

KAKEN
16560225

研 究 成 果 報 告 書

技能者による生産計画に対する工夫を支援する

高機能 CNC 装置の開発

(16560225)

平成16年度～平成17年度

科学研究費補助金（基盤研究C）研究成果報告書

平成18年5月

研究代表者 樋野 励

(名古屋大学・工学研究科・講師)

はしがき

本研究は自動化の進歩によりさらなる分業化が進むことにより、作業者による創意工夫が排斥される状況を防ぐことを目指したものである。具体的には分散スケジューリング手法を拡張することにより、個々の作業の計画の変更が全体に与える影響を把握するための手法を提案している。また自由曲面の加工に対し、CAMによる前処理を省略するための手法について基礎的な見当を行っている。

研究組織

- 研究代表者 : 樋野 励 (名古屋大学機械理工学専攻機械科学分野講師)
研究分担者 : 清水良明 (豊橋技術科学大学工学部教授)
研究分担者 : 柳 在圭 (豊橋技術科学大学工学部助手)

交付決定額 (配分額)

(金額単位: 千円)

	直接経費	間接経費	合計
平成16年度	1,600	0	1,600
平成17年度	1,100	0	1,100
平成 年度			
平成 年度			
平成 年度			
総計	2,700	0	2,700

研究発表

(1) 学会誌等

- 樋野励, 再帰的伝播法による分散型ジョブショップスケジューリング
(第4報: 協調作業を考慮に入れた情報交換手順の提案), 日本機械学会論文集(C編), vol.70, No.699, pp.3300-3307, 2004
- 樋野励, 楠見哲也, 柳在圭, 清水良明, バッファを考慮に入れたジョブショップスケジューリング, 日本機械学会論文集(C編), Vol.71, No.702, pp.685-692, 2005
- Jae Kyu YOO, Yoshiaki SHIMIZU, Rei HINO, A Sequencing Problem for Mixed-Model Assembly Line with the Aid of Relief-Man, JSME International Journal Series C, Vol.48, No.1, pp.15-20, 2005

(2) 口頭発表

- 1) 柳在圭, 許斐和彦, 樋野励, 清水良明, 補助人員を考慮した混合品種組立ラインの製品投入順序付け, 日本機械学会 生産システム部門講演会 2004, pp. 27-28, 2004
- 2) 樋野励, 楠見哲也, 柳在圭, 清水良明, バッファ容量を考慮したジョブショップスケジューリング, journal = "日本機械学会 生産システム部門講演会 2004, pp28-29, 2004
- 3) 樋野励, スケジューリングのための生産システムのモデル化(第2報 保管時間の導入), 精密工学会秋季大会学術講演会講演論文集, pp317-318, 2004
- 4) 樋野励, バッファを考慮に入れたスケジューリングに関する研究, 精密工学会春季大会学術講演会講演論文集, pp. 325-326, 2005
- 5) 樋野励, 鈴木隆志, 清水良明, バッファ容量を考慮に入れたジョブショップスケジューリング(第2報 混合整数計画問題によるモデル化), 日本機械学会 生産システム部門講演会 2005, pp. 19-20, 2005
- 6) 樋野励, 立山健太, 社本英二, 搬送車の衝突のないスケジューリングに関する一解法, 日本機械学会東海学生会第37回学生員卒業研究発表講演会講演前刷集, pp. 210-211, 2006
- 7) 樋野励, 一般リソースモデルに基づいたジョブショップスケジューリング問題, 精密工学会春季大会学術講演会講演論文集, pp. 71-72, 2005

(3) 出版物(著者名、署名、出版社、年月日)

なし

研究成果による工業所有権の出願・取得状況

なし

目次

第1章 課題の背景と目的	3
1.1 研究の到達目標	3
1.2 研究の独創性および意義	4
1.3 研究計画	4
第2章 作業者の視点に立ったスケジューリング	7
2.1 目的	7
2.2 分散スケジューリング	8
2.2.1 用語の説明	8
2.2.2 再帰的伝播法	8
2.2.3 情報交換手順	9
2.3 1機械問題	11
2.3.1 前提条件	11
2.3.2 定式化	12
2.3.3 1機械問題とジョブショップ問題	12
2.3.4 Schrage 順序	14
2.4 再帰的伝播法の拡張	17
2.4.1 情報交換手順	18
2.4.2 情報交換の例	21
2.5 対象とするスケジューリング問題	22
2.6 従来スケジューリング	23
2.7 Schrage 順序による分散スケジューリング	24
2.7.1 スケジューリング手順	24
2.8 メイクスパンと情報交換回数の評価	26
2.9 ベンチマーク問題における評価	29
2.10 まとめ	31
第3章 ユーザーインターフェイスの整備	33
3.1 目的	33
3.2 スケジューリング問題の一般化	34
3.3 工程に着目した分析	34

		1
3.3.1	合流	35
3.3.2	分岐	36
3.3.3	協調	36
3.4	作業に着目した分析	37
3.5	スケジューリング問題の記述	39
3.5.1	XML とその特徴	39
3.5.2	地図情報の記述	39
3.5.3	設備構成の記述	42
3.5.4	工程計画の記述	45
3.6	まとめ	51
第4章	直接オフセット法による工具経路生成	53
4.1	目的	53
4.2	工具指令点と加工点の関係	54
4.3	直接オフセット法の提案	56
4.4	数値計算例	59
4.5	まとめ	62
第5章	付録	63
5.1	NURBS 曲面	63
5.2	NURBS 面に対するオフセット点の高効率な計算法	64
5.3	工具刃先のモデル	66
5.3.1	基本モデル	66
5.3.2	ねじれ角を考慮したモデル	67
5.4	制御軸の変化量	70
5.4.1	直進駆動軸	70
5.4.2	回転駆動軸	73
5.5	座標変換	76
5.5.1	二次元座標変換	76
5.5.2	三次元座標変換	78
5.5.3	製品形状情報および工具姿勢の座標変換	83

第1章 課題の背景と目的

1.1 研究の到達目標

人件費の高騰により、生産の拠点が諸外国に移る中、商品企画能力や技術力については、国外への流出を何としてでも防ぎ、今後のさらなる発展のために、知恵と資本を投入すべき課題と考えられる。その際、現在の日本の技術力は、実際に加工に携わる技能者に支えられていることを忘れてはならない。たしかに、意匠性を重視した複雑な形状の加工の必然性や、高機能化を実現するための複雑な加工計画の必要性は、それらの活動を支援するためのコンピュータによる情報処理技術の発展につながっている。しかしその反面、技能者から創意工夫の機会を奪ってしまい、彼らに単なるオペレータとしての役割しか与えないという結果を生み出している。

すなわち、形状が商品の価値を決める上で重要な要因のひとつにあげられ、自由曲面の加工は不可欠なものになっている。なかでも、自由曲面の金型の製作は、商品の意匠性を決めるだけでなく、価格を決める上でも極めて重要な工程である。たとえば、直線や円弧上のエンドミル加工を取り上げれば、フライス盤に取り付けられたCNC装置には、工具径補正機能というものがある。この機能を利用して、技能者は工具摩耗による削り残しの補正をその場で修正することができる。すなわち、この機能により、作業者は、図面情報をその場で直接工作機械CNC装置への入力値として用いることができるだけでなく、実際に使用する工具の径とは異なった値を用いることにより、同じNCプログラムを用いて削り残し量の調整を容易に行なうことができる。

ところが、形態電話のように加工形状が自由曲面を含んだとたん、工具経路はCAMにより加工前に準備する必要があるため、技能者が加工現場でできることはほとんどない。つまり、自由曲面を含む場合には、製品形状ではなく製品形状のオフセット位置の情報を、工具の運動指令値としてCNC装置へ入力しなければならない。これは、直線や円弧の場合とは違い、自由曲面のオフセット形状が、製品形状と相似ではないためである。そのため、CAM (Computer Aided Design) システムによる加工前処理が不可欠であり、高価なCAMシステムの準備や操作方法の習熟が要求される。また、加工条件が代わるたびに入力情報を求め直す必要があるため、生産性の低下の原因のひとつになっている。

このように加工形状に関わらず作業者の意図する加工を自在に行なうことが望まれていることを背景としてとらえ、工作機械の知能化を行なうことにより解決する。

本研究の目的は、技能者が作業現場において、設計部門や工程管理部門が立案した計画

を修正し、また工夫を加えることを支援することができる工作機械用 CNC 装置の開発にある。開発した CNC 装置を用いて実際に工作機械を制御することは、安全面など多くの問題があるため、最終的には、工作機械メーカーの参画が不可欠だと考えている。そのため、本研究の申請期間内には、工作機械メーカーが参画の判断をするために必要な、基本アーキテクチャの構築と実現可能性を裏付ける技術情報の収集を行う。

1.2 研究の独創性および意義

本研究は、申請者が独自に開発した2つの技術の上に成り立つ。ひとつは分散スケジューリングの手法であり、もうひとつは工具経路生成手法である。いずれも、オペレータの指示を必要としない自律的な工作機械を開発するために考案したものである。これらの技術を、技能者の創意工夫を支援するために利用する。

スケジューリング手法を用いることで、計画に変更を加えた場合、影響を与えるすべての他の設備のスケジュールとの整合性を保証し、同時に、工場全体の生産計画に与える影響の大きさを知ることができる。CNC 装置にこの機能を持たせ、作業者が利用することにより、計画を変えた場合に、それがどの程度の影響を与えるかを知ることができる。言い換えれば、全体に与える効果を知ることができる。このことは、計画を変更することに対する不安を作業者から取り除くだけでなく、良い変更を加えた場合には、得られる効果を定量的に知ることができる。

工具経路生成機能は、工具の制御点を生成する手法である。例えば、工作機械の CNC 装置には、工具径補正という機能がある。この機能を用いて、CNC 装置に入力する工具の直径の値を調整することで、仕上げ代を残した加工や、工具摩耗による削り残しを修正することができる。ところが、この機能は直線や円弧の加工にしか用いることはできない。つまり、自由曲面を含む場合には、加工前に CAM システムにより作成した制御命令を、作業者が変更することはできない。申請者が開発した直接オフセット法は、基本的な制御点の生成方法に改善を加えることで、自由曲面に対してもその場で計画の修正を可能にしている。そのため、製品の形状が複雑になっても技能者の現場での工夫を生かすことができる。

1.3 研究計画

1) 基本アーキテクチャの設計

開発する CNC 装置の基本アーキテクチャを設計する。想定する基本的な構造は図 1.1 に示す通りである。作業者による入力、専門的なスケジュールや CAM の知識を必要とせず、作業者の技能を活用できるものでなくてはならない。すなわち、本研究で想定して

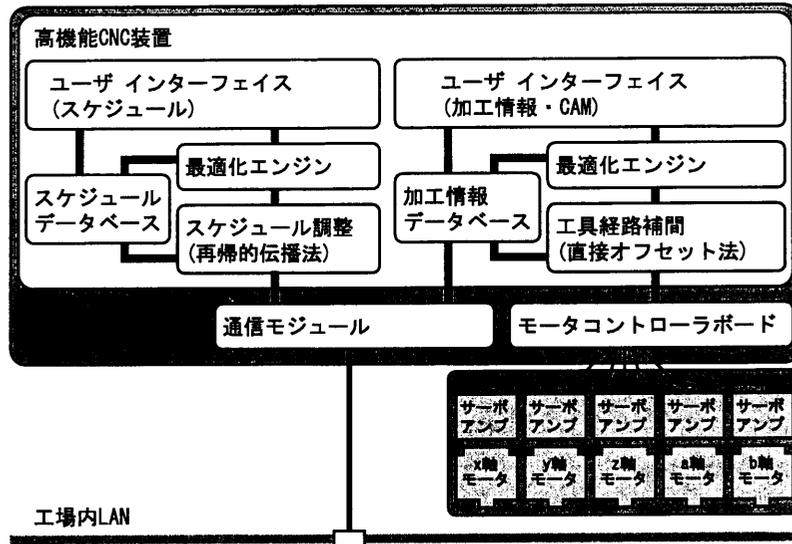


図 1.1: 開発する CNC コントローラの基本構造

いる CNC の機能は、技能者の意志決定を支援するものであり、CNC 装置が勝手に加工計画を立案するものではない。例えば本研究では、工具径補正や工具長補正など、従来から行われている直線や円弧などの単純形状の加工で用いられる一般的な加工のためのテクニックをそのまま複雑な自由曲面の加工に対しても用いることを考える。

2) スケジュール変更時の代替案の最適化

分散スケジューリングのために開発した再帰的伝播法は、作業処理時刻を変更した際に、影響を与える機械(あるいはその機械を担当する作業者に伝え、スケジュールの整合性を保証するとともに、変更をした作業者に、その変更が全体に与える影響を定量的に評価した結果を伝えることができる。

作業者の考えた変更が全体に与える影響を把握するだけでも、作業者にとっては十分に有益である。ここでさらに、作業者の変更に合わせてスケジュールの改善案を作業者に提示することで、その効果はより大きくなる。まず作業者の作業変更を評価し、変更に合わせてスケジュールの最適化を行う手法を開発する。

さらに、作業者が、現在の作業に対する変更を指示するためのユーザインターフェイスの開発を行う。ここでは、作業内容の変更による作業終了予定時刻の変更を指示することで、他の作業者との調整と全体に与える影響を作業者に通知する仕組みを作る。

また、作業変更の影響を最小限に押えるために、スケジュールの改善案を提示し、作業者の判断をうながすための仕組みを作る。

3) 加工条件の最適化アルゴリズムの開発

加工に対する技能者の工夫は、加工順序や隅部での工具の扱いなど、経験に基づいた加

工方法の判断にあり、それ以外の処理は、CNC装置によって施されるべきと考えている。特に自由曲面の加工においては、工具の姿勢制御に対しては複雑な処理が必要となる。これらの処理は、CNC装置による最適化を行うことを考える。

本研究は、CAMシステムが持つ機能をすべてCNC装置に持たせるのではなく、加工に関する専門的な知識を要する処理のみを、技能者が行うことができるようにCAMシステムと役割の分担を見直す。すなわち、被削材との干渉回避などの幾何学的な制約に対する処理は、従来通り加工前のCAMシステムによる処理を想定する。すなわち、開発するCNC装置で行う処理は、システムが設定する制約条件と、技能者が与える条件のもとで、工具姿勢などの制御条件を最適化する問題と位置付ける。

本研究で用いる直接オフセット法は、高次の補間式であるNURBS形式により表された製品形状の情報を直接参照することで、工具指令値であるオフセット点の座標と、加工点での工具姿勢情報を生成する方法である。独自のNURBS展開アルゴリズムを考案することにより、従来手法の1/40の演算時間を実現している。また、直接製品情報を取り扱うために、直線形状や円弧形状の加工と同様に、自由曲面に対しても、工具径補正および工具長補正を行なうことができる。

技能者が持つ知識に基づいて、加工情報を修正するためのユーザインターフェイスを開発する。ここでは、工具径補正機能および工具長補正機能を用いて、加工順序や削り代を調整するためのユーザインターフェイスなどを整備する。

これらの機能は、作業者がコンピュータに固有の知識を持たなくても操作できるように、従来のCNCの操作と類似のものになるように工夫する。

第2章 作業者の視点に立ったスケジューリング

2.1 目的

円滑な生産活動を行う上で、スケジューリング技術は不可欠である。一般にスケジューリングとは、機械に割り当てられた作業に対して、その処理順序を決定する行為を意味する。スケジューリングは、数学的にも最適な解を求めることが難しいため、これまでも多くの研究が行われている。しかし、単に適切なスケジュールを得ることだけが重要なわけではない。たとえば、スケジュールを立案することで、生産システムを構成する要素間の関係が明確にわかり、生産する製品が、いつ処理が始まり、いつ完成するか、また、その進捗状況を知ることが可能になる。

本研究では、スケジュールの立案を考える際、工場全体を把握することができる専門の計画者ではなく、個々の機械を実際に操作する作業者に視点を置いた検討を行う。すなわち、作業者がそれぞれの視点からスケジューリングを行うことが可能になれば、個々の作業についてやむを得ない理由により遅延が生じた際に、生産システム全体に与える影響を把握することが可能になれば、単に精神的な負担が軽減されるだけでなく、適切な対応を取ることが期待できる。

このように、個々の作業者の視点からスケジューリングを可能するためには、いわゆる分散型のスケジューリング技術の開発を行う必要がある。分散スケジューリングに関する研究はこれまでに様々な研究が行われている。例えば、企業、工場、ショップなどの階層構造を有する生産システムにおけるスケジューリング問題を取り扱う階層分散型スケジューリング [2]、また、オークション方式により機械が自身のスケジュールを決定していくオークション型スケジューリング [3]、また、製品の製造過程での加工対象物の受け渡し関係に着目した情報交換方法である再帰的伝播法を利用した再帰的伝播法によるジョブショップ型スケジューリングなどがあげられる [1]。

本研究では、この再帰的伝播法を利用し、スケジューリング問題のひとつである1機械問題に焦点を置き、作業者の支援のための手法を検討する。ここでの検討は、作業者の意志決定を速やかに行うために、分散管理下に置かれた機械が、1機械問題を求解するために必要な情報を得るための新しい再帰的伝播法を提案する。また1機械問題においての上界と下界の推定の過程で求められるクリティカルジョブに着目した分散スケジューリング手法の妥当性を評価する。

2.2 分散スケジューリング

2.2.1 用語の説明

本論文では、スケジュールの基本要素である仕事 (job) を、製品 (product) の立場からとらえた場合には工程 (process) と呼び、機械 (machine) の立場からとらえた場合には作業 (operation) と呼ぶ。使用する記号の意味は以下の通りである。

$j_{\eta,i}^{\zeta,v}$:	機械 ζ が v 番目に処理を行う作業。また、製品 η を製造するための i 番目の工程。
s :	仕事 j の処理開始時刻
f :	仕事 j の処理終了時刻
p :	仕事 j の処理時間
r :	仕事 j の処理開始可能時刻
q :	仕事 j の残り処理時間
$fmax$:	1 機械問題におけるメイクスパン (上界)
$h(Q)$	1 機械問題における下界
$\hat{s}, \hat{q}, \hat{r}$:	変更後の諸時刻

分散スケジューリングとは、対象となる生産システムを構成するシステム要素、すなわちここでは行為の主体となる作業者が全体を統括する調停者を必要とせず、互いに協調しあいながら独自にスケジュールを立案する行為である。このとき、各作業者は、すべての情報を知ることにはできず、限られた情報のみを利用してスケジュールを立案する。

このような分散スケジューリングの場合、一人の作業者がすでに決められたスケジュールを変更する場合、他の機械を操作する作業者と協調し、作業の処理時刻の整合性を保証しなければならない。そこで、本研究では、作業の処理時刻の整合性を保つために、再帰的伝播法と呼ばれる情報交換法を利用し、分散スケジューリングの解法を提案をする。

2.2.2 再帰的伝播法

スケジューリングの際には、前提条件として機械の使用順序が与えられているとする。この点に着目し、各作業者はすべての情報を知ることにはできないものの、最低限の情報として、製品を製造するためにどの作業員から加工対象物を受け取り、どの作業員に処理後の対象物を渡すかの情報を知っていることを仮定する。再帰的伝播法はこの断片的な情報だけを利用した分散スケジューリングのための手法である。

再帰的伝播法を用いて作業員が行う行為は、作業処理時刻の”変更”，その変更の”通知”，および変更に対する”報告”の3つである。変更はスケジュールの改善と、他の作業の処理時刻の変更に対してスケジュールの整合性を保つために行われる。また通知は、変更を行った場合に、他の作業員の作業に影響を与える可能性がある場合に行われる。た

だし通知は、着目する作業の次工程を処理する作業者に対してのみ行われる。報告は必要な処理がすべて終了した場合に、通知を行ってきた作業者に対して行う。このとき報告を行う作業者が、すでに他の作業者から報告を受けている場合には、その結果も合わせて報告を行う。

2.2.3 情報交換手順

生産システム内の機械は、常に他の作業者からのメッセージが送られてくるのを受け付けることが可能であると仮定する。通知を受け取ってから報告を行うまでの、再帰的伝播法による情報交換の手順を以下に示す。

Step 1: 他の作業者からメッセージが送られてくるのを待つ。メッセージを受け取った作業者を ζ 、送信元の作業者を ξ とする。受け取ったメッセージが変更に関する通知の場合、通知のメッセージに続いて、作業者 ζ は、自分が v 番目に行う作業 $j_{\eta,i}^{\zeta,v}$ の変更後の開始可能時刻 $\hat{r}_{\eta,i}^{\zeta,v}$ に関する情報を受け取る。受け取った時刻と、その作業の直前に処理を行う作業 $j_{\nu,j}^{\zeta,v-1}$ の終了時刻 $f_{\nu,j}^{\zeta,v-1}$ のうち、遅い方の時刻をその作業の処理開始時刻とする。また、システム全体に生じる影響の評価値を記憶するための変数 Δt の確保と初期化を行い、処理をStep 2に移す。 Δt は、通知を受け取るたびに確保し、報告を行う作業者ごとに区別して取り扱う。

一方、すでに他の作業者に通知を送っている場合には、作業の処理開始時刻の変更によりシステム全体に生じた変更量に関する報告を受ける場合がある。この場合は、報告された値を Δt に加算し、処理をStep 5に移す。

```

repeat
  Receive(message,  $\xi$ )
until message  $\neq$  empty
if message = "NOTICE"
then Receive( $\hat{r}_{\eta,i}^{\zeta,v}$ ,  $\xi$ )
    $\hat{s}_{\eta,i}^{\zeta,v} = \max(\hat{r}_{\eta,i}^{\zeta,v}, f_{\nu,j}^{\zeta,v-1})$ 
    $\Delta t = 0$ 
   goto Step 2
if message = "REPORT"
then Receive( $\Delta t_1, \xi$ )
    $\Delta t = \Delta t + \Delta t_1$ 
   goto Step 5

```

Step 2: 作業の終了時刻 $\hat{f}_{\eta,i}^{\zeta,v}$ を処理に要する時間 $p_{\eta,i}^{\zeta,v}$ に基づいて算出する。作業の終了時刻、すなわち、次の作業者に対する作業の開始可能時刻に変更があった場合には、Step 3を、変更が無かった場合は、Step 6の処理を行う。

```


$$\hat{f}_{\eta,i}^{\zeta,v} = \hat{s}_{\eta,i}^{\zeta,v} + p_{\eta,i}^{\zeta,v}$$

if  $f_{\eta,i}^{\zeta,v} \neq \hat{f}_{\eta,i}^{\zeta,v}$ 
then goto Step 3
else goto Step 6

```

Step 3: 処理開始時刻の変更を行った工程が、別の作業者により引き続き処理が行われる場合は Step 4 へ、変更を行った作業が、その製品の最終工程であった場合は、Step 5 に処理を移す。ここでは、対象とする作業が製品 η の i 番目の工程であることから、他の作業者 φ にとって w 番目の作業になるような、製品 η の $i+1$ の工程の存在を確認する。

```

if  $j_{\eta,i+1}^{\varphi,w} \neq \{\phi\}$ 
then goto Step 4
else goto Step 5

```

Step 4: 次に作業を行う作業者 φ に対し、変更後の仕事の開始可能時刻を通知し、Step 1 に処理を移す。

```


$$\hat{r}_{\eta,i+1}^{\varphi,w} = \hat{f}_{\eta,i}^{\zeta,v}$$

Send("NOTICE",  $\varphi$ )
Send( $\hat{r}_{\eta,i+1}^{\varphi,w}$ ,  $\varphi$ )
goto Step 1

```

Step 5: Step 3 において取り扱った仕事はその製品の最終工程の場合、あるいは、Step 1 において通知を出した作業者からの報告を受けた場合、次に自分が処理を行う作業の存在を調べる。すなわち、 $v+1$ 番目に処理を行う製品 ψ の k 番目の工程にあたる作業の存在を確認する。該当する作業がある場合、その作業の処理開始時刻の変更を行い、Step 2 において $\hat{s}_{\eta,i}^{\zeta,v}$ に対して行っている処理を、 $\hat{s}_{\psi,k}^{\zeta,v+1}$ に対して行う。作業が残っていない場合、Step 6 に処理を移す。

```

if  $j_{\psi,k}^{\zeta,v+1} \neq \{\phi\}$ 
then  $\hat{s}_{\psi,k}^{\zeta,v+1} = \max(r_{\psi,k}^{\zeta,v+1}, \hat{f}_{\eta,i}^{\zeta,v})$ 
goto Step 2
else goto Step 6

```

Step 6: 最後に変更した作業の作業終了時刻の変更量を、 Δt の値に加算する。ここで、作業者の末尾の作業の処理終了時刻に変更があった場合にのみ、 Δt_2 はゼロ以外の値をとる。Step 1 において、他の作業者から報告のあった値 Δt_1 の値を Δt に記憶しているため、結局、Step 6 において報告する Δt の値は、最初に通知のあった作業 $j_{\eta,i}^{\zeta,v}$ に対して作業者 ζ が変更を行った結果システム全体に生じた変更量になる。この値を、Step 1 の処理における通知の発信元である ξ に対して報告する。報告後は、処理を Step 1 に移し、他の作業者からのメッセージを待つ。

$$\Delta t_2 = f_{\eta,i}^{\zeta,v} - \hat{f}_{\eta,i}^{\zeta,v}$$

$$\Delta t = \Delta t + \Delta t_2$$

Send(“REPORT”, ξ)
 Send(Δt , ξ)
 goto Step 1

一方、作業の処理順序を変え、システム全体のスケジュールを改善する場合の処理手順は、前述の処理と比べて、最初の処理と最後の処理が異なるだけである。作業者は作業の搬入時刻と処理開始時刻の間に隔たりがある作業に着目し作業の処理順序を変え、着目した作業をより早い時刻に処理することで、システム全体のスケジュールの改善を試みる。次に処理の手順を示す。

Step 1': 着目する作業の処理開始時刻を変更し、Step 2 に処理を移す。

$$\hat{s}_{\eta,i}^{\zeta,v} \leftarrow s_{\eta,i}^{\zeta,v}$$

goto Step 2

Step 6': 最後に変更した作業の作業終了時刻の変更量を、 Δt の値に加算し、処理を終了する。

$$\Delta t_2 = f_{\eta,i}^{\zeta,v} - \hat{f}_{\eta,i}^{\zeta,v}$$

$$\Delta t = \Delta t + \Delta t_2$$

Stop

2.3 1 機械問題

1 機械問題とは、複数の製品を1台の機械を用いて製造する場合のスケジュールを決定する問題である。各製品は1作業であり、その作業の順序付けが問題となる。また、1 機械問題において、機械のメイクスパンは各作業の総和となり、作業を処理時間の小さい順に割り付けることで、最適順序が得られる [3] [4]。本研究では、1 機械問題を解く際に、各作業に開始可能時刻、残り処理時間という制約を与え、目的関数を機械のメイクスパンとする。また、各作業に開始可能時刻、残り処理時間の制約を加えるため、本研究で適用する1 機械問題は、有限回数での求解が難しいとされる NP 困難な問題に分類される。

2.3.1 前提条件

本論文での1 機械問題は次の前提条件を持つとする。

- 各作業に要する処理時間は既知

- 各作業は開始可能時刻を持つ
- 各作業は残り処理時間を持つ

例として, 4つの仕事に対してこれらの前提条件との関係を図2.1に示す.

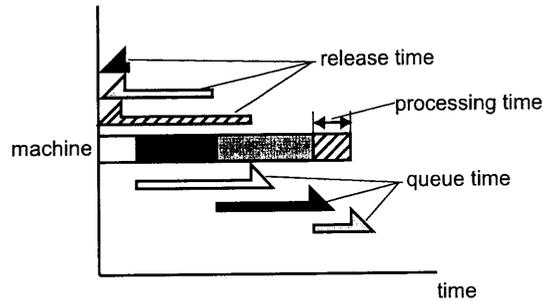


図 2.1: 1 機械問題のスケジューリング例

2.3.2 定式化

1 台の機械を用い 2 つの仕事进行处理する 1 機械問題の定式化を行う. 目的関数はメイクスパンの最小化である. 図 2.2 に 1 台の機械を用い, 二つの製品进行处理するスケジューリング例を示す.

仕事の処理開始時刻は前作業の作業開始時刻と処理時間との和と, 仕事の開始可能時刻で遅い時刻とする.

$$s_{\beta}^v = \max\{s_{\alpha}^{v-1} + p_{\alpha}^{v-1}, r_{\beta}^v\} \quad (2.1)$$

次にこの問題のメイクスパンは, 各仕事の処理開始時刻と処理時間と残り処理時刻の和の最大の時間である.

$$fmax = \max\{s^v + p^v + q^v\} \quad (2.2)$$

2.3.3 1 機械問題とジョブショップ問題

1 機械問題は, 複数の機械を対象としたジョブショップスケジューリング問題を解く際にも利用されることがある. すなわち, 1 つのジョブショップスケジューリング問題を複数

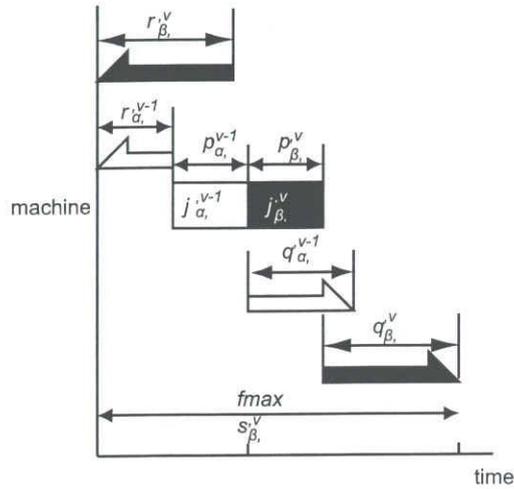


図 2.2: 1 機械問題のスケジューリング例

の 1 機械問題に分けて考えることにより, 他の機械の処理順序によって生じる制約を緩和し, 結果として効率の良い求解が期待できる.

ジョブショップ問題を 1 機械問題に置き換える際に, 例えば, 処理開始可能時刻は, 前工程の処理終了時刻とすることができる. 残り処理時間は, 最終工程の処理終了時刻と, その作業の処理終了時刻の差を残り処理時間とすることができる. 3 台の機械による 3 製品のスケジューリング問題を 3 つの 1 機械問題に分割した様子を図 2.3 に示す.

$$r_{\eta,i} = f_{\eta,i-1} \tag{2.3}$$

$$q_{\eta,i} = f_{\eta,n\eta} - f_{\eta,i} \tag{2.4}$$

また, 分散スケジューリングにおいても, 各作業の開始可能時刻, 処理時間, 残り処理時間

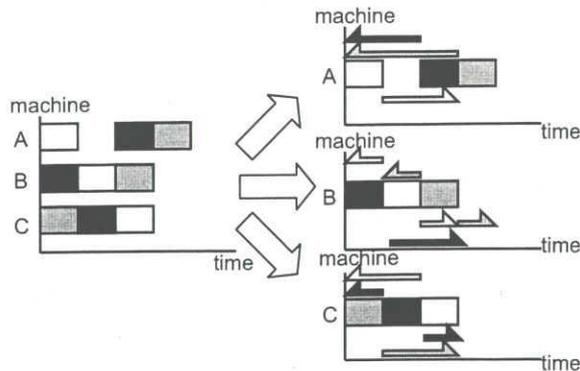


図 2.3: ジョブショップ問題から 1 機械問題への変換

を得ることが可能なため1機械問題を適用することが可能である. そのため, 1機械問題は分散管理された機械がスケジュールを立案するの上で有効であると考えられる.

2.3.4 Schrage 順序

本研究では1機械問題の解法として, Schrage 順序を使用する. Schrage は着手可能な作業から順に割り付け, 着手可能な仕事が多数存在する場合, 残り処理時間が大きい作業から順に割り付ける. 図 2.4 に n 個の作業を Schrage 順序に整列し, スケジュールした様子を示す.

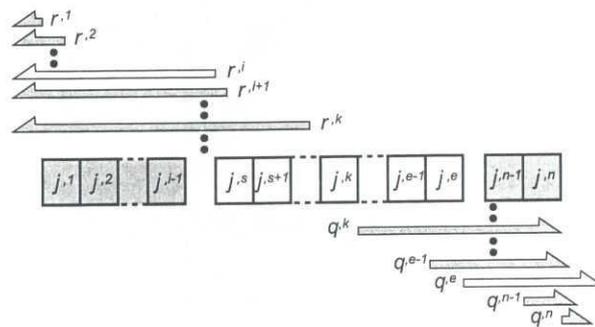


図 2.4: Schrage 順序に整列したスケジュール

この問題に関してメイクスパン $fmax$ はクリティカルパス上の仕事の集合 Q において以下の式で表すことができる. ここで, Q は仕事 $j.s$ から始まり, 仕事 $j.e$ までの仕事の集合とする. クリティカルパス上の仕事を図 2.5 に示す.

$$fmax = \max_Q \{ r^{.s} + \sum_{k \in Q} p^{.k} + q^{.e} \} \quad (2.5)$$

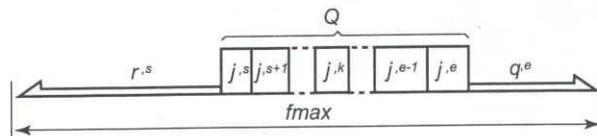


図 2.5: スケジュールのクリティカルパス

一方, このスケジュールに関して, 下界 $h(Q)$ は以下の式で表すことができる.

$$h(Q) = \min_{i \in Q} r^{.i} + \sum_{i \in Q} p^{.i} + \min_{i \in Q} q^{.e} \quad (2.6)$$

このとき、最も開始可能時刻 r が小さい仕事が最初に処理され、最も残り処理時間 q が小さい仕事が最後に処理される時、すなわち $fmax = h(Q)$ のとき、このスケジュールは最適である。もし、 $fmax > h(Q)$ ならば、クリティカルパス上の仕事の集合 Q の最後に処理される仕事 j^e の残り処理時間 q^e よりも小さな値 q を持つ仕事 j^c が存在する。この仕事をクリティカルジョブと呼ぶ。また、クリティカルパス上の作業の中で、クリティカルジョブより後方に存在する作業の集合を Q_a とする。クリティカルジョブが存在するスケジュールを図 2.6 に示す。

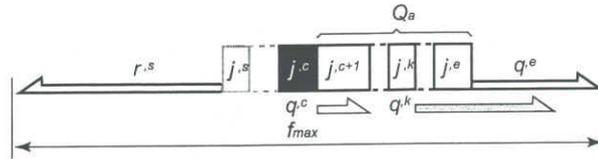


図 2.6: クリティカルジョブが存在するスケジュール

クリティカルジョブ j^c とクリティカルジョブより後方に存在する作業の集合 Q_a において、残り作業時間には以下の関係が成り立つ。

$$q^k > q^c > q^e \quad k \in Q_a \quad (2.7)$$

この関係が成り立つということは、クリティカルジョブの処理開始時刻 s^c の時点で、クリティカルジョブ j^c 以外の仕事を始めることができない。すなわち、 Q_a に属する作業のなかで、最小の開始時刻 $\min_{k \in Q} r^k$ はクリティカルジョブの処理開始時刻 s^c よりも大きいことを意味する。そのため、以下の式が成り立つ。

$$s^c = r^s + p^k + \dots + p^{c-1} < \min_{k \in Q} r^k \quad (2.8)$$

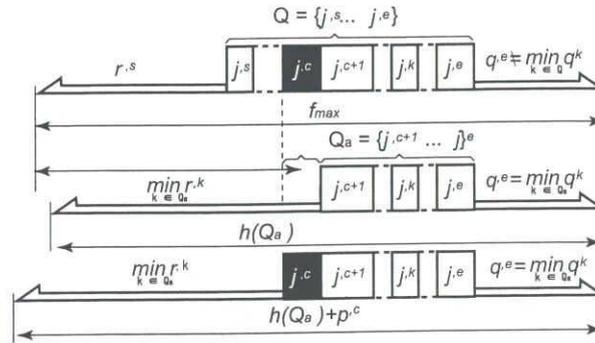
さらに、2.5 式、2.8 式より、以下の式が成り立つ。

$$fmax > h(Q_a) + p^c > h(Q_a) \quad (2.9)$$

以上のことより、最適解を f_{opt} とすると、以下の式が成り立ち、

$$\left. \begin{array}{l} h(Q_a) > f_{opt} \\ h(Q_a) + p^c > fmax \end{array} \right\} fmax - f_{opt} < p^c \quad (2.10)$$

$fmax$ と最適解 f_{opt} の差はクリティカルジョブの処理時間 p^c より小さくなる。 $fmax$ と f_{opt} とクリティカルジョブの関係を図 2.7 に示す。

図 2.7: f_{max} と f_{opt} とクリティカルジョブの関係図

すなわち, Schrage 順序を利用し, クリティカルジョブを考慮に入れることで, 上界と下界の範囲を限定することが可能である. さらに, 上界と下界の差はクリティカルジョブの処理時間 p^c 以下である.

ところで, 作業を Schrage 順序により整列し, 最適な処理順序において, クリティカルジョブが存在した場合, クリティカルジョブ j^c は, 作業の集合 Q_a のすべての作業に, 先行するか, すべての作業に後続するかのどちらかである. これは以下のことより明らかである.

2.5 式に 2.8 式を代入することより, 次式が与えられる. 次式は, クリティカルジョブが先行した場合を意味する.

$$f_{max} = s^c + p^c + \sum_{k \in Q_a} p^k + q^e \quad (2.11)$$

また, Q_a の作業の集合の中にクリティカルジョブ j^c を挿入すると, 次式が与えられる. このときのメイクスパンを f_{max}' とする.

$$f_{max}' = \min_{k \in Q_a} + \sum_{k \in Q_a} p^k + p^c + q^e \quad (2.12)$$

このとき, 2.8 式より,

$$f_{max} < f_{max}' \quad (2.13)$$

となるため, クリティカルジョブ j^c を Q_a の作業の集合の中に挿入してしまうとメイクスパンが大きくなる. クリティカルジョブが Q_a に先行した場合と, Q_a の作業の集合に挿入した場合のスケジュールを図 2.8 に示す. また, クリティカルジョブ j^c を Q_a の作業の集合に後続させた場合, 以下の条件が成り立つときに限り, メイクスパンが小さくなる.

$$\min_{k \in Q_a} r^k + q^e > s^c + p^c + q^c \quad (2.14)$$

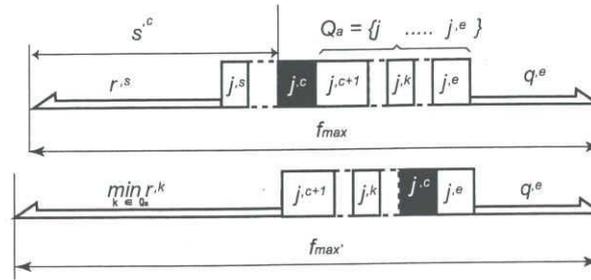


図 2.8: クリティカルジョブが Q_a に先行した場合, 及び Q_a の作業の集合に挿入した場合のスケジュール

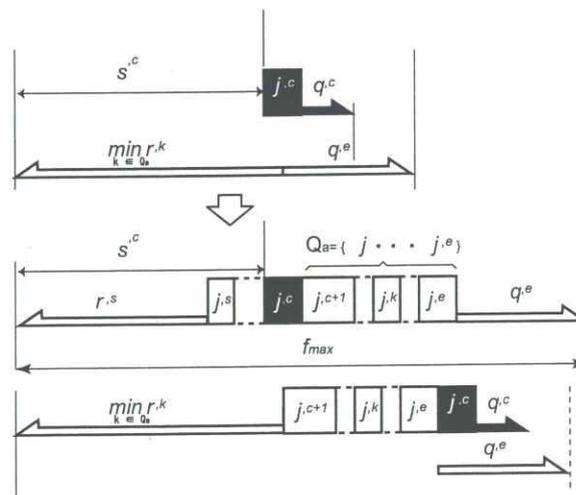


図 2.9: クリティカルジョブが Q_a に後続した場合のスケジュール

クリティカルジョブが Q_a に後続した場合のスケジュールを図 2.9 に示す。

一方, 分散スケジューリングにおいて 1 機械問題を適用する場合にも, Schrage 順序に整列し, クリティカルジョブを考慮に入れることはスケジュールを立案する上で有効であると考えられる。それは, 各機械のメイクスパンを改善することが可能である。すなわち, システム全体のメイクスパンの改善が可能である。

2.4 再帰的伝播法の拡張

分散スケジューリングにおいて, 適切な残り処理時間を知ることができれば, 1 機械問題を解く際に行う求解過程を, 個々の立場で意志決定を行うことができる作業者が利用することが可能になると考えられる。このとき, 適切な残り処理時間を得ることで, 各機械の上界値, 下界値を得ることが可能になる。上界値, 下界値を得ることで解の範囲を限定

することが可能になる。そのため、集中管理機構を必要とせずに限られた情報の中で適切な残り処理時間を得る必要がある。また、これまでの再帰的伝播法は、仕事の開始可能時刻を取得する方法に他ならない。本章では、仕事の残り処理時間の取得方法について、新たな情報交換の手順の提案を行う。

残り処理時間は各製品の最終工程の終了時刻によって決定されるため、提案する方法は各製品の最終工程の処理時刻が変更した場合に行われる。提案する方法を用いて機械が行う動作は、残り処理時間の”変更”，その変更の”通知”，及び、変更が終了したことの”報告”の3つである。変更は各仕事の残り処理時間を適切に得るために行われる。また通知は、最終工程の処理時刻が変化した場合、その仕事の前工程を受け持つ機械に最終工程の終了時刻を伝える。ただし通知は常に前工程を受け持つ機械のみ行われる。報告は、第一工程の作業の変更が終わった場合に通知のあった機械に対して行われる。すなわち、後工程を受け持つ機械に報告し、その機械の受け持つ仕事が最終工程の場合、報告を終了する。

2.4.1 情報交換手順

生産システム内の機械は、常に他の機械からのメッセージが送られてくるのを待っているものとする。通知を受け取ってから報告を行うまでの、再帰的伝播法及び、残り処理時間の調整に関する再帰的伝播法による情報交換の手順を以下に示す。

Step 1: 他の機械からメッセージが送られてくるのを待つ。メッセージを受け取った機械を ζ 、送信元の機械を ξ とする。受け取ったメッセージが変更に関する通知の場合、通知のメッセージに続いて、機械 ζ は、自分が v 番目に行う作業 $j_{\eta,i}^{\zeta,v}$ の変更後の開始可能時刻 $\hat{r}_{\eta,i}^{\zeta,v}$ に関する情報を受け取る。受け取った時刻と、その作業の直前に処理を行う作業 $j_{\nu,j}^{\zeta,v-1}$ の終了時刻 $f_{\nu,j}^{\zeta,v-1}$ のうち、遅い方の時刻をその作業の処理開始時刻とする。また、システム全体に生じる影響の評価値を記憶するための変数 Δt の確保と初期化を行い、処理をStep 2に移す。 Δt は、通知を受け取るたびに確保し、報告を行う機械ごとに区別して取り扱う。

一方、すでに他の機械に通知を送っている場合には、作業の処理開始時刻の変更によりシステム全体に生じた変更量に関する報告を受ける場合がある。この場合は、報告された値を Δt に加算し、処理をStep 5に移す。

```

repeat
  Receive(message,  $\xi$ )
until message  $\neq$  empty
if message = "NOTICE"
then Receive( $\hat{r}_{\eta,i}^{\zeta,v}$ ,  $\xi$ )
    $\hat{s}_{\eta,i}^{\zeta,v} = \max(\hat{r}_{\eta,i}^{\zeta,v}, f_{\nu,j}^{\zeta,v-1})$ 
    $\Delta t = 0$ 
  goto Step 2

```

```

if    message = "REPORT"
then  Receive( $\Delta t_1, \xi$ )
         $\Delta t = \Delta t + \Delta t_1$ 
        goto Step 5

```

Step 2: 作業の終了時刻 $\hat{f}_{\eta,i}^{\zeta,v}$ を処理に要する時間 $p_{\eta,i}^{\zeta,v}$ に基づいて算出する。作業の終了時刻, すなわち, 次の機械に対する作業の開始可能時刻に変更があった場合には, Step 3 を, 変更が無かった場合は, Step 6 の処理を行う。

```

         $\hat{f}_{\eta,i}^{\zeta,v} = \hat{s}_{\eta,i}^{\zeta,v} + p_{\eta,i}^{\zeta,v}$ 
if     $f_{\eta,i}^{\zeta,v} \neq \hat{f}_{\eta,i}^{\zeta,v}$ 
then  goto Step 3
else  goto Step 6

```

Step 3: 処理開始時刻の変更を行った工程が, 別の機械により引き続き処理が行われる場合は Step 4 へ, 変更を行った作業が, その製品の最終工程であった場合は, Step 5 に処理を移す。ここでは, 対象とする作業が製品 η の i 番目の工程であることから, 他の機械 φ にとって w 番目の作業になるような, 製品 η の $i+1$ の工程の存在を確認する。

```

if     $j_{\eta,i+1}^{\varphi,w} \neq \{\phi\}$ 
then  goto Step 4
else  goto Step 5

```

Step 4: 次に作業を行う機械 φ に対し, 変更後の仕事の開始可能時刻を通知し, Step 1 に処理を移す。

```

         $\hat{r}_{\eta,i+1}^{\varphi,w} = \hat{f}_{\eta,i}^{\zeta,v}$ 
        Send("NOTICE",  $\varphi$ )
        Send( $\hat{r}_{\eta,i+1}^{\varphi,w}$ ,  $\varphi$ )
        goto Step 1

```

Step 5: 最終工程の記号を n_η とし, Step 3 において取り扱った仕事はその製品の最終工程の場合, また, その製品の後工程を処理する機械から通知を受け取っている場合, 残り処理時間 $q_{\eta,i}^{\zeta,v}$ を変更する。残り処理時間 $q_{\eta,i}^{\zeta,v}$ は, 最終工程の終了時刻 f_{η,n_η} と, その仕事の終了時刻 $f_{\eta,i}^{\zeta,v}$ の差とする。次に, ここで対象としている作業に前工程が存在するか確認する。前工程が存在する場合, 処理を Step 6 に移す。前工程が存在しない場合は処理を Step 7 に移す。

```

if     $q_{\eta,i}^{\zeta,v} = f_{\eta,n_\eta} - \hat{f}_{\eta,i}^{\zeta,v}$ 
         $j_{\eta,i}^{\xi,u} \neq \{\phi\}$ 
goto Step 6
goto Step 7

```

Step 6: 前工程の仕事を受け持つ機械 ξ に対し、最終工程の終了時刻 f_{η,n_η} を通知し、処理をStep 8へ移す。

```
Send("QNOTICE",  $\xi$ )
Send( $f_{\eta,n_\eta}$ ,  $\xi$ )
goto Step 8
```

Step 7: Step 5で取り扱った仕事に前工程が存在しなかった場合、ここで対象としている仕事が最終工程であるか確認する。最終工程でなかった場合、後工程を受け持つ機械 φ に対して、残り処理時間の変更が終わったことを報告し、処理をStep 8へ移す。また、最終工程の場合は、対象とする製品すべての仕事に対して、残り処理時間の変更が終了しているため、処理をStep 9へ移す。

```
if  $j_{\eta,i+1}^{\varphi,w} \neq \{\phi\}$ 
then Seng("QREPORT",  $\varphi$ )
    goto Step 8
else goto Step 9
```

Step 8 受け取ったメッセージが残り処理時間の変更の通知の場合、通知のメッセージに続いて、最終工程の終了時刻 f_{η,n_η} を受け取り、処理をStep 5へ移す。

一方、残り処理時間の変更が終わった報告を受けた場合、処理をStep 7へ移す。

```
Receive(message,  $\zeta$ )
if message = "QNOTICE"
then Receive( $f_{\eta,n_\eta}$ ,  $\zeta$ )
    goto Step 5
if message = "QREPORT"
then goto Step 7
```

Step 9: Step 3において取り扱った仕事はその製品の最終工程の場合、あるいは、Step 1において通知を出した機械からの報告を受けた場合、あるいは、Step 7において取り扱った仕事はその製品の最終工程の場合、次に自分が処理を行う作業の存在を調べる。すなわち、 $v+1$ 番目に処理を行う製品 ψ の k 番目の工程にあたる作業の存在を確認する。該当する作業がある場合、その作業の処理開始時刻の変更を行い、Step 2において $\hat{s}_{\eta,i}^{\zeta,v}$ に対して行っている処理を、 $\hat{s}_{\psi,k}^{\zeta,v+1}$ に対して行う。作業が残っていない場合、Step 10に処理を移す。

```
if  $j_{\psi,k}^{\zeta,v+1} \neq \{\phi\}$ 
then  $\hat{s}_{\psi,k}^{\zeta,v+1} = \max(r_{\psi,k}^{\zeta,v+1}, \hat{f}_{\eta,i}^{\zeta,v})$ 
    goto Step 2
else goto Step 10
```

Step 10: 最後に変更した作業の作業終了時刻の変更量を, Δt の値に加算する. ここで, 機械の末尾の作業の処理終了時刻に変更があった場合にのみ, Δt_2 はゼロ以外の値をとる. Step 1 において, 他の機械から報告のあった値 Δt_1 の値を Δt に記憶しているため, 結局, Step 10 において報告する Δt の値は, 最初に通知のあった作業 $j_{\eta,i}^{\zeta,v}$ に対して機械 ζ が変更を行った結果システム全体に生じた変更量になる. この値を, Step 1 の処理における通知の発信元である ξ に対して報告する. 報告後は, 処理を Step 1 に移し, 他の機械からのメッセージを待つ.

$$\Delta t_2 = f_{\eta,i}^{\zeta,v} - \hat{f}_{\eta,i}^{\zeta,v}$$

$$\Delta t = \Delta t + \Delta t_2$$

Send("REPORT", ξ)
Send(Δt , ξ)
goto Step 1

2.4.2 情報交換の例

3 台の機械が 2 つの製品を製造する際のスケジュールを用いて, 機械が作業開始時刻を変更した際に必要な情報交換方法について説明する. 図 2.10 に示すスケジュールを初期状態とし, 図 2.11 に示すように, 機械 C が第 1 作業の開始時刻を一定量だけ遅らせた場合の情報交換の様子と, スケジュールの変更について説明する.

まず, 図 2.11 において機械 C は, 第 1 番目の作業時間を調整した後, その製品の次工程を受け持つ機械 B に対して第 2 作業の調整が必要であることを通知する. 通知を受け取った機械 B は, 図 2.12 において開始時刻の調整を行う. このとき, この仕事の次工程も次作業も存在しないため, 機械 B に対して通知を行った機械 C に結果の報告を行う.

次に図 2.13 において機械 C の第 2 作業の時刻調整を行う. ここで, この第 2 作業は最終工程であり, 時刻変化したため, 残り処理時間の調整が必要になる. そのため, この仕事の前工程を受け持つ機械に残り処理時間の変更が必要だということを通知する. 図 2.14 において, 通知を受け取った機械 B は, 第 1 作業の残り処理時間の調整を行い, 同様に前工程を受け持つ機械 A に残り処理時間の調整が必要なことを通知する. 図 2.15 において, このとき機械 A に対して, 通知を行った機械 B に残り処理時間の調整が終わったことを報告する. 図 2.16 において, 同様に, 機械 B に対して, 通知を行った機械 C に対して, 残り処理時間の調整が終わったことを報告する. この時, 機械 C の第 2 作業に着目点が戻ってきたため, 残り処理時間の調整を終了する. また, この作業は後作業も存在しないため, 時刻の調整を終了する. この時の状態を 2.17 に示す.

以上の情報交換の手順によって, 作業の開始時刻の変更をシステム内の必要な機械に伝えることができ, さらに, 各機械の作業が適切な残り処理時間を得ることが可能になる. したがって, 残り処理時間が適切に得られるため, この提案する方法により, 開始可能時刻と残り処理時間を考慮した 1 機械問題の適用が可能になる.

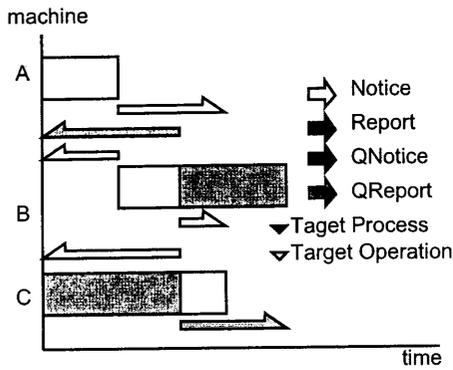


図 2.10: 初期状態

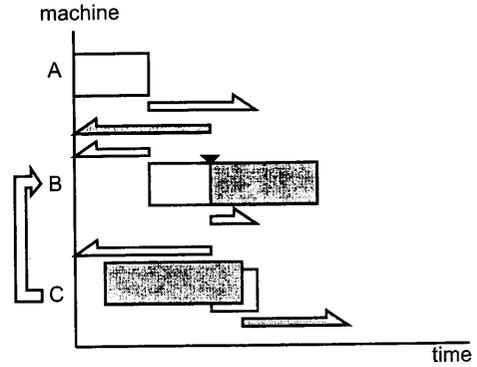


図 2.11: 開始時刻を変更と通知

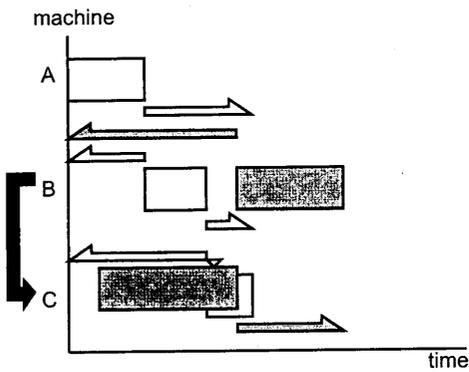


図 2.12: 開始時刻の変更と報告

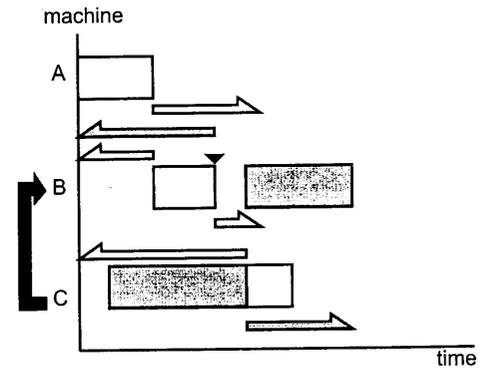


図 2.13: 開始時刻の変更と残り処理時間の調整のための通知

2.5 対象とするスケジューリング問題

本研究で対象とするスケジューリング問題はジョブショップスケジューリング問題とする。ジョブショップスケジューリング問題は、複数の製品を複数の機械を用いてス

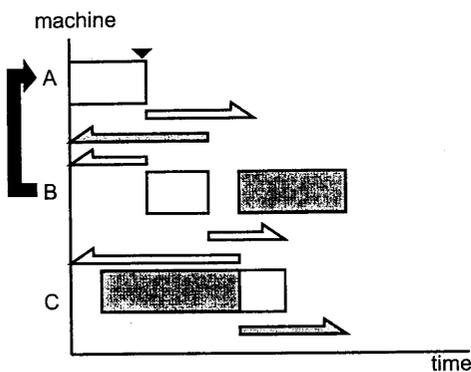


図 2.14: 残り処理時間の変更と残り処理時間の調整のための通知

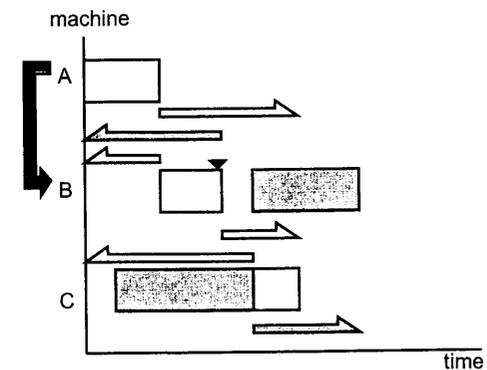


図 2.15: 残り処理時間の変更と残り処理時間の変更終了の報告

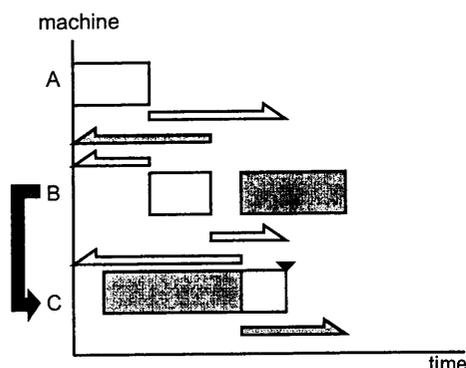


図 2.16: 残り処理時間の変更終了の報告

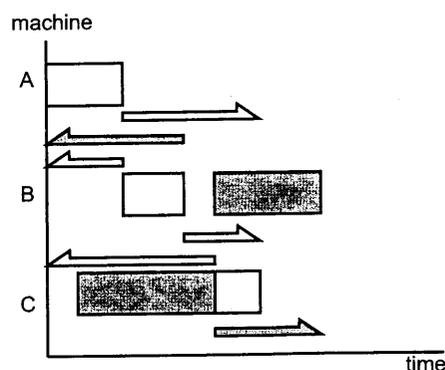


図 2.17: 最終状態

スケジュールを決定する問題である。当然、機械の操作には作業者が携わる。以下、機械を擬人化し、行為の主体とする記述を行うが、それらはすべて、それら进行操作する作業者に伝えられることになる。

また、本研究では、初期スケジュールが与えられているとする。初期スケジュールは、作業の先着順に各機械のスケジュールの末尾に作業を割り当てたスケジュールである。本研究では、トークン (token) を用いて、スケジュールの改善を行う機械を決定する。トークンとは、ある動作の実行許可を運んだり、分散システムの状態を収集したりするために利用する特別なメッセージを意味する [6]。ここでは、1 個のトークンを用い、トークンを持っている機械だけが作業順序の変更を行うものとし、その他の機械は情報交換により作業の処理開始時刻の調整を行う。また、機械は環状にネットワークを構成しており、トークンはネットワークに沿って順に渡されて行くものとする。また、機械間の情報交換は再帰的伝播法を用いて行う。本研究では、この再帰的伝播法を利用し、分散スケジューリングにおいて、新しいスケジューリングの解法を提案する。

2.6 従来のスケジューリング

従来のスケジューリングは、クリティカルパスがスケジュールのメイクスパンを決めていることから、各機械が時間的に連続している作業に着目して、その作業を試行錯誤に入れ換えることでスケジュールを改善していく手法である。以下にそのスケジューリングの解法を示す。また、各機械は最後に処理を行う予定の仕事にマークをつけておく。

Step1: トークンを受け取る。

Step2: マークをつけた作業を起点として、その作業の処理時刻 $s^{i,s}$ とひとつ前の作業の終了時刻 $f^{i,s-1}$ を比較し、時間的に連続している作業を探す。この仕事の開始可能時刻 $r^{i,s}$ より作業開始時刻 $s^{i,s}$ より遅い場合には、より早い時刻に処理を始めることができる可能性があるため、その仕事にマークを移す。

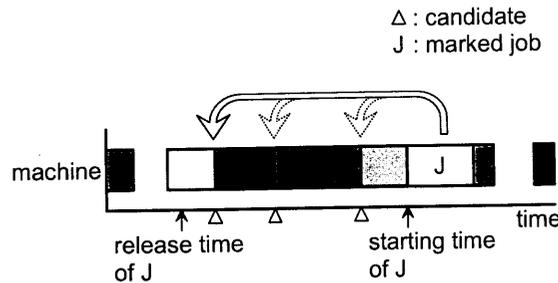


図 2.18: 適切な作業順序の探索

Step3: 図 2.18 に示すように、マークをつけた作業を開始可能時刻よりも遅い時刻内で、前方の作業と処理順序を交換する。作業の処理順序の変更は再帰的伝播法により他の機械に通知を出す。報告の結果、受け取った Δt の値がゼロまたは負の場合には、スケジュールを改善されたことを意味するため、該当する作業順序を採用する。 Δt の値が正の場合には、作業の処理順序を交換前の状態に戻す。作業順序の変更の候補が複数ある場合には、最も小さい Δt の値を与える作業順序に対して判定を行う。処理後は、マークを1つ先に処理する作業に移す。もしマークをつける作業がない場合には、末尾の作業にマークをつける。

Step4: トークンを次の機械に渡す。

2.7 Schrage 順序による分散スケジューリング

本章では、分散スケジューリングにおいて、各機械が限られた情報に基づいて独自にスケジューリングを立案するため1機械問題を適用できると考えられる。そこで1機械問題を解く際に利用される Schrage 順序に着目して、新しいスケジューリング手法を提案し、その手順を説明する。

2.7.1 スケジューリング手順

Step1: トークンを受け取る。

Step2: 機械が処理を開始する時刻を t とする。すなわち $t = 0$ とする。また、この機械が処理する全ての作業の集合を J とする。

Step3: 作業の集合 J の中で最も開始時刻が早い仕事を $j^{s,s}$ とする。

次に時刻 t において、作業の集合 J の中から処理可能な作業を選択する。すなわち、時刻 t 以下の開始可能時刻 r^s をもつ作業を選択する。また、時刻 t 以下の開始可能

時刻を持つ作業が存在しない場合は、作業の集合 J の中から最小の開始可能時刻を持つ作業 $\min_{i \in J} r^{i,s}$ を選択する。このとき、複数の作業が選択された場合は、その作業の中から最大の残り処理時間 $\max q^{i,s}$ を持つ作業を選択する。作業を選択する様子を図 2.19 に示す。

Step4: 次に選択された作業を $j^{i,v}$ とし、開始時刻 $s^{i,v}$ を変更する。変更後の開始時刻を $\hat{s}^{i,v}$ とする。この変更後の作業の開始時刻 $\hat{s}^{i,v}$ は、時刻 t と、作業 $j^{i,v}$ の開始可能時刻 $r^{i,v}$ とで、大きい時刻とする。変更後の作業 $j^{i,v}$ の開始時刻 $\hat{s}^{i,v}$ は以下の式で表すことができる。

$$\hat{s}^{i,s} = \max\{t, r^{i,s}\} \quad (2.15)$$

Step5: 作業 $j^{i,v}$ の開始時刻の変更は再帰的伝播法により他の機械に通知を出す。また、再起伝播法により、時刻の調整を行っている最中に、製品の最終工程に時刻の変化が生じた場合、 q を調整する再帰的伝播法により、適切な残り処理時間 q を得る。

Step6: 再帰的伝播法の報告の結果、受け取った Δt の値がゼロまたは負の場合は Step7 へ処理を移す。一方、受け取った Δt の値が正の場合は Step8 へ処理を移す。

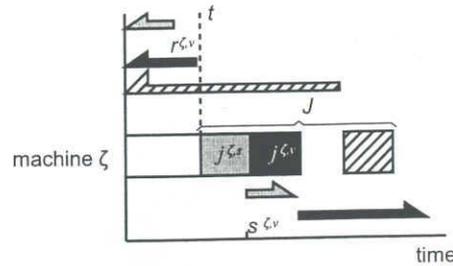
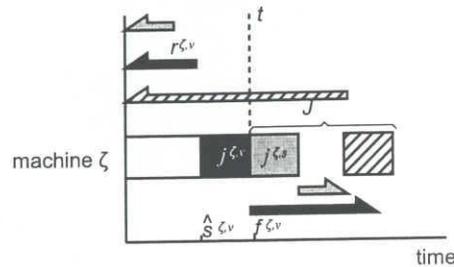
Step7: 次に、開始時刻の変更を行った作業を、作業の集合 J の中から取り除く。すなわち $J \setminus \{j^{i,v}\}$ とする。また、時刻 t の更新を行う。ここで時刻 t を開始時刻の変更を行った作業 $j^{i,v}$ の終了時刻 $f^{i,v}$ とし、処理を Step3 に移す。

一方、作業の集合が空になった場合は処理を Step10 へ移す。選択された作業の開始時刻の変更をし、時刻 t が更新された様子を図 2.20 に示す。

Step8: Step6 において、受け取った Δt の値が正の場合はスケジュールが悪くなっていることを示すため、変更後の作業 $j^{i,s}$ の開始時刻 $\hat{s}^{i,v}$ を変更前の開始時刻 $s^{i,v}$ に戻す。さらに、再帰的伝播法により、開始時刻を元に戻したことを通知し、処理を Step9 へ移す。

Step9: 次に、時刻 t において最も開始時刻が早い作業 $j^{i,s}$ を、作業の集合 J の中から取り除く。すなわち $J \setminus \{j^{i,s}\}$ とする。また、時刻 t の更新を行う。ここで時刻 t を作業 $j^{i,s}$ の終了時刻 $f^{i,s}$ とし、処理を Step3 に移す。一方、作業の集合が空になった場合は処理を Step10 へ移す。この時の時刻 t が更新された様子を図 2.21 に示す。

Step10: トークンを次の機械に渡す。

図 2.19: 時刻 t における作業の選択図 2.20: 開始時刻の変更と時刻 t の更新

2.8 メイクスパンと情報交換回数の評価

本研究で提案するクリティカルジョブを考慮に入れたスケジューリング手法の有効性を評価するために、従来法と集中型の一般に良い解が得られるとされているディスパッチングルールである最早開始時刻優先ルールとの比較を行った。最早開始時刻優先ルールは開始時刻が早い順に仕事を割り付けスケジュールを立案する。対象とした問題はジョブショップスケジューリング問題であり、一様乱数を用いて作成した。また作業の先着順に割り当てた状態を、スケジュールの初期状態とした。例として、機械数5台、製品数5個のときの初期スケジューリングを図2.22に示す。

ここでは、機械数10台、10から100個の各製品数の問題について、100回、スケジューリングを行い、そのメイクスパンの平均をとり、従来法と集中型の最早開始時刻優先ルール、提案法との比較を行った。従来法の終了条件はメイクスパンに変化がなくなった時に、スケジューリングを終了し、提案法の終了条件は、各機械の上界と下界が等しくなったとき、スケジューリングを終了する。図2.23に従来法、集中型の最早開始時刻優先ルール、提案法とのメイクスパンの比較をしたグラフを示す。図2.23において、提案法と従来法を比較すると最大で約30%改善されていることがわかる。さらに、提案法は、集中型の最早開始時刻優先ルールと比較しても、同程度のメイクスパンが得られていることがわかる。また、従来法と集中型の最早開始時刻優先ルールのメイクスパンを比較すると、製品数が増加するにつれ、メイクスパンの差が大きくなっていくことがわかる。提案法と集中型の最早開

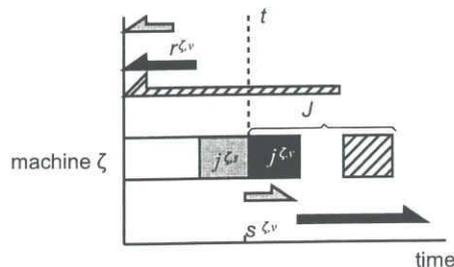
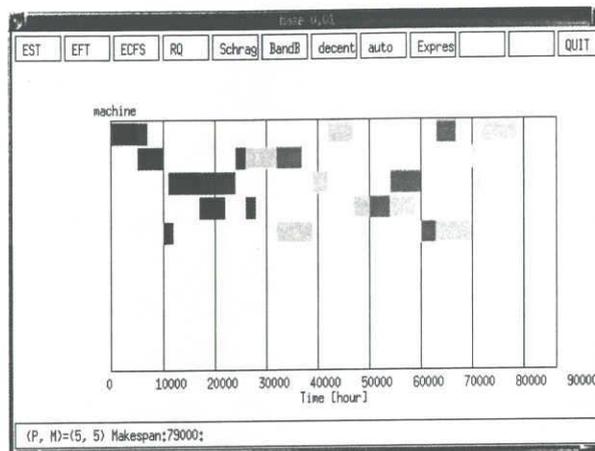
図 2.21: 時刻 t の更新

図 2.22: 初期スケジューリングの例

始時刻優先ルールメイクスパンを比較すると、製品数が増加しても同程度のメイクスパンが得られていることがわかる。

次に、機械数 10 台、10 から 100 個の各製品の問題について、100 回、スケジューリングを行い、機械間の情報交換回数を、従来法と提案法において比較を行った。図 2.24 に提案法と従来法との情報交換回数の比較を行った。図 2.24 において、提案法の情報交換回数は従来法よりも最大で 138 倍多くなっていることがわかる。そこで、提案法の情報交換回数の増加の理由を確かめるため、提案法において、作業の交換の際に生じた情報交換回数と、残り処理時間を調整する際に生じた情報交換の回数、及び、従来法の情報交換回数の比較を行い、図 2.25 に結果を示す。また、従来法は残り処理時間を考えていないため、情報交換は作業の交換のときのみ行われる。図 2.25 において、提案法の情報交換回数において、残り処理時間の調整の際に行われる情報交換回数が 70% を占めており、作業の交換の際に行われる情報交換回数は 30% を占めている。そのため、提案法において、情報交換回数の増大は残り処理時間の調整を行う情報交換に起因していると考えられる。また、従来法の情報交換回数（作業の交換の際に行われる情報交換回数）と提案法の作業の交換の際に行われる情報交換回数を比較すると、提案法の作業の交換の際に行われる情報交換回数は、従

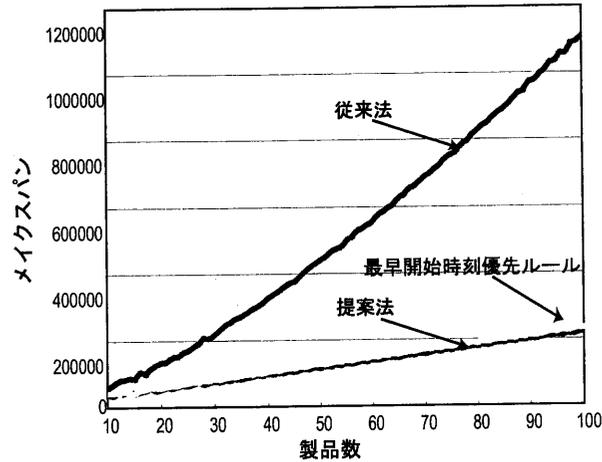


図 2.23: メイクスパンの比較

来法に比べ、40倍多くなっていることがわかる。この原因として考えられることは、従来法においては、各機械が連続している作業を探し、その連続している作業だけを交換して、探索している。しかし、提案法では、各機械のすべての作業を対象として、作業の交換を行うため情報交換回数が多くなっていると考えられる。

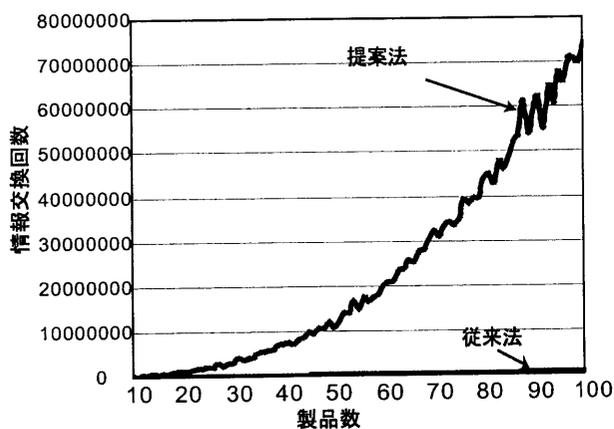


図 2.24: 情報交換回数の比較

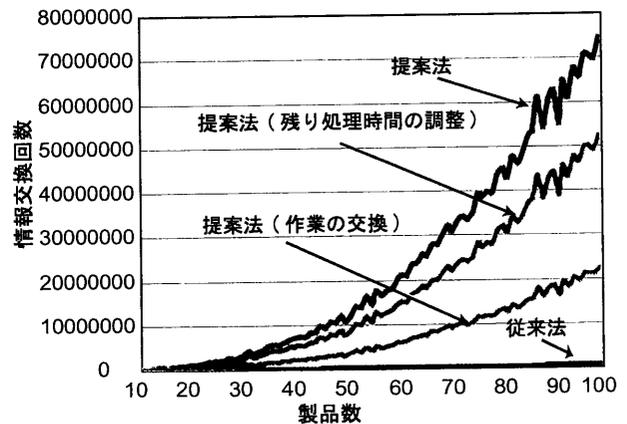


図 2.25: 作業の交換の際の情報交換回数と残り処理時間の調整の際の情報交換回数の比較

次に、提案法において、通信回数がメイクスパンにどの程度影響を与えているか確かめるために、従来法と提案法において通信回数を同じにして、メイクスパンの比較を行った。図 2.26 に従来法と提案法においての通信回数を示し、その時の通信回数において、図 2.27 に従来法、集中型の最早開始時刻優先ルール、提案法とのメイクスパンの比較を示す。図

2.27において、提案法のメイクスパンは従来法のメイクスパンに比べ30%程度悪くなっていることがわかる。また、提案法のメイクスパンは集中型の最早開始時刻優先ルールに比べ、6倍程度悪くなっていることがわかる。そのため提案法は、情報交換回数がメイクスパンに大きな影響を与えていると考えられる。

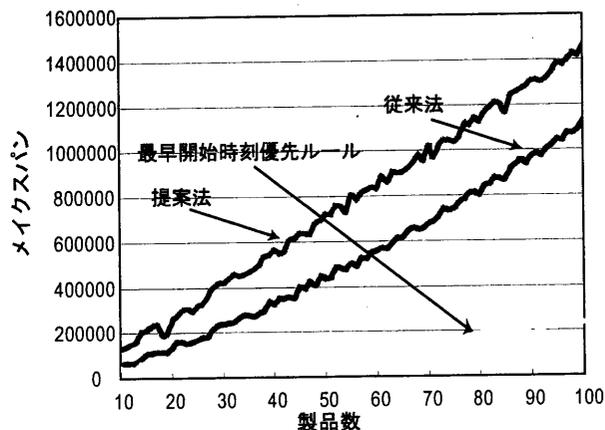
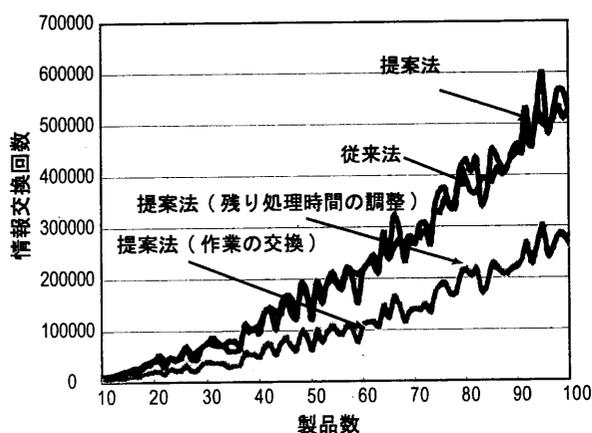


図 2.26: 従来法と提案法の情報交換回数を同じにした時の情報交換回数の比較

図 2.27: 従来法と提案法の情報交換回数を同じにした時のメイクスパンの比較

以上の結果により、提案法は、情報交換回数を増やすことで、メイクスパンが良くなり、効率的なスケジュールを立案することが可能である。また、提案法は製品数、機械数が増えることにより、作業の交換の際に生じる情報交換回数が増加し、残り処理時間を調整する際に生じる情報交換回数が増加するため、全体の情報交換回数は指数関数的に増加する傾向がある。

2.9 ベンチマーク問題における評価

本研究で提案したクリティカルジョブを考慮した分散スケジューリング手法と、従来法において、一般に解くことが難しいとされている Muth & Thomson の 10 製品 10 機械のジョブショップスケジューリング問題 (最適解:930) を対象にして比較した。表 2.1 に従来法と提案法のマイクスパンと情報交換回数を示す。また、図 2.28, 図 2.29 に従来法と提案法のスケジューリング結果を示す。

従来法は最適解と比べ、42% の誤差であり、提案法は 25% の誤差に収まった。そのため、提案法が有効であるといえる。提案法が従来法と比べ有効である理由は次のことが考えられる。Muth & Thomson の問題はフローショップ問題に似ていることが特徴である。図 2.30 に機械数が 2 台、製品数 3 個のフローショップ問題のスケジューリング例を示す。図 2.30 に示す例を利用し、提案法が有効であることを示す。図 2.30 において、機械 A の第 1 作業と第 2 作業を入れ換えたとする。この時の状態を図 2.31 に示す。従来法は機械 B に

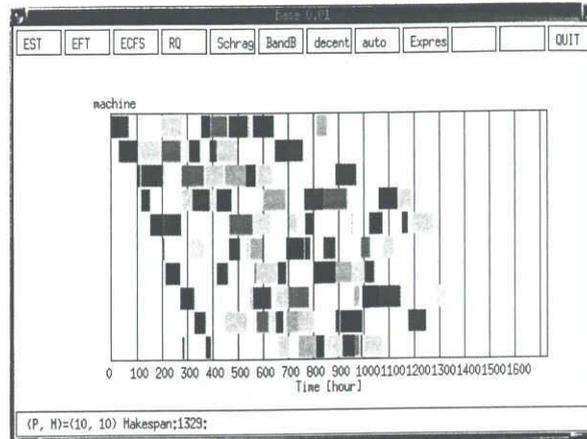


図 2.28: 従来法のスケジュール

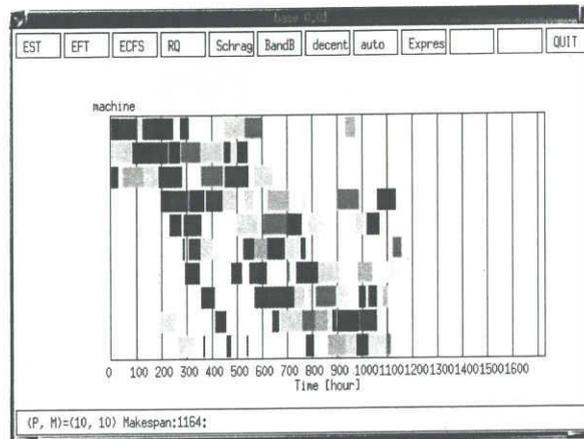


図 2.29: 提案法のスケジュール

トークンを渡された場合、機械Bの第2, 第3の作業が対象となるため、入れ替わることはなく、スケジュールは良くなる。提案法においては、機械Bにトークンを渡された場

表 2.1: 提案法と従来法の比較

	従来法	提案法
メイクスパン	1329	1164
情報交換回数	12928	279468
情報交換回数(作業の交換)	12928	88686
情報交換回数(残り処理時間の調整)	—	190782

合、開始可能時刻、残り処理時間を考え、開始可能時刻が早い順に割り付けるため、図 2.32 に示すように処理順序が機械 A の処理順序と同じになる。そのため、提案法が有効であるといえる。

以上のことより、提案法は各機械が開始可能時刻、残り処理時間を考慮に入れてスケジュールを立案するため、局所解に陥りやすい問題においても有効であるといえる。

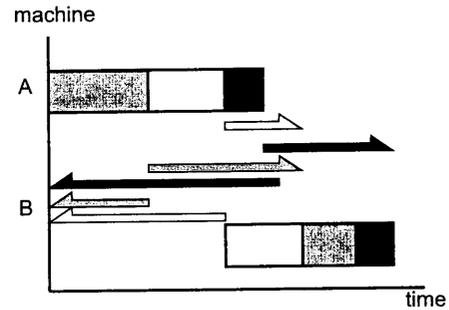
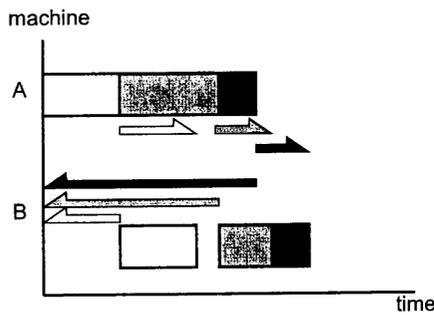


図 2.30: フローショップ問題の初期スケジュール 図 2.31: 機械 A の第 1 作業と第 2 作業の交換後のスケジュール

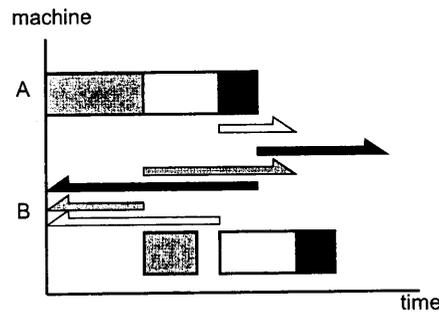


図 2.32: 提案法によるスケジュール

2.10 まとめ

本章では、作業者が与えられた仕事に対して、処理順序や処理内容を変更することを支援することを想定し、分散スケジューリング手法の導入を行った。特に、本研究ではすでに申請者が開発済みの手法を単に利用するだけでなく、あらたな拡張を加えることにより、より効率的な意志決定支援の手法として利用することができるように検討した。

具体的には、スケジューリング問題のひとつである 1 機械問題に焦点をおき、クリティカルジョブを考慮にいれた分散スケジューリング手法の提案をした。

まず、分散スケジューリングにおいて、分散管理下に置かれた機械は限られた情報に基づいてスケジューリングを立案するため、各機械のスケジューリングには各作業が開始可能時刻、残り処理時間を持っている1機械問題を適用した。

次に、分散スケジューリングにおいて、スケジュールを変更する際には、他の機械と協調し、処理する作業の処理時刻の整合性を保つ必要がある。この機械との情報交換手法として、再帰的伝播法を利用した。しかし、本研究で適用した1機械問題は、残り処理時間を考慮に入れている。この残り処理時間は、自分の作業の処理を終了した後、製品が完成するまでの時間を示している。そのため、スケジュールを立案する上で各機械の処理する作業が適切な残り処理時間を得ることは重要であると考えられる。そのため、適切な残り処理時間を得ることができる再帰的伝播法を提案した。具体的には、各仕事の残り処理時間は、製品の最終工程によって決まるため、最終工程の終了時刻が変化したときに、前工程を受け持つ機械にその終了時刻を伝えるといった情報交換を行う。

次に、分散管理下に置かれた機械が、1機械問題の解法の1つである Schrage 順序を利用して、クリティカルジョブを考慮したスケジューリング手法を提案した。具体的には各機械を Schrage 順序を元に整列する。さらに、クリティカルパスを探しクリティカルジョブを決定し、そのクリティカルジョブをクリティカルパス上の最後に割り付け解の評価し、作業順序を決定する。この探索を各機械において繰り返し行い、スケジュールを立案する。

次に提案した分散スケジューリングの有効性を確認するために、数値計算を行った。数値計算では、提案法と従来法と集中型の最早開始時刻優先ルールにおいて、メイクスパンの比較を行った。また、提案法と、従来法において情報交換回数の比較を行った。この数値計算により、提案法のメイクスパンが従来法より、良い結果が得られていることを確認し、集中型の開始時刻優先ルールと同等のメイクスパンが得られていることを確認した。さらに、提案法の情報交換回数が、製品数の増加により、指数関数的に増加することを確認した。また、10機械10製品のベンチマーク問題を対象とし、提案法と従来法の比較を行った。この結果により、一般に難しいとされている問題においても、提案法は従来法よりも、良い解が得られることを確認した。

これらの機能を工作機械、すなわち工作機械を制御する CNC 装置が備えることにより、従来の工作機械では不可能な作業への支援が可能になると考えられる。

第3章 ユーザーインターフェースの整備

3.1 目的

工程計画は、製品の製造方法、すなわち、製品製造の際の使用設備と使用順序および処理時間を決めることである。一方、スケジューリングは、工程計画によって定められたそれぞれの工程について、処理を開始する時刻を決める行為である。したがって、スケジューリングは工程計画を前提条件とする最適化問題と見なすことができる。

スケジューリングに対しては、これまでも多くの検討が行われている [8-12]。しかし、従来のスケジューリング問題は、どちらかといえば、数学的な意味での最適性を求めた結果、実際の生産活動に利用するためにはまだ、多くの課題が残されている。その理由の一つは、従来のスケジューリング問題では、加工機械による処理のみをスケジューリングの対象としている点にある。実際の生産活動では、加工機械以外にも作業員や搬送装置も生産に携わっている。

たとえば、作業員は、加工機械による処理の前後において、材料の機械への取付け、取外しを行ったり、安全上や品質管理上、加工機械の処理開始時や終了時の確認などを行う。また、作業員は以前のように一台の機械のみを取扱うのではなく、複数の機械を取扱うことが一般的となっている。しかし、作業員が機械を取扱うような生産形態は、従来のスケジューリング手法では取扱うことができない。従来のスケジューリング手法では、処理時間が短い場合には省略することが多く、ましてや、複数台の加工機械を取扱うような作業員は考慮に入れていないためである。

また、搬送装置に関しても、機械による処理時間に比べ、搬送時間が短かったり、生産設備の配置や搬送経路などを考慮に入れる必要があるために、従来のスケジューリング手法での取扱いは難しい。

本研究では、スケジューリングの対象を加工機械のみならず、搬送車や作業員に拡張したスケジューリング問題を扱う。具体的には、生産設備間の加工物の受渡し関係について詳細な分析を行い、理論上は全ての生産形態を取扱うことができるようなスケジューリング問題の一般化を行う。この一般化によりこれまでのスケジューリング問題が対象としている加工機械に加え、複数の機械を操作する作業員も考慮に入れたスケジューリングが可能である。本研究では、スケジューリング問題の一般化により、取扱いが可能になった機械、搬送車、作業員が連携して行う作業を考慮に入れた場合を想定し、特にユーザーインターフェースを開発する上で基礎となる、問題の記述方法について整理する。具体的には、構造化記述言語の一つである XML を用いた生産計画の記述法を採用する。

3.2 スケジューリング問題の一般化

ここでは、前章までで取り扱ったスケジューリング問題を拡張し、複数の機械が同時に必要とされるような作業を対象とする。すなわち、ひとりの作業者が複数の機械の操作をするような場合、機械だけあるいは作業者だけを対象としたスケジューリングでは、実質的に意味をなさない。

そのため、前章で示した記述法に加え以下の表現を導入する。すなわち、一般的なスケジューリング問題が対象とする1対1の機械間の関係から、多対1あるいは1対多の関係へと拡張する。さらに、単独の機械により行なわれる作業だけでなく、他の機械との連携が求められるいわゆる協調作業についても考察を加える。そのため、添字がより多くの意味を含む。例えば、要素 e から成る集合を $\{e\}$ と記述し、複数の設備により同時に処理される仕事を次のように表現する。ただし、添字が表す情報が意味を持たない場合には、記載を省略することもある。

$j_{\alpha,i}^{\{\zeta_s\},v_s}$: 複数の生産設備 $\{\zeta_s\}$ が各々 v_s 番目に処理する作業
また、製品 α を製造するための i 番目の工程
($s = 1, 2, \dots, n_{\alpha,i}$ $n_{\alpha,i}$ は使用される生産設備台数)

3.3 工程に着目した分析

従来のスケジューリング問題では、工程の受渡しは、1対1の機械間で行なわれることを前提としている。さらに、機械が一度に処理できる作業を一つに仮定した場合に想定されるガントチャートを図3.1に示す。この前提条件の下では、機械は、自分が行なう直前の作業 (previous operation) の終了時刻か、着目する作業が他の機械から受渡される時刻のいずれか遅い時刻にならないと、着目する作業の処理を開始することができない。図3.1から理解できるように、作業がその機械に受渡される時刻は、他の機械によって処理される直前の工程 (previous process) にの終了時刻に等しい。

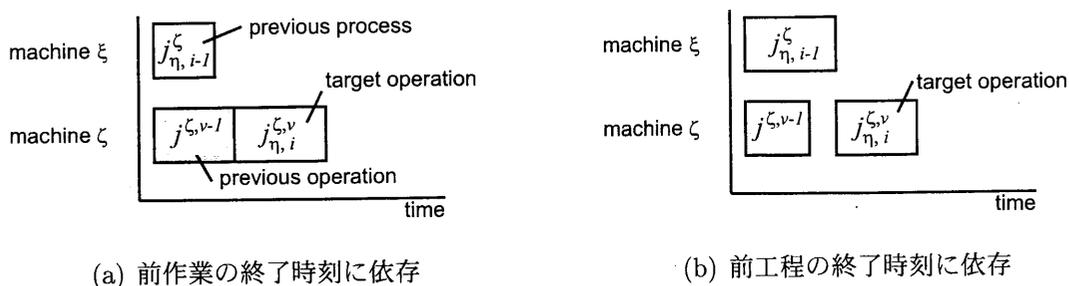


図 3.1: 1 対 1 の機械間で受渡される工程の処理開始時刻

例えば、機械 ζ が v 番目に処理を予定している作業が製品 η の i 番目の工程とし、この作業の処理開始時刻を求める場合を例にあげる。直前の工程が、機械 ξ によって処理される場合、以下の式によって処理開始時刻が求められる。

$$r_{\eta,i}^{\zeta,v} = f_{\eta,i-1}^{\xi} \quad (3.1)$$

$$s_{\eta,i}^{\zeta,v} = \max(r_{\eta,i}^{\zeta,v}, f_{\eta,i}^{\xi,v-1}) \quad (3.2)$$

ここで、 $\max()$ は、括弧内の時刻のうち、大きい方の値を返す関数である。また、処理終了時刻は次式によって求められる。

$$f_{\eta,i}^{\zeta,v} = s_{\eta,i}^{\zeta,v} + p_{\eta,i}^{\zeta,v} \quad (3.3)$$

本研究では、様々な製造工程のスケジュールを対象とするために、工程の前後において複数の機械が用いられる三つのケースを考える。

3.3.1 合流

複数の機械によって処理される工程を1台の機械が引き継いで処理する場合を合流と呼ぶ。例えば、複数の部品が1台の機械で組立られる工程が合流にあたる。この場合、machine ζ を製品の組立を行なう機械、machine ξ_1 、machine ξ_2 は半製品の加工を行なう機械とすると、machine ξ_1 とmachine ξ_2 の加工が終了しないと、machine ζ は組立を行なうことができない。合流を表すガントチャートの一例を図3.2に示す。

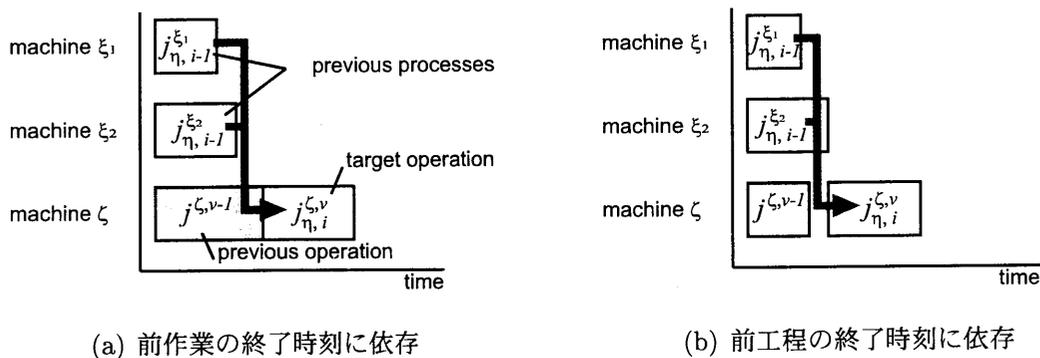


図 3.2: 合流する工程の処理開始時刻

図3.2 から理解できるように、機械は、前工程を処理する機械のうち、工程の受渡し時刻が最も遅い時刻にならないと処理を開始することはできない。作業の処理開始時刻は次式によって求められる。

$$r_{\eta,i,j}^{\zeta} = f_{\eta,i-1}^{\xi_j} \quad (j = 1, 2, \dots) \quad (3.4)$$

$$r_{\eta,i}^{\zeta,v} = \max(\{r_{\eta,i,j}^{\zeta}\}) \quad (3.5)$$

$$s_{\eta,i}^{\zeta,v} = \max(r_{\eta,i}^{\zeta,v}, f^{\zeta,v-1}) \quad (3.6)$$

3.3.2 分岐

1台の機械によって処理される工程が複数の機械によって個別に引き継がれる場合を分岐と呼ぶ。例えば、複数の部品を運んで来て、異なる機械に配る場合が分岐にあたる。この場合、machine ξ を複数の部品を運ぶ搬送装置、machine ζ_1 、machine ζ_2 をそれらの部品を必要とする加工機械とすると、搬送装置が部品を所定の位置に運び終わらないと、machine ζ_1 、machine ζ_2 は処理を始めることができない。分岐を表すガントチャートの一例を図3.3に示す。

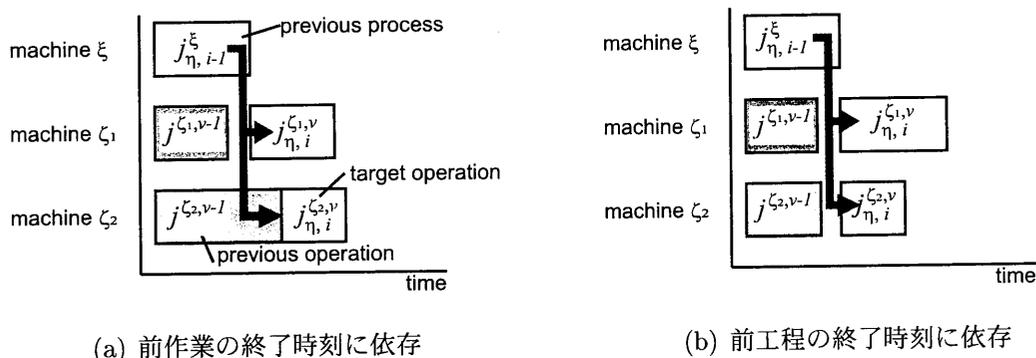


図 3.3: 分岐する工程の処理開始時刻

この場合は、個々の機械については、1対1の関係と考えてよい。作業の処理開始時刻は次式によって求められる。

$$r_{\eta,i}^{\{\zeta_j\}} = f_{\eta,i-1}^{\xi} \quad (j = 1, 2, \dots) \quad (3.7)$$

$$s_{\eta,i}^{\{\zeta_j\},v} = \max(r_{\eta,i}^{\{\zeta_j\}}, f^{\{\zeta_j\},v-1}) \quad (3.8)$$

3.3.3 協調

対象となる工程を複数の機械により同時に処理する作業を協調という。例えば、製品をロボットを用いて機械に取付けるといった連携作業は、この協調にあたる。また本研究課題である作業者は常に機械と強調して作業をする立場にある。協調を表すガントチャートの一例を図3.4に示す。

図3.4から理解できるように、このような作業の処理開始時刻は機械間で常に同じ時刻になる。すなわち、一緒に作業を行なう機械の中で最も遅く作業を開始することができる

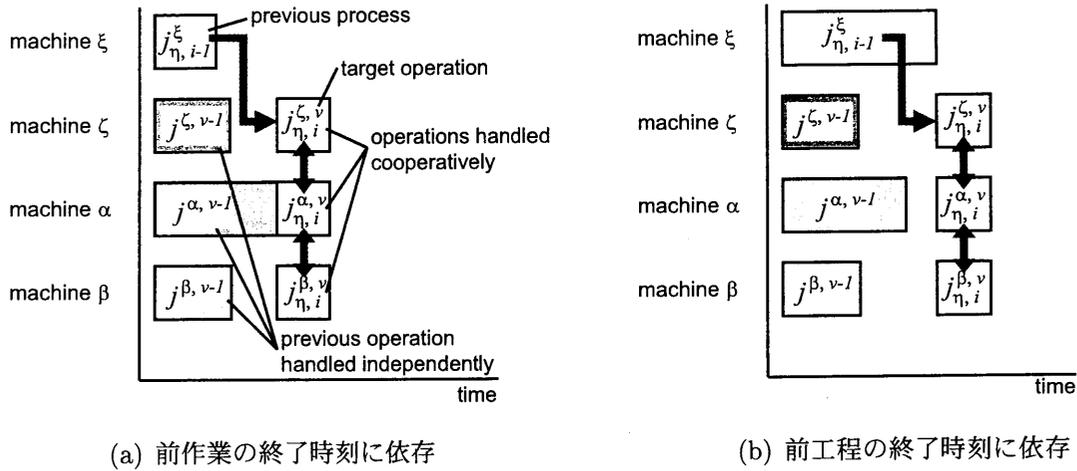


図 3.4: 協調する工程の処理開始時刻

時刻に作業の処理開始時刻を一致させる必要がある。作業の処理開始時刻は次式によって求められる。

$$r_{\eta, i}^{\zeta, v} = f_{\eta, i-1}^{\xi} \tag{3.9}$$

$$\hat{s}_{\eta, i}^{\zeta, v} = \max(r_{\eta, i}^{\zeta, v}, f_{\zeta, v-1}^{\zeta}) \tag{3.10}$$

$$s_{\eta, i}^{\zeta, v} = \max(\hat{s}_{\eta, i}^{\zeta, v}, s_{\eta, i}^{\alpha}, s_{\eta, i}^{\beta}) \tag{3.11}$$

3.4 作業に着目した分析

スケジュールは工程計画を前提として立案されることはすでに述べた通りである。しかし、工程計画によって明確にされない作業についても考慮しなければ、現実的なスケジュールを立案することはできない。本節では、工程計画に付随して処理される作業を中心に分析を行なう。

機械が処理を行なう作業は、工程計画によって定められる作業と、与えられた作業に付随して生じる作業に分類される。このような作業をここでは、前処理 (pre-operation) および後処理 (post-operation) と呼び、前作業との間および次作業との間に行なうことを仮定する。前処理および後処理を含む作業のスケジュール例を図 3.5 に示す。この図に示されるように、一つの作業に対して、複数の前処理と複数の後処理が行なわれる可能性がある。

前処理や後処理は、主体となる作業の処理順序を変えた場合には、同様に処理開始時刻を調整する必要がある。前作業や後作業の処理開始時刻の決定には2通りの状況が考えられる。まずひとつは図 3.6(a) に示すように、先行する作業の処理が終了すると同時に前作業の処理を開始する方法である。この場合の処理開始時刻は、次式によって与えられる。

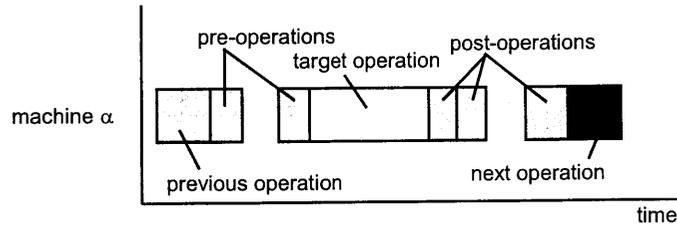
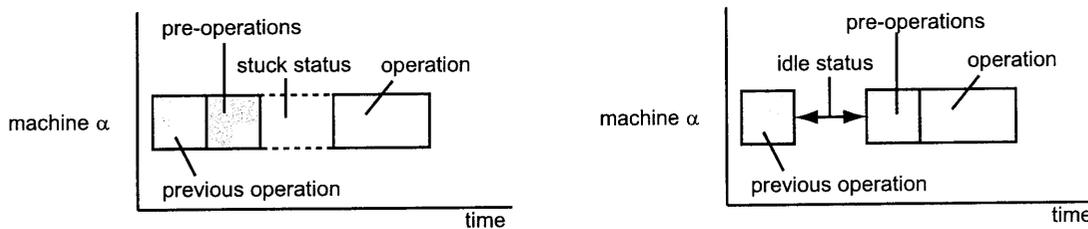


図 3.5: 前処理および後処理を含む作業



(a) 先行する作業の終了時刻に依存

(b) 主体となる作業の開始時刻に依存

図 3.6: 前作業の処理開始時刻

$$\hat{s}_{\eta,i}^{\alpha,v} = f^{\alpha,v-1} \quad (3.12)$$

$$s_{\eta,i}^{\alpha,v} = \max(r_{\eta,i}^{\alpha,v}, \hat{s}_{\eta,i}^{\alpha,v} + \hat{p}_{\eta,i}^{\alpha,v}) \quad (3.13)$$

ここで、 \hat{s} および \hat{p} はそれぞれ、前処理の処理開始時刻および処理時間を表す。

もうひとつの方法は、図 3.6(b) に示すように、主体となる作業の処理開始時刻に間に合うように、前処理の開始時刻を決める方法である。前処理と主体となる作業の間には、別の作業を行なうことができないため、本研究では、この時間を不動時間 (stuck time) と呼び、図 3.6(b) に示される遊び時間 (idle time) と区別して取扱う。この場合は、いったん式 (2.12) および式 (2.13) に従って求めた作業の処理時刻を用いて、前処理の処理開始時刻を次式に従って求め直す。

$$\hat{s}_{\eta,i}^{\alpha,v} = s_{\eta,i}^{\alpha,v} - \hat{p}_{\eta,i}^{\alpha,v} \quad (3.14)$$

前節で考察した工程を受渡す機械間の依存関係は、前処理や後処理に対しても当てはまる。例えば、搬送車によって搬送される製品を、ロボットを用いて機械へ取付ける場合、この作業は前述した複数の機械によって、同時に処理される前処理として取扱われる。

3.5 スケジューリング問題の記述

スケジューリング問題を作成し、計算機により計算を行わせるためには、スケジューリング問題を、計算機が読み込み可能なデータ形式で記述する必要がある。本研究では、スケジューリング問題をXML形式で記述することにより、計算機への読み込みを可能にしている。

3.5.1 XMLとその特徴

XML(Extensible Markup Language)は、汎用的なデータ記述言語である。XMLは文書記述言語としてISOが標準化したSGML(Standard Generalized Markup Language)から派生したものであるが、特にインターネット上でのデータ交換を意識して設計されている。そのため、現在では文書の記述だけでなく、インターネット上で交換可能なあらゆるデータの記述に使われている。XMLがさまざまな分野で利用されている理由は、XMLの持つ特徴がCSV(Comma Separated Value)などといったこれまでのデータ記述形式にはない利点をもたらすからである。XMLの特徴を以下に述べる。

- テキスト形式で記述するため、OSに依存せず様々な環境で利用できる。
- タグの名前を自由に決めることができるため、タグの名前を使って情報内容などを明確に表現することができる。
- タグの階層構造を自由に定義することができる。

スケジューリング問題は、以下の4つの部分により構成されている。すなわち、これらの問題の記述方法を定めることが、ユーザインターフェイスを開発する前段階として重要である。

1. XML宣言：XML文書であることを宣言
2. 地図情報：工場内の地図に関するデータを記述
3. 設備構成：生産を行う設備に関するデータを記述
4. 工程計画：製品の製造方法に関するデータを記述

3.5.2 地図情報の記述

地図情報部分には、工場内の経路情報を記述する。地図情報の記述で使用するタグの一覧を表3.1に示す。

roadmap

表 3.1: 地図情報の記述で使用するタグ一覧

タグ名	概要
<roadmap>	地図情報の記述開始と終了を宣言
<node>	地図上の節点を定義
<connection>	節点間の接続状態を定義

書式 <roadmap> … </roadmap>

説明 地図情報の記述におけるトップレベルのタグである。<roadmap> は地図情報の記述開始、</roadmap> は記述終了を宣言する。

属性 なし

node

書式 <node ref=" 節点番号" x="x 座標" y="y 座標" z="z 座標" type=" 節点の種類" />

説明 地図上の節点を定義する。

属性 ref 属性には、節点番号を指定する。x 属性、y 属性、z 属性には、地図上の座標を指定する。地図の座標は、x-z 座標系であるので、y 属性には"0" を指定する。

connection

書式 <connection start=" 始点節番号" end=" 終点節番号" />

説明 節点と節点の間の接続状態を定義する。

属性 start 属性には、始点の節番号を数字で指定する。end 属性には、終点の節番号を指定する。

地図情報の記述例

図 3.7 は、工場の地図情報を視覚的に表したものである。この工場の地図情報を記述したものを以下に示す。

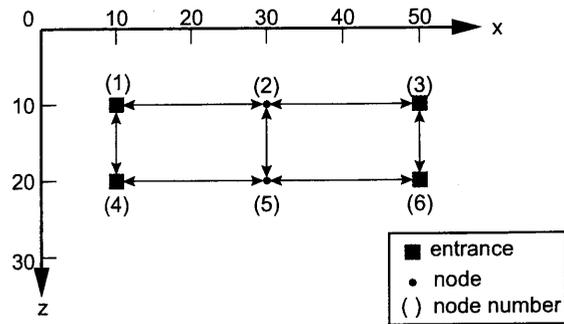


図 3.7: 工場の地図の一例

```

<roadmap>
  <node ref="1" x="10" y="0" z="10" type="entrance" />
  <node ref="2" x="30" y="0" z="10" type="node" />
  <node ref="3" x="50" y="0" z="10" type="entrance" />
  <node ref="4" x="10" y="0" z="20" type="entrance" />
  <node ref="5" x="30" y="0" z="20" type="node" />
  <node ref="6" x="50" y="0" z="20" type="entrance" />
  <connection start="1" end="2" />
  <connection start="1" end="4" />
  <connection start="2" end="1" />
  <connection start="2" end="3" />
  <connection start="2" end="5" />
  <connection start="3" end="2" />
  <connection start="3" end="6" />
  <connection start="4" end="1" />
  <connection start="4" end="5" />
  <connection start="5" end="2" />
  <connection start="5" end="4" />
  <connection start="5" end="6" />
  <connection start="6" end="3" />
  <connection start="6" end="5" />
</roadmap>

```

3.5.3 設備構成の記述

設備構成部分には、生産システムを構成する生産設備に関する情報を記述する。設備構成の記述で使用するタグの一覧を表 3.2 に示す。

表 3.2: 設備構成の記述で使用するタグ一覧

タグ名	概要
<structure>	設備構成の記述開始と終了を宣言
<resource>	生産設備を定義
<type>	生産設備の種類を定義
<ability>	生産設備の処理能力を定義
<map>	生産設備の位置情報を定義

<structure>

書式 <structure> … </structure>

説明 設備構成の記述におけるトップレベルのタグである。<structure> は設備構成の記述開始、</structure> は記述終了を宣言する。

属性 なし

<resource>

書式 <resource name=" 設備番号" port="0"> … </resource>

説明 生産設備を定義する。一つの設備は、<resource> と</resource> に包含される情報によって定義される。

属性 name 属性には、設備番号を指定する。例えば、"machine1" など。port 属性は、TCP/IP に従った通信を行う場合に使用するポート番号を指定する。

<type>

書式 <type ref=" 設備の種類" />

説明 生産設備の種類を定義する。

属性 ref 属性には、設備の種類を指定する。加工機械や棚には"PRODUCER"、搬送車や作業者には"MOVER" を指定する。

<ability>

書式 <ability ref="能力値" />

説明 生産設備の処理能力を定義する。

属性 ref 属性には、設備の処理能力値を指定する。<type> の種類指定が"PRODUCER"であれば処理能力値、"MOVER"であれば移動速度を指定する。

<map>

書式 <map name="ファイル名" location="節点番号" />

説明 生産設備の位置情報を定義する。

属性 name 属性には、地図情報が記述されているファイル名を指定する。location 属性には、設備を配置する地図情報内の節点番号を指定する。

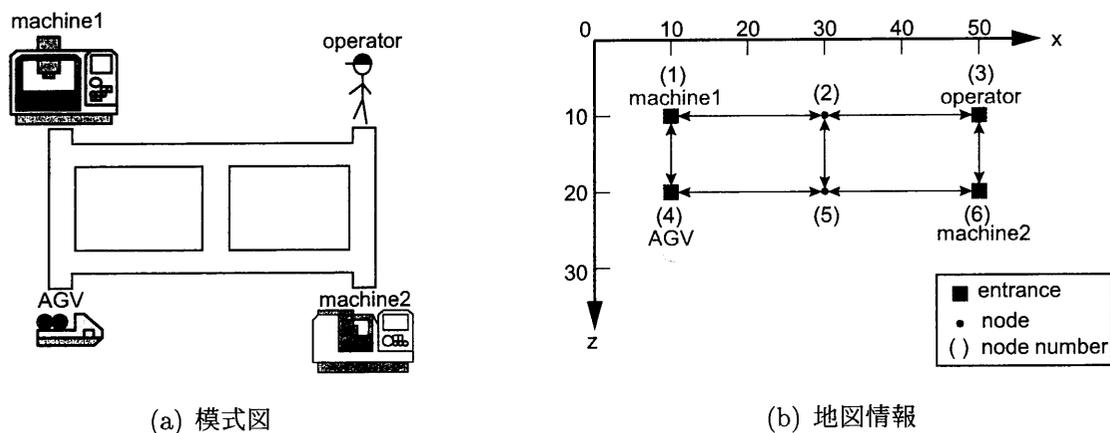
設備構成の記述例

図 3.8: 設備構成の一例

図 3.8 は、加工機械が 2 台、作業者が 1 人、搬送車が 1 台で構成される生産システムの一例を示している。図 3.8(a) は設備構成を模式的に示したものである。また、図 3.8(b) は、地図情報を示している。図 3.8 に示す設備構成を記述したものを以下に示す。

```
<structure>
  <resource name="machine1" port="0" >
    <type ref="PRODUCER" />
    <ability ref="250" />
    <map name="example.map" location="1" />
  </resource>
  <resource name="machine2" port="0" >
    <type ref="PRODUCER" />
    <ability ref="250" />
    <map name="example.map" location="6" />
  </resource>
  <resource name="operator" port="0" >
    <type ref="MOVER" />
    <ability ref="0.01" />
    <map name="example.map" location="3" />
  </resource>
  <resource name="AGV" port="0" >
    <type ref="MOVER" />
    <ability ref="0.01" />
    <map name="example.map" location="4" />
  </resource>
</structure>
```

3.5.4 工程計画の記述

工程計画部分には、製品の製造方法に関する情報を記述する。工程計画の記述で使用するタグの一覧を表 3.3 に示す。

表 3.3: 工程計画の記述で使用するタグ一覧

タグ名	概要
<processplan>	工程計画の記述開始と終了を宣言
<product>	製品を定義
<process>	工程を定義
<resource>	工程の処理を行う設備を定義
<duration>	工程の処理時間を定義
<predecessor>	先行する工程の情報を定義
<successor>	後続する工程の情報を定義
<site>	工程を処理する場所を定義

<processplan>

書式 <processplan version="バージョン"> … </processplan>

説明 工程計画の記述におけるトップレベルのタグである。<processplan> は設備構成の記述開始、</processplan> は記述終了を宣言する。

属性 version 属性には、工程計画のバージョンを指定する。現在のバージョンは"0.0"である。

<product>

書式 <product name="製品番号" /> … </product>

説明 製品を定義する。一つの製品は、<product> と </product> に包含される情報によって定義される。

属性 name 属性には、製品番号を指定する。

<process>

書式 <process name="工程番号" /> … </process>

説明 工程を定義する。一つの工程は、`<process>` と `</process>` に包含される情報によって定義される。

属性 `name` 属性には、工程番号を指定する。

`<resource>`

書式 `<resource name=" 設備番号" />`

説明 工程の処理を行う設備を定義する。

属性 `name` 属性には、工程の処理を行う設備番号を指定する。

`<duration>`

書式 `<duration time=" 処理時間" />`

説明 工程の処理時間を定義する。

属性 `time` 属性には、処理時間を指定する。

`<predecessor>`

書式 `<predecessor ref=" 工程番号" />`

説明 先行する工程の情報を定義する。

属性 `ref` 属性には、先行する工程番号を指定する。先行工程がない場合(第一工程)は、"NONE"と指定する。

`<successor>`

書式 `<successor ref=" 工程番号" />`

説明 後続する工程の情報を定義する。

属性 `ref` 属性には、後続する工程番号を指定する。後続工程がない場合(最終工程)は、"NONE"と指定する。

`<site>`

書式 `<site at=" 設備番号" />`

説明 工程を処理する場所を定義する。

属性 `at` 属性には、工程を処理する設備番号を指定する。

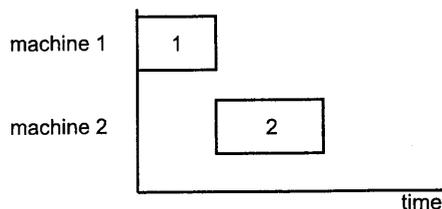


図 3.9: 工程の受渡しが1対1の機械間で行われる場合の工程計画の一例

工程計画の記述例

(1) 工程の受渡しが1対1の機械間で行われる場合

図 3.9 は、2 工程で処理される製品の工程計画をガントチャート上に示したものである。図からわかるように、工程 1(第一工程)は機械 1 で、工程 2(最終工程)は機械 2 で処理される。このとき、工程 1 と工程 2 を処理する機械間の関係は 1 対 1 である。このように、工程の受渡しが 1 対 1 の機械間で行われる場合の工程計画を記述したものを以下に示す。

```

<processplan version="0.0" >
  <product name="1" >
    <process name="1" >
      <resource name="machine1" />
      <duration time="3000" />
      <predecessor ref="NONE" /> ← 第一工程：先行工程なし
      <successor ref="2" />
      <site at="machine1" />
    </process>
    <process name="2" >
      <resource name="machine2" />
      <duration time="4000" />
      <predecessor ref="1" />
      <successor ref="NONE" /> ← 最終工程：後続工程はなし
      <site at="machine2" />
    </process>
  </product>
</processplan>

```

(2) 合流を含む場合

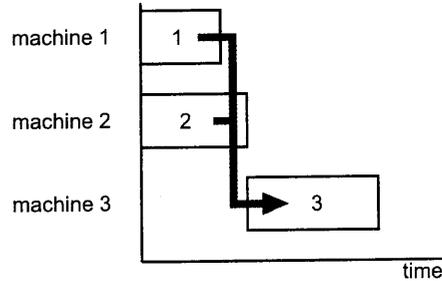


図 3.10: 合流を含む工程計画の一例

図 3.10 は、合流を含む工程計画をガントチャート上に示したものである。図からわかるように、工程 1 と工程 2 の次工程は共に工程 3 であり、合流する工程を示している。このような、合流を含む工程計画を記述したものを以下に示す。

```

<process name="1" >
  <resource name="machine1" />
  <duration time="3000" />
  <predecessor ref="NONE" />
  <successor ref="3" />
  <site at="machine1" />
</process>
<process name="2" >
  <resource name="machine2" />
  <duration time="4000" />
  <predecessor ref="NONE" />
  <successor ref="3" />
  <site at="machine2" />
</process>
<process name="3" >
  <resource name="machine3" />
  <duration time="5000" />
  <predecessor ref="1" />
  <predecessor ref="2" />
  <successor ref="NONE" />
  <site at="machine3" />
</process>

```

(3) 分岐を含む場合

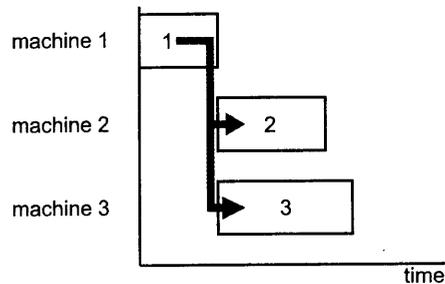


図 3.11: 分岐を含む工程計画の一例

図 3.11 は、分岐を含む工程計画をガントチャート上に示したものである。図からわかるように、工程 1 の次工程は工程 2 と工程 3 であり、分岐する工程を示している。このような、分岐を含む工程計画を記述したものを以下に示す。

```

<process name="1" >
  <resource name="machine1" />
  <duration time="3000" />
  <predecessor ref="NONE" />
  <successor ref="2" />
  <successor ref="3" />
  <site at="machine1" />
</process>
<process name="2" >
  <resource name="machine2" />
  <duration time="4000" />
  <predecessor ref="1" />
  <successor ref="NONE" />
  <site at="machine2" />
</process>
<process name="3" >
  <resource name="machine3" />
  <duration time="5000" />
  <predecessor ref="1" />
  <successor ref="NONE" />
  <site at="machine3" />
</process>

```

(4) 協調を含む場合

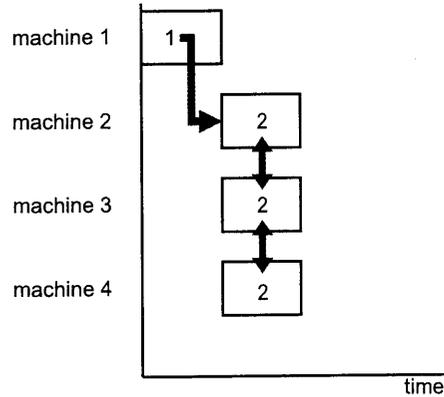


図 3.12: 協調を含む工程計画の一例

図 3.12 は、協調を含む工程計画をガントチャート上に示したものである。図からわかるように、工程 2 を処理する機械は機械 1、機械 2 および機械 3 であり、協調する工程を示している。このような、協調を含む工程計画を記述したものを以下に示す。

```

<process name="1" >
  <resource name="machine1" />
  <duration time="3000" />
  <predecessor ref="NONE" />
  <successor ref="2" />
  <site at="machine1" />
</process>
<process name="2" >
  <resource name=" machine2" />
  <resource name=" machine3" />
  <resource name=" machine4" />
  <duration time="4000" />
  <predecessor ref="1" />
  <successor ref="NONE" />
  <site at="machine2" />
</process>

```

3.6 まとめ

本研究では、作業者の工夫を支援するために、個々の作業に関する計画、特にスケジュールの変更がシステム全体に与える影響を把握するために分散スケジュールの手法を採用している。また、より現実的な生産活動に対応するために、複雑な工程計画を扱う工夫をしている。本章では、そのような複雑な工程計画や生産条件を記述するために、構造化記述言語のひとつである、XML による記述を定義した。実際に、XML で記述された情報を解読し、スケジュールリングソフトウェアが利用するためのプログラムは作成済みである。さらにユーザが入力する際の手間を軽減するためのユーザインターフェイスの開発が必要と考えている。

第4章 直接オフセット法による工具経路生成

4.1 目的

エンドミル加工に代表される回転工具による機械加工を行う場合、工作機械の数值制御装置が扱う工具の移動目標位置と、実際に工具刃先が材料の除去を行う加工点の位置とは一般に工具半径分だけ異なる。そのため、目的の製品形状に対して、一定距離だけ隔てたいわゆるオフセット位置を工具経路とする必要がある。自由曲面のオフセット位置を求めることにはさまざまな問題点があるため、工具経路の生成問題は機械加工を行う場合の重要な問題として認識されている。

自由曲面に対する工具経路生成に対しては、これまでに多くの検討が行われている。これらの研究は、加工面となる製品形状を参照して必要な工具位置を逐次求めるものと、あらかじめ求めておいた製品形状のオフセット面を参照するものとに分類することができる。

たとえば、加工面を逐次参照する方法として、近藤らは反転した工具の先端位置を工具経路とする逆オフセット法を提案している [14]。また、逆オフセット法の問題点を解決する K オフセット法 [15] や、あらかじめ与えた移動経路に沿って工具を移動させ、工具と曲面が干渉する場合に工具を中心軸方向に移動させる方法 [16] などが提案されている。

オフセット面を参照する方法としては、NURBS (Non Uniform Rational B-Spline) 補間によって表現されたオフセット面を利用して工具経路を生成する方法 [17] [18] が提案されている。この方法は加工対象となる製品形状のオフセット面をあらかじめ求めておき、オフセット曲面に沿って工具の移動経路を CNC 装置の内部で生成するものである。

両者の中間的な手法として、Sata [19] らは加工面の代表的な点とその点における法線ベクトルの情報からオフセット点を求め、それらを通過する Bézier 曲線を工具の移動経路とする手法を提案している。

加工面を参照する手法は、短時間に精度の良い工具位置を求めることに注力しており、特に加工中に工具経路を生成することにより、いわゆる”倣い加工”と同じ視点から加工を行うことを試みている [20]~[23]。これらの方法では、離散化された多面体で表現される加工面を対象にするものが多く、許容される誤差と情報量の点から離散化を適切に行なう必要がある [24]。

一方、あらかじめ求めておいたオフセット面を高次の補間式を用いて表現する手法は、少ない情報量で複雑かつ滑らかな自由曲面を表現することが可能なため、CNC 装置への

入力情報が少なくてすむという利点がある。その反面、オフセット面を求めるための特別なCAM(Computer Aided Manufacturing) システムが必要になることに加え、オフセット面を所定の補間式で近似する際に誤差が生じる可能性がある。さらに、使用する工具の径を変えるたびに工具経路であるオフセット面を求め直す必要がある [25]。

いずれの方法においても、制御装置に用いる演算装置の記憶容量と演算速度が大きな制約条件となっている。特に計算量は大きな制約条件として意識され、そのため両方の特徴を組み合わせた方法、すなわち高次の補間式で記述された製品形状を逐次参照する方法はあまり検討されていない。ところが昨今の情報処理技術の進歩には目覚ましいものがあり、特に計算に要する時間は大幅な短縮化が実現されている。また、設計段階で木型や粘土などによる実物大模型(mockup)を用いることは少なくなり [26]、設計の当初から製品形状は電子情報化されていることが多い。その場合、表現可能な形状の豊富さから、NURBS 補間などの高次補間式が多く用いられている。そこで、高次補間式で表現された製品の形状情報をそのまま CNC 装置の入力情報として用いることができれば、製造効率の向上に寄与できると考えられる。

本論文は、NURBS 補間に代表される高次の補間式により表現された製品形状から直接工具制御位置であるオフセット点を求め、同時に加工点の位置情報や法線方向情報などを利用して、加工目的に適した工具経路を生成する方法(直接オフセット法と呼ぶ)を提案する。本報では、提案する手法の基本手順を説明し、加工によって得られる製品形状を数値計算を用いて評価することにより、その特徴について述べる。

4.2 工具指令点と加工点の関係

工具の運動軌跡を制御するために CNC 装置が指令値として利用する空間上の座標と、工具刃先が材料を除去する箇所(箇所)の座標の関係を理解することは、工具経路を生成する上で極めて重要である。図 4.1 は、汎用的なボールエンドミル工具による自由曲面の切削加工を模式的に表している。

ボールエンドミル工具を用いた切削加工の場合、加工点における加工面の法線方向からボールノーズ半径(radius of ball nose)だけ離れた位置を工具の運動指令値として用いる。この指令値は工具上では、図 4.1 に示されるように、加工点からボールノーズ半径分だけ離れた工具軸中心上にある。なお、図 4.1 は、加工点における法線を含む任意の平面内で一定角度だけ工具回転軸を傾けた状態を示している。

工具の運動指令値の集合は、加工面である製品形状と至るところで一定距離だけ隔てられたオフセット曲面を形成する。本論文では、次式で与えられるオフセット曲面 [27] を取り扱う。

$$S^{\circ}(u, v) = S(u, v) + r \cdot P(u, v) \quad (4.1)$$

ここで、 $S(u, v)$ は、媒介変数 (u, v) に対応する曲面上の一点を表す位置ベクトルであ

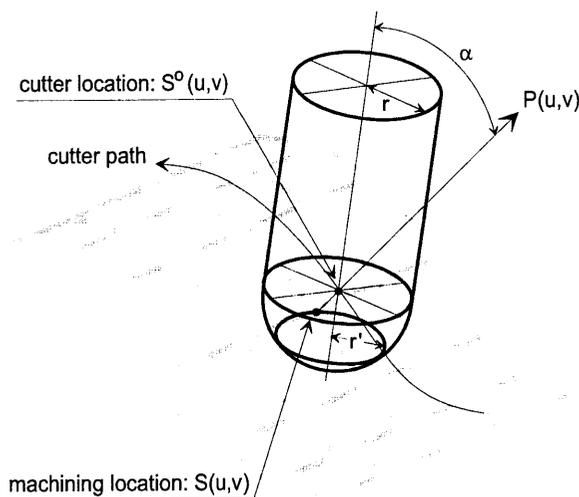


図 4.1: Ball end milling of free form surface

り, $P(u, v)$ は, $S(u, v)$ における単位法線ベクトル, $S^o(u, v)$ はオフセット曲面上の位置ベクトルを表している. r は, オフセット量であり, ここではボールノーズ半径に相当する.

NURBS 補間に代表される高次補間式によって自由曲面を表現する手法は, 対象とする形状を多面体の集まりで表現する手法に比べ, 扱わなくてはならない情報量の少なさと形状精度の高さの点で優れている. その一方で, 計算には多くの乗除算を含む実数演算が必要になり, 演算速度が問題になる場合には用いることができない. 実際, CNC 内部に専用の補間演算を行う機構を設けている場合でも, 入力として, 製品形状を表す情報ではなく, 工具が通過するオフセット形状を表現する情報が用いられる [17] [18]. あらかじめ求めたオフセット面を利用する手法の問題点 [25] は冒頭で触れているが, ここではさらに工具指令点と加工点の立場からの問題点を以下に指摘する.

図 4.2 に, 凹型および凸型の形状に対して, 工具の制御位置であるオフセット位置と加工点の関係を模式的に示す. ここでは簡単のために, 2次元平面上の自由曲線とそのオフセット曲線の関係を示しているが, 3次元の自由曲面の場合にも同様のことが言える. 工具を一定の送り速度で移動させる場合には, 一般にこのオフセット位置での工具制御位置が等間隔で移動することになる. 図中 c_i は, CNC 装置が指令値として与える工具位置を一定の時間ごとに示している. また, c_i を工具が通過するときの加工面上の工具刃先位置を m_i により示している. 図 4.2 では, 工具径の位置を破線の円により表示してある. この円は, 3次元の自由曲面の場合, 工具のボールノーズ半径に相当する. またオフセット位置上の接線および法線方向も合わせて表示してある.

図 4.2 から理解できるように, 一定の速度で工具指令位置がオフセット位置上を通過する場合, 加工点は加工面上を不均一な速度で通過する. たとえば, 凹形状の金型を想定した場合, 凹部の曲率の大きい箇所では, 周辺部に比べ工具 1 刃当たりの送り量が大きくなり, 工具送り方向の削り残しに因る粗さが大きくなる [28] ことが考えられる. 逆に, 凸形

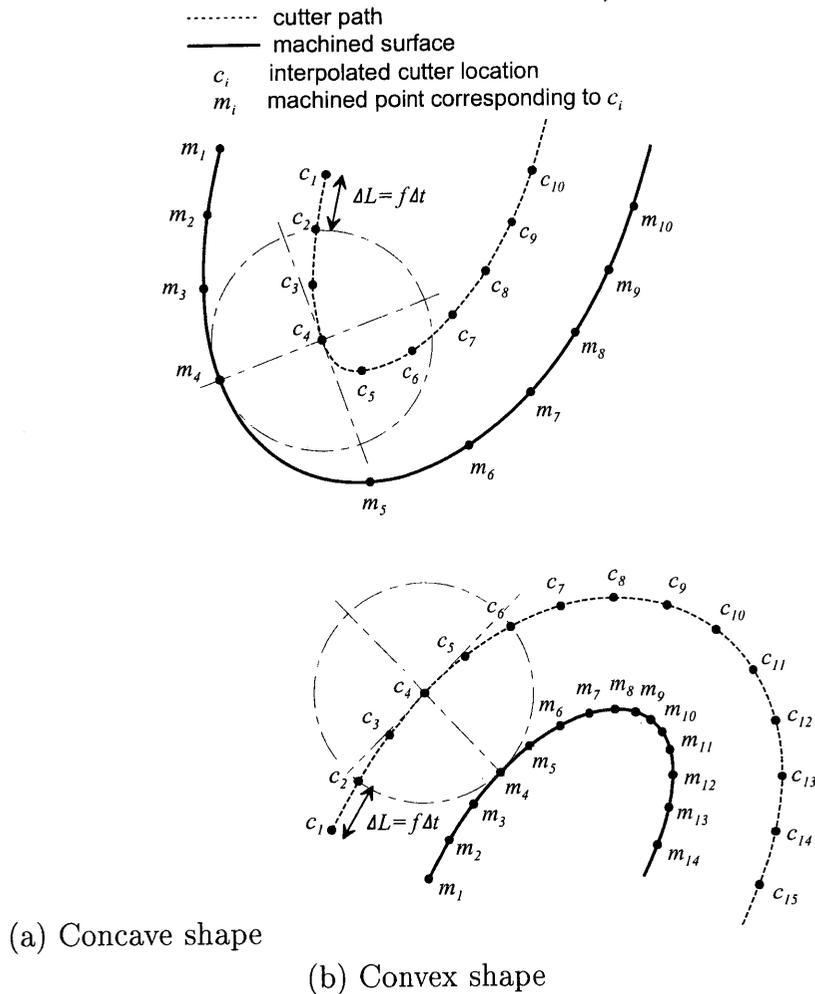


図 4.2: Relationship between cutter locations interpolated on offset surface and corresponding machining points

状の部分では送り速度が小さくなり、切り取り厚さが減少する。このことは、加工硬化が生じやすい材料に対して少ない送り量を避ける必要がある場合などに問題になる。

4.3 直接オフセット法の提案

工具指令位置としてあらかじめ求めておいたオフセット曲面を用いる方法は、前述のように多くの欠点を伴う。製品の曲面形状の表現方式として高次の補間式を用いた場合に、オフセット曲面を指令値とする理由は、前節でも述べたように CNC 装置における演算時間に因ることが考えられる。しかし、演算装置が持つ処理能力の向上に加え、計算手順を工夫する(付録参照)ことにより、演算時間の問題は、十分に解決できることが期待できる。そこで、本論文では、製品形状を表すパラメータの情報を直接 CNC 装置への入力情報

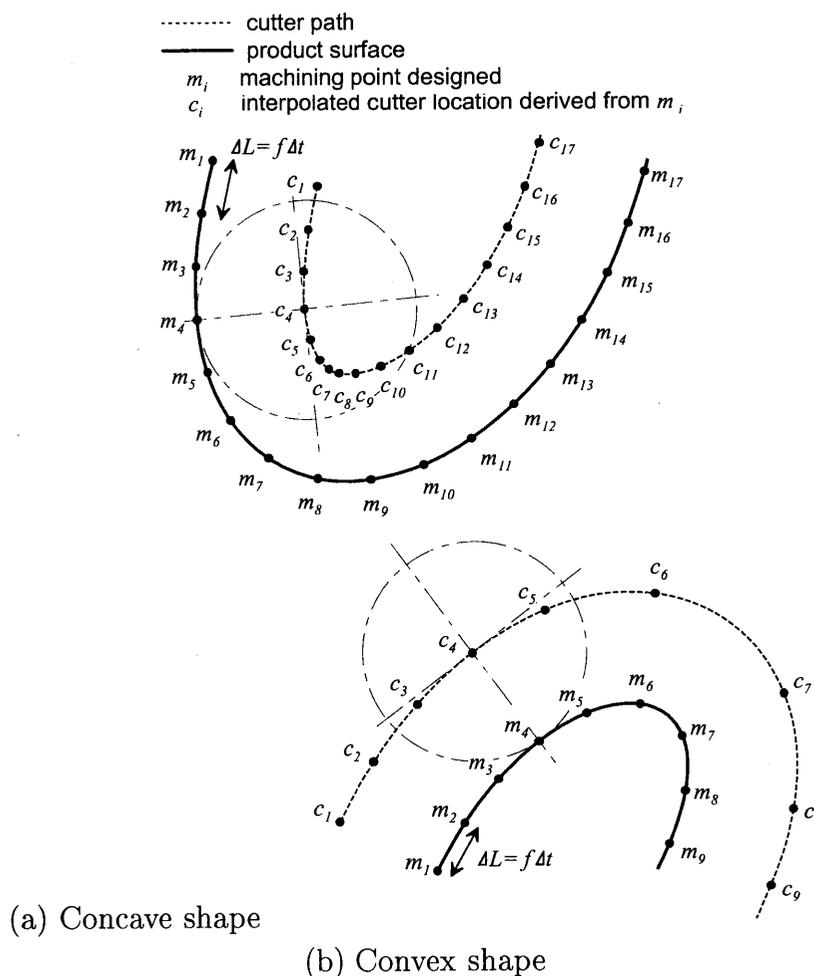


図 4.3: Relationship between cutter locations and corresponding machining points determined by proposed method

報とし、工具指令位置を生成する方法を提案する。

図 4.3 は、工具の指令位置と加工位置の関係を模式的に示している。本手法では、CNC 装置への入力情報として、直接加工面の位置情報を与えることを前提としているため、加工面上を一定の速度で通過するように加工位置を決めるのに適している。この場合、曲率に応じて工具指令位置の送り速度が変化することになる。

なお、加工前にあらかじめ求めたおいたオフセット面上で工具を一定の速度で移動させる場合でも、工具の回転数を変化させることにより、加工点における工具一刃辺りの切り取り厚さを一定にすることが可能である。しかしそのためには、オフセット面での工具位置に対応する加工点の位置情報が必要になり、結局、計算負荷を少なくできる利点がない。

図 4.4 に、高次の補間式として NURBS 補間により表現されたオフセット形状を利用する従来法と、本論文で提案する手法のそれぞれについて必要となる手順を模式的に示

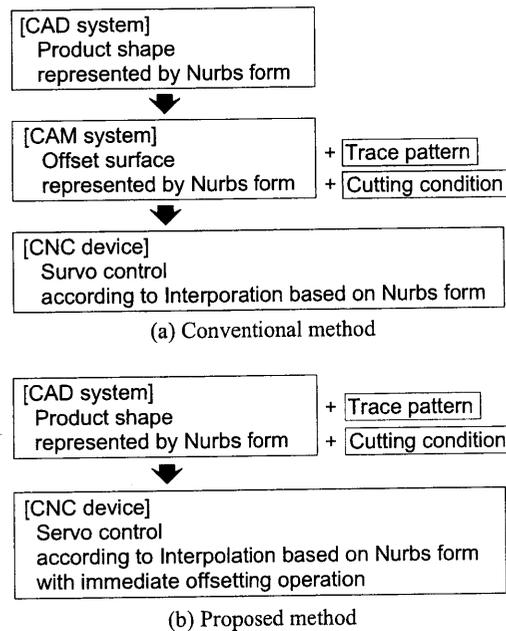


図 4.4: Diagram of procedure for machining of free form surface

す。加工を行なうためには、加工面に対して工具の走査形態 (trace pattern) や加工条件 (cutting condition) などの情報が必要となる。前者は等高線上での工具の移動や直線上での一方向の工具移動の様式などを指定するものである。また後者は、主軸回転数や送り速度などの条件を意味する。これらの情報がCAMシステムへの入力とされる場合もあるが、ここではいずれも工作機械の数値制御装置への入力情報として扱っている。ここで対象とする2通りの方法の異なる点は、CAMシステムの有無とCNC装置内部における処理にある。あらかじめオフセット位置を求める従来法においては、CAMシステムによるオフセット面形状を求めることが必要となるだけでなく、CNC装置内部においてもNURBS補間に基づいた処理が求められる。一方、提案する手法は、CAMシステムによる加工前の処理を必要としないかわりに、NURBS補間に基づいた補間時において直接オフセット位置を求める処理が必要になる。

加工面の位置情報に基づいて、工具の位置を制御する本手法の手順は次のとおりである。時刻 t における加工点の位置を与える媒介変数の値を (u, v) とすると、時間 Δt 後の加工位置および工具指令位置は次の手順により求めることができる。ここで、加工点における工具刃先の送り速度を f とする。また $|S(u', v') - S(u, v)|$ を2点 $S(u, v)$ および $S(u', v')$ 間の距離とする。

Step 1 $|S(u', v') - S(u, v)| = f\Delta t$ を満たす $S(u', v')$ を求める。

Step 2 $S(u', v')$ に対するオフセット点、 $S^o(u', v')$ を式 (4.1) に従って求める。

Step 3 工具を座標 $S^o(u', v')$ に移動させる.

この方法は、加工前にオフセット曲面を求める必要が無く、以下の特徴があげられる.

- (1) CAD システムからの情報を直接 CNC 装置への入力として利用することが可能 (オフセット曲面を求めるための特別な CAM システムが不要).
- (2) 使用する工具のボールノーズ半径を変更しても、CNC 装置に入力する加工形状情報の修正が不要.
- (2') 工具摩耗などによる工具径の変化に対して、工具径補正の機能を CNC 装置に付与することが可能.
- (3) オフセット曲面を求める際、切り込み方向への計算誤差が無いいため、製品形状の幾何精度を保つことが可能.

金型によって成形される製品は、通常一方向へ抜き取られる。このことは、その方向からの工具の進入が可能であることを意味する。従って、金型で対象となる形状は、ほとんどが3軸のフライス盤によって加工可能である自由曲面である。しかし、3軸のフライス盤によって加工を行った場合、製品の加工場所によって、加工に関与する工具刃先の位置が変化する。そのため、刃先の回転半径が変化し、結果として切削速度を一定に保つことができなくなる。すなわち加工点における切削速度は、工具の回転速度および送り速度だけでなく、工具姿勢の影響も受けることになる。この場合、条件によっては表面粗さが変化したり、工具寿命が短くなったりする [29]。しかし、多軸のフライス盤を用いて工具軸の方向を制御することができる場合には、加工に用いる刃先の位置を制御することが可能になる。この場合、削り残しによる加工誤差 [28] [30] を低減することや、加工面に形成される凹凸模様を制御する [31] ことが可能である。このような工具の姿勢を制御するためには、加工点における法線方向の情報が不可欠である。本手法はオフセット位置を求める際に、加工点の位置情報に加え、加工点における法線方向の情報を算出しなくてはならないが、この情報は、工具姿勢の制御を行うために用いることができる。これらのことから次の特徴をあげることができる。

- (4) オフセット点の情報を求める際に求められる加工点での法線情報は、工具姿勢の制御を行なうための情報として利用可能.
- (5) 加工面上での実質的な切削速度を制御することが可能.

4.4 数値計算例

本研究で用いる工具経路生成法の妥当性を確認するために、計算機シミュレーションによる検証を行なう。ここでは4つの制御点から与えられるNURBS曲線を対象とし、この

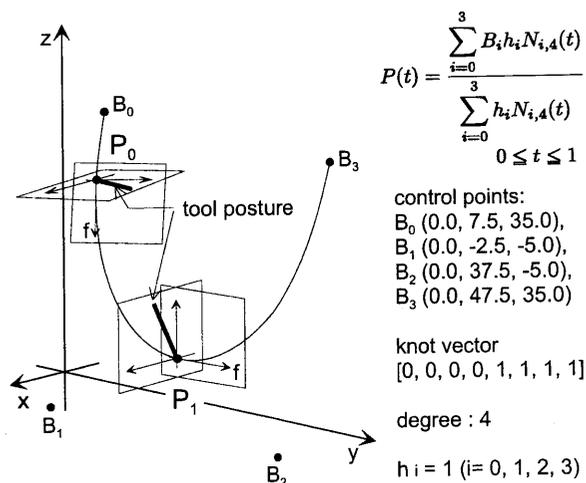


図 4.5: Cutting pass and evaluated locations

表 4.1: Cutting conditions

Tool radius R	5 mm
Helix angle	0 °
Number of teeth	2
Feed rate	0.25 mm/tooth
Pick feed	0.5 mm
Tilting angle	20 °

曲線を一方向に掃引してできる曲面の加工を考える。図 4.5 に、工具 1 工程で加工する形状を模式的に示す。このとき図に示すように、加工点における工具送り方向に垂直な平面内において、法線方向と常に一定の角度を維持するように工具の傾き角度 (tilting angle) を制御する。ここでの解析は、自由曲線上を工具が姿勢を変化させながら移動する点を除き、是田らによる研究 [28] と同じ手法である。また、製品形状に含まれる曲率を十分に加工することができる径の工具を使用する [33] ものとし、加工点以外での工具と被削材の干渉については考慮しない。計算に使用した加工条件を表 4.1 に示す。

図 4.6 に、オフセット曲線上を工具が等速度で移動する場合の計算結果を示す。ここでは、図 4.5 に示す曲率の異なる点 P_0 および点 P_1 における加工面の形状を示し、図中には、解析結果から求めた工具送り方向に生じる凹凸の間隔を併記している。図 4.6(a) および (b) には、製品形状情報を含めた加工面の様子を示し、同図 (c) および (d) は、製品形状の情報を取り除き、表面粗さを求めた結果を示す。このとき凹凸の最大高さを、切削送り方向およびピックフィード方向に分けて評価した結果を併せて記している。

図 4.6 から、比較的曲率の小さい点 P_0 においても加工面上では、設定値と異なる送り

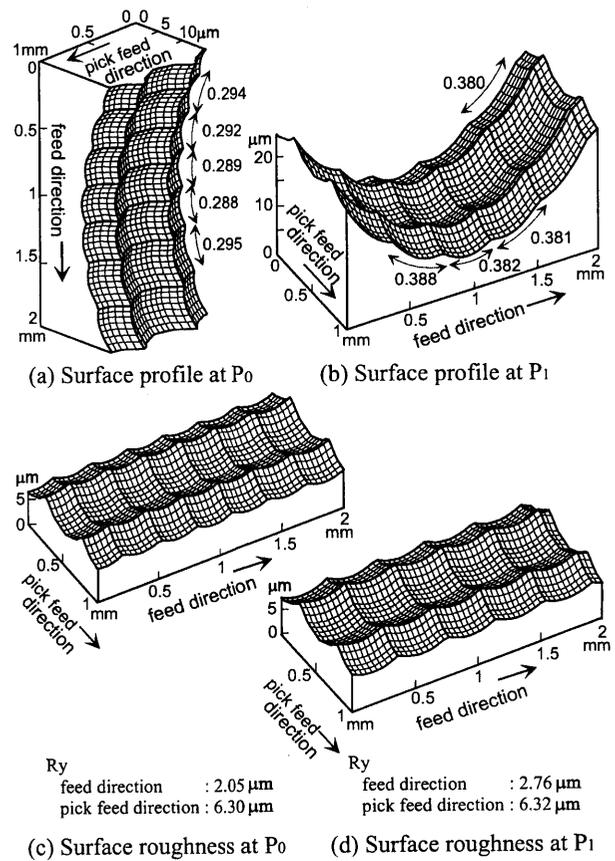


図 4.6: Calculation results of surface profile and roughness machined with use of ball end mill moving on offset positions at constant feed speed

量で加工されることになり、さらに曲率が大きくなる点 P_1 においては、約 1.5 倍の送り速度で加工していることがわかる。このように、あらかじめ求めておいたオフセット曲面を工具の指令値として CNC 装置への入力値とする場合、異なった送り速度で加工が行われ、均質な加工面粗さを得ることができないことが確認できる。

図 4.7 は、製品形状を CNC 装置への入力情報とした場合に得られる加工面の様子を算出した結果を示している。前章で述べたように、この手法では、製品面上において単位時間ごとに等距離になる座標を求め、対応するオフセット位置を工具が移動する。図から理解できるように、曲率が変わっても、設定通りの間隔で送り方向に凹凸が生じていることが確認できる。

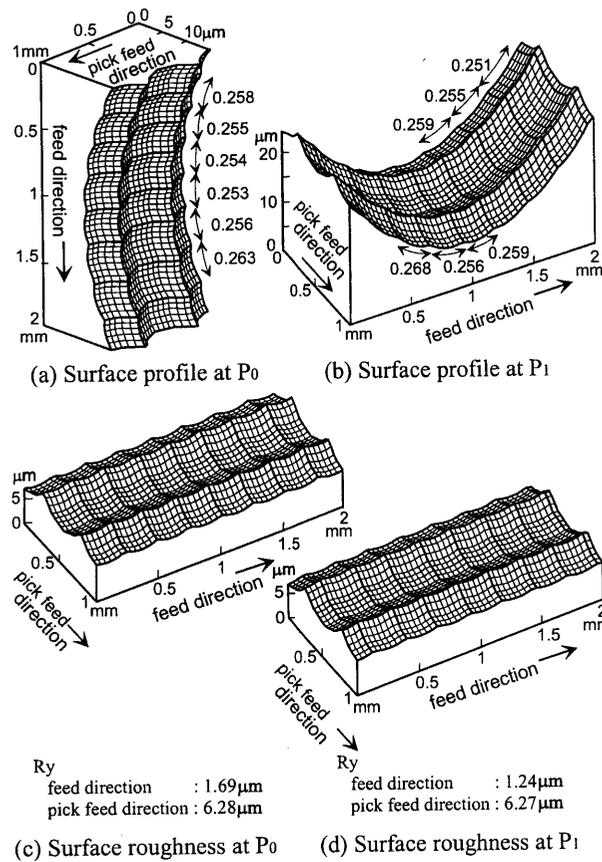


図 4.7: Calculation results of surface profile and roughness machined with use of ball end mill moving on product shape positions at constant feed speed

4.5 まとめ

自由曲面をボールエンドミルによって加工する際に、NURBSに代表される高次の補間式によって表現される製品形状をそのままCNC装置への入力値として用いる直接オフセット法と名付けた手法を提案した。本論文では、あらかじめ求めておいたオフセット形状を用いる従来の手法に比べ、提案する手法が多くの特長をもつことを明らかにした。特に、加工面性状に関しては、計算機シミュレーションによる解析を行ない、その優位性を確認した。今後、提案する手法を用いた数値制御装置の開発を行なう予定である。

第5章 付録

5.1 NURBS 曲面

用語の説明

$N_{i,k}(u)$	k 位の i 番目の基底関数
$M_{j,l}(v)$	l 位の j 番目の基底関数
x, y	u 方向および v 方向のノットベクトル
h	同次座標ベクトル
B_{ij}	定義多角形頂点の位置ベクトル
d	オフセット量
$S(u, v)$	パラメータ (u, v) に対する曲面上の 1 点を表す位置ベクトル
$P(u, v)$	$S(u, v)$ における単位法線ベクトル
$S^0(u, v)$	$S(u, v)$ に対するオフセット曲面

オフセット曲面 と NURBS 曲面

点 (u, v) における x 方向, および y 方向の曲面 $S(u, v)$ 上の接線ベクトルはそれぞれ, $\vec{t}_u(1, 0, \frac{\partial}{\partial u}S(u, v))$, $\vec{t}_v(0, 1, \frac{\partial}{\partial v}S(u, v))$ により与えられる. これら 2 つの接線ベクトルの外積をとって, 法線ベクトル \hat{P} は以下のように与えられる.

$$\begin{aligned} \hat{P}(u, v) &= \det \begin{bmatrix} i & j & k \\ 1 & 0 & \frac{\partial}{\partial u}S(u, v) \\ 0 & 1 & \frac{\partial}{\partial v}S(u, v) \end{bmatrix} \\ &= -\frac{\partial}{\partial u}S(u, v)i - \frac{\partial}{\partial v}S(u, v)j + k \end{aligned} \quad (5.1)$$

これより, 単位法線ベクトルは次式で与えられる.

$$P(u, v) = \hat{P}(u, v) / |\hat{P}(u, v)| \quad (5.2)$$

一方, NURBS 曲面は次式で与えられる [32].

$$S(u, v) = \frac{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{ij} h_{ij} N_{i,k}(u) M_{j,l}(v)}{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{ij} N_{i,k}(u) M_{j,l}(v)} \quad (5.3)$$

$$x_{min} \leq u < x_{max}, \quad 2 \leq k \leq n+1$$

$$y_{min} \leq v < y_{max}, \quad 2 \leq l \leq m+1$$

ここで、 $N_{i,k}(u)$ および $M_{j,l}(v)$ は、基底関数あるいは混ぜ合わせ (Blending) 関数と呼ばれ、Cox-DeBoor の帰納式によって次式で与えられる。

$$N_{i,k}(u) = \frac{u - x_i}{x_{i+k-1} - x_i} N_{i,k-1}(u) + \frac{x_{i+k} - u}{x_{i+k} - x_{i+1}} N_{i+1,k-1}(u)$$

$$N_{i1}(u) = \begin{cases} 1 & (x_i \leq u < x_{i+1}) \\ 0 & (\text{それ以外の場合}) \end{cases} \quad (5.4)$$

$$M_{j,l}(v) = \frac{v - y_j}{y_{j+l-1} - y_j} M_{j,l-1}(v) + \frac{y_{j+l} - v}{y_{j+l} - y_{j+1}} M_{j+1,l-1}(v)$$

$$M_{j1}(v) = \begin{cases} 1 & (y_j \leq v < y_{j+1}) \\ 0 & (\text{それ以外の場合}) \end{cases} \quad (5.5)$$

単位法線ベクトルを求める際に必要な偏微分 $\frac{\partial}{\partial u} S(u, v)$, および $\frac{\partial}{\partial v} S(u, v)$ は、式 (5.3) を形式的に u あるいは v によって偏微分することによって得られる。

5.2 NURBS 面に対するオフセット点の高効率な計算法

NURBS 曲面を取り扱う多くの文献で、NURBS 曲面を用いる際に必要となる基底関数の計算式として Cox-DeBoor の帰納式が示されている。プログラムの作成上は、関数の再帰呼び出しによって実装することができ、簡潔に記述することができるという利点がある。しかし、再帰呼び出しは、関数呼び出し毎に局所的な記憶領域が消費されるため、計算の際に要求される記憶領域が一般的な繰り返し計算に比べ大きくなる。さらに関数呼び出しが繰り返し行われるため、処理時間の点からは不利であることが知られている。

ここでは、Cox-DeBoor の帰納式に従って計算を行う場合に、再帰呼び出しを行わずに、単純な繰り返し計算によって混ぜ合わせ関数を求める方法について整理する。特に、NURBS 曲面を計算する場合には、同位の連続する複数の基底関数の値を計算する必要が

ある。この際重複する計算を省略することで、計算に要する時間を大幅に短縮することが可能になる。

なお、例題としては、3位の有理化されていない非一様な B-Spline 曲線によって定義される曲線取り扱うが、基底関数および定義多角形頂点の定義から判断できるように、ここで述べる方法は、容易に n 位の NURBS 表現された自由曲面に拡張が可能である。

有理化されていない B-Spline 曲線は次式で与えられる。

$$P(u) = \sum_{i=1}^{n+1} B_i N_{i,k}(u)$$

ここで、 B_i は、 $N+1$ 個の定義多角形頂点の位置ベクトル、 $N_{i,k}$ は、正規化された B-Spline 基底関数であり、NURBS 自由曲面と同様に、式 (5.4) により与えられる。

さて、媒介変数 u (ただし、 $x_i \leq u < x_{i+1}$) に対して、曲線上の 1 点 $P(u)$ を求めることを考える。式 (5.4) より、有効な値をもつ 2 位の基底関数は、 $N_{i-1,2}(u)$ と $N_{i,2}(u)$ の 2 つだけである。具体的には、値 u に対して次式で求められる。

$$N_{i-1,2}(u) = \frac{x_{i+1} - u}{x_{i+1} - x_i}, \quad N_{i,2}(u) = \frac{u - x_i}{x_{i+1} - x_i}$$

この結果から、有効な値を持つ 3 位の B-スプライン基底関数は、 $N_{i-2,3}(u)$ 、 $N_{i-1,3}(u)$ 、 $N_{i,3}(u)$ の 3 つだけになり、次式で求められる。

$$\begin{aligned} N_{i-2,3}(u) &= \frac{x_{i+1} - u}{x_{i+1} - x_{i-1}} N_{i-1,2}(u), \\ N_{i-1,3}(u) &= \frac{u - x_{i-1}}{x_{i+1} - x_{i-1}} N_{i-1,2}(u) + \frac{x_{i+2} - u}{x_{i+2} - x_i} N_{i,2}(u), \\ N_{i,3}(u) &= \frac{u - x_i}{x_{i+2} - x_i} N_{i,2}(u). \end{aligned} \quad (5.6)$$

結局、3 階の NURBS 曲線の場合、 $x_i \leq u < x_{i+1}$ の範囲に対しては以下の計算を行うだけで良い。

$$P(u) = B_{i-2} N_{i-2,3}(u) + B_{i-1} N_{i-1,3}(u) + B_i N_{i,3}(u)$$

実際の計算においては次のように適当な一時変数を用いることで、必要な除算の回数を減らすことができる。

たとえば、式 (5.6) の計算過程においては以下のような演算を行うと良い。

$$temp_1 = \frac{N_{i,2}(u)}{x_{i+2} - x_i}, \quad temp_2 = \frac{N_{i-1,2}(u)}{x_{i+1} - x_{i-1}}$$

一方、式 (5.4) に従って高位の基底関数の値から順次求めていく場合には、結果的にゼロになる値に対しても相当数の乗除算を行うことになる。ここで示したように低位の基底関数の値から順次高位基底関数の値を求めることにより、このような無

表 5.1: Examples of calculation time required

	coordinates on surface	coordinates on offset
Cox-deBoor	96.0 μ sec	368 μ sec
Evaluated method	5.28 μ sec	11.2 μ sec

CPU: Ultra SPARC II 300MHz(SUN Microsystems, Inc.)

無駄な計算をあらかじめ排除することになり、より高速な演算が可能になる。

この手順による計算は、本論文で利用するオフセット座標の計算のときに、特に有効である。法線を求める際には、各点における接線ベクトルを求める必要があり、そのためには、基底関数の1次導関数を求める必要がある。1次導関数を求める際には、1つの基底関数の値が下位の4つの基底関数から求められることになり、さらに多くの演算が必要になる。しかし、同様の手法によって計算に必要な除算を減らすことができるだけでなく、必要な値の大部分は、曲面上の座標を計算する際に求めた基底関数の値を利用することができる。

表 5.1 に 4 位の NURBS 自由曲面上の 1 点の座標およびオフセット点の座標を求めるために要する時間を実測した結果を示す。計算に要する時間は、使用する計算機の性能および実装の仕方によって異なるため、参考値に過ぎないが、Cox-deBoor の帰納式に従った場合に比べ、短時間での計算が可能であることが確認できる。

5.3 工具刃先のモデル

ここでは、ボールエンドミルを用いた切削シミュレーションの方法について説明する。

5.3.1 基本モデル

ねじれ角を考慮しない工具を基本モデルとする。N 枚刃の工具の基本モデルを考えた時、工具座標系 $O-X_t Y_t Z_t$ において工具の切れ刃のうちの 1 つを $X_t Z_t$ 平面にあるものとし、これを 1 番目の切れ刃とする。i 番目の切れ刃曲線上の任意の点を A (ボール部分)、B (シリンダー部分) とすると A, B の座標は OA と Z_t 軸のなす角を θ_1 、工具半径を R、B の z 座標 Z_1 を使って以下のように表される。例として図 5.1 に 2 枚刃の基本モデルを示す。

$$\begin{aligned}
 A &= \{ R \sin \theta_1 \cos(2\pi(i-1)/N) \quad R \sin(2\pi(i-1)/N) \quad -R \cos \theta_1 \} \\
 B &= \{ R \cos(2\pi(i-1)/N) \quad R \sin(2\pi(i-1)/N) \quad Z_1 \}
 \end{aligned}
 \tag{5.7}$$

b. 切れ刃曲線の導出

図5.3, 5.4に工具座標系と切れ刃の位置の定義を示す. 図3.4.3(a)にあるように切れ刃曲線のボール部分における任意の点をA'とし, OA'と X_t 軸のなす角を θ_2 , A'のz座標を Z_2 , とするとA'における位置ベクトル \vec{r} は以下ようになる.

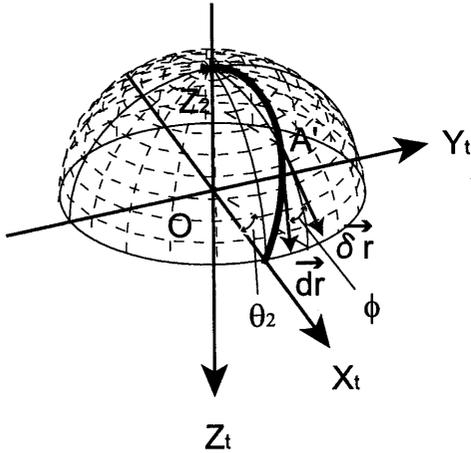


図 5.3: ボール部分の工具座標系

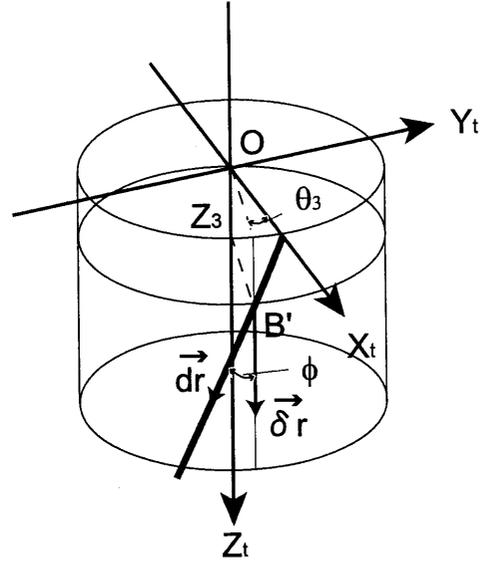


図 5.4: シリンダー部分の工具座標系

$$\vec{r} = \left(\sqrt{R^2 - Z_2^2} \cos \theta_2 \quad \sqrt{R^2 - Z_2^2} \sin \theta_2 \quad Z_2 \right) \quad (5.8)$$

\vec{r} をz方向、 θ_2 方向にそれぞれ微分したものを \vec{r}_z, r_{θ_2} とすると,

$$\vec{r}_z = \frac{\partial \vec{r}}{\partial z} = \left(\frac{-Z_2 \cos \theta_2}{\sqrt{R^2 - Z_2^2}} \quad \frac{-Z_2 \sin \theta_2}{\sqrt{R^2 - Z_2^2}} \quad 1 \right)$$

$$r_{\theta_2} = \frac{\partial \vec{r}}{\partial \theta_2} = \left(-\sqrt{R^2 - Z_2^2} \sin \theta_2 \quad \sqrt{R^2 - Z_2^2} \cos \theta_2 \quad 0 \right)$$

ここで, 切れ刃の接線ベクトル $\vec{d}\vec{r}$ と経線の接線ベクトル $\vec{\delta}\vec{r}$ を考えると,

$$\vec{d}\vec{r} = \vec{r}_z dz + r_{\theta_2} d\theta_2, \quad \vec{\delta}\vec{r} = \vec{r}_z \delta z$$

$\vec{d}\vec{r}$ と $\vec{\delta}\vec{r}$ の内積を考えて,

$$\vec{dr} \cdot \vec{\delta r} = |\vec{dr}| \cdot |\vec{\delta r}| \cos \phi$$

これより,

$$d\theta_2 = \frac{R}{R^2 - Z_2^2} \tan \phi dz$$

$$\int d\theta_2 = R \tan \phi \int_0^z \frac{dz}{R^2 - Z_2^2} = \frac{1}{2} \tan \phi \int_0^z \left(\frac{1}{R + Z_2} + \frac{1}{R - Z_2} \right) dz$$

$$\theta_2 = \frac{1}{2} \tan \phi \ln \frac{R + Z_2}{R - Z_2} \quad (5.9)$$

式 (5.8), (5.9) より, ボール部分の切れ刃曲線が得られる.

また, シリンダー部分においても図 5.4 にあるように切れ刃曲線のシリンダー部分における任意の点を B' とし, OB' と X_t 軸のなす角を θ_3 , B' の z 座標を Z_3 とすると位置ベクトル \vec{r}_2 は以下のようなになる.

$$\vec{r}_2 = (R \cos \theta_3 \quad R \sin \theta_3 \quad Z_3) \quad (5.10)$$

ボール部分と同様に θ_2 を求めると,

$$\theta_2 = \frac{Z_3}{R} \tan \phi \quad (5.11)$$

式 (5.10), (5.11) より, シリンダー部分の切れ刃曲線も求められる.

式 (5.7)~(5.11) より, ねじれ角を考慮した切れ刃上の点 $P_t(x, y, z)$ は, 工具座標系 $O-x_t y_t z_t$ において以下のように表される.

(藤) ボール部分

$$\begin{aligned} x &= R \sin \theta_1 \cos(\theta_2 + 2\pi(i-1)/N) \\ y &= R \sin \theta_1 \sin(\theta_2 + 2\pi(i-1)/N) \\ z &= -R \cos \theta_1 \end{aligned} \quad (5.12)$$

ただし,

$$\theta_2 = \frac{1}{2} \tan \phi \ln \frac{1 - \cos \theta_1}{1 + \cos \theta_1}$$

(討) シリンダー部分

$$\begin{aligned} x &= R \cos(\theta_3 + 2\pi(i-1)/N) \\ y &= R \sin(\theta_3 + 2\pi(i-1)/N) \\ z &= Z_3 \end{aligned} \quad (5.13)$$

ただし,

$$\theta_3 = \frac{Z_3}{R} \tan \phi$$

式(5.12),(5.13)をもとに描かれた1枚刃工具モデルを図5.5に示す.

5.4 制御軸の変化量

5.4.1 直進駆動軸

ここでは、直進駆動軸の変化量の算出方法について述べる。図5.6にテーブル上の機械座標系 $O-X_m Y_m Z_m$ を示す。各直進駆動軸の位置は、加工点 p が指定された工具姿勢をとるためにテーブルが回転、傾斜することによって移動した点になる。したがって、テーブルの回転、傾斜後の点 p の位置を算出することで、各直進駆動軸の位置を算出することができる。以下では、テーブルの回転、傾斜後の点 p の位置を算出する方法について述べる。

図5.7に図5.6の Z_m 軸の正方向から見た図を示す。加工点 p が C 軸の回転により c だけ回転した点を p' とすると p' は、

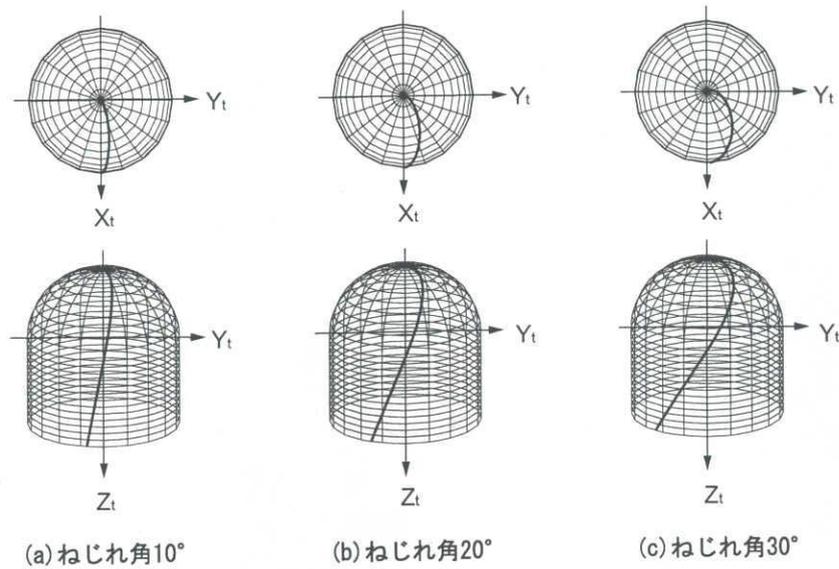


図 5.5: 工具モデル

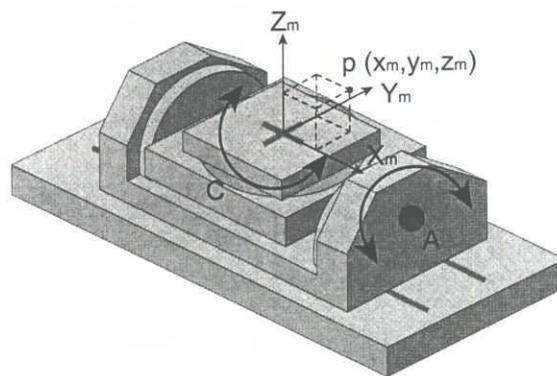


図 5.6: テーブル上の機械座標系

$$\begin{aligned} \begin{Bmatrix} x_m' \\ y_m' \\ z_m' \end{Bmatrix} &= \begin{Bmatrix} \sqrt{x_m^2 + y_m^2} \cos(\phi - c) \\ \sqrt{x_m^2 + y_m^2} \sin(\phi - c) \\ z_m \end{Bmatrix} \\ &= \begin{Bmatrix} \sqrt{x_m^2 + y_m^2} (\cos \phi \cos c + \sin \phi \sin c) \\ \sqrt{x_m^2 + y_m^2} (\sin \phi \cos c - \cos \phi \sin c) \\ z_m \end{Bmatrix} \end{aligned}$$

となる. ここで, ϕ は, p と X_m 軸とが成す角である. また, $\sin \phi$ と $\cos \phi$ は,

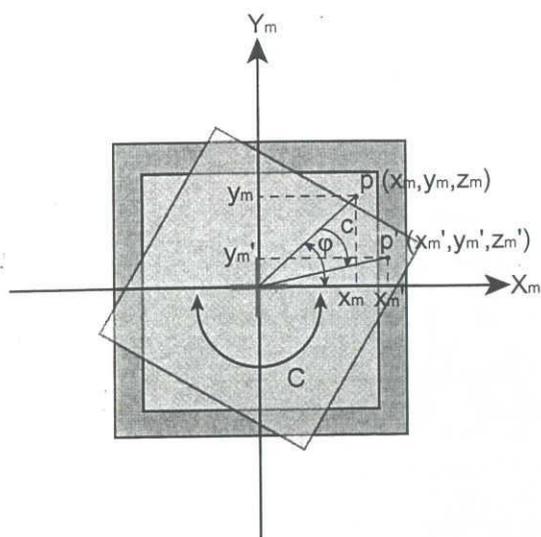


図 5.7: C 軸の回転による加工点の移動

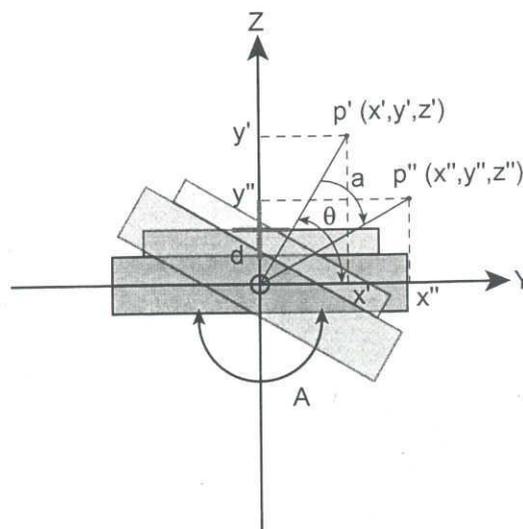


図 5.8: A 軸の回転による加工点の移動

$$\sin \phi = \frac{y_m}{\sqrt{x_m^2 + y_m^2}}$$

$$\cos \phi = \frac{x_m}{\sqrt{x_m^2 + y_m^2}}$$

と表すことができる。

次に、図 5.8 に図 5.6 の X_m 軸の正方向から見た図を示す。座標系は A 軸の回転中心を原点とし、図 5.6 に示す機械座標系 $O - X_m Y_m Z_m$ の原点との距離を d とする。したがって点 p' は、

$$\begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} = \begin{Bmatrix} x_m' \\ y_m' \\ z_m' + d \end{Bmatrix}$$

と表すことができる。このとき、加工点 p' が A 軸の回転により a だけ回転した点を p'' とすると p'' は、

$$\begin{Bmatrix} x'' \\ y'' \\ z'' \end{Bmatrix} = \begin{Bmatrix} x' \\ \sqrt{y'^2 + z'^2} \cos(\theta - a) \\ \sqrt{y'^2 + z'^2} \sin(\theta - a) \end{Bmatrix}$$

$$= \left\{ \begin{array}{c} x' \\ \sqrt{y'^2 + z'^2}(\cos \theta \cos a + \sin \theta \sin a) \\ \sqrt{y'^2 + z'^2}(\sin \theta \cos a - \cos \theta \sin a) \end{array} \right\}$$

となる。ここで、 θ は、 p' と Y 軸とが成す角である。また、 $\sin \theta$ と $\cos \theta$ は、

$$\sin \theta = \frac{z'}{\sqrt{y'^2 + z'^2}}$$

$$\cos \theta = \frac{y'}{\sqrt{y'^2 + z'^2}}$$

と表すことができる。

したがって、テーブルの旋回、傾斜後の点 p の位置 p'' 、すなわち直進駆動軸の位置は、

$$\left\{ \begin{array}{c} x_m'' \\ y_m'' \\ z_m'' \end{array} \right\} = \left\{ \begin{array}{c} x_m' \\ \sqrt{y_m'^2 + (z_m' + d)^2}(\cos \theta \cos a + \sin \theta \sin a) \\ \sqrt{y_m'^2 + (z_m' + d)^2}(\sin \theta \cos a - \cos \theta \sin a) \end{array} \right\}$$

となる。よって、ある時刻 t における直進 3 軸の値を X_t, Y_t, Z_t 、次の制御点である $t + \delta t$ における直進 3 軸の値を $X_{t+\delta t}, Y_{t+\delta t}, Z_{t+\delta t}$ とすると、直進駆動軸の変化量は、

$$\left\{ \begin{array}{c} \Delta X \\ \Delta Y \\ \Delta Z \end{array} \right\} = \left\{ \begin{array}{c} X_{t+\delta t} - X_t \\ Y_{t+\delta t} - Y_t \\ Z_{t+\delta t} - Z_t \end{array} \right\}$$

となる。

5.4.2 回転駆動軸

ここでは、回転制御軸の変化量の算出方法について述べる。

工具姿勢を表す工具回転軸の単位ベクトルは、方向余弦ベクトル $\vec{e} = (e_{mx}, e_{my}, e_{mz})$ の形で収まっている。この値から工作機械上で同じ工具姿勢をとるような回転 2 軸の値に変換する必要がある。先述したように、ボールエンドミル加工の場合、主軸が回転するので工具軸周りには方向性がない。したがって、姿勢を与えるのに二つの回転運動があればよい。5 軸制御マシニングセンタでは、機械の構造からくる回転 2 軸の構成によって姿勢表現が自ずと決定され、いくつかの Type に分類することができる。

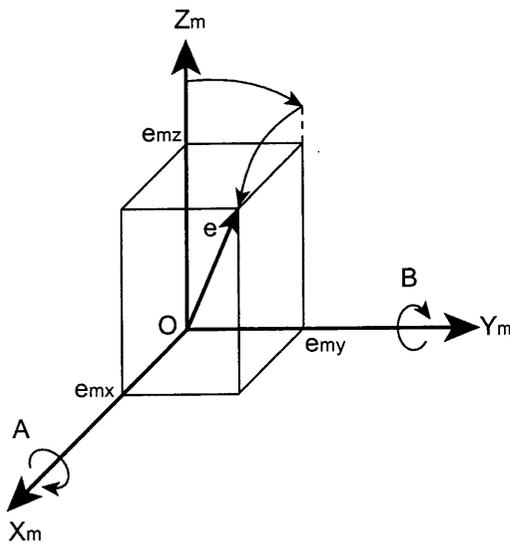


図 5.9: Type1

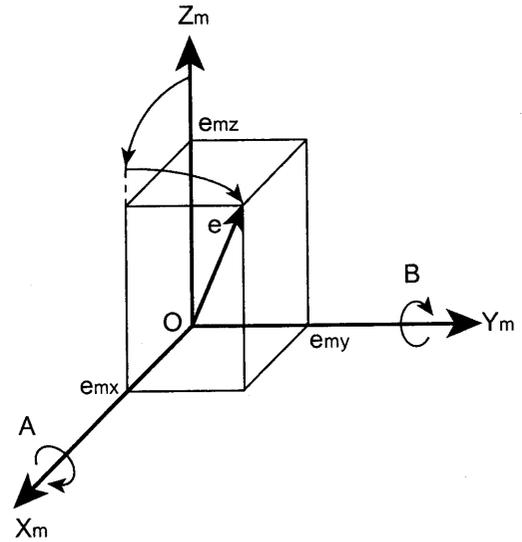


図 5.10: Type2

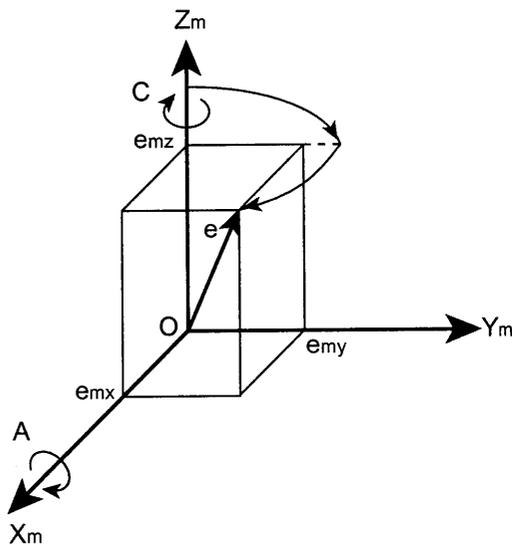


図 5.11: Type3

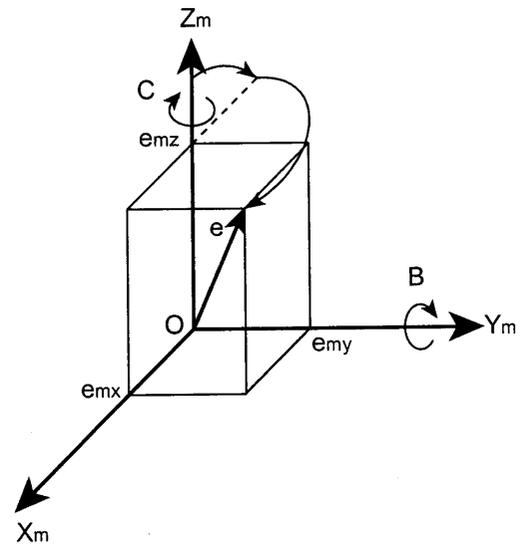


図 5.12: Type4

5軸制御マシニングセンタの制御軸のうち、3軸は従来通り X, Y, Z 軸に平行な並進制御軸である。残りの2軸は回転軸であるが、A, B, Cのうちいずれか二つを選択することになる。従来の3軸制御の状態である工具軸が Z 軸を向いている状態を初期値として、回転2軸の姿勢構成の組合せは図 5.9~図 5.12 に示す4つが考えられる。

Type1 : A, B 軸 (A が工具に近い側, B が工作物に近い側)

Type2 : B, A 軸 (B が工具に近い側, A が工作物に近い側)

Type3 : A, C 軸

Type4 : B, A 軸

AB の組合せには 2 通りあるが, AC, BC の組合せはそれぞれ 1 通りである. これは工具軸が初期状態で Z 軸を向いているため, 姿勢構成の際の回転順序が $A \rightarrow C$, $B \rightarrow C$ の 1 通りしかないためである. 以下, 角度に関してはその正負は工具軸の動きを主体として定め, 座標軸の正の方向から原点を見て時計回りを正としている. ここで, $\text{sgn}(a)$ は実数 a の正負を決める関数で以下の値を返すものとする.

$$\text{sgn}(a) = \begin{cases} 1 & (a \geq 0) \\ -1 & (a < 0) \end{cases}$$

(1) Type1

AB 軸の構成で工作物に近い側に B 軸があるものである. この軌道の順序は, 回転軸を工具側から順に並べた順序である. この形式ではテーブル上面 ($e_{mz} \geq 0$) のすべての姿勢をとるために, A 軸, B 軸ともに 180° の回転範囲を必要とする.

$$\begin{aligned} A &= \text{sgn}(e_{my}) \cos^{-1} \sqrt{e_{mx}^2 + e_{mz}^2} \frac{180}{\pi} \\ B &= \begin{cases} -\text{sgn}(e_{mx}) \cos^{-1} \frac{e_{mz}}{e_{mx}^2 + e_{mz}^2} \frac{180}{\pi} & (|e_{my}| \neq 1) \\ 0 & (|e_{my}| = 1) \end{cases} \end{aligned}$$

(2) Type2

Type1 のような軌道进行を考えると対称性から逆順の軌道も考えられ, これは Type1 と姿勢構成の軌道の順序が逆になったものであり, 同様に,

$$\begin{aligned} A &= \begin{cases} \text{sgn}(e_{my}) \cos^{-1} \frac{e_{mz}}{e_{my}^2 + e_{mz}^2} \frac{180}{\pi} & (|e_{mx}| \neq 1) \\ 0 & (|e_{mx}| = 1) \end{cases} \\ B &= \text{sgn}(e_{mx}) \cos^{-1} \sqrt{e_{my}^2 + e_{mz}^2} \frac{180}{\pi} \end{aligned}$$

(3) Type3

この Type は空間の極座標の形式と類似しており, Type4 とともにこの回転軸構成が一般的である. この形式ではテーブル上面 ($e_{mz} \geq 0$) のすべての姿勢をとるために, A 軸 90° , C 軸ともに 360° の回転範囲を必要とする.

$$A = \cos^{-1} e_{mz} \frac{180}{\pi}$$

$$C = \operatorname{sgn}(e_{mx}) \cos^{-1} \frac{e_{my}}{\sqrt{e_{mx}^2 + e_{my}^2}} \frac{180}{\pi}$$

(4) Type4

Type3 と同様に,

$$B = \cos^{-1} e_{mz} \frac{180}{\pi}$$

$$C = \operatorname{sgn}(e_{my}) \cos^{-1} \frac{-e_{mx}}{\sqrt{e_{mx}^2 + e_{my}^2}} \frac{180}{\pi}$$

以上の方法により、回転2軸の値を算出することができる。したがって、例えばType1の場合、ある時刻 t における回転2軸の値を A_t, B_t 、次の制御点である $t + \delta t$ における回転2軸の値を $A_{t+\delta t}, B_{t+\delta t}$ とすると、回転駆動軸の変化量は、

$$\begin{Bmatrix} \Delta A \\ \Delta B \end{Bmatrix} = \begin{Bmatrix} A_{t+\delta t} - A_t \\ B_{t+\delta t} - B_t \end{Bmatrix}$$

となる。

5.5 座標変換

CNC装置に入力される製品形状情報は、CAD座標系上の値であるため、実際に工作機械を駆動させるには、これを絶対座標系（機械座標系）に座標変換する必要がある。また、本研究においては、工具姿勢を加工面座標系で定義するため、次の制御点における工具姿勢を決定し、工作機械の各制御軸の変化量を求める際にも、座標変換を行う必要がある。以下では、二次元座標変換、三次元座標変換および本研究で実際に行う座標変換について説明する。

5.5.1 二次元座標変換

a) 並進運動に対する座標変換

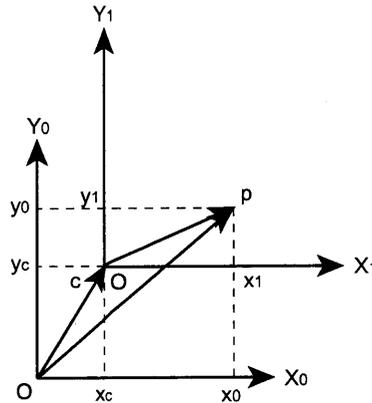


図 5.13: 並進運動に対する二次元座標変換

ここでは、二次元における並進運動に対する座標変換について説明する。図 5.13 に示すように、 $O-x_0y_0$ と $O-x_1y_1$ の 2 つの座標系を定める。 $O-x_0y_0$ 座標系に対して $O-x_1y_1$ 座標系の原点の位置ベクトル \vec{c} を (x_c, y_c) とする。

このとき x_0 は \vec{p} から Y_0 軸に下ろした垂線の長さであるので、 x_1, x_c を用いると、

$$x_0 = x_1 + x_c$$

が得られる。同様に \vec{p} から X_0 に下ろした垂線の長さから、

$$y_0 = y_1 + x_c$$

が得られる。したがって、この座標変換は

$$\begin{Bmatrix} x_0 \\ y_0 \end{Bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix}$$

と表すことができる。

b) 回転運動に対する座標変換

ここでは、二次元における回転運動に対する座標変換について説明する。図 5.14 に示すように、 $O-x_0y_0$ と $O-x_1y_1$ の 2 つの座標系を定める。 2 つの座標系の成す角を図に示すように、 X_0 と X_1 の成す角 θ とする。 なお、回転角 θ は、反時計回りに回転した場合を正とする。

このとき x_0 は \vec{p} から Y_0 軸に下ろした垂線の長さであるので、 x_1, y_1 を用いると、

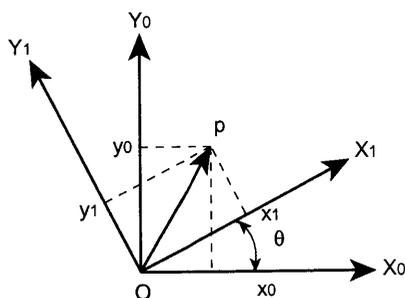


図 5.14: 回転運動に対する二次元座標変換

$$x_0 = x_1 \cos \theta - y_1 \sin \theta$$

が得られる。同様に p から X_0 に下ろした垂線の長さから、

$$y_0 = x_1 \sin \theta + y_1 \cos \theta$$

が得られる。したがって、この座標変換は

$$\begin{Bmatrix} x_0 \\ y_0 \end{Bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix}$$

と表すことができる。ここで

$$E = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

を座標変換マトリクスという。

5.5.2 三次元座標変換

a) 並進運動に対する座標変換

ここでは、三次元における並進運動に対する座標変換について説明する。図 5.15 に示すように、 $O-X_0Y_0Z_0$ と $O-X_1Y_1Z_1$ の2つの座標系を定める。 $O-X_0Y_0Z_0$ 座標系に対する $O-X_1Y_1Z_1$ 座標系の原点の位置ベクトル \vec{c} を (x_c, y_c, z_c) とする。

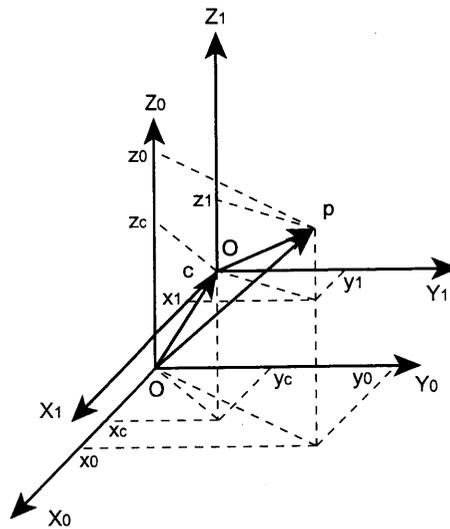


図 5.15: 並進による三次元座標変換

このとき x_0 は \vec{p} を X_0Y_0 平面に投影したもののから Y_0 軸に下ろした垂線の長さであるので, x_1, x_c を用いると,

$$x_0 = x_1 + x_c$$

が得られる. 同様に \vec{p} を X_0Y_0 平面に投影したもののから X_0 に下ろした垂線の長さから,

$$y_0 = y_1 + x_c$$

\vec{p} から X_0Y_0 平面に下ろした垂線の長さから,

$$z_0 = z_1 + z_c$$

が得られる. したがって, この座標変換は

$$\begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$$

と表すことができる.

b) 回転運動に対する座標変換

ここでは三次元における回転運動に対する座標変換について説明する. 図 5.16 に示すように, $O-x_0y_0z_0$ と $O-x_1y_1z_1$ の2つの座標系を定める. 2つの座標系の成す角を図に示すように, Y_0 と Y_1 の成す角 φ とする. なお, 回転角 φ は回転座標軸の終端側から見て反時計回りに回転した場合を正とする.

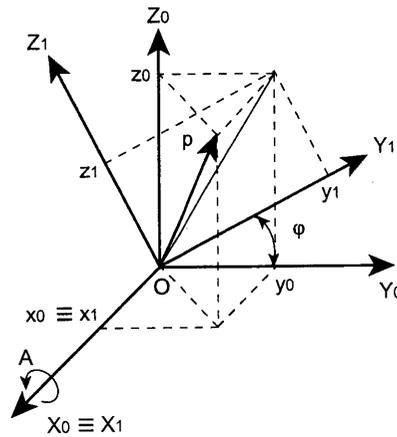


図 5.16: A 軸の回転による三次元座標変換

このとき y_0 は \vec{p} を Y_0Z_0 平面に投影したもののから Z_0 軸に下ろした垂線の長さであるので, y_1, z_1 を用いると,

$$y_0 = y_1 \cos \varphi - z_1 \sin \varphi$$

が得られる. 同様に \vec{p} を Y_0Z_0 平面に投影したもののから Y_0 軸に下ろした垂線の長さから,

$$z_0 = y_1 \sin \varphi + z_1 \cos \varphi$$

また, A 軸の回転であるので x_0 は,

$$x_0 = x_1$$

となる. したがって, この座標変換は

$$\begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$$

と表すことができる。ここで、

$$E_A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.14)$$

を A 軸回転による三次元座標変換の座標変換マトリクスという。

同様に図 5.17 および図 5.18 に示すように、2つの座標系の成す角をそれぞれ Z_0 と Z_1 の成す角 ψ , X_0 と X_1 の成す角 θ とする。

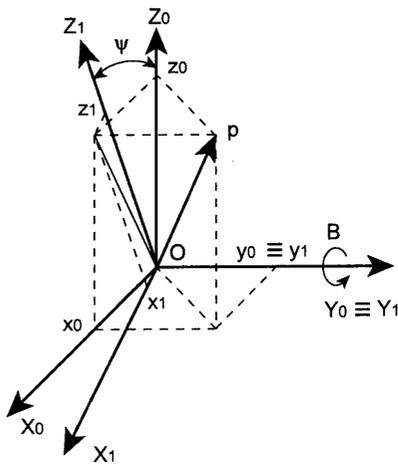


図 5.17: B 軸の回転による三次元座標変換

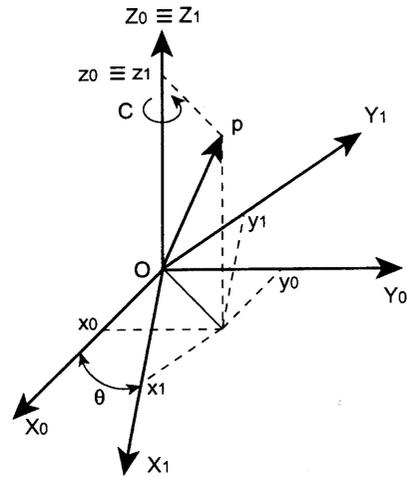


図 5.18: C 軸の回転による三次元座標変換

このとき B 軸の回転による座標変換は、

$$\begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix} = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$$

と表すことができる。ここで、

$$E_B = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.15)$$

を B 軸回転による三次元座標変換の座標変換マトリクスという。また、C 軸の回転による座標変換は、

$$\begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$$

と表すことができる。ここで、

$$E_C = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.16)$$

をC軸回転による三次元座標変換の座標変換マトリクスという。

2つの軸の回転運動による座標変換は式(5.14)~(5.16)の座標変換マトリクスの組み合わせにより行うことができる。例えば、図5.19に示すように、 $O-X_0Y_0Z_0$ 、 $O-X_1Y_1Z_1$ 、 $O-X_2Y_2Z_2$ の3つの座標系を定め、C、Aの順に軸を回転運動させて座標変換を行う。 $O-X_1Y_1Z_1$ と $O-X_2Y_2Z_2$ の2つの座標系の成す角を X_0 と X_1 の成す角 θ 、 $O-X_0Y_0Z_0$ と $O-X_1Y_1Z_1$ の2つの座標系の成す角を Y_0 と Y_1 の成す角 φ とする。

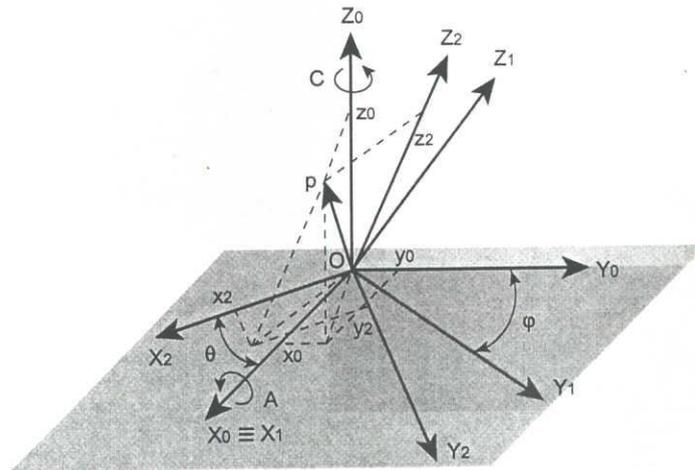


図 5.19: 2軸回転による三次元座標変換

このような2つの軸の回転運動による座標変換は、座標変換マトリクスを順に掛けることで行うことができる。したがって、C、A軸の回転による座標変換は式(5.14)、(5.16)を用いると、

$$\begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix} = E_C E_A \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix} \\
&= \begin{bmatrix} \cos \theta & -\sin \theta \cos \varphi & \sin \theta \sin \varphi \\ \sin \theta & \cos \theta \cos \varphi & -\cos \theta \sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}
\end{aligned}$$

となる。このように、式(5.14)~(5.16)の座標変換マトリクスを組み合わせることで、2つの軸の回転運動、さらに3つの軸の回転運動における3次元座標変換を行うことができる。

5.5.3 製品形状情報および工具姿勢の座標変換

CNC装置に入力される製品形状情報は、図5.20に示すようにCAD座標系 $O-X_c Y_c Z_c$ 上の値である。このため、実際に工作機械を駆動させるには、これを絶対座標系(機械座標系) $O-X_m Y_m Z_m$ に座標変換する必要がある。

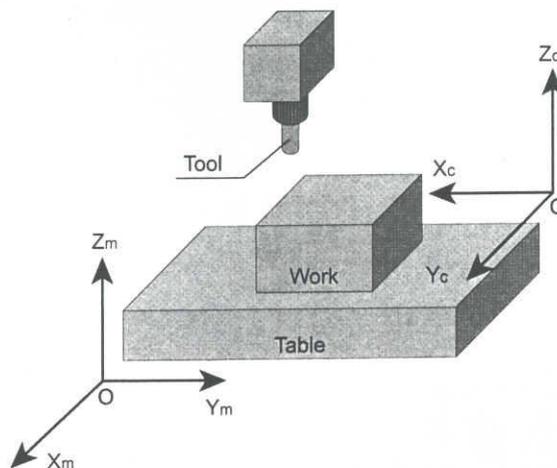


図 5.20: CAD 座標系と機械座標系

また、本研究においては、図5.21に示すように工具姿勢を、加工点における法線ベクトルを Z_s 軸と一致させ、送り方向を X_s 軸と一致させて、 Z_s 軸を基準とする傾き角 α と、 Z_s 軸周りの回転角 β を用いて加工面座標系 $O-X_s Y_s Z_s$ で定義する。したがって、図5.22に示すように、ある時刻 t における加工面の座標系と $t + \delta t$ における加工面の座標系は異なるため、次の制御点における工具姿勢を決定する際には、絶対座標系である機械座標系 $O-X_m Y_m Z_m$ に変換する必要がある。さらに、工作機械の各制御軸の変化量を決定する際にも、工具姿勢が機械座標系で表現されている必要がある。

ここでは、CAD座標系で表現されている製品形状情報と加工面座標系で表現されている工具姿勢を絶対座標系である機械座標系に変換する方法について説明する。

CAD座標系から機械座標系への座標変換と加工面座標系から機械座標系への座標変換は同様にして行うことができるので、以下ではCAD座標系と加工面座標系を $O-XYZ$ として説明する。図5.23に示すように、絶対座標系である機械座標系 $O-X_mY_mZ_m$ に対して相対座標系 $O-XYZ$ の原点の位置ベクトルを \vec{o} とする。このとき、X、Y、Zの各軸の並進運動における座標変換とA、B、Cの各軸の回転運動における座標変換を行うことで $O-XYZ$ から $O-X_mY_mZ_m$ に変換することができる。回転運動における座標変換は2.2節で述べたように、座標変換マトリクスを順に掛けることで行うことができるので、A、B、Cの各軸をどのような順に回転させるかによってその方法が異なる。回転3軸の組み合わせは次のようなものが考えられる。

Type1 : A, B, C 軸の順に回転運動

Type2 : A, C, B 軸の順に回転運動

Type3 : B, A, C 軸の順に回転運動

Type4 : B, C, A 軸の順に回転運動

Type5 : C, A, B 軸の順に回転運動

Type6 : C, B, A 軸の順に回転運動

以下では、各Typeにおける座標変換方法について説明する。

(1) Type1

まず、相対座標系 $O-XYZ$ の原点と機械座標系 $O-X_mY_mZ_m$ の原点を一致させて、回転運動による座標変換を行う。図5.24～図5.26にABC各軸の回転運動による座標変換を示す。

A軸の回転運動による座標変換は、A軸を φ だけ回転させて相対座標系 $O-XYZ$ のZ軸を X_mZ_m 平面に一致させる。このときの座標変換マトリクス E_1 は式(5.14)より、

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.17)$$

となる。ここで、A軸の回転角 φ は、X、Y、Z軸の単位ベクトルをそれぞれ \vec{i} 、 \vec{j} 、 \vec{k} とすると、 \vec{k} を Y_mZ_m 平面に投影したものと Z_m 軸とが成す角である。したがって、 $\sin \varphi$ と $\cos \varphi$ は、

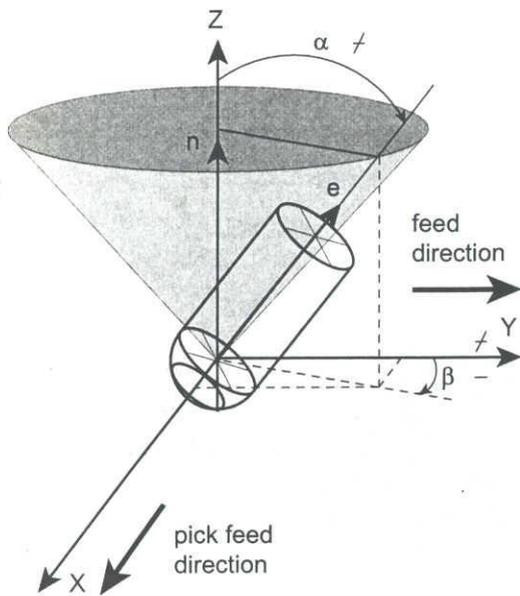


図 5.21: 工具姿勢の定義

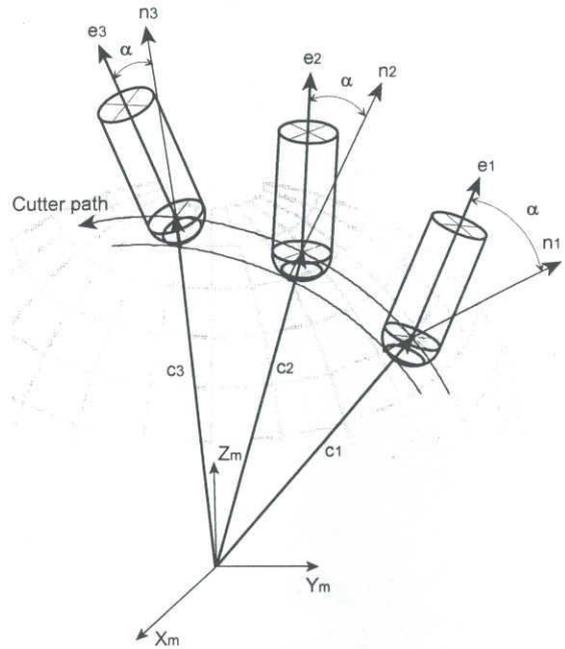


図 5.22: 自由曲面における工具軌跡

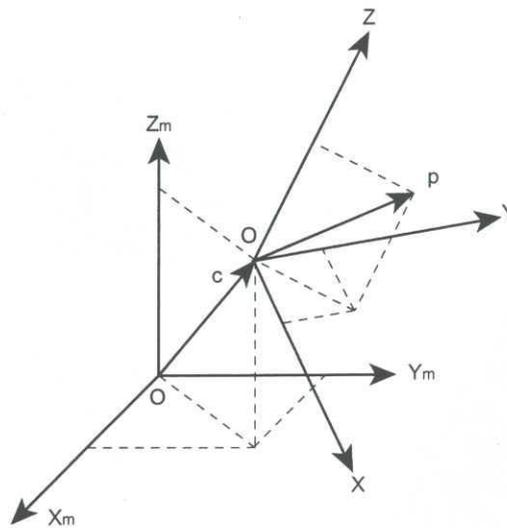


図 5.23: 相対座標系と絶対座標系の関係

$$\sin \varphi = \frac{y_{mk}}{\sqrt{y_{mk}^2 + z_{mk}^2}}$$

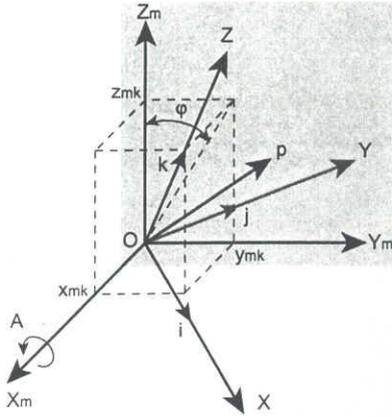


図 5.24: A 軸の回転運動

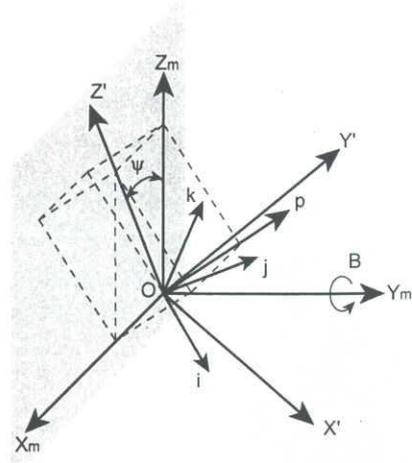


図 5.25: B 軸の回転運動

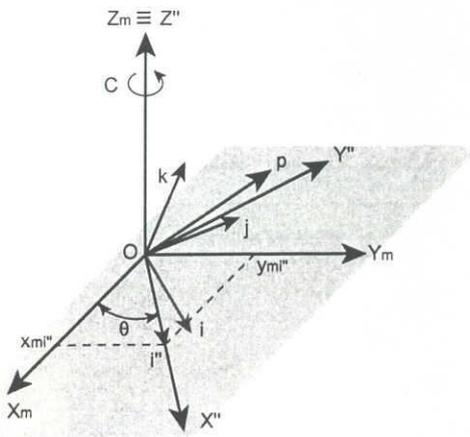


図 5.26: C 軸の回転運動

$$\cos \varphi = \frac{z_{mk}}{\sqrt{y_{mk}^2 + z_{mk}^2}}$$

と表すことができる。

次に B 軸の回転による座標変換は、A 軸の回転運動によってできた座標系を $O-X'Y'Z'$ とすると、B 軸を ψ だけ回転させて Z' 軸を Z_m 軸に一致させる。このときの座標変換マトリクス E_2 は式 (5.15) より、

$$E_2 = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.18)$$

となる。ここで、B軸の回転角 ψ は、 Z' 軸と Z_m 軸の成す角である。 Z' 軸は Z 軸を X_m 軸を中心に回転させてできた軸であるので、結局 ψ は、 Z 軸と \vec{k} を $Y_m Z_m$ 平面に投影したものと成す角である。したがって、 $\sin \psi$ と $\cos \psi$ は、

$$\sin \psi = \frac{x_{mk}}{\sqrt{x_{mk}^2 + y_{mk}^2 + z_{mk}^2}}$$

$$\cos \psi = \frac{\sqrt{y_{mk}^2 + z_{mk}^2}}{\sqrt{x_{mk}^2 + y_{mk}^2 + z_{mk}^2}}$$

と表すことができる。

次にC軸の回転による座標変換は、B軸の回転運動によってできた座標系を $O-X''Y''Z''$ とすると、C軸を θ だけ回転させて全ての軸を一致させる。このときの座標変換マトリクス E_3 は式(5.16)より、

$$E_3 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

となる。ここで、C軸の回転角 θ は、 X'' 軸と X_m 軸の成す角である。したがって、 X'' 軸の単位ベクトルを \vec{i}'' とすると $\sin \theta$ と $\cos \theta$ は、

$$\sin \theta = \frac{y_{mi}''}{\sqrt{x_{mi}''^2 + y_{mi}''^2}} \quad (5.20)$$

$$\cos \theta = \frac{x_{mi}''}{\sqrt{x_{mi}''^2 + y_{mi}''^2}} \quad (5.21)$$

と表すことができる。また、 \vec{i}'' は \vec{i} を X_m 軸を中心に回転させ、さらに Y_m 軸を中心に回転させてできた軸である。ここで、 \vec{i} を X_m 軸を中心に回転させたものを \vec{i}' とすると \vec{i}' は、

$$\begin{aligned} \begin{Bmatrix} x_{mi}' \\ y_{mi}' \\ z_{mi}' \end{Bmatrix} &= \begin{Bmatrix} x_{mi} \\ \sqrt{y_{mi}^2 + z_{mi}^2} \cos(\alpha - \varphi) \\ \sqrt{y_{mi}^2 + z_{mi}^2} \sin(\alpha - \varphi) \end{Bmatrix} \\ &= \begin{Bmatrix} x_{mi} \\ \sqrt{y_{mi}^2 + z_{mi}^2} (\cos \alpha \cos \varphi + \sin \alpha \sin \varphi) \\ \sqrt{y_{mi}^2 + z_{mi}^2} (\sin \alpha \cos \varphi - \cos \alpha \sin \varphi) \end{Bmatrix} \end{aligned}$$

となる。ここで、 α は、 \vec{i} を $Y_m Z_m$ 平面に投影したものと Y_m 軸と成す角である。また、 $\sin \alpha$ と $\cos \alpha$ は、

$$\sin \alpha = \frac{z_{mi}}{\sqrt{y_{mi}^2 + z_{mi}^2}}$$

$$\cos \alpha = \frac{y_{mi}}{\sqrt{y_{mi}^2 + z_{mi}^2}}$$

と表すことができる。さらに、 \vec{i}'' は、

$$\begin{aligned} \begin{Bmatrix} x_{mi}'' \\ y_{mi}'' \\ z_{mi}'' \end{Bmatrix} &= \begin{Bmatrix} \sqrt{z_{mi}^2 + x_{mi}^2} \sin(\beta - \psi) \\ y_{mi} \\ \sqrt{z_{mi}^2 + x_{mi}^2} \cos(\beta - \psi) \end{Bmatrix} \\ &= \begin{Bmatrix} \sqrt{z_{mi}^2 + x_{mi}^2} (\sin \beta \cos \psi - \cos \beta \sin \psi) \\ y_{mi} \\ \sqrt{z_{mi}^2 + x_{mi}^2} (\cos \beta \cos \psi + \sin \beta \sin \psi) \end{Bmatrix} \end{aligned}$$

となる。ここで、 β は、 \vec{i}' を $Z_m X_m$ 平面に投影したものと Z_m 軸とが成す角である。また、 $\sin \beta$ と $\cos \beta$ は、

$$\sin \beta = \frac{x_{mi}'}{\sqrt{z_{mi}^2 + x_{mi}^2}}$$

$$\cos \beta = \frac{z_{mi}'}{\sqrt{z_{mi}^2 + x_{mi}^2}}$$

と表すことができる。

したがって、相対座標系 $O-XYZ$ から機械座標系 $O-X_m Y_m Z_m$ への回転運動による座標変換は、式 (5.17)~(5.19) より、

$$\begin{Bmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{Bmatrix} = E_1 E_2 E_3 \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} \quad (5.22)$$

となる。

さらに、式 (5.22) に並進運動による座標変換を加える。並進運動による座標変換は、2.1 節で述べたように機械座標系 $O-X_m Y_m Z_m$ に対する相対座標系 $O-XYZ$ の原点の位置ベクトル \vec{c} (c_{mx} , c_{my} , c_{mz}) を足せばよい。したがって、Type1 における座標変換は、

$$\begin{aligned}
& \begin{pmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{pmatrix} \\
= & E_1 E_2 E_3 \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{pmatrix} \\
= & \begin{bmatrix} \cos \psi \cos \theta & -\cos \psi \sin \theta & \sin \psi \\ \sin \varphi \sin \psi \cos \theta + \cos \varphi \sin \theta & -\sin \varphi \sin \psi \sin \theta + \cos \varphi \cos \theta & -\sin \varphi \cos \psi \\ -\cos \varphi \sin \psi \cos \theta + \sin \varphi \sin \theta & \cos \varphi \sin \psi \sin \theta + \sin \varphi \cos \theta & \cos \varphi \cos \psi \end{bmatrix} \\
& \times \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{pmatrix}
\end{aligned}$$

と表すことができる。

(2) Type2

同様にA, C, Bの順に回転運動させたときの座標変換を行う。図5.27～図5.29にABC各軸の回転運動による座標変換を示す。

A軸の回転運動による座標変換マトリクス E_1 は式(5.14)より,

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.23)$$

となる。ここで, $\sin \varphi$, $\cos \varphi$ は,

$$\begin{aligned}
\sin \varphi &= \frac{z_{mj}}{\sqrt{y_{mj}^2 + z_{mj}^2}} \\
\cos \varphi &= \frac{y_{mj}}{\sqrt{y_{mj}^2 + z_{mj}^2}}
\end{aligned}$$

と表すことができる。

次にC軸の回転運動による座標変換マトリクス E_2 は式(5.16)より,

$$E_2 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.24)$$

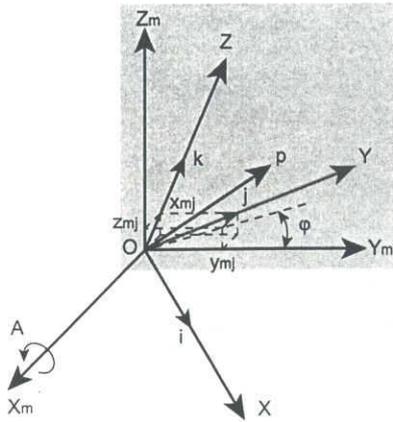


図 5.27: A 軸の回転運動

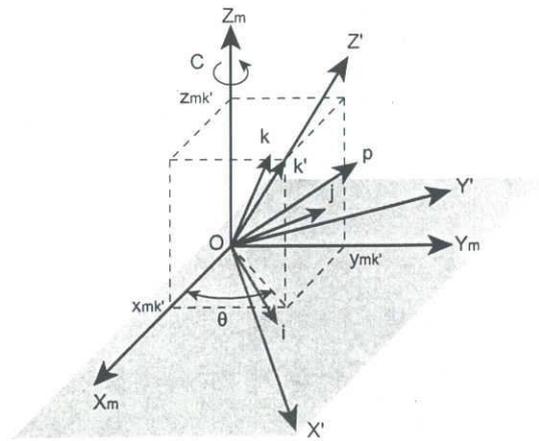


図 5.28: C 軸の回転運動

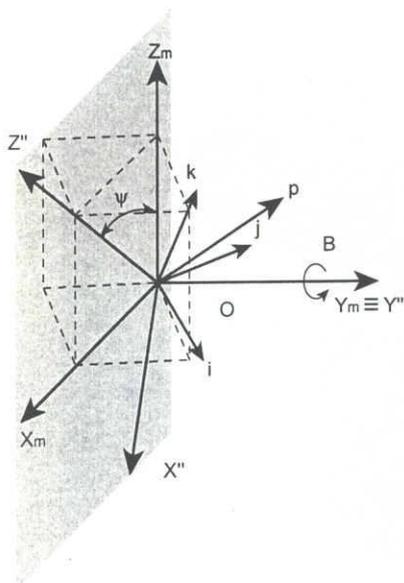


図 5.29: B 軸の回転運動

となる。ここで、 \vec{k} を X_m 軸を中心に回転させたものを \vec{k}' とすると $\sin \theta$, $\cos \theta$ は、

$$\sin \theta = \frac{y_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2}}$$

$$\cos \theta = \frac{x_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2}}$$

と表すことができる。また、 \vec{k}' は、

$$\begin{aligned} \begin{pmatrix} x_{mk'} \\ y_{mk'} \\ z_{mk'} \end{pmatrix} &= \begin{pmatrix} x_{mk} \\ \sqrt{y_{mk}^2 + z_{mk}^2} \cos(\alpha - \varphi) \\ \sqrt{y_{mk}^2 + z_{mk}^2} \sin(\alpha - \varphi) \end{pmatrix} \\ &= \begin{pmatrix} x_{mk} \\ \sqrt{y_{mk}^2 + z_{mk}^2} (\cos \alpha \cos \varphi + \sin \alpha \sin \varphi) \\ \sqrt{y_{mk}^2 + z_{mk}^2} (\sin \alpha \cos \varphi - \cos \alpha \sin \varphi) \end{pmatrix} \end{aligned}$$

となる。ここで、 α は、 \vec{k} を $Y_m Z_m$ 平面に投影したものと Y_m 軸とが成す角である。また、 $\sin \alpha$ と $\cos \alpha$ は、

$$\sin \alpha = \frac{z_{mk}}{\sqrt{y_{mk}^2 + z_{mk}^2}}$$

$$\cos \alpha = \frac{y_{mk}}{\sqrt{y_{mk}^2 + z_{mk}^2}}$$

と表すことができる。

次に B 軸の回転運動による座標変換マトリクス E_3 は式 (5.15) より、

$$E_3 = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.25)$$

となる。ここで、 $\sin \psi$ 、 $\cos \psi$ は、

$$\sin \psi = \frac{\sqrt{x_{mk'}^2 + y_{mk'}^2}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

$$\cos \psi = \frac{z_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

と表すことができる。

したがって、Type2 における座標変換は、式 (5.23)~(5.25) より、

$$\begin{aligned}
& \begin{Bmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{Bmatrix} \\
= & E_1 E_2 E_3 \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} + \begin{Bmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{Bmatrix} \\
= & \begin{bmatrix} \cos \theta \cos \psi & -\sin \theta & \cos \theta \sin \psi \\ \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi & \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi \end{bmatrix} \\
& \times \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} + \begin{Bmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{Bmatrix}
\end{aligned}$$

となる。

(3) Type3

同様にして B, A, C の順に回転運動させたときの座標変換を行う。図 5.30～図 5.32 に ABC 各軸の回転運動による座標変換を示す。

B 軸の回転運動による座標変換マトリクス E_1 は式 (5.15) より,

$$E_1 = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.26)$$

となる。ここで, $\sin \psi$, $\cos \psi$ は,

$$\begin{aligned}
\sin \psi &= \frac{x_{mk}}{\sqrt{x_{mk}^2 + z_{mk}^2}} \\
\cos \psi &= \frac{z_{mk}}{\sqrt{x_{mk}^2 + z_{mk}^2}}
\end{aligned}$$

と表すことができる。

次に A 軸の回転運動による座標変換マトリクス E_2 は式 (5.14) より,

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.27)$$

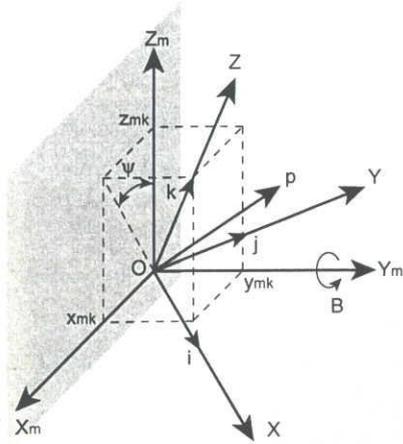


図 5.30: B 軸の回転運動

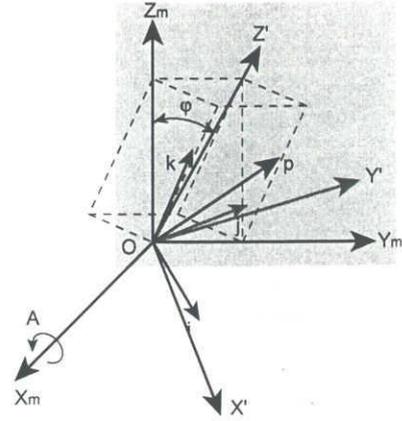


図 5.31: A 軸の回転運動

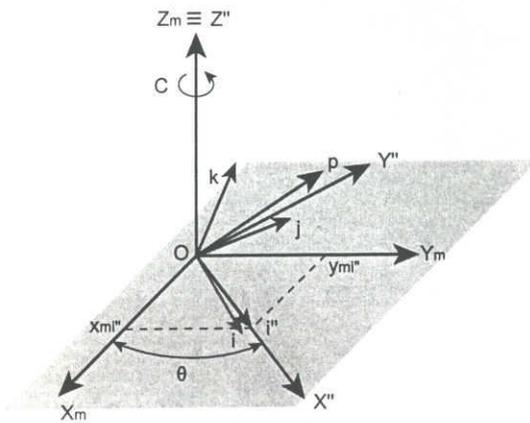


図 5.32: C 軸の回転運動

となる。ここで、 $\sin \varphi$, $\cos \varphi$ は、

$$\sin \varphi = \frac{y_{mk}}{\sqrt{x_{mk}^2 + y_{mk}^2 + z_{mk}^2}}$$

$$\cos \varphi = \frac{\sqrt{x_{mk}^2 + z_{mk}^2}}{\sqrt{x_{mk}^2 + y_{mk}^2 + z_{mk}^2}}$$

と表すことができる。

次に C 軸の回転運動による座標変換マトリクス E_3 は式 (5.16) より、

$$E_3 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.28)$$

となる。ここで、 \vec{i} を Y_m 軸周りに回転させ、 X_m 軸周りに回転させたものを \vec{i}'' とすると、 $\sin \theta$ 、 $\cos \theta$ は、

$$\sin \theta = \frac{y_{mi}''}{\sqrt{x_{mi}''^2 + y_{mi}''^2}}$$

$$\cos \theta = \frac{x_{mi}''}{\sqrt{x_{mi}''^2 + y_{mi}''^2}}$$

と表すことができる。また、 \vec{i}' は、

$$\begin{aligned} \begin{Bmatrix} x_{mi}' \\ y_{mi}' \\ z_{mi}' \end{Bmatrix} &= \begin{Bmatrix} \sqrt{z_{mi}^2 + x_{mi}^2} \sin(\alpha - \psi) \\ y_{mi} \\ \sqrt{z_{mi}^2 + x_{mi}^2} \cos(\alpha - \psi) \end{Bmatrix} \\ &= \begin{Bmatrix} \sqrt{z_{mi}^2 + x_{mi}^2} (\sin \alpha \cos \psi - \cos \alpha \sin \psi) \\ y_{mi} \\ \sqrt{z_{mi}^2 + x_{mi}^2} (\cos \alpha \cos \psi + \sin \alpha \sin \psi) \end{Bmatrix} \end{aligned}$$

となる。ここで、 α は、 \vec{i} を $Z_m X_m$ 平面に投影したものと Z_m 軸とが成す角である。また、 $\sin \alpha$ と $\cos \alpha$ は、

$$\sin \alpha = \frac{x_{mi}}{\sqrt{z_{mi}^2 + x_{mi}^2}}$$

$$\cos \alpha = \frac{z_{mi}}{\sqrt{z_{mi}^2 + x_{mi}^2}}$$

と表すことができる。さらに、 \vec{i}'' は、

$$\begin{aligned} \begin{Bmatrix} x_{mi}'' \\ y_{mi}'' \\ z_{mi}'' \end{Bmatrix} &= \begin{Bmatrix} x_{mi}' \\ \sqrt{y_{mi}'^2 + z_{mi}'^2} \cos(\beta - \varphi) \\ \sqrt{y_{mi}'^2 + z_{mi}'^2} \sin(\beta - \varphi) \end{Bmatrix} \\ &= \begin{Bmatrix} x_{mi}' \\ \sqrt{y_{mi}'^2 + z_{mi}'^2} (\cos \beta \cos \varphi + \sin \beta \sin \varphi) \\ \sqrt{y_{mi}'^2 + z_{mi}'^2} (\sin \beta \cos \varphi - \cos \beta \sin \varphi) \end{Bmatrix} \end{aligned}$$

となる。ここで、 β は、 \vec{i} を $Y_m Z_m$ 平面に投影したものと Y_m 軸とが成す角である。また、 $\sin \beta$ と $\cos \beta$ は、

$$\sin \beta = \frac{z_{mi'}}{\sqrt{y_{mi'}^2 + z_{mi'}^2}}$$

$$\cos \beta = \frac{y_{mi'}}{\sqrt{y_{mi'}^2 + z_{mi'}^2}}$$

と表すことができる。

したがって、Type3における座標変換は、式(5.26)~(5.28)より、

$$\begin{aligned} & \begin{pmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{pmatrix} \\ = & E_1 E_2 E_3 \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{pmatrix} \\ = & \begin{bmatrix} \cos \psi \cos \theta + \sin \psi \sin \varphi \sin \theta & -\cos \psi \sin \theta + \sin \psi \sin \varphi \cos \theta & \sin \psi \cos \theta \\ \cos \varphi \sin \theta & \cos \varphi \cos \theta & -\sin \varphi \\ -\sin \psi \cos \varphi + \cos \psi \sin \varphi \sin \theta & \sin \psi \sin \theta + \cos \psi \sin \varphi \cos \theta & \cos \psi \cos \varphi \end{bmatrix} \\ & \times \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{pmatrix} \end{aligned}$$

となる。

(4) Type4

同様にしてB, C, Aの順に回転運動させたときの座標変換を行う。図5.33~図5.35にABC各軸の回転運動による座標変換を示す。

B軸の回転運動による座標変換マトリクス E_1 は式(5.15)より、

$$E_1 = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.29)$$

となる。ここで、 $\sin \psi$, $\cos \psi$ は、

$$\sin \psi = \frac{z_{mi}}{\sqrt{x_{mi}^2 + z_{mi}^2}}$$

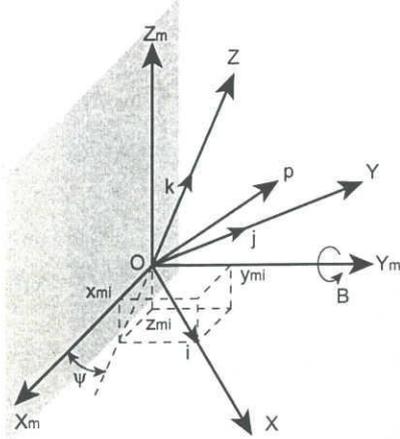


図 5.33: B 軸の回転運動

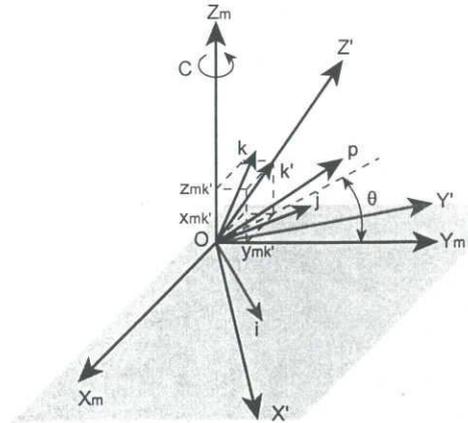


図 5.34: C 軸の回転運動

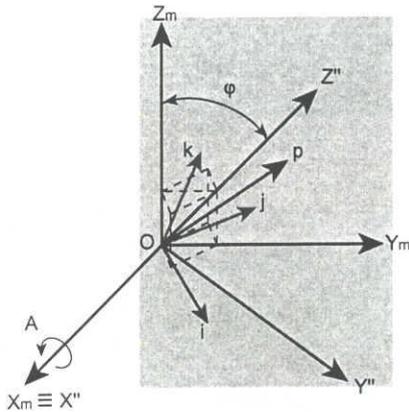


図 5.35: A 軸の回転運動

$$\cos \psi = \frac{x_{mi}}{\sqrt{x_{mi}^2 + z_{mi}^2}}$$

と表すことができる。

次に C 軸の回転運動による座標変換マトリクス E_2 は式 (5.16) より、

$$E_2 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.30}$$

となる。ここで、 \vec{k} を Y_m 軸を中心に回転させたものを \vec{k}' とすると $\sin \theta$, $\cos \theta$ は、

$$\sin \theta = \frac{x_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2}}$$

$$\cos \theta = \frac{y_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2}}$$

と表すことができる。また、 \vec{k}' は、

$$\begin{aligned} \begin{Bmatrix} x_{mk'} \\ y_{mk'} \\ z_{mk'} \end{Bmatrix} &= \begin{Bmatrix} \sqrt{z_{mk}^2 + x_{mk}^2} \sin(\alpha - \psi) \\ y_{mk} \\ \sqrt{z_{mk}^2 + x_{mk}^2} \cos(\alpha - \psi) \end{Bmatrix} \\ &= \begin{Bmatrix} \sqrt{z_{mk}^2 + x_{mk}^2} (\sin \alpha \cos \psi - \cos \alpha \sin \psi) \\ y_{mk} \\ \sqrt{z_{mk}^2 + x_{mk}^2} (\cos \alpha \cos \psi + \sin \alpha \sin \psi) \end{Bmatrix} \end{aligned}$$

となる。ここで、 α は、 \vec{k} を $Z_m X_m$ 平面に投影したものと X_m 軸とが成す角である。また、 $\sin \alpha$ と $\cos \alpha$ は、

$$\sin \alpha = \frac{x_{mk}}{\sqrt{z_{mk}^2 + x_{mk}^2}}$$

$$\cos \alpha = \frac{z_{mk}}{\sqrt{z_{mk}^2 + x_{mk}^2}}$$

と表すことができる。

次に A 軸の回転運動による座標変換マトリクス E_3 は式 (5.14) より、

$$E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.31)$$

となる。ここで、 $\sin \varphi$ 、 $\cos \varphi$ は、

$$\sin \varphi = \frac{\sqrt{x_{mk'}^2 + y_{mk'}^2}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

$$\cos \varphi = \frac{x_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

と表すことができる。

したがって、Type4における座標変換は、式(5.29)~(5.31)より、

$$\begin{aligned}
 & \begin{pmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{pmatrix} \\
 = & E_3 E_2 E_1 \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{pmatrix} \\
 = & \begin{bmatrix} \cos \psi \cos \theta & -\cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi & \cos \psi \sin \theta \sin \varphi + \sin \psi \cos \varphi \\ \sin \theta & \cos \theta \cos \varphi & -\cos \theta \sin \varphi \\ -\sin \psi \cos \theta & \sin \psi \sin \theta \cos \varphi + \cos \psi \sin \varphi & -\sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi \end{bmatrix} \\
 & \times \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{pmatrix}
 \end{aligned}$$

となる。

(5) Type5

同様にしてC, A, Bの順に回転運動させたときの座標変換を行う。図5.36~図5.38にABC各軸の回転運動による座標変換を示す。

C軸の回転運動による座標変換マトリクス E_1 は式(5.16)より、

$$E_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.32)$$

となる。ここで、 $\sin \theta$, $\cos \theta$ は、

$$\begin{aligned}
 \sin \theta &= \frac{x_{mj}}{\sqrt{x_{mj}^2 + y_{mj}^2}} \\
 \cos \theta &= \frac{y_{mj}}{\sqrt{x_{mj}^2 + y_{mj}^2}}
 \end{aligned}$$

と表すことができる。

次にA軸の回転運動による座標変換マトリクス E_2 は式(5.14)より、

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.33)$$

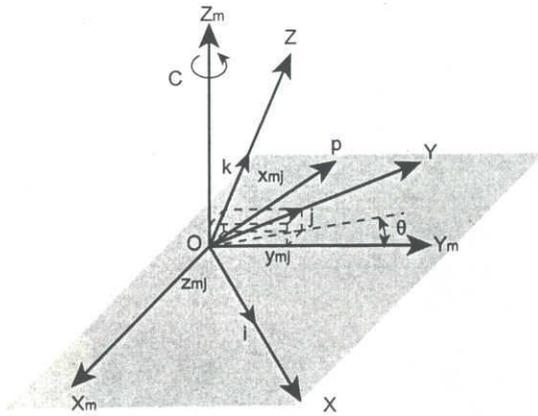


図 5.36: C 軸の回転運動

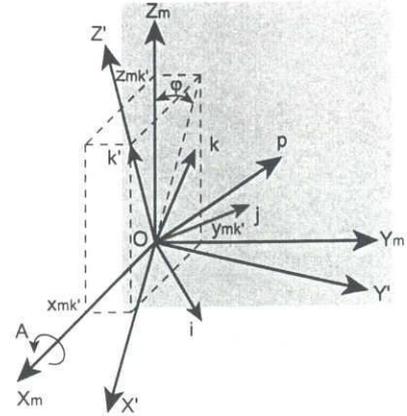


図 5.37: A 軸の回転運動

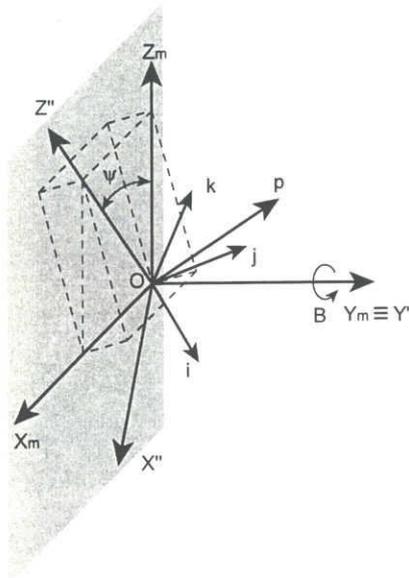


図 5.38: B 軸の回転運動

となる. ここで, \vec{k} を Z_m 軸を中心に回転させたものを \vec{k}' とすると $\sin \varphi$, $\cos \varphi$ は,

$$\sin \varphi = \frac{y_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2}}$$

$$\cos \varphi = \frac{z_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2}}$$

と表すことができる。また、 \vec{k}' は、

$$\begin{aligned} \begin{Bmatrix} x_{mk'} \\ y_{mk'} \\ z_{mk'} \end{Bmatrix} &= \begin{Bmatrix} \sqrt{x_{mk}^2 + y_{mk}^2} \cos(\alpha - \theta) \\ \sqrt{x_{mk}^2 + y_{mk}^2} \sin(\alpha - \theta) \\ z_{mk} \end{Bmatrix} \\ &= \begin{Bmatrix} \sqrt{x_{mk}^2 + y_{mk}^2} (\cos \alpha \cos \theta + \sin \alpha \sin \theta) \\ \sqrt{z_{mk}^2 + x_{mk}^2} (\sin \alpha \cos \theta - \cos \alpha \sin \theta) \\ z_{mk} \end{Bmatrix} \end{aligned}$$

となる。ここで、 α は、 \vec{k} を $X_m Y_m$ 平面に投影したものと X_m 軸とが成す角である。また、 $\sin \alpha$ と $\cos \alpha$ は、

$$\sin \alpha = \frac{y_{mk}}{\sqrt{x_{mk}^2 + y_{mk}^2}}$$

$$\cos \alpha = \frac{x_{mk}}{\sqrt{x_{mk}^2 + y_{mk}^2}}$$

と表すことができる。

次に B 軸の回転運動による座標変換マトリクス E_3 は式 (5.15) より、

$$E_3 = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.34)$$

となる。ここで、 $\sin \psi$ 、 $\cos \psi$ は、

$$\sin \psi = \frac{z_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

$$\cos \psi = \frac{\sqrt{x_{mk'}^2 + y_{mk'}^2}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

と表すことができる。

したがって、Type5 における座標変換は、式 (5.32)~(5.34) より、

$$\begin{Bmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{Bmatrix}$$

$$\begin{aligned}
&= E_3 E_2 E_1 \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} + \begin{Bmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{Bmatrix} \\
&= \begin{bmatrix} \cos \theta \cos \psi - \sin \theta \sin \varphi \sin \psi & -\sin \theta \cos \varphi & \cos \theta \sin \psi + \sin \theta \sin \varphi \cos \psi \\ \sin \theta \cos \psi + \cos \theta \sin \varphi \sin \psi & \cos \theta \cos \varphi & \sin \theta \sin \psi - \cos \theta \sin \varphi \cos \psi \\ -\cos \varphi \sin \psi & \sin \varphi & \cos \varphi \cos \psi \end{bmatrix} \\
&\quad \times \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} + \begin{Bmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{Bmatrix}
\end{aligned}$$

となる。

(6) Type6

同様にして C, B, A の順に回転運動させたときの座標変換を行う。図 5.39～図 5.41 に ABC 各軸の回転運動による座標変換を示す。

C 軸の回転運動による座標変換マトリクス E_1 は式 (5.16) より,

$$E_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.35)$$

となる。ここで, $\sin \theta$, $\cos \theta$ は,

$$\sin \theta = \frac{y_{mi}}{\sqrt{x_{mi}^2 + y_{mi}^2}}$$

$$\cos \theta = \frac{x_{mi}}{\sqrt{x_{mi}^2 + y_{mi}^2}}$$

と表すことができる。

次に B 軸の回転運動による座標変換マトリクス E_2 は式 (5.15) より,

$$E_2 = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \quad (5.36)$$

となる。ここで, \vec{k} を Z_m 軸を中心に回転させたものを \vec{k}' とすると $\sin \psi$, $\cos \psi$ は,

$$\sin \psi = \frac{x_{mk'}}{\sqrt{x_{mk'}^2 + z_{mk'}^2}}$$

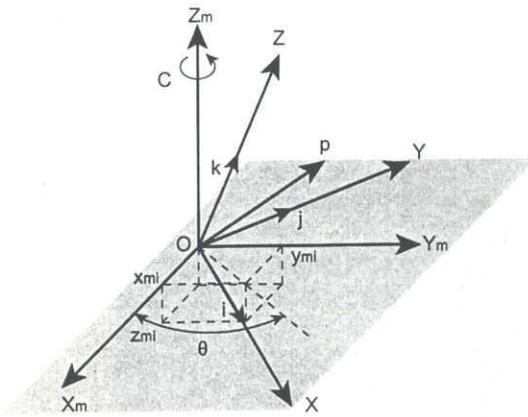


図 5.39: C 軸の回転運動

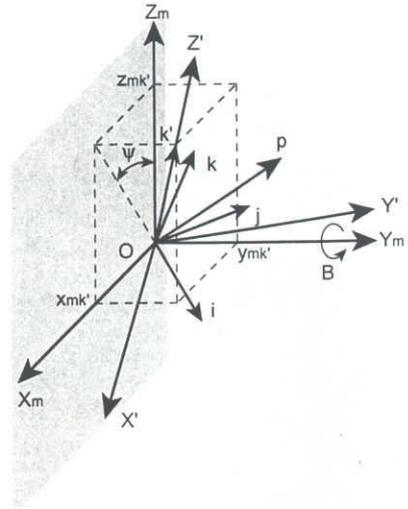


図 5.40: B 軸の回転運動

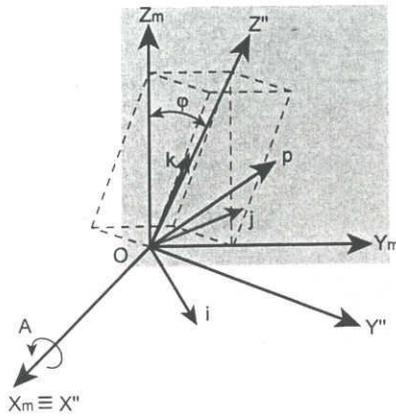


図 5.41: A 軸の回転運動

$$\cos \psi = \frac{z_{mk'}}{\sqrt{x_{mk'}^2 + z_{mk'}^2}}$$

と表すことができる。また、 \vec{k}' は、

$$\begin{Bmatrix} x_{mk'} \\ y_{mk'} \\ z_{mk'} \end{Bmatrix} = \begin{Bmatrix} \sqrt{x_{mk}^2 + y_{mk}^2} \cos(\alpha - \psi) \\ \sqrt{x_{mk}^2 + y_{mk}^2} \sin(\alpha - \psi) \\ z_{mk} \end{Bmatrix}$$

$$= \left\{ \begin{array}{c} \sqrt{x_{mk}^2 + y_{mk}^2}(\cos \alpha \cos \psi + \sin \alpha \sin \psi) \\ \sqrt{z_{mk}^2 + x_{mk}^2}(\sin \alpha \cos \psi - \cos \alpha \sin \psi) \\ z_{mk} \end{array} \right\}$$

となる。ここで、 α は、 \vec{k} を $X_m Y_m$ 平面に投影したものと X_m 軸とが成す角である。また、 $\sin \alpha$ と $\cos \alpha$ は、

$$\sin \alpha = \frac{y_{mk}}{\sqrt{x_{mk}^2 + y_{mk}^2}}$$

$$\cos \alpha = \frac{x_{mk}}{\sqrt{x_{mk}^2 + y_{mk}^2}}$$

と表すことができる。

次にA軸の回転運動による座標変換マトリクス E_3 は式(5.14)より、

$$E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \quad (5.37)$$

となる。ここで、 $\sin \varphi$, $\cos \varphi$ は、

$$\sin \varphi = \frac{y_{mk'}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

$$\cos \varphi = \frac{\sqrt{x_{mk'}^2 + z_{mk'}^2}}{\sqrt{x_{mk'}^2 + y_{mk'}^2 + z_{mk'}^2}}$$

と表すことができる。したがって、Type6における座標変換は、式(5.35)~(5.37)より、

$$\begin{aligned} & \begin{Bmatrix} p_{mx} \\ p_{my} \\ p_{mz} \end{Bmatrix} \\ &= E_1 E_2 E_3 \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} + \begin{Bmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{Bmatrix} \\ &= \begin{bmatrix} \cos \theta \cos \psi & -\sin \theta \cos \varphi + \cos \theta \sin \psi \sin \varphi & \sin \theta \sin \varphi + \cos \theta \sin \psi \cos \varphi \\ \sin \theta \cos \psi & \cos \theta \cos \varphi + \sin \theta \sin \psi \sin \varphi & -\cos \theta \sin \varphi + \sin \theta \sin \psi \cos \varphi \\ -\sin \psi & \cos \psi \sin \varphi & \cos \psi \cos \varphi \end{bmatrix} \\ & \quad \times \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} + \begin{Bmatrix} c_{mx} \\ c_{my} \\ c_{mz} \end{Bmatrix} \end{aligned}$$

となる。

関連図書

- [1] 樋野 励, 森脇俊道: 再帰的伝播法による分散型生産ジョブショップスケジューリング; 日本機械学会論文集 C, Vol.66, No.647, pp2449-2455(2000)
- [2] 樋野 励, 森脇俊道: 再帰的伝播法による分散型生産ジョブショップスケジューリング (第2報:階層構造を有するシステムのスケジューリング); 日本機械学会論文集 C, Vol.66, No.651, pp3791-3798(2000)
- [3] 杉村延広: 分散型生産システムの動向ースケジューリングの観点から, システム/制御/情報, Vol.46, No.5, pp269-275(2002)
- [4] 関口恭毅, 木瀬洋:スケジューリング理論の基礎と応用ースケジューリング問題の分枝限定法; システム/制御/情報,Vol.45,No.2,pp97-104(2001)
- [5] J.Carlier and E.Pinson: An algorithm for solving the job-shop problem; Management Science, Vol.35,No.2,pp.164-176
- [6] 亀田恒彦, 山下雅史, 分散アルゴリズム, 近代科学社,1994
- [7] 坂本碩也, 生産管理入門 第2版, 理工学社, 2000.
- [8] John F.Muth and Gerald L.Thompson, Industrial sheduling, Prentice Hall, Inc., Englewood Clifs, New Jersey, 1963.
- [9] 関根知明, スケジューリングの理論, 日刊工業新聞社, 1971.
- [10] J. Carlier and E. Pinson, An algorithm for solving the job-shop problem, Manegement Science, Vol.35, No.2, pp.167-176, 1989.
- [11] 樋野 励, 森脇俊道, スケジューリングに基づく生産設備の再配置 (第1報)-設備配置に対する評価値としてのスケジュールの導入-, 精密工学会誌, Vol.69, No.5, Page655-659, 2003
- [12] 樋野 励, 再帰的伝播法によるジョブショップスケジューリング (第4報: 協調作業を考慮にいた情報交換手順の提案), 日本機械学会論文集 C, Vol.70, No.699, Page3300-3307, 2004

- [13] 園川隆夫, 伊藤謙治, 生産マネジメントの手法, 朝倉書店, 1996.
- [14] 近藤司, 岸波建史, 斎藤勝政, 逆オフセット法を基にした形状加工処理, 精密工学会誌, Vol. 54, No. 5, pp971-976, 1988
- [15] Minghui Hao, 水垣善夫, 坂本正史, 拡張 Z-map モデルによる CAM システムの開発 — オフセット面生成法の一提案 —, 精密工学会誌, Vol. 60, No.2, pp275-279, 1994
- [16] 入口健二, 加藤清敬, 曲面細分割による工具経路生成方法, 精密工学会誌, Vol. 64, No. 11, pp1618-1622, 1998
- [17] 高橋悌二, 加藤清敬, NURBS 補間機能をもった CNC の開発, 1993 年度精密工学会秋季大会学術講演会講演論文集, pp57-58, 1993
- [18] 宮田光人, 島淳, (特集) 最近の FA 制御とネットワーク技術, 「高速高精度曲線補間技術 NURBS」, 精密工学会誌, Vol. 65, No.9, pp 1263-1266, 1999
- [19] T.Sata, F.Kimura, N.Okada and M. Hosaka, A New Method of NC Interpolation for Machining the Sculptured Surface, Annals of the CIRP, Vol. 30, 1, 1981
- [20] 鈴木裕, 山崎和雄, 星鐵大郎, 二川正康, 実時間金型加工用数値制御システムの開発 (第 1 報) — 曲面交差線高速算出アルゴリズムとその応用について —, 精密工学会誌, Vol.53, No.8, pp1189-1194, 1987
- [21] 鈴木裕, 山崎和雄, 星鐵大郎, 黒宮義久, 実時間金型加工用数値制御システムの開発 (第 2 報) — 実時間工具経路生成について — 精密工学会誌, Vol.53, No.12, pp1953-1958, 1987
- [22] 鈴木裕, 黒田豊, 坂本正史, 元村直行, 実時間金型加工用数値制御システムの開発 (第 3 報) — 逆オフセット法に基づく CAM の実時間処理化 —, 精密工学会誌, Vol.57, No.3, pp473-478, 1991
- [23] 白瀬敬一, 近藤高茂, 岡本満, 若松栄史, 荒井栄司, NC プログラミングを必要としない機械加工のための仮想倣い加工システムの開発, (自律型 NC 工作機械のための実時間工具経路生成), 日本機械学会論文集 (C 編), Vol.66, No.644, pp1368-1373, 2000
- [24] 趙 曉明, 葛東方, 堤正臣, 周曉光, メッシュ曲面加工用 CAM システムに関する研究 — 加工精度, 法線誤差および格子間隔の関係 —, 精密工学会誌, Vol. 61, No.7, pp 954-958, 1995
- [25] 加藤清敬, 入口健二, 高橋宣行, 面補間による数値制御方式, 精密工学会誌, Vol. 66, No.12, pp 1879-1884, 2000

- [26] 後藤孝行, 三好隆志, 高谷裕浩, 枝光毅彦, 曲率を考慮した B-スプライン曲線の測定点データへのあてはめ, 精密工学会誌, Vol. 60, No.7, pp 964-968, 1994
- [27] Les A. Piegl, Wayne Tiller, Computing offsets of NURBS curves and surfaces, Computer-Aided Design 31, pp147-156, 1999
- [28] 是田規之, 江川庸夫, 黒田基文, 渡部健, 伊井良治, ボールエンドミル加工における表面粗さの生成過程の解析, 精密工学会誌, Vol. 59, No.9, pp 1537-1542, 1993
- [29] 高津戸光雄, 貴志浩三, ボールエンドミルによる傾斜最適送りの実現, 日本機械学会論文集 (C 編), Vol.56, No.521, pp189-194, 1990
- [30] 趙 暁明, 堤正臣, 是田規之, 葛東方, 5 軸制御加工におけるボールエンドミルの最適傾斜角決定方法- 仕上げ面粗さを基準にした球面の場合 -, 精密工学会誌, Vol. 63, No.7, pp 992-996, 1997
- [31] 斎藤明德, 趙 暁明, 堤正臣, ボールエンドミル加工における曲面上の凹凸模様の形成方法, 精密工学会誌, Vol. 66, No.12, pp 1963-1967, 2000
- [32] Devid F. Rogers, J. Alan Adams, 河合彗 監訳, コンピュータ グラフィックス第 2 版 日刊工業新聞社, 1997
- [33] 青村茂, 自由曲面形状切削における工具干渉領域探索法, 日本機械学会論文集 (C 編), Vol.56, No.532, pp3430-1373, 2000