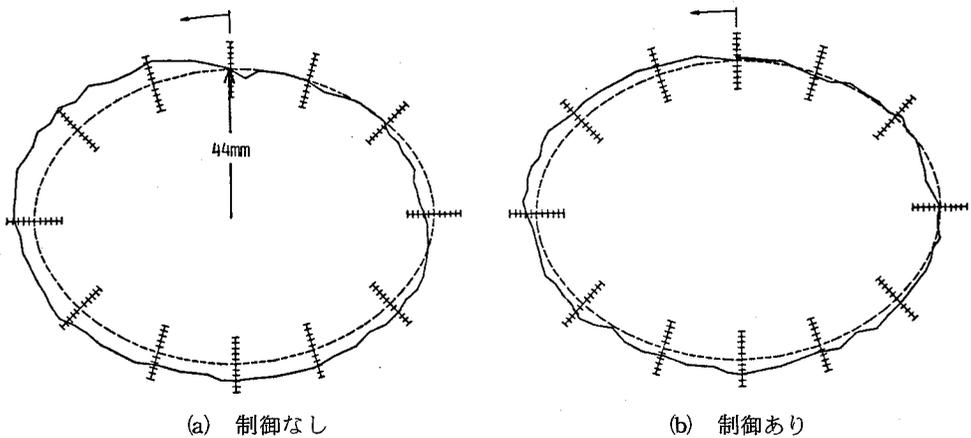


図 4.5.0 楕円形状カムの輪郭測定結果 (1目盛 10 μm)

(c) 凹部を持つカム加工の場合

円端直動従節の板カムの運動曲線は $\theta$ をカムの回転角とすると、次式で与えられる。

$$\begin{aligned}
 0 \leq \theta < \frac{2}{3}\pi \text{ のとき } & f(\theta) = 25 + \frac{5}{2} \left( 1 - \cos \frac{3}{2}\theta \right) \\
 \frac{2}{3}\pi \leq \theta < \frac{5}{6}\pi \text{ のとき } & f(\theta) = 15 \left\{ 1 + \cos 3 \left( \theta - \frac{2}{3}\pi \right) \right\} \\
 \frac{5}{6}\pi \leq \theta < \pi \text{ のとき } & f(\theta) = 15 \left\{ 1 - \sin 3 \left( \theta - \frac{5}{6}\pi \right) \right\} \\
 \pi \leq \theta < \frac{7}{6}\pi \text{ のとき } & f(\theta) = 15 \left\{ 1 - \cos 3 \left( \theta - \pi \right) \right\} \\
 \frac{7}{6}\pi \leq \theta < \frac{4}{3}\pi \text{ のとき } & f(\theta) = 15 \left\{ 1 + \sin 3 \left( \theta - \frac{7}{6}\pi \right) \right\} \\
 \frac{4}{3}\pi \leq \theta < 2\pi \text{ のとき } & f(\theta) = 25 + \frac{5}{2} \left\{ 1 + \cos \frac{3}{2} \left( \theta - \frac{4}{3}\pi \right) \right\}
 \end{aligned} \tag{4.6}$$

ただし、基円半径は49mm、従節ローラ半径は10mmとしている。計算による温度変化と熱変形量の予測値ならびに加工時の温度変化の測定値を図4.5.1に示す。ここにあげた3例とも実測温度は予測値より大きく観測されており、有限要素法による予測モデルにおいて、流入熱量を少ない目に見積っているか、空気への熱伝達率を大きく取りすぎているらいがあるが、適応制御加工ではこれらの点あまり問題にならずに済んでいる。これがまた適応制御加工の目的でもある。ギャップ・センサーによる輪郭形状測定結果を図4.5.2に示す。

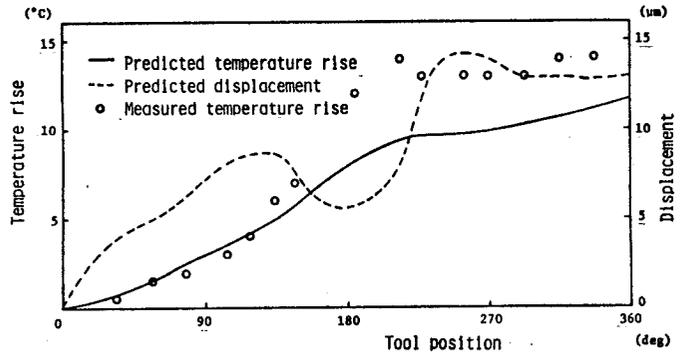


図 4.5.1 凹部を持つカム加工における温度変化

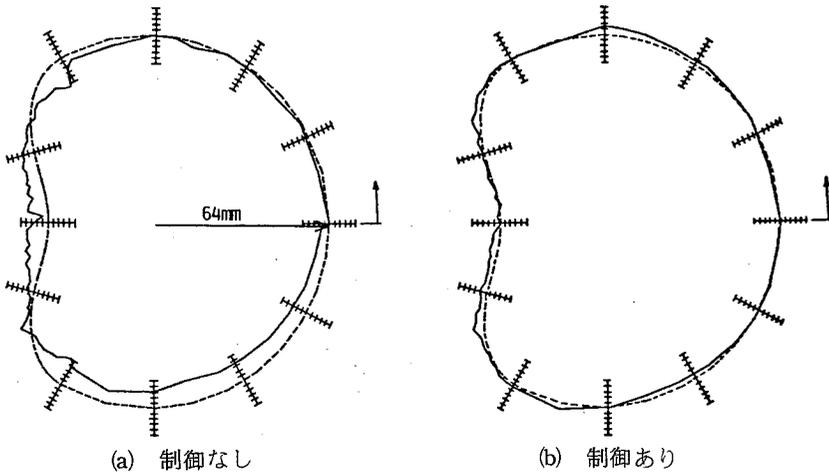


図 4.5.2 凹部を持つカムの輪郭測定結果 (1目盛10 $\mu$ m)

以上3種類の形状について述べたが、何れの場合も制御の効果が明らかであり本方式の有効性が確かめられた。

## 4.6 結 言

CNC装置を用いることの利点は、切削過程の途中で実時間で計算機の演算能力を利用できることである。本章では板カムをモデルに工作物熱変形に対する工具径路補正を実時間で実行する、適応制御実験について述べた。まず、各種の計測器をミニコンピュータに接続するためのアナログ入力インターフェース並びにデジタル入力インターフェースの設計法について述べ、続いてNCフライス盤の持つ位置決め精度に言及し、實際上その誤差は $\pm 10\mu\text{m}$ と見なしてよいことを確かめた。本実験システムの特徴は以下の通りである。

- (1) 単に計算だけによる熱変形推定に頼らず、加工時の温度測定も行って適応制御を実施している。制御を行わない場合 $40\mu\text{m}$ ないし $50\mu\text{m}$ の形状精度しか得られないが、制御を行うことによってこれが $10\mu\text{m}$ ないし $20\mu\text{m}$ に改善され、制御の効果が明らかになっている。
- (2) 加工対象が板カムであることを考慮し、各輪郭上の点における法線の偏角を制御用データの中に含ませることにより、工具径路補正を精度良く容易に実行している。
- (3) ミニコンピュータには大きな負荷がかからないようになっており、小型計算機の低価格化と共にCNCによる実時間制御の実用化を容易にしている。
- (4) 工作物である板カムの輪郭形状測定もNC装置とミニコンピュータにより自動化されており、回転するギャップ・センサーを利用して法線方向から見た形状誤差の評価を行っている。この方法はとくに板カムの輪郭測定に適している。
- (5) ギャップ・センサーによる輪郭形状測定を行う場合、NC装置の位置決め誤差の介入が避けられないので、リニア・エンコーダを用いてセンサーの位置を監視し、これに基づいて誤差評価を行っている。このことにより、加工と計測に同一機械を使うことの弊害を防止している。

フライス輪郭切削で、工具径路を実時間で修正して加工精度を向上させようとする試みはまだ少ない。しかし、CNC装置の普及とNCによる高精度加工の要求により、本章で述べた手法ならびにこれと工具径の実時間補正も含めた工具径路の実時間制御は次第に実用化されて行くと考えられる。

## 参 考 文 献

- 1) 人見勝人、中村信人：切削加工の最適化に関する研究、日本機械学会論文集、37巻、301号(1971)、pp.1795-1802。
- 2) 竹山秀彦、関口 博、高田孝次：旋削加工における最適化の手法、精密機械、Vol.36、No.5(1970)、pp.311-317。
- 3) 谷 和男、水野 豊、竹中規雄：旋削の最適化および適応制御、日本機械学会論文集、44巻、377号(1978)、pp.366-373。
- 4) 貴島 徹、稲崎一郎、米津 栄：旋削加工における最適化適応制御、日本機械学会論文集、44巻、385号(1978)、pp.3251-3259。

- 5) 佐田登志夫、塩田泰仁：旋削加工の適応制御、精密機械、Vol.46、No.7(1980)、pp.805-810.
- 6) 竹内芳美、坂本正史、佐田登志夫：NC旋盤の工具熱変形補正による加工精度の向上、精密機械、Vol.46、No.11(1980)、pp.1392-1397.
- 7) D.A.Milner：Adaptive Control of Feedrate in the Milling Process, Int. J.Mach.Tool Des. Res., Vol.14(1974)、pp.187-197.
- 8) M.Weck, E.Verhaag and M.Gather：Adaptive Control for Face-Milling Operations with Strategies for Avoiding Chatter-Vibrations and for Automatic Cut Distribution, Annals of the C.I.R.P., Vol.24(1975)、pp.405-409.
- 9) K.Nakazawa：Improvement Adaptive Control of Milling Machine by Non-Contact Cutting Force Detector, Proceedings of the 16th M.T.D.R. Conference (1975)、pp.109-116.
- 10) R.Bedini and P.C.Pinotti：A Hardwired Logic for Adaptive Control of a Milling Machine, Int. J. of M.T.D.R., Vol.16(1976)、pp.193-207.
- 11) C.Boer, M.C.De Malherbe and R.Venter：Adaptive Control Optimization for a Numerically Controlled Milling Process, Proceedings of the 18th International M.T.D.R. Conference (1977)、pp.665-671.
- 12) D.H.Maltby and H.R.Marten：Software for a Computer Numerical Control System Designed for Adaptive Control Research, Proceedings of the 19th International M.T.D.R. Conference(1978)、pp.19-26.
- 13) J.Peklenik：Geometrical Adaptive Control of Manufacturing Systems, Annals of the C.I.R.P., Vol.18(1970)、pp.265-272.
- 14) I.Wasiukiewicz：The Improvement in Accuracy of High Precision Machine Tools by Means of Adaptive Control, Proceedings of the 15th International M.T.D.R. Conference (1974)、pp.79-87.
- 15) 本多庸悟、竹山秀彦、安部哲夫：計算機・適応制御旋削加工システムの試み(第2報) - 切削力および寸法精度に関する適応制御 -、精密機械、Vol.40、No.3(1974)、pp.244-249.
- 16) 山県敬一、牧之内三郎、難波義治、藤田正道、浦上豊蔵：板カム輪郭フライス切削における工作物熱変形の適応制御、精密機械、Vol.47、No.6(1981)、pp.675-680.
- 17) Y.Namba and H.Tsuwa：Geometrical Adaptive Control in Profile Milling by CNC System, Proceedings of the 17th International M.T.D.R. Conference (1976)、pp.67-74.
- 18) NECミニコンピュータ NEAC-M4 入出力システム、日本電気.
- 19) NECミニコンピュータ NEAC-M4 ハードウェア説明書、ML-4890 汎用入出力インターフェース、日本電気.

- 20) 温度センサー、温度コントローラ用ICの応用回路、集積回路技術資料 IEA-523, 日本電気.
- 21) JIS C 1602 熱電対, 日本規格協会(1974).
- 22) 松永輝雄: 光電式リニアエンコーダとその精度, Nikon Technical Journal, No.1(1973), pp.26-38.
- 23) 小林淳人, 岡本清和, 山浦弘紀: 数値制御用駆動部DSサーボシステム, NEC日本電気技報, No.99(1970).
- 24) 井川直哉, 水本 洋: しゅう動体の停止挙動 - 工作機械の位置決め精度の基礎研究(第1報) -, 精密機械, Vol.41, No.8(1975), pp.771-777.
- 25) 水本 洋, 松原十三生, 井川直哉: すべり案内面の動摩擦力変動と位置決め精度 - 工作機械の位置決め精度の基礎研究(第2報) -, 精密機械, Vol.44, No.6(1978), pp.653-659.
- 26) 沢田 潔, 木村浩哉: NC機の位置決め精度の測定について, 日本機械学会誌, 75巻, 637号(1972), pp.246-253.
- 27) 第1章参考文献3), 5), 6).
- 28) 牧之内三郎, 難波義治, 山県敬一, 吉池 毅: 板カム加工の工作物熱変形のオンライン制御(第2報) - 有限要素法による補正法の解析と実験 -, 昭和51年度精機学会春季大会学術講演会前刷(1976) pp.31-32.
- 29) 山県敬一, 牧之内三郎: DNCシステムによる複雑な形状部品の輪郭切削について, 精機学会昭和40年度関西地方定期学術講演会前刷(1974), pp.137-140.

## 総 括

集積回路技術の発達に伴い、計算機ハードウェアの進歩には著しいものがあり、マイクロコンピュータやミニコンピュータは低価格化が進み、大型計算機もタイムシェアリング・システムの普及で手軽に使えるようになって来ている。それだけに用途も広がる一方で、その利用技術やプログラムの開発に多くの労力が要求される。本研究では加工の省力化を目的として、大型計算機のタイムシェアリング・システムやミニコンピュータを用いているが、この場合も例外ではない。数値制御による切削加工を応用対象として、コンピュータをいかに効率よく利用するかが重要な課題である。以上の理由から、本論文は第1編と第2編に分けられており、第1編では加工を応用対象としながらも、計算機のソフトウェアの組立て方に重点が置かれている。第1編はさらに大きく二つに分けられ、第1章と第2章では応用プログラムの作成法を論じ、第3章と第4章ではミニコンピュータの基本ソフトウェアの作成法を論述している。第2編は加工の適応制御という応用に重点を置いて述べられている。

本研究の実験に用いたTSSオンラインCNC(DNC)システムは、大型計算機とミニコンピュータと数値制御装置が階層構造をもって接続され、いろいろな実験に効果的に利用できるように工夫されている。単にデータ伝送が自動的に行われるだけでなく、適宜必要に応じて大型計算機とミニコンピュータの処理能力が活用でき、経済的に大きな処理能力が得られる。特にTSSのファイル・システムはそのエディタと共に加工情報を保持するのに便利である。またミニコンピュータは本来制御用計算機として開発されたものであり、リアルタイム処理に適しているので適応制御などに効果的に利用できる。以下に本論文を通観して主要な論点を要約する。

- (1) 第1編第1章では取り扱う部品の形状に制限を加えた場合、簡便な図形処理により容易にNCテープが作成できることを明らかにしている。図形要素は直線と円に限られているが、これらで定義できる部品も数多くあり、ポケット・カッティングも可能にしているので十分用途は広いと考えられる。形状に制約を加えることを逆に利用すると、汎用プロセッサとしての特性は失われるが、限定された応用領域の上で専用プロセッサとして効率の良いものを作成することができる。ここで述べた応用例の場合、安価な蓄積型ブラウン管を効果的に利用し、管面上で図形や輪郭を直接定義することができ、例えばAPTのように言語を用いて手順を書き下すことから開放される。このような点は計算機の応用プログラム作成の本質的な問題である。重要なことはどのような応用領域を設定するかという事であろう。実際これまでに、旋削加工に限ったNCテープ作成のプロセッサが開発されているのもこの種の例である。

グラフィック・ディスプレイ端末(蓄積型ブラウン管とキーボード)を用いて作成されたNCテープのイメージは、タイムシェアリング・システムのファイルに保存される。このファイルはオンラインCNCシステムと共用になっており、紙テープを用いることなく直接ファイル内のNC指令に従って加工を実施することができる。また、さらに重要なことは、加工条件にかかわる指令を機械室の端末を用いて挿入できることである。これにより設計時には定められない加工条件を、実際に加工を行うときに決定したり修正したりすることが可能である。設計と加工が一つのシステムで統合されていることに重要な意味があるのである。

- (2) 第1編第2章では板カムの設計問題とNCテープの作成について述べている。NCテープ作成の面から見ると、これも一つの限られた応用領域で、設計の過程から直接輪郭に関するNC指令が得ら

れるという際立った特徴を持つ。したがって、この章では設計問題を中心に述べている。板カムの輪郭は、カムの機構（直動、揺動等）にかかわる計算式に、従節変位曲線（のぼり行程、くだり行程）の関数を組み合わせて決定される。ここでは FORTRAN の副プログラムの概念が有効に使える。すなわち、カム曲線の関数に名前をつけてライブラリ化しておく、その名前の引用だけで機構の計算プログラムに結合できる。解析的にはカムの輪郭はパラメトリック曲線で表現されるので、具体的に輪郭を得るには、パラメータを微量ずつ変えた点列を求める。これを許容公差内で円弧補間すると円弧列で輪郭の近似が得られる。NC 装置に円弧補間機能があれば、円弧列はそのまま加工データとなる。設計と加工データの生成が密着した形式で処理される特別な応用領域である。

- (3) 第1編第3章では小型計算機の実時間処理向けに、新しいデータ処理形態を提案している。実時間処理では複数個のファンクション・プログラムの擬似並行処理とそれらの間の同期操作が不可欠である。この機能を簡潔に実現するため、新しい概念として t-unit のストリームを導入し、タスク管理やイベント管理の機能を t-unit の操作だけで統一的に遂行している。ユーザ側は t-unit の扱い方如何で平易な形式の逐次処理のプログラムから、バッファ・キューを含む複雑なプログラムまで、その作成法に自由度が与えられ、モニター側は煩雑な同期操作の一部をユーザ側にまかせることによって、それ自身は簡潔になる。

各ファンクション・プログラムは独立に作成され、相互の間の結合は t-unit を用いて動的に行われる。したがって、システム・ジェネレーションが不要であり、プログラムの差し換えを容易にしている。割込み処理を含むプログラムについても、一般のファンクション・プログラムと同様に差し換えが自由であり、広い応用範囲に適用できるように考慮されている。

- (4) 第1編第4章では TSS オンライン CNC (DNC) システムの組み立てについて述べている。タイムシェアリング・システムの端局にミニコンピュータを用いる場合、通信回線の伝送制御手順に対す制御プログラムが不可欠で、これを中心に各種デバイスの制御プログラムが準備される。デバイス制御プログラムでは割込み処理とミニコンピュータ (NEAC-M4) の入出力命令を駆使し、直接ハードウェアにかかわる機能を主プログラム側で使う必要のないようにしておく。主プログラムからこれら制御プログラムを呼び出して利用するのは、第3章で述べた t-unit の操作で行う。

数値制御装置も一種のデバイスと見なして、ミニコンピュータからのデータ伝送を可能にする。標準的なデバイスの他に、アナログ入力インターフェースやリニア・エンコーダからのデータの読み込みなど、適応制御実施のための機器も接続されており、これらを使いこなすためのプログラムも用意されている。また、デバイス制御の他に高精度の算術演算プログラムも必要であり 10 進約 13 けた (2 進仮数部 46 ビット) の倍精度演算プログラムも用意されている。

- (5) 第2編第1章ではフライス輪郭切削における工作物への流入熱量の推定を試みている。この値は加工条件に依存する。円筒フライスの切削過程をそのまま解析するのは極めて困難であるから、この切削過程を近似モデルでおきかえて解析する。まず多刃断続切削を均一化して平均切削抵抗を求める。これを熱の問題としてみると、実際は断続的に流入して来る熱量を連続移動熱源でおきかえたことになるが、工作物全体の温度上昇を考える場合には、このおきかえは支障を来たさないと思われる。次に発熱量とその工作物への流入割合の解析は 2 次元切削モデルで行う。ここで、フライス切削では切削厚さが均一でないことが問題になるが、これも近似モデルとしてフライス切削における最大切削厚さの  $\frac{1}{2}$  の値を採用して均一化する。最後に工作物への流入熱量の割合は、M. C. Shaw

の解析法に従って求める。以上の手順により、切削条件をパラメータとする流入熱量の予測式が得られる。実験値とのつき合わせも行っているが、近似モデルとして単純化していることや、雰囲気、工具の刃先状態など不確定な要因が介在するため、理論値と実験値が完全に合致しているとはいえない。しかしながら、適応制御のための予測値として、理論式を用いるのは有効であると思われる。

- (6) 第2編第2章と第3章では、板カム輪郭加工時の工作物の熱的挙動を調べるため、有限要素法によるシミュレータについて述べている。輪郭切削における熱の問題のモデルは、熱源が指定された送り速度で輪郭外周を移動する移動連続熱源問題である。工作物が板状であることを考慮して厚さ方向については有限要素への分割を行わず、応力問題は平面応力問題として扱い、計算の簡略化を図っている。ただし、温度分布の計算では四面体要素を用いて上下面からの熱の放出を考慮してある。非常常問題であるのでWilson-Cloughの方法を適用する。シミュレーションの結果は、後の適応制御のための予測値として用いられるので、熱源(工具)が有限要素節点を通過するときのその節点の変位と中心点の温度をファイルに保存して制御用データとする。これを用いることによって工作物熱変形に対する適応制御が容易になる。
- (7) 第2編第4章では、実時間で工具径路を補正する適応制御について述べている。適応制御の方式を述べるに先立ち、これに必要な各種の計測器をミニコンピュータに接続するためのインターフェースの設計について述べている。また計測法自体についても、第1編で述べた概要を補い、より詳しく具体的に記述している。また、NCフライス盤が本来持っている位置決め精度を吟味し、実際的な位置決め誤差を $\pm 10\mu\text{m}$ とみなしてよいことを確かめている。以上の基礎の上に適応制御方式を論じている。

有限要素法によるシミュレータでは、正確な工作物への流入熱量は分っておらず、仮の予測値を用いて温度分布と節点変位が求められている。したがって、実際の加工にあたっては、工作物の一点(板カムモデルの回転中心)で温度を監視し、シミュレータによる予測温度とつき合わせて流入熱量補正のスケールリング・ファクタを定める。これに基づいて節点変位の補正を行い、工作物熱変形を考慮した工具径路を実時間で決定する。この適応制御機能はミニコンピュータとNC装置の結合体(CNC)によって実現される。実際に加工実験を行い、適応制御の実施によって工作物の熱変形に起因する形状精度の劣化を防止できることを明らかにしている。

また、本実験では加工後の工作物を工作機械に取りつけたまま、輪郭形状精度をリニア・エンコーダの監視のもとでギャップ・センサーを用いて自動測定を行う新しい試みもなされている。今後オンラインCNCの実用化に伴って、設計、加工、計測の一貫した過程の自動化が重要になって来ると考えられる。

## 謝 辞

本研究の遂行にあたり、終始御指導と御鞭撻を賜りました大阪大学牧之内三郎教授ならびに津和秀夫教授、井川直哉教授、手塚慶一教授に深甚の謝意を表すとともに、有益な御教示を賜りました大阪大学川辺秀昭教授、山田朝治教授ならびに長谷川嘉雄教授に厚く感謝の意を表します。またとくに、計算機の利用技術については京都大学萩原宏教授の御教示に、そして機械加工については大阪大学難波義治助教授の御助言に負う所が大であり、牧之内研究室の方々からは多くの御協力を頂きました。記して厚く御礼申し上げます。