

論文 / 著書情報
Article / Book Information

題目(和文)	マーク流れ線図による非連続生産システムの制御とシミュレーションに関する研究
Title(English)	
著者(和文)	高橋宏治
Author(English)	Koji Takahashi
出典(和文)	学位:工学博士, 学位授与機関:東京工業大学, 報告番号:甲第1627号, 授与年月日:1985年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:甲第1627号, Conferred date:1985/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

B-43

マーク流れ線図による

非連続生産システムの制御とシミュレーションに関する研究

A Study on Control and Simulation for Discrete Production Systems

by Mark Flow Graph

1984年 12月

指導教官 長谷川 健介 教授

提出者 大学院博士課程 制御工学専攻

2050 高橋 宏治

A Study on Control and Simulation for Discrete Production Systems

by Mark Flow Graph

Kouji TAKAHASHI

(Supervised by Prof. K.HASEGAWA)

The paper deals with a systematic method of control and simulation for discrete production systems.

Discrete production systems, for example, production lines in mechanical industries, have the properties of asynchronism, ordering, concurrency and conflict, so that unsafeness and deadlock will be apt to occur in the systems. Petri net is one of the effective means to represent such systems.

For applying it to the design, analysis, and control of the systems, the guarantee of safeness and the notation of input/output of signals from/to machines should be required. The authors proposed Mark Flow Graph (MFG) as a new kind of graph deduced from the Petri net so as to satisfy the above requirements.

From the fact that the control of a system represented by MFG can be realized directly by executing the corresponding MFG as the control algorithm, a MFG-based programmable controller was developed with a microprocessor. Also a equipment was developed based on a personal computer to support synthesis and debugging of MFG, and to load the program into the controller. Using them, a control system was implemented for a miniature production system model, and their effectiveness was confirmed by executing tasks specified.

To assure the good running condition of the system and to improve the efficiency are key issues of system design. For their estimation, a quantitative simulator was developed based on modified MFG, in which the concept of time and macro description were introduced. The simulator developed was used to simulate a real production system and to modify it by the obtained result.

In order to control and simulate a discrete production system with MFG, it is necessary to represent the system in MFG. When its specification is complicated, it will become a tremendous hard work. The authors proposed an algorithm to synthesize a well-behaved Petri net which is able to generate a given firing sequence, and implemented it in the computer. As the Petri net generated can be translated into a MFG, the algorithm is the basis of the automatic synthesis of MFG.

【 論 文 要 旨 】

本研究は、機械工業などの非連続生産システムを対象として、体系的な制御技術、およびその設計や解析に必要なシミュレーション手法の確立をめざすものである。

非連続生産システムにおいてシーケンス制御が工程制御の主体であるが、現状ではほとんどの場合、その設計や解析を経験的な手法に頼っている。近年、対象とするシステムの大規模化や複雑化、また制御に用いるプログラマブルコントローラやコンピュータの発展に伴い、システムの取り扱いや制御は複雑になってきている。その際、システムは同時性・並列性をもち、動作続行不能・追突・暴走などの現象を発生する可能性がある。また、合理的な設計や運用のため階層構造をとることや、フレキシブルなシステムとするためにプログラマブルとする場合も増加している。そのため、非連続生産システムをその制御系も含めて体系的に取り扱う必要が生じてきている。

このための新しい手法として、マーク流れ線図(MFG)を提案した。MFGは、対象とするシステムを事象と状況の交互の接続としてとらえモデル化するもので、ペトリネットの概念に非連続生産システム独自の『安全性』と『信号の入出力機能の表現力』をもたせたグラフ手法である。このMFGについて定義を行ない、その規則、数式による記述、行列による記述、基本的な諸性質について述べ、また、MFGにより非連続生産システムを表現する手順を示した。

MFGはコントローラも含めたシステムのモデルであり、これと実際のシステム構成要素の動作を一致させることが制御につながる。この実現のため、制御対象の機器と「信号の授受」を行ないながらプログラムされたMFGを実行する、「MFGにもとづいたプログラマブルコントローラ」をマイクロプロセッサを用いて試作した。また、そのプログラムとなるMFGの開発やデバッグを支援するために、グラフィカルにプログラミングやシミュレーションを行ないそれをコントローラにロードする、プ

プログラムローダをパーソナルコンピュータにより作成した。なお、制御実行中にはモニタとしても機能する。これらの装置を用いて生産ラインの模型の制御実験を行なった結果、仕様どおりの制御が実現できた。この実験は小規模なものであるが、原理的には大規模な実際のシステムと同じものであり、MFGにより非連続生産システムの制御が適切に行なえることを確認した。

ところで、非連続生産システムの設計の際に重要な問題である「動作状況」や「効率」は、システムの実態に即した「定量的な挙動」から得られるが、解析的に求めることは困難でありシミュレーションが必要となる。そこで制御との一貫性も考慮し、MFGに時間の概念を導入し、また表現の繁雑さを避けるためのマクロ表現を定義し、MFGによる定量的なシミュレーションを行なうシミュレータを作成した。これを用いて、実在の非連続生産システムのシミュレーションを行ない、さらにこのデータをもとに立案した増産計画の妥当性についても考察した結果を示し、非連続生産システムのシミュレーション手法としてMFGが有効であることを確認した。

このようなMFGによる非連続生産システムの制御、あるいはシミュレーションを行なうためには、システムをMFGで記述する必要があるが、複雑な仕様の場合この作業は手間のかかるものであり自動生成が望まれる。そのために、非連続生産システムでの作業仕様を点弧系列とみなし、その点弧系列をもつペトリネットを生成するアルゴリズムを提案した。このアルゴリズムは、接続行列を拡張した重複接続行列を用いて「経路の生成」と「競合の調停」を行ない、仕様を満たすペトリネットを生成するものであり、その実行例を示した。そして、ここで生成されたペトリネットをMFGに変換できることを示し、非連続生産システムをモデル化するMFGの自動生成の基礎とした。

【 目 次 】

第 1 章 緒言	1
1. 1 研究の概要	1
1. 2 論文の構成	2
第 2 章 非連続生産システム	4
2. 1 非連続生産システムと非連続システム	4
2. 2 非連続生産システムの性質と現象	5
2. 3 非連続生産システムの構成要素の分類	8
2. 4 非連続生産システムの階層性	9
2. 5 非連続生産システムの表現手法に必要な事項	12
第 3 章 マーク流れ線図 (M F G)	13
3. 1 ベトリネットと M F G	13
3. 2 マーク流れ線図の構成と性質	13
3.2.1 マーク流れ線図の構成	13
3.2.2 マーキングとその遷移	16
3.2.3 マーク流れ線図の基本的性質	17
3. 3 マーク流れ線図の数式による記述	18
3.3.1 マーク流れ線図の数式表現	18
3.3.2 トランジションの点弧とマーキングの遷移の数式表現	19
3. 4 マーク流れ線図の行列による記述	23
3.4.1 行列の定義	23

3.4.2	マーク流れ線図の行列表現	25
3.4.3	トランジションの点弧とマーキングの遷移の行列表現	26
3.5	マーク流れ線図における構造とそれに伴う性質	27
3.5.1	ソースとシンク	27
3.5.2	競合関係および競合ボックス	28
3.5.3	パス・マーク流れ線図	30
3.6	マーク流れ線図におけるデッドロック	32
3.6.1	デッドロックと活性	32
3.6.2	デッドロックに関する条件	32
3.7	マーク流れ線図の特徴	35
第4章	MFGによる非連続生産システムの表現	36
4.1	MFGと非連続生産システムとの対応	36
4.1.1	作業工程とMFG	36
4.1.2	システム内の動作・機能とMFG	37
4.2	MFGによるシステム表現の手順	39
4.2.1	例とするシステム	39
4.2.2	工程や作業の接続	40
4.2.3	機器の接続	40
4.2.4	受け入れ容量や区間などの制御条件の規定	41
4.2.5	機器制御の記述	42
第5章	MFGによる非連続生産システムの制御	43
5.1	MFGによる制御の実現	43

5.2	MFGによる制御のための準備	43
5.2.1	アービタ(調停器)	43
5.2.2	MFGの表形式の記述	44
5.3	MFGにもとづいたプログラマブルコントローラ	47
5.3.1	動作原理	47
5.3.2	ハードウェア	51
5.4	プログラム支援装置	53
5.4.1	装置の概要	53
5.4.2	MFGの入力	54
5.4.3	MFGのシミュレーション	55
5.4.4	MFGコントローラへの プログラムのローディングとモニタ	58
5.5	実験	59
5.5.1	実験システム	59
5.5.2	実験の手順	63
5.5.3	実験結果	64
5.6	今後の課題	64
第6章	MFGによる非連続生産システムのシミュレーション	66
6.1	非連続生産システムのシミュレーションとMFG	66
6.2	MFGへの時間の導入	67
6.2.1	時間の導入	67
6.2.2	時間ボックス	67
6.2.3	時限トランジション	69

6.2.4	時間の導入によるMFGの性質の変化	70
6.3	MFGへのマクロなボックスの導入	70
6.3.1	マクロなボックス	70
6.3.2	容量ボックス	71
6.3.3	パレタイジングボックス	72
6.3.4	デパレタイジングボックス	73
6.3.5	表現例	74
6.4	シミュレータ	76
6.5	実例	79
6.5.1	対象とした非連続生産システム	79
6.5.2	シミュレーション(I)	83
6.5.3	シミュレーション(II)	85
6.6	今後の課題	88
 第7章 非連続生産システムをモデル化するMFGの自動生成の基礎		89
7.1	MFGの自動生成	89
7.2	取り扱う問題	89
7.3	重複接続行列	90
7.4	基本経路の生成	91
7.4.1	基本手順	92
7.4.2	重複の取り扱い(I)	94
7.4.3	重複の取り扱い(II)	95
7.4.4	並列区間の取り扱い	96
7.5	競合の調停	98

7. 6	実行例	1 0 0
7. 7	M F G への変換	1 0 4
7. 8	今後の課題	1 0 4
第 8 章 結言		1 0 6
8. 1	本研究のまとめ	1 0 6
8. 2	今後の課題	1 0 9
【謝辞】		1 1 1
【参考文献】		1 1 2

第 1 章 緒 言

1.1 研究の概要

生産におけるオートメーションをながめてみると、その対象として化学プラントなどの連続生産システムと、機械工業などの非連続生産システムに大別できる。また、これらのシステムにおける工程の制御技術について考えてみると、連続生産システムではフィードバック制御が主体であるのに対し、非連続生産システムではシーケンス制御が主体となっている。本研究では、後者の非連続生産システムを対象として、体系的な制御技術、およびその設計・解析のためのシミュレーション手法の確立をめざしている。

非連続生産システムの制御技術の中心となるシーケンス制御は、オートメーションの基礎的な技術であり、古くから産業や社会のいたる所で用いられている。しかし、その設計や解析に対して従来は論理代数や順序回路論を基礎とした手法があるにすぎず、ほとんどの場合、実用化の過程で積み上げられた定石や経験的な手法に頼っているのが現状である。1-1)

ところが近年、対象とするシステムが大規模化かつ複雑化し、また制御に用いるプログラマブルコントローラやコンピュータも発展し、システムの取り扱いや制御はますます複雑なものとなってきている。その際、システムは同時性・並列性をもち、比較的簡単な構造のシステムの場合にはおこらない動作続行不能・追突・暴走などの現象を発生する可能性が生じている。これらの発生をなくし設計や運用の合理性を考えるため、構成上階層構造をとることも多い。さらに、最近では多品種少量生産の要求に対応して、フレキシブルなシステムとするためにプログラマブルである場合が増加している。

そのため、非連続生産システムをその制御系も含めて体系的に取り扱う必要が生じてきており、これに対応するための新しい手法としてマーク流れ線図(MFG)を導入する。本研究では特に、MFGによる非連続生産システムの制御とシミュレーションの手法を確立することを主たるテーマとしている。さらに、非連続生産システムの制御系の自動設計への足掛りとして、システムをモデル化するMFGの自動生成の基礎についても言及する。

1.2 論文の構成

第2章では、本研究で対象とする非連続生産システムを event-condition system としてとらえ、これがもっている非同期性・順序性・並列性などの性質や、追突・動作続行不能などの現象について考察する。また、システムのもつ階層性についての検討や、システム構成要素の機能の分類を行なう。これらにもとづいて、非連続生産システムの表現手法に要求される事項を整理する。

第3章では、前章での考察にもとづいてペトリネット(Petri net)の概念にシーケンス制御系独自の性質や機能を付加し、本研究の中心手法であるマーク流れ線図の定義を行ない、その規則と諸性質について述べる。

第4章では、MFGで非連続生産システムを表現する手順について述べる。

第5章では、MFGにより直接に非連続生産システムの制御を行なうために、マイクロプロセッサを用いてMFGにもとづいたプログラマブルコントローラを試作する。また、このコントローラのためのプログラムローダとモニタをパーソナルコンピュー

タにより作成し、グラフィカルにプログラミングやその検証が行なえる制御装置とする。これを用いて、非連続生産システムの模型の制御実験を行ない、MFGが制御手法として有効であることを確認する。

第6章では、非連続生産システムの設計や解析に必要な定量的なデータを得るためにMFGに時間の概念を導入し、また表現の繁雑さを避けるためのマクロなボックスを定義し、これにもとづいたシミュレータを作成する。これを用いて、実在の非連続生産システムのシミュレーションを行ない、このデータをもとに増産計画を立案しその妥当性について考察した結果を示す。この例より、非連続生産システムのシミュレーション手法としてMFGが有効であることを確認する。

第7章では、非連続生産システムでの作業仕様を点弧系列とみなし、その点弧系列をもつペトリネットを生成するアルゴリズムを提案し、実行例を示す。これがMFGに変換できることより、非連続生産システムをモデル化するMFGの自動生成の基礎とする。

第8章では、本研究のまとめを行ない、今後の課題や応用などについて述べる。

第 2 章 非連続生産システム

2.1 非連続生産システムと非連続システム

機械工業などの生産システムでは、その生産工程は配列された機械に依存する各種工程の接続により構成されており、システムを流れる各被加工物はそれらの工程を順次進行していく。その際、ある工程から次の工程に進むにあたり、「その工程が終了した後に次の工程が開始される」という工程間の区切りが必ず存在している。このような『区切り』という非連続性をもつことから、『非連続生産システム』とよばれる。

この非連続生産システムにおける動作の進行について、「・・・加工－組立－・・・」を例として考えてみると、

・・・

加工開始 という事象の発生

加工中 という状況の保持

加工終了 という事象の発生

組立待ち という状況の保持

組立開始 という事象の発生

組立中 という状況の保持

組立終了 という事象の発生

・・・

という『事象』の発生と『状況』の保持が交互に現れていることがわかる。このように事象と状況の交互の接続で活動しているシステムを、『非連続システム (event - condition system)』というが、非連続生産システムはその一つである。なお、事象の発生は状況によって決まる。したがって、「状況」を事象発生のための「条件」と

みることができる。

2.2 非連続生産システムの性質と現象

非連続生産システムは、つぎにあげる非連続システムとしての性質をもつ。

【非同期性】

条件さえ整えば、事象はいつでも発生する。同期は制御の都合で与えられており、システムの挙動は本来非同期である。

【順序性】

ある状況に入る前に発生すべき事象があり、その状況の後に発生すべき事象がある。また、ある事象が発生するための前提条件があり、その事象が発生した後の後提条件がある。このように、事象あるいは状況の間に前後関係がある。

【並列性】

1つのシステム内で複数の状況が保持され、それぞれが独立に別々の事象を発生しつぎの状況に移行することができる。

【競合性】

ある状況の移行が複数の事象に関係し、これらの発生の早遅により異なる状況に至る。

また、特異な現象として、

【追突】

ある状況が保持されているにもかかわらず、さらにその直前の事象が発生し、多重にその状況を保持しようとする現象。なお、追突をおこさないことを『安全』という。

【続行不能】

いくつかの状況と事象がからみあい、新たな事象を発生できなくなり、その状況を抜け出すことができない現象。システムは動作の続行が不可能である。

などをおこすことがある。

これらの現象を具体的に、Fig.2-1 に示す加工機、ローディング／アンローディング両用ロボット、搬入・搬出コンベアで構成されたシステムを例として考えてみる。

(1) Fig.2-2(a)のように、ワークピースAがローディングを待っているにもかかわらず次のワークピースBが進入してくると、Aにぶつかり規定の位置をずらす。これが追突である。

(2) Fig.2-2(b)のように、加工機で加工中にロボットが次にロードするワークピースをつかんで待機していると、加工が終了してもワークピースのアンローディングが行なえず、システムが停止し先に進めない。これが続行不能である。

非連続生産システムでは、これらの現象がおこることは許されないことである。

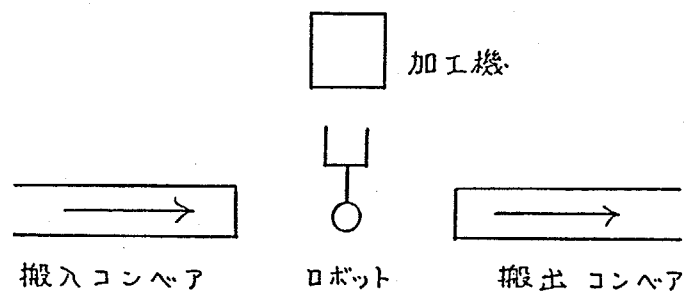
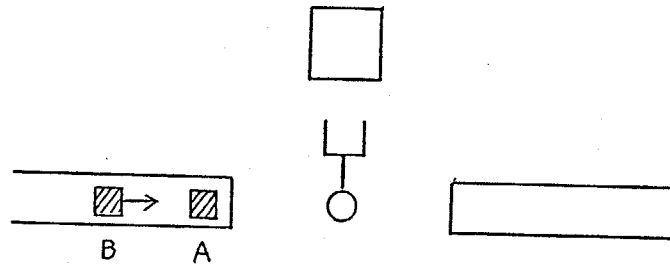
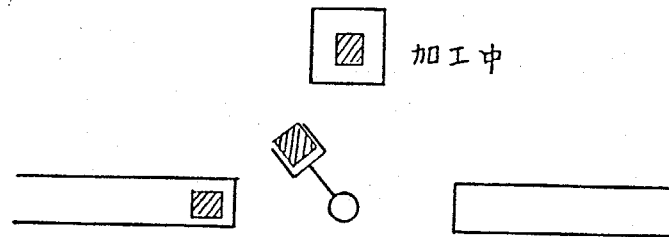


Fig.2-1 非連続生産システムの例



(a) 追突



(b) 続行不能

Fig.2-2 追突と続行不能

2.3 非連続生産システムの構成要素の分類

非連続生産システムの構成要素における動作や機能について、つぎのように分類できる。

【操作 (operation)】

加工や処理など、被加工物に対して作業を行なうことを『操作』とよぶ。搬送も操作の1つとみなせる。

【結合 (join)】

組立作業などでは、いくつかの部品を合せて1つのものにする。このような動作を『結合』とよぶ。結合は、それに必要な前提条件がすべて満たされると開始される。そして『結合中』という状況に移る。

【分割 (division)】

組立作業とは逆に、1つの材料からいくつかの品物を切り出す作業がある。このような動作を『分割』とよぶ。分割では、『分割中』という状況が終了し分割が完了すると、その後戻条件がすべて満たされる。

【集合 (gathering)】

いくつかのワークピースを1枚のパレットに載せ、以後それを単位として扱うという動作がある。このような動作を『集合』とよぶ。集合では、あらかじめ設定された回数だけ前提事象が発生するまで待ち、その数に達するとその後戻事象が発生することができる。そして、後戻事象が発生するまでは、前提事象の発生は抑止される。

【分離 (separation)】

集合とは逆に、1枚のパレット上にあるいくつかのワークピースを1つずつ取り出すような動作がある。このような動作を『分離』とよぶ。分離ではその前提事象が1

回発生すると、あらかじめ設定された回数の後堤事象を発生させることができ、これが完了するまで前提事象の発生は抑止される。

【分岐 (branching)】

作業や工程の進行において、その行き先がいくつもありそのうちのどれか1つを選ぶという場合がある。このような動作を『分岐』とよぶ。

【合流 (junction)】

分岐とは逆に、ある1つの工程がいくつかの工程の受け入れ先であり、そのうちのどれか1つを選ぶという場合がある。このような動作を『合流』とよぶ。

【容量 (capacity)】

バッファのように、流入側と流出側との間に入り、複数の“もの”が滞留できる要素を『容量』とよぶ。

【波滞 (jam)】

ある被加工物が、その工程による処理は終了し次の工程に移行できるにもかかわらず、先の工程が詰っていて前に進めないでいるような状態を『波滞』とよぶ。

【停留 (pause)】

ある被加工物が、工程により操作を受けるために留まっているような状態を『停留』とよぶ。ものの流れを中心にみた場合は流れの一時停止であるが、工程や作業についてみると動作中である。

2.4 非連続生産システムの階層性

非連続生産システムにおける『状況』をながめてみると、その中がまた『事象と状況の接続』となっている場合がみられる。たとえば、ある『加工作業中』という状況は、

ワークピース 取付け開始
ワークピース 取付け中
ワークピース 取付け完了
加工機 始動待ち
加工機 始動
加工機 運転中
加工機 停止
ワークピース 取外し待ち
ワークピース 取外し開始
ワークピース 取外し中
ワークピース 取外し完了

というように、さらに詳細な事象と状況の接続で表現される。

また、システム全体の動作についても、

システムの動作は、いくつかのラインの動作により構成されている
各ラインの動作は、いくつかのステーション動作により構成されている
各ステーションの動作は、いくつかの機械の動作により構成されている
各機械の動作は、いくつかの動作部の動作により構成されている
.....

というように、各段階をおって詳細なものへと分解される。 [fig.2-3]

非連続生産システムは、このような『階層性』をもっている。

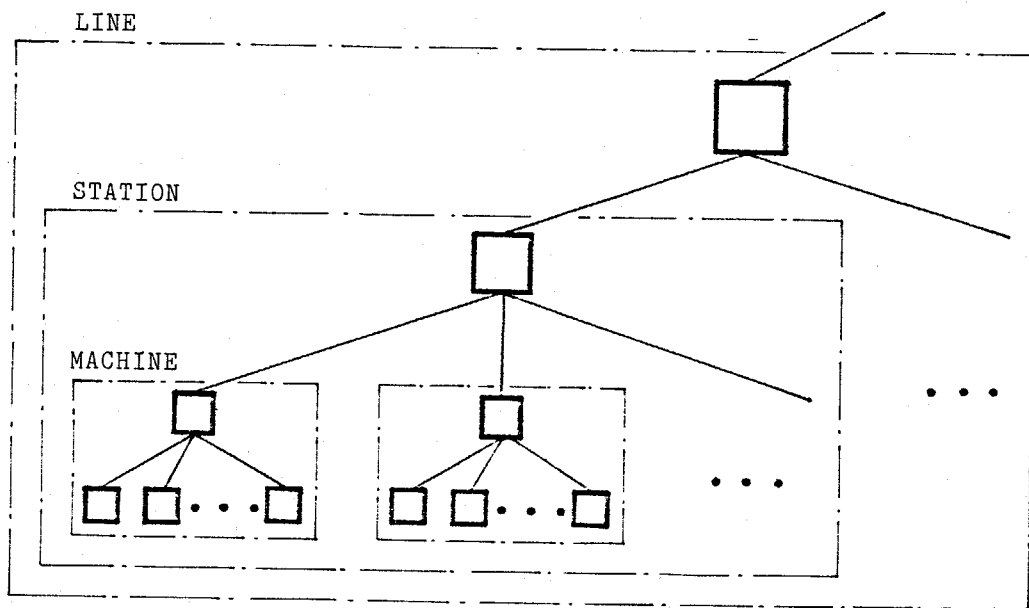


Fig.2-3 非連続生産システムの階層性

2.5 非連続生産システムの表現手法に必要な事項

以上の非連続生産システムを的確にわかりやすく表現するためには、つぎのような条件を満たす表現手法が必要となる。

- (1) システムのもつ性質、機能、動作などを正確に表現できること
- (2) 階層性に対応して、概念レベルから詳細レベルまで表現できること
- (3) 関係者間の情報交換に有効であること
- (4) 設計・解析に有効な数学的表現ができること

また、非連続生産システムの制御も含めてその表現を行なう場合には、それに加えて、

- (5) 追突がおこらない すなわち『安全』であること
- (6) 機器との信号の入出力の表現ができること

が要求される。

第 3 章 マーク流れ線図 (M F G)

3.1 ペトリネットと M F G

前章で述べた非連続生産システムの表現手法に要求される条件のうち、非連続システムにかかわる条件を満たすものの一つとして、非同期・同時進行的な要素からなるシステムのモデル化のために作られたペトリネット (Petri net) がある。3-1) この概念を非連続生産システムの制御や設計・解析に導入すると、非連続生産システムを事象と状況の接続によるグラフでモデル化し、このモデルが動くことがシステムの挙動に対応することになる。その際、ペトリネットのもつ解析能力やモデル化能力にくわえて、非連続生産システムの独自の性質の表現能力が必要となる。そこで、ペトリネットの概念に「安全性」と「信号の入出力機能の表現力」をもたせた新しいグラフ手法として、マーク流れ線図を提案する。

3.2 マーク流れ線図の構成と性質

3.2.1 マーク流れ線図の構成

マーク流れ線図 (Mark Flow Graph 以下 MFG) は、つぎの構成要素からなる。

(a) ボックス (box) 状況を表現するもので、グラフ上では Fig.3-1(a) の四角形で示される。

(b) トランジション (transition) 事象を表現するもので、グラフ上では Fig.3-1(b) の棒で示される。

(c) 方向枝 (directed arc) 状況とそれを区切る事象の前後関係を表現するためにボックスとトランジションを接続するもので、グラフ上では Fig.3-1(c) の矢印で

示される。なお、ボックスあるいはトランジションからみて、方向枝の出る側を出力側、入る側を入力側という。

(d) マーク (mark) 状況が保持されていることを表現するもので、グラフ上では Fig.3-1(d) のように黒丸で示されボックス内におかれる。

(e) ゲート信号枝 (gating arc) トランジションに対応する事象の発生を許可あるいは抑止するもので、許可枝と抑止枝があり、さらに信号の出所により内部信号枝と外部信号枝に分類される。

許可枝 (permissive arc) は、Fig.3-1(e)に示すようにトランジションに付く黒丸を先端にもつ枝で、信号が "1" のときに接続されているトランジションに対応する事象の発生を他の許可条件との論理積の形で許可する。

抑止枝 (inhibitor arc) は、Fig.3-1(f)の白丸の付いた枝で示され、信号が "1" のときに接続されているトランジションに対応する事象の発生を他の抑止条件との論理和の形で抑止する。

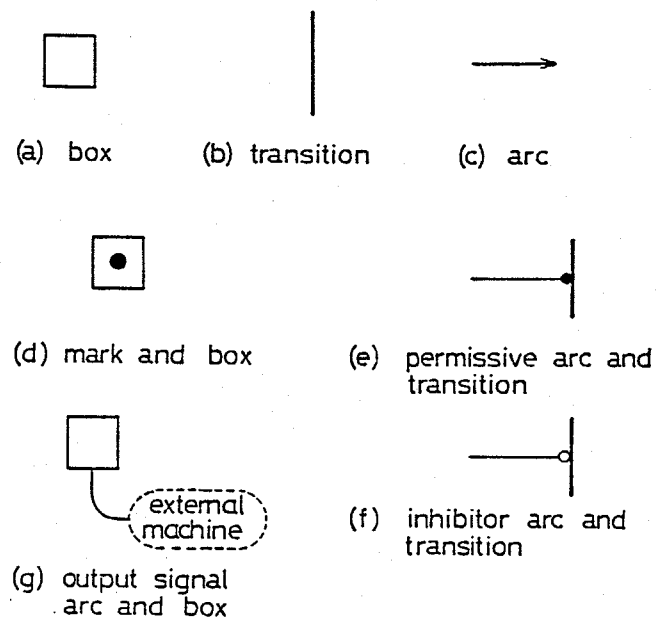


Fig.3-1 マーク流れ線図の構成要素

内部信号枝 (internal signal arc) の信号源はボックスであり、ボックスにマークがある場合は "1"、マークがない場合は "0" である。

外部信号枝 (external signal arc) は、信号源がグラフの外部にあるもの、すなわち外部の機器からの 2 値信号入力を示すものである。

(f) 出力信号枝 (output signal arc) は、ボックスからグラフの外部の機器に 2 値信号を出力するものであり、Fig.3-1(g) のようにボックスと外部機器を線で結ぶことにより表わされる。信号は、出力されるボックスにマークがあるとき "1"、ないときは "0" である。

ボックスとトランジションは、必ず交互に方向枝で接続される。また、ボックスおよびトランジションに出入りする枝の数には制限がない。ただし、1 組のトランジション・ボックス間あるいは 1 組のトランジション・外部信号源間の枝はたかだか 1 本である。 Fig.3-2 に例を示す。

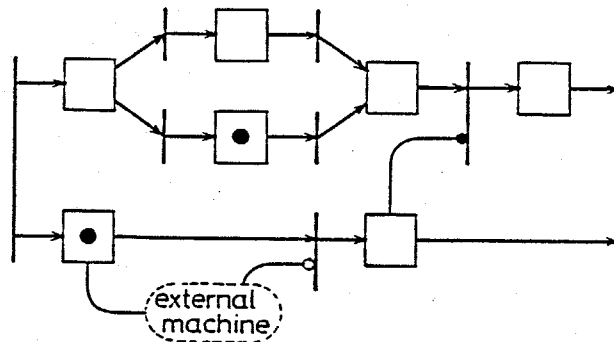


Fig.3-2 マーク流れ線図の例

3.2.2 マーキングとその遷移

【マーキング】

MFG では、システムの状態はマークの配置により表現される。これをマーキング (marking)、また初期状態のマーキングを初期マーキング (initial marking) という。

初期マーキングにおいて、ボックス内にはたかだか一つのマークしかおいてはならない。なお、マーク一つもない初期マーキングを0-初期マーキングという。システムの挙動は、事象の発生による状態の遷移で示される。これを、MFG で表わすために、事象の発生に対応するトランジションの点弧の規則をつぎのように定めている。

【可点弧】

あるトランジションが、

- (1) マークの入った出力側ボックスをもたない
- (2) マークのっていない入力側ボックスをもたない
- (3) 許可状態にない内部許可枝をもたない
- (4) 抑止状態にある内部抑止枝をもたない

という条件をすべて満たしているとき、そのトランジションは可点弧である (enabled) という。なお、可点弧でないことを非可点弧である (disabled) という。

【点弧】

可点弧であるトランジションが、許可状態にない外部許可枝および抑止状態にある外部抑止枝をもたないとき、そのトランジションは特別な場合(注)を除き直ちに点弧する (fire)。

(注) 競合のある場合については 3.5 で述べる。また、時間的遅れを与えることもある。

点弧すると、そのトランジションのすべての入力側ボックス内にあるマークは消滅し、かつすべての出力側ボックス内にマークが発生する。Fig.3-3に例を示す。

なお、点弧は無限小時間内に行なわれるものとする。よって、同時に複数個のトランジションが点弧することはないので、MFG内の点弧は時間的に不連続であり前後関係がある。この系列を点弧系列(firing sequence)と呼び、これにより区切られた時間の順序づけをタイムシーケンス(time sequence)と呼ぶ。

3.2.3 マーク流れ線図の基本的性質

以上のような規則により、MFG はつぎの基本的性質をもつ。

【安全】

初期マーキングにおいては、ボックス内にはたかだか一つのマークしかおくことが許されていない。また、可点弧および点弧に関する条件より、すでにマークをもつボックスにさらにマークを発生することはない。よって、任意の時点ですべてのボックス内にはたかだか一つのマークしか存在しない。つまり、MFG は本質的に安全(safe) 3-2)であり、MFG で表わされるシステムに追突がおこることはない。

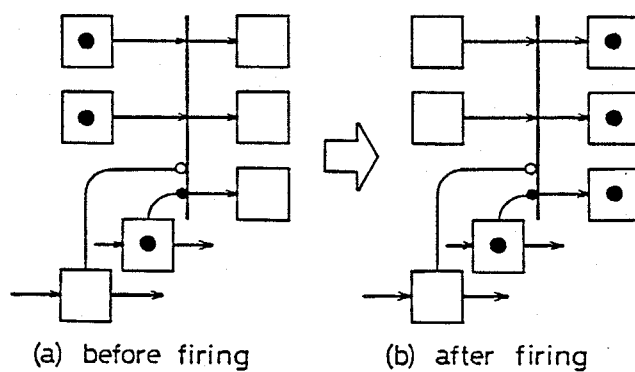


Fig.3-3 点弧

3.3 マーク流れ線図の数式による記述

上で述べた MFG およびマークの挙動の数式による記述を以下に示す。

3.3.1 マーク流れ線図の数式表現

マーク流れ線図 MFG は、6-タプル

$$\text{MFG} = (B, T, A, G_I, G_E, S)$$

で表わされる 2 部有向グラフ (bipartite directed graph) である。ここで、

$B = \{B_1, \dots, B_i\} (i \geq 1)$ はボックスの集合

$T = \{T_1, \dots, T_j\} (j \geq 1)$ はトランジションの集合

$A = A_I \cup A_O$ は方向枝の集合

ただし、

$A_I = \{A_{Ik}, \dots, A_{Ik}\} (k \geq 0)$ はトランジションへの入力方向枝の集合

$A_O = \{A_{Ol}, \dots, A_{Ol}\} (l \geq 0)$ はトランジションからの出力方向枝の集合

$G_I = \{G_{Im}, \dots, G_{Im}\} (m \geq 0)$ は内部信号枝の集合

$G_E = \{G_{En}, \dots, G_{En}\} (n \geq 0)$ は外部信号枝の集合

$S = \{S_1, \dots, S_p\} (p \geq 0)$ は出力信号枝の集合

である。また、関数 $C(x)$ をノード y からノード z に向かう枝 x の入出力の接続関係

(x, y) を与えるものとし、

集合 $U = \{u_i | i=1 \sim m\}$ と集合 $V = \{v_j | j=1 \sim n\}$ の

直積 $U \times V$ を

順序対 (u_i, v_j) の全体集合 $\{(u_i, v_j) | i=1 \sim m, j=1 \sim n\}$

を与えるものとする、

$$A'_i = \{C(A_{iq}) | q=1 \sim k\} \subseteq B \times T$$

$$A'_0 = \{C(A_{0r}) | r=1 \sim l\} \subseteq T \times B$$

$$G'_i = \{C(G_{i\cdot}) | u=1 \sim m\} \subseteq B \times T$$

$$G'_E = \{C(G_{E\cdot}) | v=1 \sim n\} \subseteq D \times T$$

$$S' = \{C(S_w) | w=1 \sim p\} \subseteq B \times M$$

ただし、

D は外部信号源の集合

M は外部受信端の集合

である。ここで、 A_i, A_0, G_i, G_E, S の要素はその機能も含めた枝を意味していたのに対し、 $A'_i, A'_0, G'_i, G'_E, S'$ の要素は枝の入出力の接続関係を表す順序対である。このとき、

$X = \{(a, b)\}$ に対して $X^{-1} = \{(b, a)\}$ とすると、3.2.1で述べた枝に関する制限より

$$A'_i \cap A'^{-1}_0 = A'_i \cap G'_i = A'^{-1}_0 \cap G'_i = \phi$$

でなくてはならない。

マーキング μ は、

$$\mu: B \rightarrow \{0, 1\}$$

なる関数である。なお、初期マーキングは μ_0 とする。

3.3.2 トランジションの点弧とマーキングの遷移の数式表現

ボックス B_i のマークの有無を示す論理変数をボックス変数 b_i とおき、

$$b_i = \begin{cases} 0: \text{マークがないとき} \\ 1: \text{マークがあるとき} \end{cases}$$

とする。また、トランジション T_j が可点弧であるか否かを示す論理変数をトランジション変数 t_j とおき、

$$t_j = \begin{cases} 0: \text{可点弧でないとき} \\ 1: \text{可点弧であるとき} \end{cases}$$

とする。さらに、ゲート信号枝 G_1 による信号を表す論理変数を g_1 とする。

任意のトランジション T_j は、Fig.3-4に示すように入力側に M 個のボックス、出力側に N 個のボックスが方向枝により接続され、 Q 本の内部許可枝、 R 本の内部抑止枝、 U 本の外部許可枝、 V 本の外部抑止枝をもつ。トランジション T_j について、

- $b_{j,m}^i (m=1 \sim M)$ を入力側のボックス変数
- $b_{j,n}^o (n=1 \sim N)$ を出力側のボックス変数
- $q_{j,q}^{iP} (q=1 \sim Q)$ を内部許可枝の信号変数
- $q_{j,r}^{iI} (r=1 \sim R)$ を内部抑止枝の信号変数
- $q_{j,u}^{eP} (u=1 \sim U)$ を外部許可枝の信号変数
- $q_{j,v}^{eI} (v=1 \sim V)$ を外部抑止枝の信号変数
- k をタイムシーケンス番号

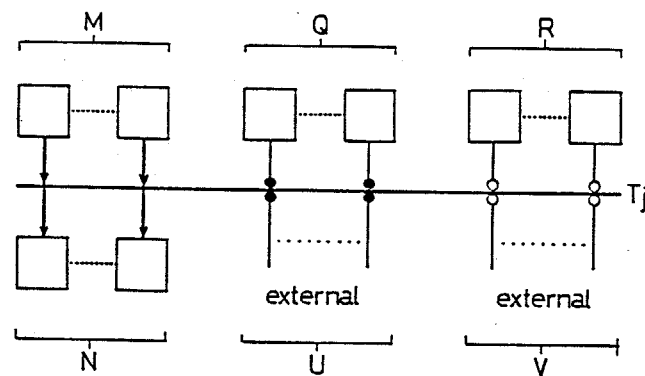


Fig.3-4 トランジションへの接続

とし、

\vee を論理変数の論理和

\wedge を論理変数の論理積

$$\bigcap_{i=1}^m a_i = a_1 \wedge a_2 \wedge \dots \wedge a_m,$$

$$\bigcup_{i=1}^m a_i = a_1 \vee a_2 \vee \dots \vee a_m$$

とすると、可点弧条件式および外部条件式は、

$$t_j(k) = \bigcap_{m=1}^M b'_{j,m}(k) \wedge \bigcap_{n=1}^N \overline{b''_{j,n}(k)} \wedge \bigcap_{q=1}^Q g_{j',q}(k) \wedge \bigcap_{r=1}^R \overline{g_{j'',r}(k)} \quad (3-1)$$

$$g_j^F = \bigcap_{u=1}^U g_{j',u}^{FP} \wedge \bigcap_{v=1}^V \overline{g_{j'',v}^{FT}} \quad (3-2)$$

となる。また、 T_j の点弧によるマーク消滅式およびマーク発生式は、

$$\begin{cases} b'_{j,m}(k+1) = b'_{j,m}(k) \wedge \overline{(t_j(k) \wedge g_j^F)} & (m=1 \sim M) \\ b''_{j,n}(k+1) = b''_{j,n}(k) \vee (t_j(k) \wedge g_j^F) & (n=1 \sim N) \end{cases} \quad (3-3)$$

で表わされる。

さてここで、あるトランジションの可点弧条件への入力側・出力側ボックスの寄与を考えてみると、(3-1)式より

$$t'_j(k) = \bigcap_{m=0}^M b'_{j,m}(k) \wedge \bigcap_{n=0}^N \overline{b''_{j,n}(k)} \quad (3-4)$$

となる。ところで、ペトリネットにおいてプレース P_i 内のトークンの有無を表わす論理変数を p_i とおき、あるトランジションの可点弧条件を(3-1)式と同形式で表わすと、

$$t_j(k) = \bigcap_{l=0}^L p_{j,l}(k) \quad (3-5)$$

となる。そこで、 $b_{j_m}^1$ を表わす、すなわち $b_{j_m}^1$ 個のトークンをもつプレースを $P_{j_m}^1$ 、 $\overline{b_{j_n}^0}$ を表わすプレースを $P_{j_n}^2$ とし、これらをトランジション T_j の入力プレースとしたペトリネットを考えると、

$$t_j(k) = \prod_{m=0}^M p_{j_m}^1(k) \wedge \prod_{n=0}^N p_{j_n}^2(k) \quad (3-6)$$

となり、(3-4)式と等しくなる。

$$(\because p_{j_m}^1 = b_{j_m}^1, p_{j_n}^2 = \overline{b_{j_n}^0})$$

いま、ボックス B がトランジション T_j の出力側および T_{j+1} の入力側に接続されているとすると、これらのプレースは b, \overline{b} を表現するものであり、また (3-6) 式を満たすので Fig.3-5 のように接続される。これより、ボックスは入力側トランジションの制御と出力側トランジションの制御の二面性をもっていることがわかる。

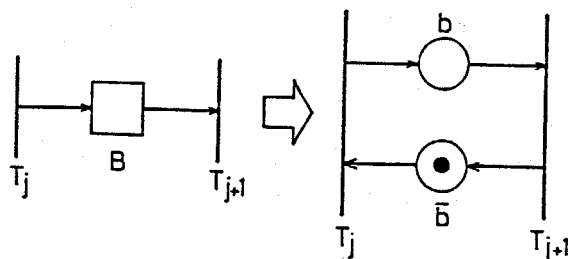


Fig.3-5 ボックスと等価なペトリネット

3.4 マーク流れ線図の行列による記述

MFG は、行列を用いてグラフ全体に関して一括して表現することもできる。

3.4.1. 行列の定義

いま MFG が、 m 個のボックスと n 個のトランジションから構成され、 q 個の外部信号源および r 個の外部受信端をもっているとする。これらの接続関係を表わす行列をつぎのように定義する。なお、3.2.1 に述べた枝に関する制限より、その要素は 0 か 1 となる。

【入力側接続行列 C-】

各トランジションの入力側に接続されるボックスを表わす $n \times m$ の行列で、

$$C-[j, i] = \begin{cases} 1 & : \text{トランジション } T_j \text{ の入力側にボックス } B_i \text{ が接続されている} \\ 0 & : T_j \text{ の入力側に } B_i \text{ は接続されていない} \end{cases}$$

とする。

【出力側接続行列 C+】

各トランジションの出力側に接続されるボックスを表わす $n \times m$ の行列で、

$$C+[j, i] = \begin{cases} 1 & : T_j \text{ の出力側に } B_i \text{ が接続されている} \\ 0 & : T_j \text{ の出力側に } B_i \text{ は接続されていない} \end{cases}$$

とする。

【内部許可枝行列 D+】

各トランジションへの内部許可枝の信号源となるボックスを表わす $n \times m$ の行列で、

$$D+[j, i] = \begin{cases} 1 & : T_j \text{ への内部許可枝が } B_i \text{ から接続されている} \end{cases}$$

$$\left\{ \begin{array}{l} 0 : T_j \text{ への内部許可枝が } B_i \text{ から接続されていない} \end{array} \right.$$

とする。

【内部抑止枝行列 D-】

各トランジションへの内部抑止枝の信号源となるボックスを表わす $n \times m$ の行列で、

$$D-[j,i] = \left\{ \begin{array}{l} 1 : T_j \text{ への内部抑止枝が } B_i \text{ から接続されている} \\ 0 : T_j \text{ への内部抑止枝が } B_i \text{ から接続されていない} \end{array} \right.$$

とする。

【外部許可枝行列 H+】

各トランジションへの外部許可枝の信号源を表わす $n \times q$ の行列で、

$$H+[j,i] = \left\{ \begin{array}{l} 1 : T_j \text{ への外部許可枝が外部信号源 } M_i \text{ から接続されている} \\ 0 : T_j \text{ への外部許可枝が } M_i \text{ から接続されていない} \end{array} \right.$$

とする。

【外部抑止枝行列 H-】

各トランジションへの外部抑止枝の信号源を表わす $n \times q$ の行列で、

$$H-[j,i] = \left\{ \begin{array}{l} 1 : T_j \text{ への外部抑止枝が } M_i \text{ から接続されている} \\ 0 : T_j \text{ への外部抑止枝が } M_i \text{ から接続されていない} \end{array} \right.$$

とする。

【出力信号枝行列 Q】

各ボックスからの信号出力先を表わす $r \times m$ の行列で、

$$Q[j,i] = \left\{ \begin{array}{l} 1 : B_i \text{ から出力信号枝が外部受信端 } U_j \text{ に接続されている} \\ 0 : B_i \text{ から出力信号枝が } U_j \text{ に接続されていない} \end{array} \right.$$

とする。

3.4.2 マーク流れ線図の行列表現

3.4.1 で定義した行列をもとに下記の行列を導入する。

【接続行列 C】

$$C = C+ - C-$$

と定義することにより、トランジションとボックスの接続関係が記述される。なお、

C はつぎのように解釈する。

$$C[j, i] = \begin{cases} -1 : T_j \text{ の入力側に } B_i \text{ が接続されている} \\ 0 : T_j \text{ と } B_i \text{ とは接続されていない} \\ 1 : T_j \text{ の出力側に } B_i \text{ が接続されている} \end{cases}$$

【内部ゲート枝行列 D】

$$D = D+ - D-$$

と定義することにより、各トランジションへの内部信号枝の信号源となるボックスが記述される。なお、D はつぎのように解釈する。

$$D[j, i] = \begin{cases} -1 : T_j \text{ への内部抑止枝が } B_i \text{ から接続されている} \\ 0 : T_j \text{ へは内部信号枝が } B_i \text{ から接続されていない} \\ 1 : T_j \text{ への内部許可枝が } B_i \text{ から接続されている} \end{cases}$$

【外部ゲート枝行列 H】

$$H = H+ - H-$$

と定義することにより、各トランジションへの外部信号枝の信号源が記述される。な

お、H はつぎのように解釈する。

$$H[j, i] = \begin{cases} -1 : T_j \text{ への外部抑止枝が } M_i \text{ から接続されている} \\ 0 : T_j \text{ へは外部信号枝が } M_i \text{ から接続されていない} \\ 1 : T_j \text{ への外部許可枝が } M_i \text{ から接続されている。} \end{cases}$$

以上の行列と、さきに定義した出力信号枝行列により MFG が表現される。

3.4.3 トランジションの点弧とマーキングの遷移の行列表現

まず、論理変数の要素をもつ行列の演算子として、"&", "*"をつぎのように定義する。

A, B, C を $m \times n$ の論理変数行列、D を $n \times p$ の論理変数行列とするとき、

(1) $C = A \ \& \ B$ にたいし

$$C[i, j] = A[i, j] \ \& \ B[i, j]$$

(2) $E = A \ * \ D$ にたいし

$$E[i, j] = \bigcup_{k=1}^m (A[i, k] \ \& \ D[k, j])$$

と演算の規則を定める。

これより、可点弧なトランジションを表わす可点弧ベクトル e_1 および 外部ゲート条件が成立しているトランジションを表わす外部ゲートベクトル e_2 は、

$$e_1 = \overline{(C^- * \overline{\mu^t})} \ \& \ \overline{(C^+ * \mu^t)} \ \& \ \overline{(D^+ * \overline{\mu^t})} \ \& \ \overline{(D^- * \mu^t)}$$
$$e_2 = \overline{(H^+ * \overline{\mu^t})} \ \& \ \overline{(H^- * \mu^t)}$$

となる。

また、点弧するトランジションを表わす点弧ベクトル f は、

$$f = e_1 \ \& \ e_2$$

となり、マーキングの遷移はつぎの式で与えられる。

$$\mu' = \mu + (f^t * C)$$

ただし、競合 [3.5.2参照] が生じている場合には、 f において競合するトランジションのうち、どれか1つが選択されなくてはならない。

3.5 マーク流れ線図における構造とそれに伴う性質

3.5.1 ソースとシンク

MFG のトランジション T_j の入力側ボックスの集合および出力側ボックスの集合を、それぞれ $\cdot T_j, T_j^{\cdot}$ で表わす。また、ボックス B_i の入力側トランジションの集合および出力側トランジションの集合を、それぞれ $\cdot B_i, B_i^{\cdot}$ で表わす。このとき、

$\cdot T_j = \phi$ なる T_j をトランジション・ソース

$\cdot B_i = \phi$ なる B_i をボックス・ソース

といい、これらを総称してソース (source) という。

$T_j^{\cdot} = \phi$ なる T_j をトランジション・シンク

$B_i^{\cdot} = \phi$ なる B_i をボックス・シンク

といい、これらを総称してシンク (sink) という。これらの例を Fig.3-6に示す。

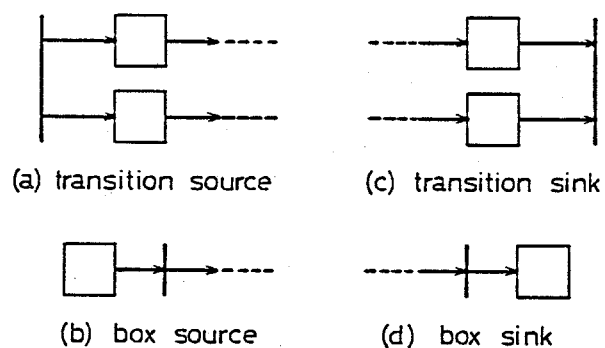


Fig.3-6 ソースとシンク

3.5.2 競合関係および競合ボックス

$T_1 \cap T_2 \cap \dots \cap T_n = \emptyset$ なるトランジション T_1, T_2, \dots, T_n

または、

$T_1 \cap T_2 \cap \dots \cap T_n = \emptyset$ なるトランジション T_1, T_2, \dots, T_n

は、それぞれ互いに競合関係にあるという。

また、集合 S の要素の個数を $\text{card}(S)$ で表わすことにすると、

$\text{card}(B_i) > 1$ あるいは $\text{card}(B_o) > 1$

なるボックスを競合ボックス (conflict box) とよび、競合ボックスの入力側で競合関係にあるときを入力競合ボックス、出力側の場合を出力競合ボックスという。Fig.3-7に例を示す。

競合関係にあるトランジションのうち複数個が同時に点弧しようとするとき、これらのトランジションは競合し (conflict)、つぎのようにふるまう。

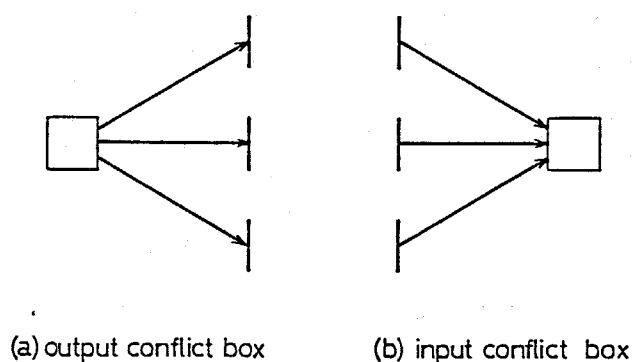


Fig.3-7 競合ボックス

《定理 3-1》 競合定理

ある競合ボックスに接続されているトランジションが競合するとき、競合するトランジションのうち任意の一つのみが点弧でき、その一つが点弧すると残りは非可点弧となる。

(証明)

点弧は無窮小時間内に行なわれ同時に複数個のトランジションが点弧することはないので、まず競合するトランジションのうち任意の一つが点弧する。この点弧により、その競合ボックスにマークが発生(入力競合ボックスの場合)あるいは消滅(出力競合ボックスの場合)し、他のトランジションの可点弧条件は満たされなくなる。■

このように、競合ボックスはマークの流れの分岐や合流を行なう機能をもっている。

Fig.3-8 に例を示す。

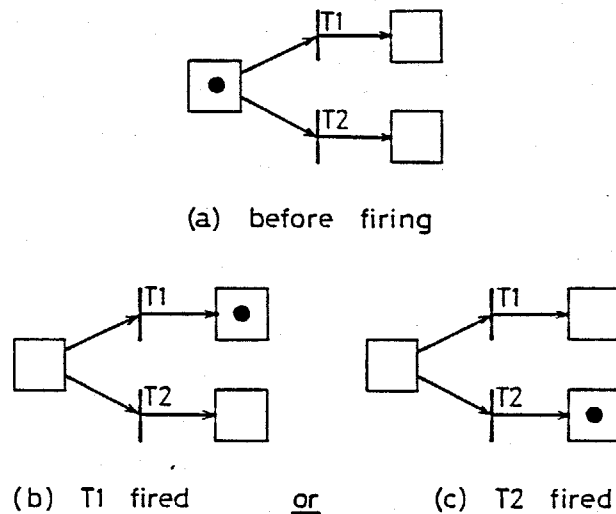


Fig.3-8 競合定理

3.5.3 パス・マーク流れ線図

Fig.3-9 のように、有向ループをもたないMFG をパス・マーク流れ線図 (Path-Mark Flow Graph 以下 P-MFG) という。

P-MFG 中にはソースからシンクまでの有向パスがあり、この途中のボックスあるいはトランジションからソースに近い側を上流、シンクに近い側を下流とよぶ。

《定理 3-2》 反転定理

あるボックスに接続されている方向枝の方向をすべて反転しても、そのボックス内のマーク有無およびそのボックスから出力されている信号枝（ゲート信号枝、出力信号枝）の許可・抑止の論理を反転すれば、このボックスが接続されているトランジションの可点弧条件は反転前と変わらない。

(証明)

反転するボックスのボックス変数を b_r とすると、このボックスと方向枝で接続されているトランジションの可点弧条件式中の b_r は、方向枝の反転により \bar{b}_r にかわるが、マークの有無を反転することにより $\overline{(\bar{b}_r)} = b_r$ となる。また、信号枝に関してはマークの有無を反転することにより \bar{b}_r となるが、信号枝の論理を反転することにより $\overline{(\bar{b}_r)} = b_r$ となる。よって、反転する前と可点弧条件は同じである。■

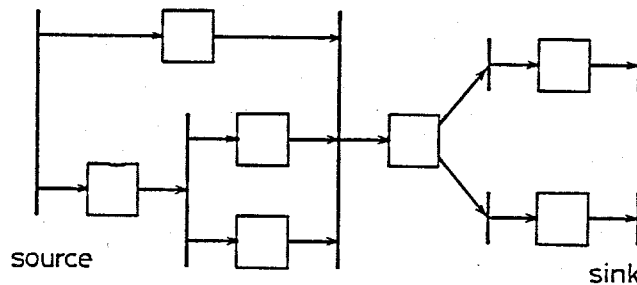


Fig.3-9 パス・マーク流れ線図の例

この定理を用いると、制御の情報の伝達方向を任意に設定でき、また本来初期マーキングでマークの必要なボックスがあっても、これを反転することにより0-初期マーキングとすることができる。Fig.3-10に例を示す。

《定理 3-3》

競合ボックスをもたないMFGは、P-MFGに必ず変換できる。

(証明)

競合ボックスをもたないMFGは、トランジションをノードとし、ボックスとそれにつく方向枝をまとめてアークとする有向グラフとみなせる。このグラフの任意の木を考え、その各ノードについて上流側より下流側のほうが大きくなるように番号を割り当てることができる。さらに、この木の補木の枝つまりリンクを、必要な場合には反転定理を用いてノード番号の小さいものから大きいものとなる方向にすべて接続することができる。このときの基本閉路はすべて有向閉路でないことは明らかである。これは、いいかえれば初めのMFGに等価な、すなわち対応する各トランジションの可点弧条件が等しく、したがって常にもとのMFGと等しい点弧系列を発生するP-MFGを得たことにほかならない。■

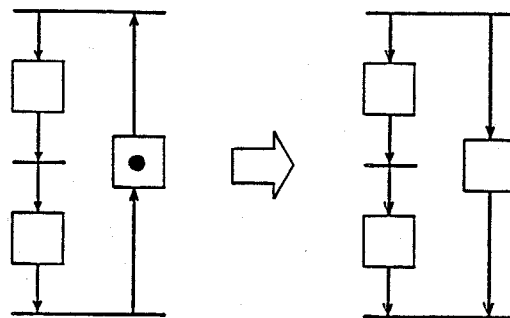


Fig.3-10 反転定理

3.6 マーク流れ線図におけるデッドロック

3.6.1 デッドロックと活性

MFGにおいて、あるマーキングから始めて順次トランジションの点弧を行なっていたとき、あるトランジションがそれ以後可点弧とならなくなることもある。この現象をデッドロック (deadlock) 現象とよび、そのMFGはデッドロックであるという。なお、デッドロック状態に陥らないことを活性状態である (live) という。

MFGでのデッドロックは、その原因から二つに分けられる。

【条件デッドロック】

不適切な初期マーキングが原因で生じたデッドロックで、ある適切な初期マーキングを与えれば活性状態となるものを条件デッドロック (conditional deadlock) という。

【構造デッドロック】

作られたグラフの構造自体に起因し、どのような初期マーキングを与えても活性状態にならないデッドロックを構造デッドロック (structural deadlock) という。

ペトリネットにおいて有界でない場合、これと同一構造 (ただしプレースをボックスとする) をもつMFGは構造デッドロックとなる。なぜならば、有界でないペトリネットではグラフ全体でのトークンの総数も有界ではないが、ボックスは容量が1に限定されたプレースに相当するので、同一構造のMFGでのマークの総数はボックス数以下でなくてはならず、したがってこれを越えるような点弧をおこすことができないからである。

3.6.2 デッドロックに関する条件

ここでは、競合ボックスと内部信号枝をふくまないMFGに限定してデッドロックについて考察する。

《定理 3-4》

競合ボックス、ボックス・ソース、ボックス・シンクおよび内部信号枝をもたない 0-初期マーキングの P-MFG は活性状態にある。(活性に関する十分条件)

(証明)

デッドロック・活性問題に関しては、外部ゲート信号枝および出力信号枝は関係がないのでこれらを取り去って考える。また、3.3.2で述べたようにボックスとトランジションの接続関係は、Fig.3-5に示すペトリネットで表現されるので、このMFG全体はマークトグラフ(MG)3-2)で表わされる。MGに関する定理、

「マーキングが活性状態にあるための必要十分条件は、MGのおののの有向閉路のトークンの数が少なくとも1であることである。」3-2)

を適用する。

いま、P-MFGであるからすべての方向枝は上流から下流に向いている。そして、0-初期マーキングであるから、このMFGに対応するMGにおいては上流から下流に向いたアークに接続されたすべてのプレースにトークンがなく、その逆向きのアークに接続されたすべてのプレースにトークンがある。ゆえに、このMGのおののの有向閉路には少なくとも1つのトークンが存在するので活性状態であり、もとのMFGも活性状態にある。■

《定理 3-5》

競合ボックス、ボックス・ソース、ボックス・シンクおよび内部ゲート信号枝をもたないMFGにおいて、このMFGがデッドロックであるための必要十分条件は、マークが入っているボックスすべてに反転定理を適用し、0-初期マーキングにしたときに有向閉路が存在することである。

(証明)

〈十分条件〉 前項の証明と同様に、このMFGに対応するMGで考える。いま、もと

のMFGで0-初期マーキングの有向閉路が存在したならば、Fig.3-11に示すように1つもトークンをもたない有向閉路が生じる。よって、このMGのマーキングは活性状態になく、もとのMFGはデッドロックである。

〈必要条件〉 定理3-4の対偶をとる。■

これより、上記の制約のMFGでは、反転定理によりP-MFGに変換したときに、0-初期マーキングとなるように初期マーキングを与えれば活性状態にすることができる。

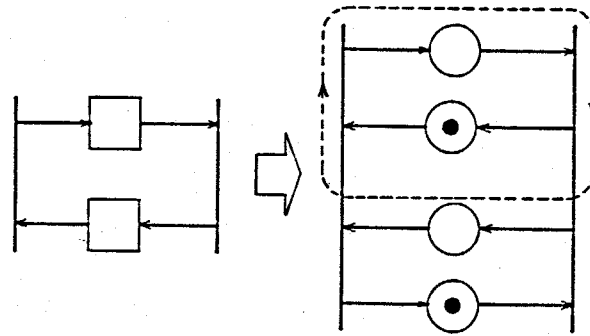


Fig.3-11 マークをもたない有向閉路

3.7 マーク流れ線図の特徴

MFG は安全なグラフであり、ゲート信号枝や出力信号枝をもたないMFG は容量を1に制限したペトリネット、あるいは Fig.3-5に示したボックスを1つのトークンと1対のブレースで置きかえたペトリネットに対応する。このことから、MFGはペトリネットがもつ解析能力やモデル化能力などを受け継いでいるといえる。さらに、MFGはつぎの特徴をもっている。

(1) 非連続生産システムの制御には安全（衝突がおこらないというグラフ理論的な意味での安全）が不可欠であるが、安全の条件を特に意識せずに設計・解析ができる。

(2) ボックス内にはたかだか1つのマークしか存在しない、すなわち0か1であるので2値論理により取り扱える。

(3) 非連続生産システムの制御は、制御部と対象機器との間で信号を授受しながら進行してゆく。つまり、制御部はある作業を実行させる信号を出力し、機器からの作業の完了確認信号を受け取りつぎの段階に進む。MFGでは、作業をボックスに割り当てここから状態信号を出力し、作業の切れ目をトランジションに割り当て、トランジションにゲート信号枝で機器からの信号を入力するという方法により、制御部と制御対象の接続も含めて非連続生産システムの制御系を適切に表現できる。

(4) 設計にあたってボックスに作業を割り当てる際、まずマクロな作業で表現しつぎにその作業を詳細に記述する、すなわちマクロな作業を表わすボックスの内部をミクロな作業で記述したMFGで表現するという階層的取り扱いが可能である。

第4章 M F G による

非連続生産システムの表現

前章で提案した M F G を用いて、具体的に対象とする非連続生産システムを表現することにより、そのシステムの挙動を解析し、また制御システムの設計が可能になる。そこで、M F G の代表的な構成要素であるトランジションとボックスを、どのようにシステム内の動作や機能に対応させるかについて考察し、例によってその表現手順を示す。

4.1 M F G と非連続生産システムとの対応

4.1.1 作業工程と M F G

M F G は非連続生産システムを、『事象と状況の接続』としてとらえその制御則も含めてモデル化するものである。非連続生産システムの基本構造は、工程や作業の順序関係としてとらえることができ、その制御は「機器の稼働」と「工程や作業」との関係づけ、および工程や作業の進行条件の規定としてとらえることができる。

工程や作業のつながりや進行を表わす場合に、工程や作業を事象として取り扱うか、状況として取り扱うかの問題がある。これは、事象のとらえかたとして primitive event と non-primitive event の2通りあることによる。

被加工物を中心に考えると、工程や作業は事象として扱う方がわかりやすい。なぜなら、

「ある被加工物は、『材料』という状態から加工されて『半製品』という状態になり、仕上げ作業を受けて『製品』という状態にいたる。」

というように、被加工物にとっては、作業はその状態に変化を与える事象にすぎない

からである。

一方、システムの制御や機械の動作を中心に考えると、

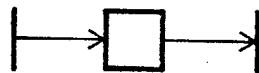
「ある作業は、『作業開始』という事象で始まり、『作業中』という状況となり、『作業終了』という事象で終る。」

というようにとらえなければ、機械の起動や停止などの制御に関わる事項が表現されない。

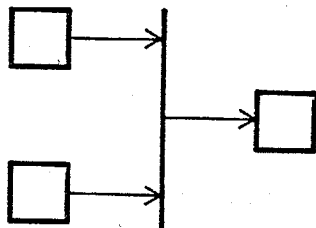
そこで、MFGでは非連続生産システムをその制御も含めて表現するために、工程や作業などを「その作業を実行中である」という状況として表わし、その前後に「その作業の開始」や「終了」を示す事象が接続されるものとする。

4.1.2 システム内の動作・機能とMFG

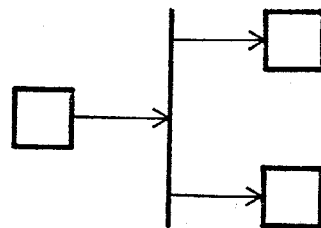
2.3 で分類したシステム内での動作や機能のMFGによる表現を、以下に示す。



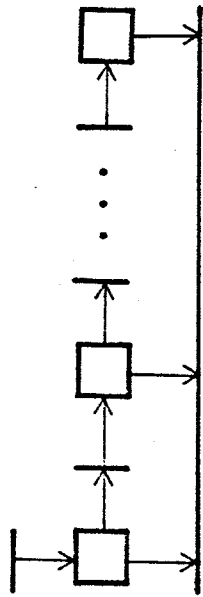
(1) 操作



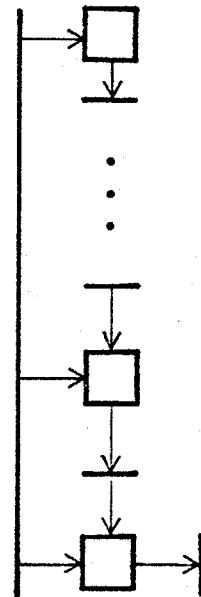
(2) 結合



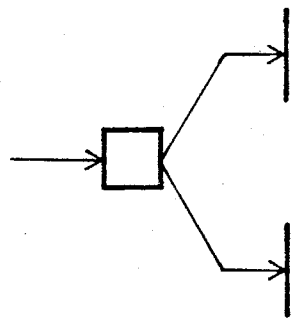
(3) 分割



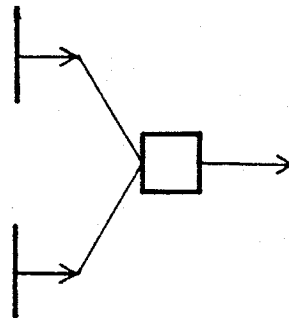
(4) 集合



(5) 分離



(6) 分岐



(7) 合流



(8) 容量 (FIFOの場合)

4.2 MFGによるシステム表現の手順

MFGによりシステムの表現を行なう手順を、簡単な例題を用いて説明する。

4.2.1 例とするシステム

対象とするシステムは、Fig.4-1 に示す加工機、ロボット、搬入／搬出両用コンベアにより構成されている。また、その動作仕様はつぎのとおりとする。

- ① コンベア CV により、ワークピースが搬入される (CI)
- ② ロボット RB が、ワークピースを加工機にロードする (LD)
- ③ 加工機 M が、ワークピースを加工する (PR)
- ④ ロボット RB が、加工機 M からワークピースをアンロードし、コンベア CV に載せる (UL)
- ⑤ コンベア CV がワークピースを搬出する (CO)

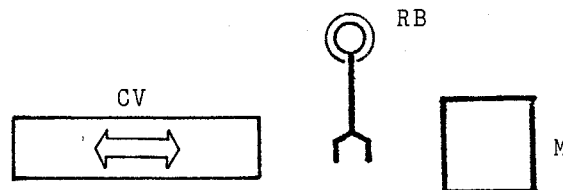


Fig.4-1 非連続生産システムの例

4.2.2 工程や作業の接続

まず、必要なレベルに応じて工程や作業をボックスに割り当てる。このボックスを工程や作業の順序関係に従って、方向枝によりその開始・終了を表わすトランジションを介して接続する。 [Fig.4-2]

4.2.3 機器の接続

つぎに、システム構成要素の各機器をボックスに割り当てる。このボックスは、その機器の使用状況に対応しており、『使用中』あるいは『準備完了』を表わすものである。このボックスを、その機器が使用される範囲、すなわち「その機器を使い始める工程や作業」の開始を表わすトランジションと、「その機器を使い終る工程や作業」の終了を表わすトランジションとの間に接続する。 [Fig.4-3]



Fig.4-2 工程や作業の接続

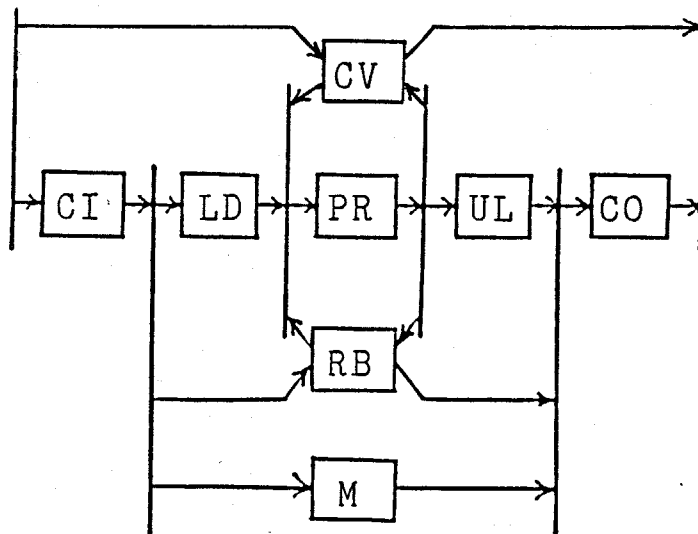


Fig.4-3 機器の接続

4.2.4 受け入れ容量や区間などの制御条件の規定

システムの構造や動作の仕様によっては、受け入れられるワークピースの数や同時に行なえる作業の数が制限される区間をもつ場合がある。この例の場合コンベアが搬入・搬出を兼ねているため、加工機で1つめのワークピースを加工している間に次のワークピースの搬入を開始すると、初めのワークピースの搬出ができなくなり続行不能状態になる。よってこの例では、搬入から搬出までが受け入れ容量が1の区間であり、これを規定するボックスをその区間に接続する。 [Fig.4-4]

また、そのほか必要に応じて制御条件の規定を行なう。

この段階のMFGは、システムにおける「工程や作業の進行条件」、およびこれと「機器の使用」との関係を規定するものであり、システムのコーディネートに用いることができる。

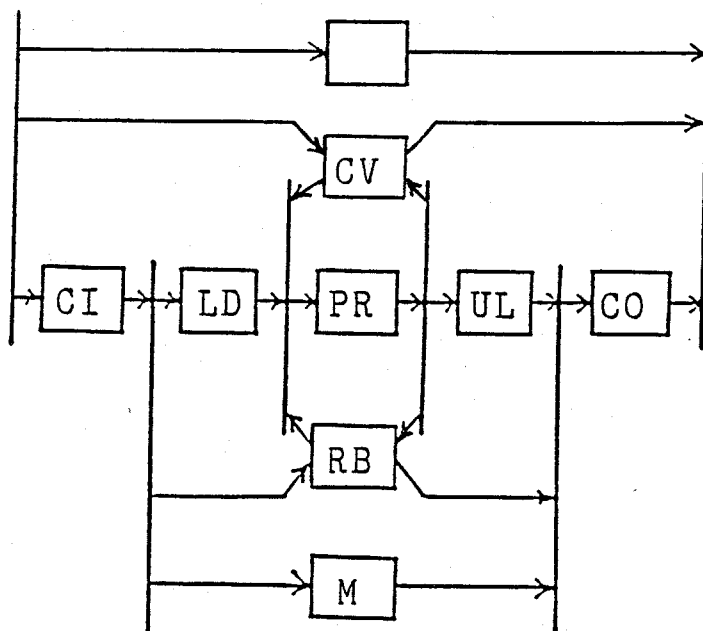


Fig.4-4 制御条件の規定

4.2.5 機器制御の記述

実際の詳細な機器制御まで表現する場合、工程や作業を各機器に応じてそれがもつ単位動作に分解し、さらに機器との信号の授受を記述することが必要となる。ある1つの工程や作業は、単位動作を表わすボックスの結合で表わされる。

そのボックスから出力信号枝を対象機器に接続し、動作命令の送り出しを記述する。また、各単位動作の開始・終了を表わすトランジションへ機器からゲート信号枝を接続し、動作完了信号の受け取りを記述する。

このほか、必要に応じて出力信号枝やゲート信号枝を加える。

上記の例でLDという工程を、単位動作に分解したものを Fig.4-5に示す。

この段階のMFGは、機器制御の詳細についてまで規定しており、システム制御用のプログラムとして用いることができる。

以上のようにして、非連続生産システムはMFGにより表現される。

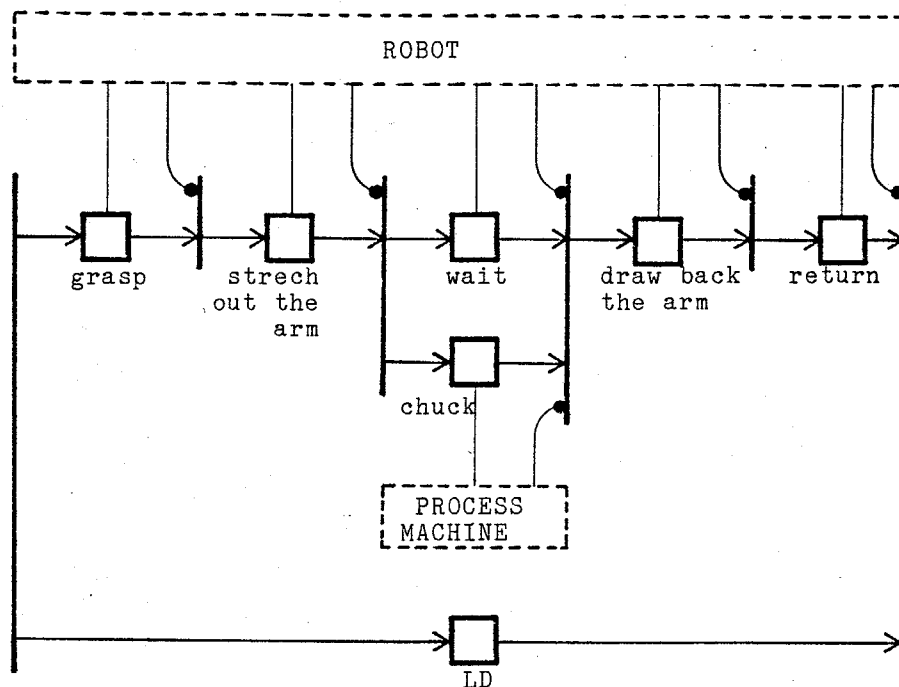


Fig.4-5 機器制御の記述

第5章 MFGによる

非連続生産システムの制御

5.1 MFGによる制御の実現

前章までの考察より、非連続生産システムはその制御則も含めて、MFGにより適切に表現されることが明らかになった。このように、MFGはコントローラも含めたシステムのモデルであり、そのモデルと実際のシステム構成要素の動作を一致させることが制御につながる。具体的にいうと、システム構成要素の機器からのゲート信号により、MFGにおけるトランジションの点弧やマーキングの遷移を実際のシステムの進行に同期させる。そして、MFGのマーキングに従って動作命令に相当する信号を対象の機器に出力し、MFGの状態に実際のシステムの動作状況を一致させる。

これを実現するためには、MFGを実行する制御装置が必要となる。従来のプログラマブルコントローラを利用する場合には、システムを表現するMFGを論理式やラダー図等に変換してプログラムする必要がある。この方法によると、実際の動作を監視する場合にMFGとの対応がつけにくい、仕様変更を行なう場合は変更されたMFGを再び変換しプログラムしなおさなければならない、などの問題もある。そこで、MFGによりプログラムされ、直接にMFGを実行し制御を行なう『MFGにもとづいたプログラマブルコントローラ』を試作し、またそのプログラム支援装置を作成する。

5.2 MFGによる制御のための準備

5.2.1 アービタ（調停器）

MFGの競合ボックスではマークの流れの分岐あるいは合流が行なわれるが、3.5で述べたように競合関係にあるトランジションのうち複数個が点弧可能な場合、それらのうち任意の1つのみが点弧し他は非可点弧となる。グラフ上の約束としてはこれでよいが、MFGをプログラマブルコントローラやコンピュータにより実行する場合には、点弧させるトランジションの選択が必要となる。もしこれを怠ると、1つずつトランジションの点弧条件を調べては点弧するという方式(逐次点弧方式)ではテスト順序に依存し、すべてのトランジションの点弧条件を調べてから点弧するという方式(同時点弧方式)では競合していてもそれらが全部点弧するという誤動作につながる。そこで、競合ボックスにおけるマークの流れを制御する、すなわちトランジションの競合を調停するために、つぎのような基本アービタ(調停器)を導入する。

基本アービタは、競合ボックスに接続された競合関係にあるトランジションそれぞれに、一定の順序にしたがって平等な割合で点弧の『権利』を割り当てる。ただし、点弧の権利が割り当てられたトランジションが点弧可能でない場合は、その権利は消滅するものとする。

制御プログラムとしてのMFGの競合ボックスには、この基本アービタが内在されているものと考え、グラフ上には特にこれを記述しないことにする。なお、任意の仕様で競合ボックスにおけるマークの流れを制御する場合には、グラフ上に仕様に応じたアービタをグラフとして付加する。この場合、基本アービタも同時に機能するが、これは最低限の条件であるのでなんら問題はない。

5.2.2 MFGの表形式の記述

MFGを制御プログラムとするとき、図的表現をどのように取り扱うかが問題となる。その表現方法の1つに3.4で示した行列表現がある。ここでは、MFGの進行はトランジションの点弧に依存することに着目し、この行列の概念をもとにトランジシ

ジョンを中心とした表による記述を導入する。

【接続関係表】

これは、そのトランジションに方向枝により接続されているボックスの番号を、入力側・出力側の区別をつけて表にしたものである。番号が正の場合はそのボックスがこのトランジションの出力側に接続されていることを表わし、負の場合には入力側に接続されていることを表わす。いいかえれば、このトランジションが点弧するための前提条件と、点弧による後堤条件を表にしたものと考えられる。

【ゲート信号枝表】

これは、そのトランジションに接続されているゲート信号枝の信号源の番号を、内部・外部の区別および許可・抑止の区別をつけて表にしたものである。内部信号枝の場合は" B "、外部信号枝の場合は" E "が番号の前に付く。また、許可枝と抑止枝の区別は、抑止枝には" - "を付けることにより行なう。

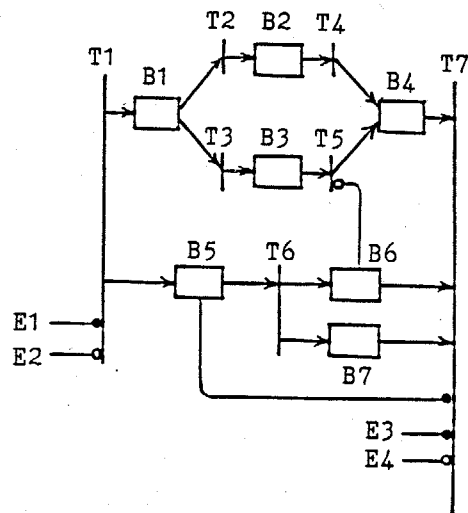
制御プログラムとしての M F G では、競合関係にあるトランジションに対して基本アービタが内部的に付加されている。そこで、これを記述するためにアービタ表を導入する。

【アービタ表】

これは、互いに競合関係にあるトランジションの番号を表にしたもので、点弧の権利の割り当てはこの表の順番で行なわれる。

Fig.5-1 に例を示す。このような形式をとるとグラフとの対応もよく、取り扱いも

容易である。



(1) Connection table

T NO.	Box NO.		
T 1	1	5	
T 2	-1	2	
T 3	-1	3	
T 4	-2	4	
T 5	-3	4	
T 6	-5	6	7
T 7	-4	-6	-7

(2) Gating arc table

T NO.	B NO. or E NO.			
T 1	E 1	-E 2		
T 5	-B 6			
T 7	B 5	E 3	-E 4	

(3) Arbiter table

NO.	Transition NO.	
1	2	3
2	4	5

Fig.5-1 M F G の表形式の記述例

5.3 MFGにもとづいたプログラマブルコントローラ

MFGの表記述によりプログラムされ、このMFGにより直接に制御を実現する『MFGにもとづいたプログラマブルコントローラ』をマイクロプロセッサを用いて試作した。これを、MFGコントローラと呼ぶことにする。 [Fig.5-2]

5.3.1 動作原理

MFGコントローラは、プログラムされたMFGの表記述をもとに、同時点弧方式を用いてMFGを実行する。その手順の概略は、

- (1) 外部ゲート信号をラッチする
- (2) 各トランジションの可点弧条件を調べる
- (3) 競合が生じていればそれを調停する
- (4) トランジションの点弧とマーキングの遷移を行なう
- (5) マーキングにしたがった信号出力状態とする

を繰り返すものである。以下、その詳細について説明する。

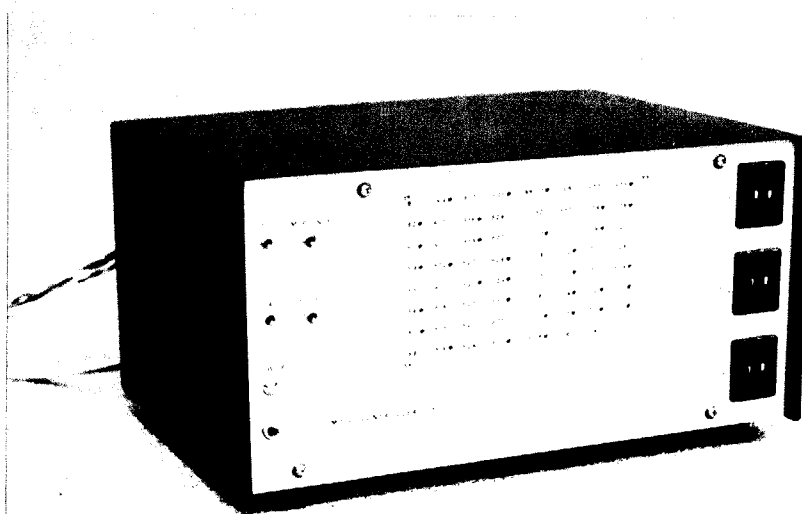


Fig.5-2 MFGコントローラ

【トランジションの点弧条件のテスト】

MFGコントローラは、接続関係表およびゲート信号枝表を用いて、各トランジションの点弧可能性をつぎのように調べる。

- ① そのトランジションの入力側に、マークの入っていないボックスがあるか？
- ② そのトランジションの出力側に、マークの入っているボックスがあるか？
- ③ そのトランジションは、信号が"0"である許可枝をもっているか？
- ④ そのトランジションは、信号が"1"である抑止枝をもっているか？

すべてのテスト結果が"NO"であれば、そのトランジションは点弧可能である。また、どれか1つでも"YES"であれば直ちに点弧不能と判定する。このフローチャートをFig.5-3に示す。なお①～④の各テスト内においても、ボックスや信号でどれか1つでも"YES"であれば、残りのものに対するテストは行わずに直ちに抜け出す。

これをすべてのトランジションに対して行ない、点弧可能なトランジションを探し出す。

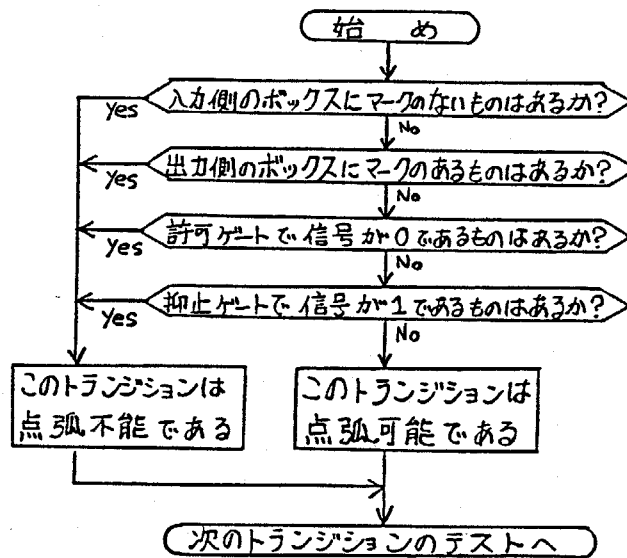


Fig.5-3 点弧条件のテスト

【競合の調停】

プログラムされたMFGが競合ボックスをもっているとき、競合が生じる場合があるので、各競合をアービタ表にもとづいてつぎの手順により調停する。

① 競合関係にあるトランジションのうち、点弧可能なものはあるか？

* もしなければ、調停の必要はない。

② 点弧の権利の割り当て先を指すポインタで指示されたトランジションは、点弧可能か？

* もしそうでなければ、ポインタを点弧可能なトランジションが見つかるまで進める。

③ 競合関係にあるトランジションのうち、ポインタで指されたもの以外を点弧不能とする。

④ 次回の調停のためにポインタを1つ進めておく。

このフローチャートをFig.5-4に示す。

これを、すべての競合関係に対し行なうことにより、競合が回避される。

【点弧とマーキングの遷移】

点弧条件のテストおよび競合の調停の結果にもとづき、接続関係表を利用してつぎの手順により、各トランジションを点弧しマーキングを遷移させる。

① このトランジションは点弧可能か？

* もしそうでなければ、点弧は行なわない。

② このトランジションの入力側のすべてのボックスから、マークを消去する。

③ このトランジションの出力側のすべてのボックスに、マークを生起する。

このフローチャートをFig.5-5に示す。

これをすべてのトランジションに対し行ない、1回の走査が完了する。

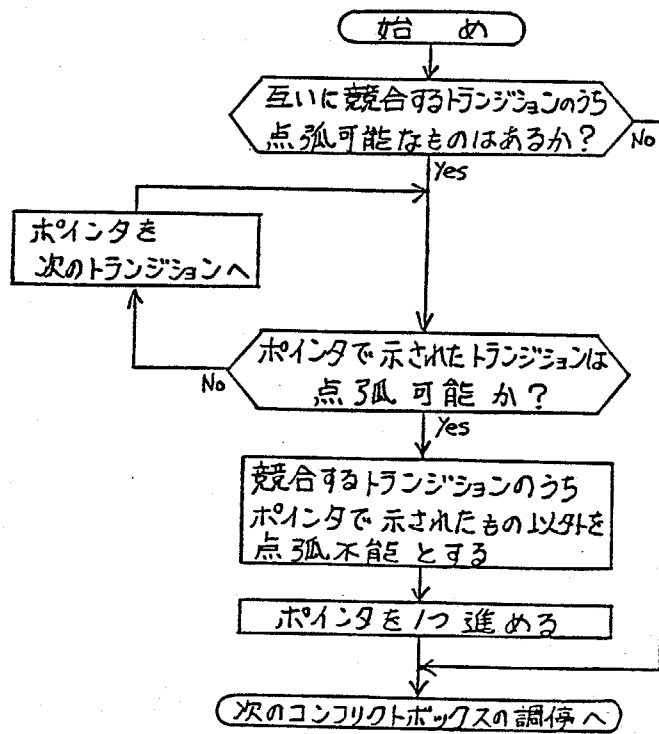


Fig.5-4 競合の調停

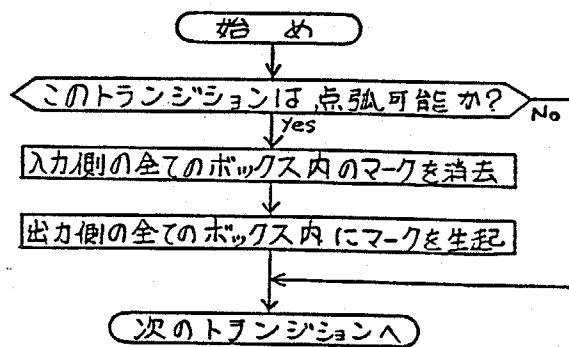


Fig.5-5 点弧

5.3.2 ハードウェア

今回試作したMFGコントローラは、

- * マイクロプロセッサMC68B09 (2 MHz, 最小命令実行時間1 μ sec)
- * 読み出し/書き込みメモリ 4Kバイト
- * 読み出し専用メモリ 2Kバイト
- * ラッチ 入力用 32ビット/出力用 32ビット
- * 絶縁型入出力インタフェース 32入力/32出力
- * LEDインディケータ
- * 通信回線インタフェース (RS-232c)

より構成されている。構成図を、Fig.5-6に示す。

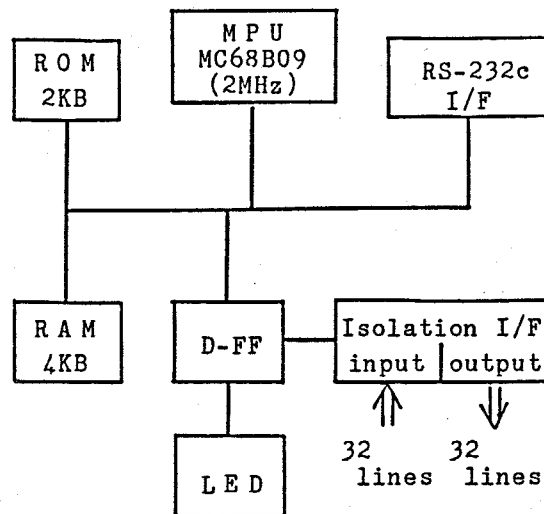


Fig.5-6 ハードウェア構成

システム制御のプログラムである表形式で記述されたMFGは、その初期マーキングとともに通信回線により入力され、読み出し／書き込みメモリに記憶される。また、マイクロプロセッサの処理プログラム、すなわちMFGの実行規則は読み出し専用メモリに記憶されている。

外部ゲート信号の入力あるいは出力信号枝による出力は、ラッチおよび絶縁型インタフェースを介して対象機器と接続される。この入出力信号の状態およびマーキングは、LEDインディケータに表示される。なお、マーキングは必要に応じて通信回線インタフェースを介して読み出すことができる。

このコントローラが取り扱えるMFGは、600の内部接続、32入力（外部ゲート信号源）、および32出力（出力信号送出先）以内のものである。

処理時間は、点弧条件の1つのテストに8 μ sec、マーク1つの生起・消滅に9 μ secである。1回の走査時間は、グラフの大きさや構造およびその時点でのマーキングに依存するが、最大でも6msecであり通常は1msec以内である。

5.4 プログラム支援装置

5.4.1 装置の概要

MFGコントローラへのプログラミングを容易に行なうためには、プログラムの開発やそのデバッグを支援する装置が重要である。特にMFGの場合、グラフ手法であるので図的に扱えることが望ましい。このために、グラフィックCRT、ライトペン、フロッピディスクドライバ、プリンタ、通信回線インタフェースをもつパーソナルコンピュータを用いて、MFGコントローラのためのプログラム支援装置を作成した。

[Fig.5-7]

この装置は、

- (1) MFGコントローラのプログラムとなるMFGの開発およびそのシミュレーション
- (2) MFGコントローラへのプログラムのローディング
- (3) MFGコントローラのモニタ

の機能をもつ。

これを、MFGプログラマ/モニタとよぶことにする。

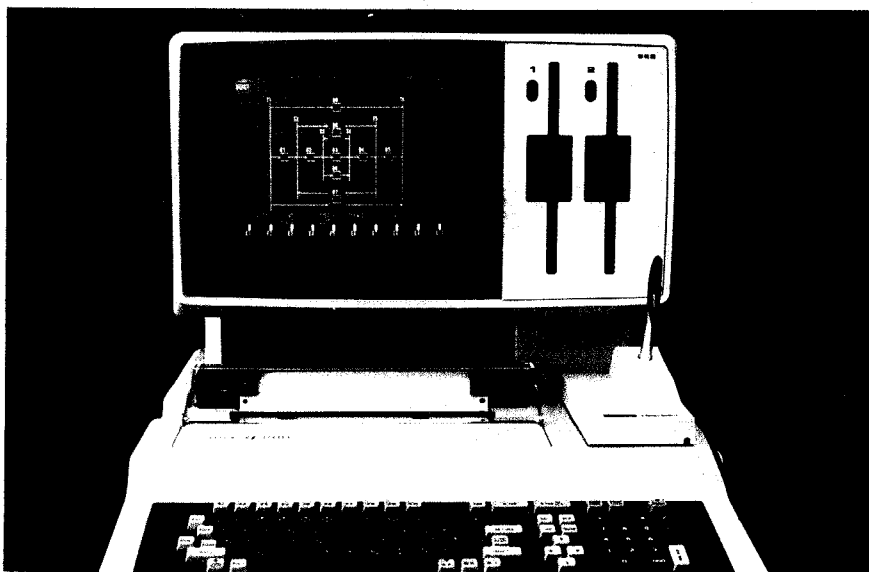


Fig.5-7 MFGプログラマ/モニタ

5.4.2 MFGの入力

MFGはグラフ手法であるので、その入力はグラフを描き込むという方法がプログラマにとっては分りやすい。そこで本装置では、ライトペンを用いてグラフィックCRT上にMFGを描き込んで入力する。これは、Fig.5-8のようにCRT上の指示点を利用してMFGの各要素を記入してゆくもので、消去や修正も同様にして容易に行なえる。その際、グラフの構造に誤りがあれば警告を発するため、誤りが防止される。また、競合ボックスがある場合にはアービタが自動的に内部に付加される。なおゲート信号枝の接続については、キーボードよりトランジション番号とその信号源の番号を入力することにより行なう。

これらのMFGは、フロッピディスクに保存することができ、必要なときすぐに引き出すことができる。

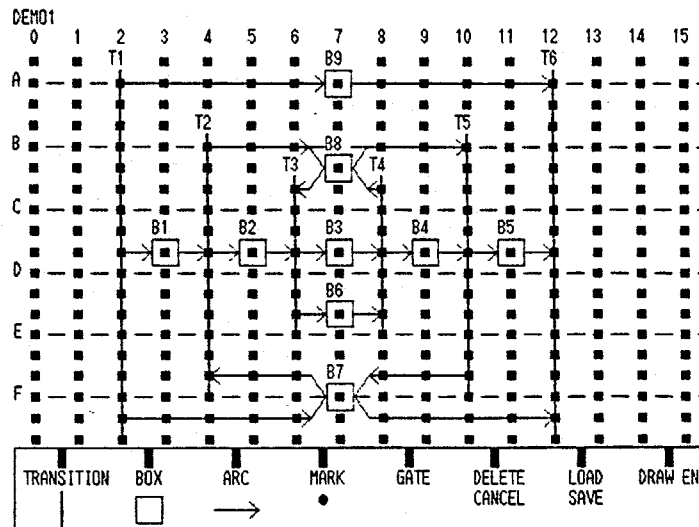


Fig.5-8 MFGの入力

5.4.3 MFGのシミュレーション

入力したMFGが正しく動作するか否か、つまりプログラムに誤りがないかどうかを、実際の制御を行なう前に確認することは重要なことである。MFGでは並列性・同時性・競合などがあるため、人手によりその検証を行なうことは手間のかかるものであり、また複雑なグラフの場合には間違える可能性もある。そこで、入力されたMFGに対して、計算機シミュレーションを行なうことにする。

このシミュレーションは、MFGコントローラの実行アルゴリズムとほぼ同様な手順でMFGを実行するもので、外部ゲート信号は機器の代わりにファンクションキーにより与える。実行時には、トランジションの点弧やマークの動きをCRT上のMFGグラフに表示し、またデッドロックやゲート信号待ちの判定も行なう。このため、動作の確認が直視的に行なえ、変更すべき箇所の発見が容易である。シミュレーション手順を以下に示す。

- (1) ボックスの入出力関係による各トランジションの可点弧条件を調べる。
- (2) 内部ゲート信号による各トランジションの可点弧条件を調べる。
- (3) この時点で点弧可能なトランジションが1つもなければ、マーキングによるシステムの停止でありデッドロックと判定する。そして、グラフの構造あるいは初期マーキングを変更するよう警告を発し修正を促す。
- (4) デッドロックでなければ、外部ゲート信号による各トランジションの可点弧条件を調べる。
- (5) この時点で点弧可能なトランジションが1つもなければやはりシステムは停止するが、外部ゲート信号の条件が成立すれば再び動きだせるので、デッドロックとは区別し外部ゲート待ちと判定する。そして、その表示を行ない信号の入力を促し、条件の成立を待つ。
- (6) その後、競合ボックスに接続されるトランジションにおいて競合が生じてい

るか否かを調べ、もし生じているならばアービタによる調停を行なう。

(7) 点弧可能なトランジションをCRT上のグラフに表示する。

(8) 点弧可能なトランジションを点弧し、マーキングを遷移させる。それとともに、CRT上のグラフのマークを移動させる。

(9) 以上を繰り返す。

フローチャートをFig.5-9に示す。

このシミュレーションにより、MFGの誤りや仕様を満たしていない箇所を発見し、ライトペンによりMFGを修正し再びシミュレーションを行なうことにより、デバッグが行なえるものである。

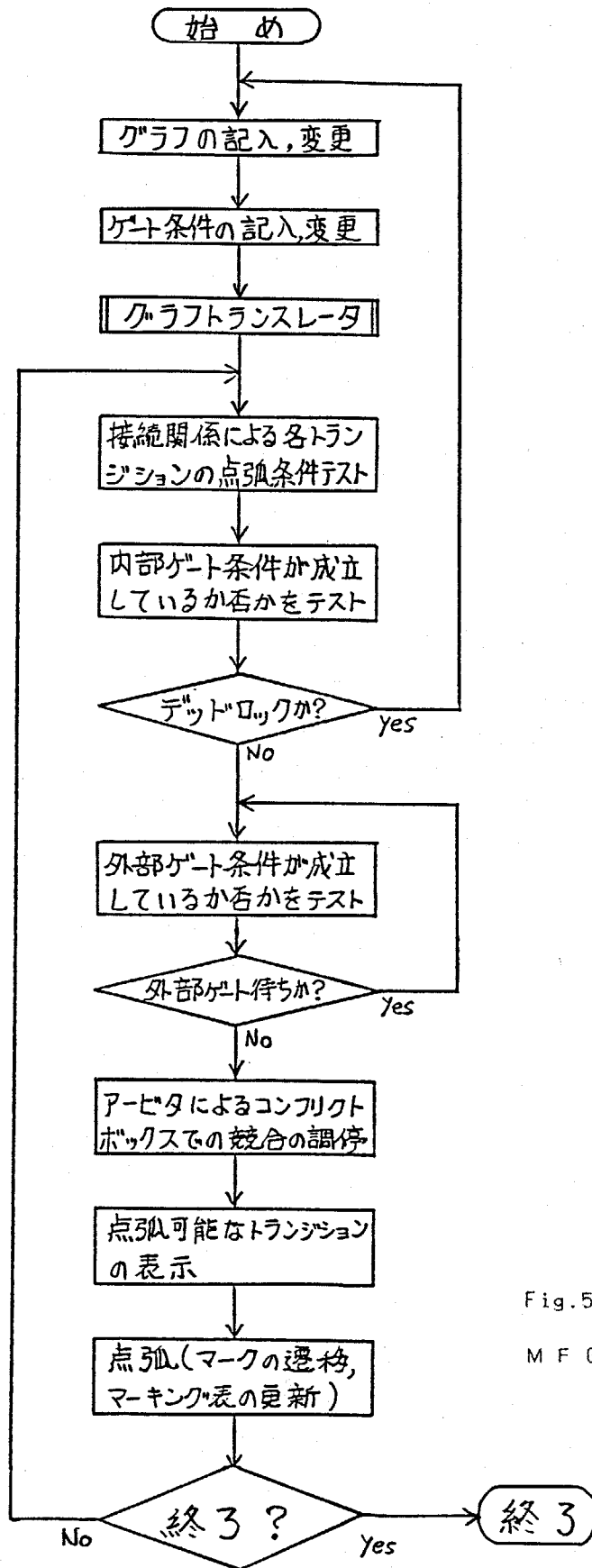


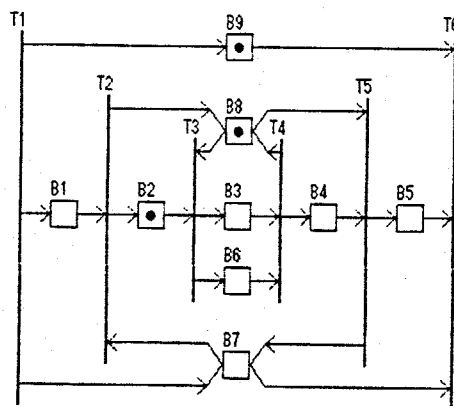
Fig.5-9

MFGのシミュレーション

5.4.4 MFGコントローラへのプログラムのローディングとモニタ

MFGコントローラは、MFGの表形式によりプログラムされる。そのため、本装置は完成したMFGを表形式に変換して、通信回線によりMFGコントローラに転送し、プログラムのローディングを行なう。

また、MFGコントローラが制御を実行している間、その動作状況はマーキングとして知ることが出来るが、このままではシステムとの対応がつきにくい。そこで、本装置がMFGコントローラのマーキングをサンプルし、それをCRT上のMFGグラフに表示する。このようにして、動作状況のモニタが行なえる。



動作状況のモニタ

5.5 実験

以上のコントローラとプログラミング装置を用いて、生産ラインの模型の制御実験を行なった。

5.5.1 実験システム

この実験システムは、

- (1) MFGコントローラ
- (2) MFGプログラマ/モニタ
- (3) 生産ラインの模型 [Fig.5-10]

により構成されている。MFGコントローラとMFGプログラマ/モニタは、通信回線 (RS-232c) で接続されている。生産ラインの模型は、2台のロボット、3台のコンベア、2台のワークステーション、および8個のフォトスイッチによる検出器からなる。ロボットはそれぞれ制御用プログラマブルコントローラをもち、単位動作があらかじめプログラムされている。これらの生産ラインの構成要素それぞれとMFGコントローラは、絶縁型インタフェースを介して入出力線により接続されている。システムの構成図をFig.5-11に示す。

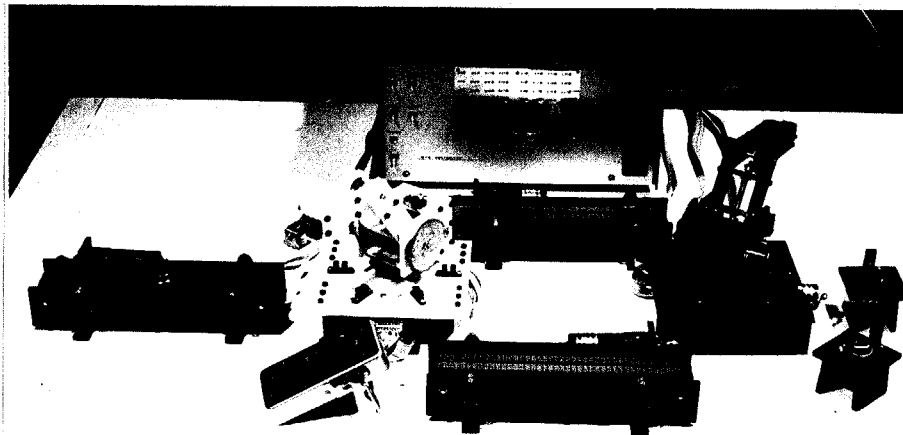


Fig.5-10 生産ラインの模型

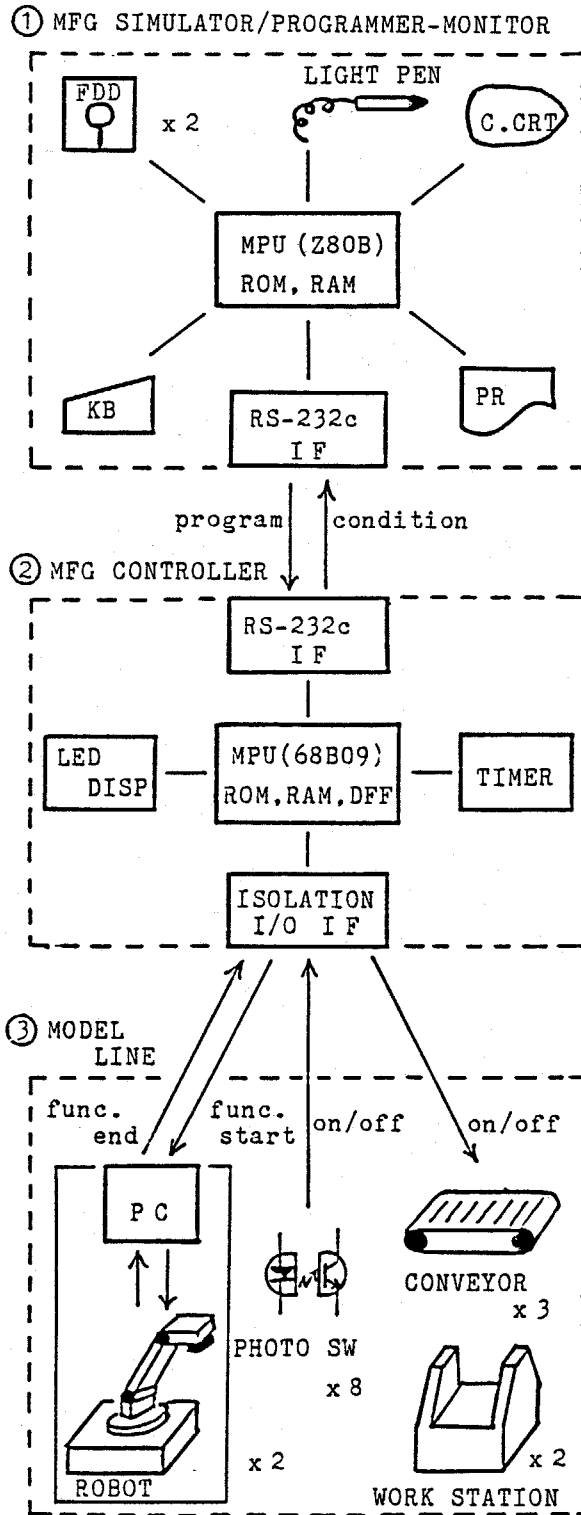


Fig.5-11 システム構成

制御対象の生産ラインはFig.5-12に示す配置をもち、その動作仕様はつぎのとおりである。

- ① ワークピースが E 12 に到着する
 - ② コンベア CV1 が E 11 まで搬送する
 - ③ ロボット RB1 が E 14 に移し換える
 - ④ コンベア CV2 が E 13 まで搬送する
 - ⑤ ロボット RB2 がワークステーション PR1 または PR2 のどちらか空いている方にロードする
 - ⑥ ワークステーションが加工する
 - ⑦ ロボット RB2 がアンロードし E 17 に置く
 - ⑧ コンベア CV3 が E 16 まで搬送する
 - ⑨ ロボット RB1 が搬出する
- * なお、効率を上げるために、デッドロックにならない範囲で可能なかぎり多くのワークピースを受け入れるものとする。

この条件のもとでの制御方法を M F G で表わしたものが Fig.5-13 である。

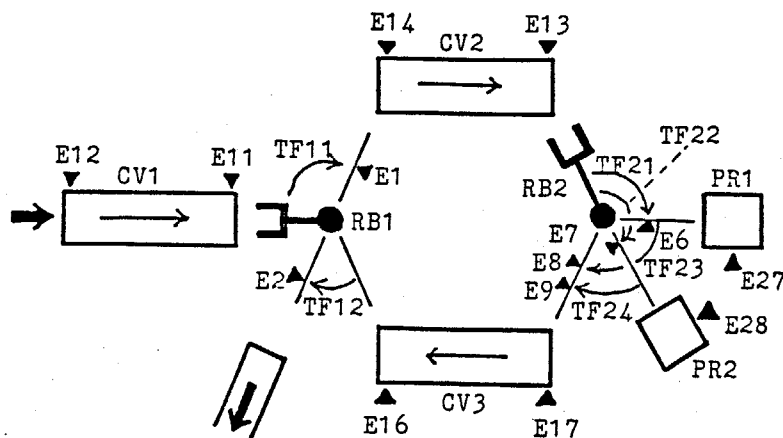


Fig.5-12 生産ラインの配置

5.5.2 実験の手順

実験は以下の手順にしたがって行なった。

- (1) MFGをライトペンによりMFGプログラマ/モニタのCRT上に描き、またゲート信号枝の接続関係も入力する。
- (2) このMFGのシミュレーションをプログラマ/モニタにより行ない、動作仕様を満たしているかどうかを確認する。もし誤りがあれば、それを修正し再びシミュレーションを行なう。これを、誤りがなくなるまで繰り返す。
- (3) 完成したMFGをプログラマ/モニタによりFig.5-14に示す表形式に変換し、通信回線を通してMFGコントローラにロードする。これにより、MFGコントローラのプログラミングが完了する。

CONNECTION				GATE INPUT			
T	BOX NO.			T NO.:	B NO.	OR	E NO.
1	12			T 1 :	-B20	-E11	E12
2	-12	20		T 2 :	E11		
3	1	-20	25	T 3 :	-B21	-E14	-E 1
4	-1	21	-25	T 4 :	E 1		
5	14	-21		T 5 :	-B26	-E13	
6	-14	26		T 6 :	E13		
7	6	24	-26	T 7 :	-B17	-B27	-E 6
8	-6	17	-24	T 8 :	E 6		
9	8	24	-27	T 9 :	-B22	-E15	-E17 -E 8
10	-8	22	-24	T10 :	E 8		
11	7	24	-26	T11 :	-B18	-B28	-E 7
12	-7	18	-24	T12 :	E 7		
13	9	24	-28	T13 :	-B22	-E15	-E18 -E 9
14	-9	22	-24	T14 :	E 9		
15	15	-22		T15 :	-B23	-E16	
16	-15	23		T16 :	E16		
17	2	-23	25	T17 :	-E 2		
18	-2	-25		T18 :	E 2		
19	-17	27		T19 :	E17		
20	-18	28		T20 :	E18		

ARBITER			
NO.	TRANSITION		
1	10	14	
2	7	9	11 13
3	8	10	12 14
4	3	17	
5	4	18	
6	7	11	

Fig.5-14 プログラムされた内容

(4) MFGコントローラは入力されたMFGにしたがって、生産ラインの機器からの信号の入力を行ない、トランジションを点弧しマーキングを遷移する。その結果のマーキングにもとづいて、機器に信号を出力する。これを可能なかぎり速く繰り返すことにより、制御を実現する

(5) この間、プログラマ/モニタはコントローラのマーキングをサンプルし、そのCRT上のMFGグラフにマークを表示する。これにより制御対象のラインの動作状況をモニタし、また万一システムに異常が生じたときにその発生場所の発見を容易に行なう。

(6) システムの動作仕様の変更が必要となった場合は、MFGを修正しコントローラに再ロードする。なお、システムの動作プログラムであるMFGは、フロッピディスクに保存され必要に応じて取り出す。

5.5.3 実験結果

以上の手順により実験を行なった結果、仕様どおりの制御が実現できた。またこのようにラインの模型を実際に動かすことにより、グラフ上のシミュレーションでは見落しがちな「動作中」という状況と「待ち」という状況の区別を、制御時にはその対象となる機器に応じてはっきりさせなければならないことが明確になった。

また、模型を用いた実験はシステムの実動シミュレーションに相当し、これはプログラムの開発や教育に有効なものであることもわかった。

5.6 今後の課題

MFGによりプログラムされ、このMFGを直接に実行するプログラマブルコントローラを試作し、またそのプログラム支援装置も作成した。これらを利用して模型の

生産ラインの制御実験を行なった。この実験は小規模な模型のラインの制御であるが、原理的には大規模な実際のシステムと同じものであり、MFGにより非連続生産システムの制御が適切に行なえることを確認したといえる。

今後の課題としては、非連続生産システムのもつ階層性に着目した階層制御に、これを発展させることがあげられる。

第 6 章 M F G による

非連続生産システムのシミュレーション

6.1 非連続生産システムのシミュレーションと M F G

非連続生産システムでは、システム内を多数のワークピースが次々と個別に流れていくという並列性や非同期性があり、またその流れ方が複雑な場合も多い。そのため、システム構成要素である各機械の動作条件の規定、すなわち工程の制御や管理をどのようにするかが、システム設計の際の重要な問題である。その設計が不適切であると、正常に動いても機械の稼働率が低くシステム全体として効率の悪い場合があり、また仕様を満たさないことや誤動作をおこす場合もある。最悪の場合は、システムの動作の続行が不能というデッドロック状態に陥る可能性もある。そのため、非連続生産システムの設計あるいは改善を行なう場合、これらの点に注目してシステムの実態に即した定量的な挙動の確認を行ない、さらにシステム全体としての生産能力や各機械の稼働率などを知ることが重要である。

ところが、2.2 で述べた非連続生産システムの諸性質のため、その挙動において過渡現象や脈動を不規則に生じる場合が多くみられ、解析的にこれらを求めることは困難である。また、異常時におけるシステムのふるまいを知りたいという要求もあり、非連続生産システムのシミュレーションを定量的に行なうことが必要とされる。

前章までの考察により、M F G は非連続生産システムを適切に表現でき、さらにそれを用いて制御が行なえることが確認されている。このことより、システムの設計時から M F G を用いれば、その結果を制御にまで一貫して利用することが可能であるといえる。そのために、システムの設計あるいは改善を行なう際に必要となる定量的なシミュレーションを、M F G により行なう手法について提案する。

6.2 MFGへの時間の導入

6.2.1 時間の導入

非連続生産システムをMFGにより表現しマーキングの遷移をみることにより、そのシステムの定性的な挙動、すなわち作業やワークピースの流れの順序やその進行条件などを確認することができる。しかし実際のシステムでは、作業の進行の制御にタイマが用いられる場合がしばしばあり、また加工や搬送などの作業にそれぞれ所要の時間がかかるため、その挙動は時間に依存したものになる。システムの実際の挙動を把握し定量的な扱いをする場合には、時間も考慮してシステムをシミュレートする必要があり、MFGに時間の概念を導入することが不可欠である。

6.2.2 時間ボックス

これは、状況の時間経過 (time lapse) を表わすもので、経過すべき時間をボックスに割り当てる。[Fig.6-1]

ボックス内にマークが入ると、割り当てられた時間が経過するまで拘束されて移動できないものとし、拘束時間がすぎた後は普通のボックス内のマークと同じように、接続されているトランジションの可点弧条件に寄与する。すなわち、この場合の可点弧条件は、

『そのトランジションの入力側に時間的な拘束が解除されたマークの入っていないボックスがなく、かつ出力側にマークの入ったボックスがないとき』

となる。時間ボックスによる点弧のようすを Fig.6-2に示す。

これを用いて、非連続生産システムにおける時間経過をシミュレートすることができる。

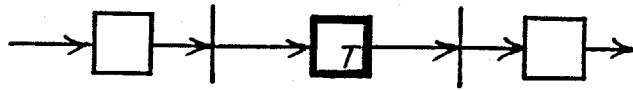


Fig.6-1 時間ボックス

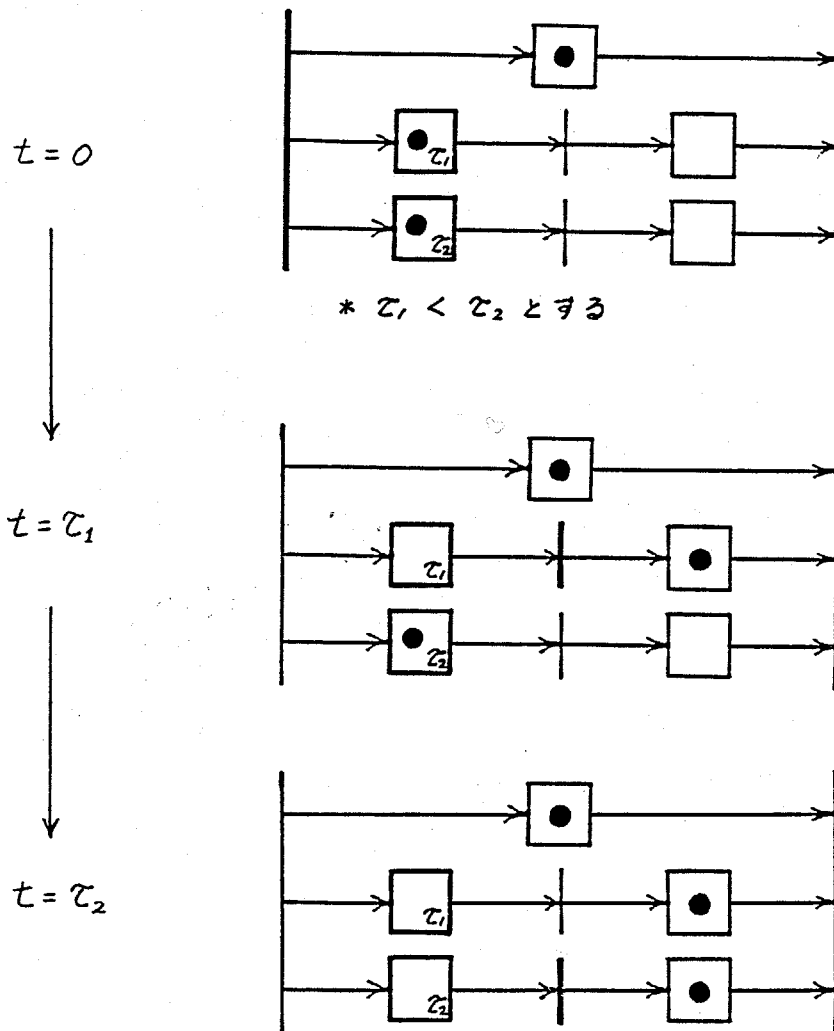


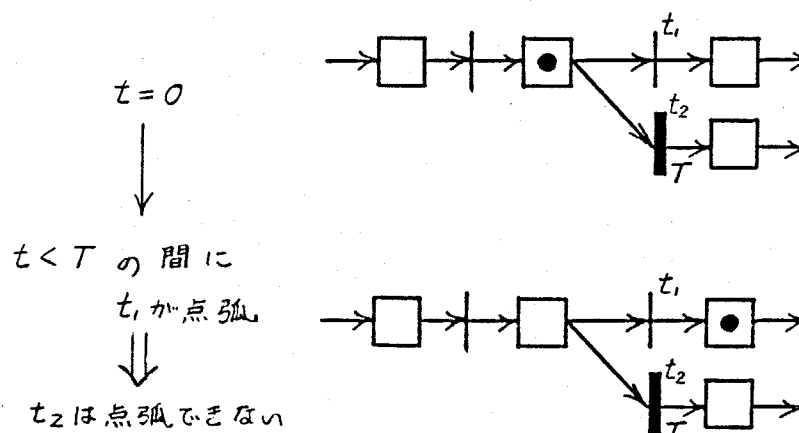
Fig.6-2 時間ボックスにおけるマークの動き

6.2.3 時限トランジション

これは、その点弧条件が指定持続時間 (specified duration) T だけ維持できるか否かを判定し、維持できた場合は点弧する機能をもつトランジションである。

時限トランジションは、可点弧条件が成立し、その条件が割り当てられた時間だけ維持されると、その時点で時間的抑止を解除し点弧する。なお、途中で可点弧条件が満たされなくなった場合、維持できなかつたとして点弧条件はリセットされるものとする。時限トランジションにおける点弧のようすを Fig.6-3に示す。

(a)



(b)

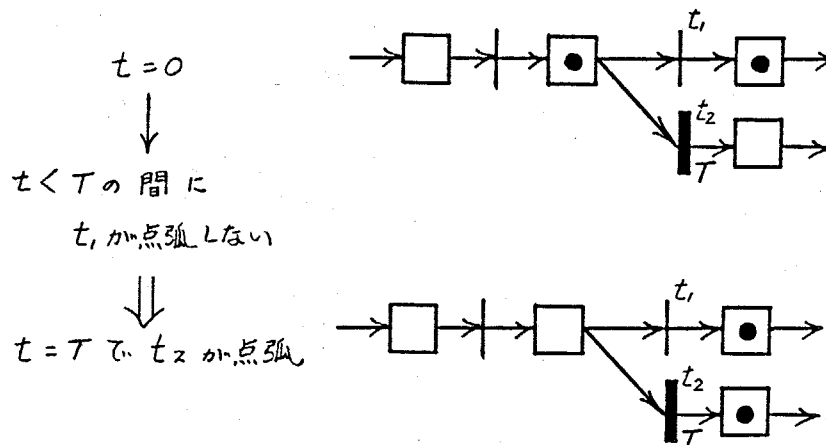


Fig.6-3 時限トランジションの点弧

6.2.4. 時間の導入によるMFGの性質の変化

時間を導入したことにより、つぎのようにMFGの性質の一部が変わる。

(1) 時間ボックスに対する反転定理の不成立

時間ボックスに対しては、反転定理[3.5.3参照]は成立しない。なぜなら、時間ボックスではマークの時間的拘束が行なわれるため、ボックス内にマークが入ってきても一定時間はその出力側トランジションの可点弧条件に寄与しない。そのため、方向枝の向きを反転すると、マークの有無および信号枝の論理を反転してもそのボックスの入力側と出力側の可点弧条件は変わってしまう。

(2) 時間の割り当てによるデッドロックの回避

MFG内のボックスあるいはトランジションに時間を割り当てると、マーキングの遷移が時間に依存するため、その系列を規定することも可能である。競合ボックスをもつMFGの場合、この性質を利用して競合ボックスに接続されたトランジションの点弧順序を制御することもでき、それによりデッドロックを回避することができる場合がある。

6.3 MFGへのマクロなボックスの導入

6.3.1 マクロなボックス

MFGにより非連続生産システムをモデル化した場合、システム内を流れるワークピースはマークで表わされ、またワークピースの占める空間的な所在場所、たとえばワークステーションやコンベアあるいは製品置場などは、ボックスまたは複数のボックスの結合で表わされる。同様に制御の流れも表わされるが、進行条件などの規定においても複数のボックスとトランジションの結合で表わされることがある。このように、ある1つの機能や要素を表わすのに、いくつかのボックスやトランジションを用いる

場合がよくみられる。そこで、外部と信号の授受が行なわれない部分や詳細な内部動作についての状況が必要でない部分については、グラフが繁雑になるのを避けるために、よく現れる機能や要素をマクロなボックスとして定義する。

6.3.2 容量ボックス

非連続生産システムの中には、バッファのように容量が規定された要素があり、また加工ユニットにおいても同時に受け入れられるワークピースの数が制限されている場合がある。バッファ内の状態を特に問題としない場合や受け入れ数の制限を規定する場合、Fig.6-4(a)に示すように容量に相当する個数のボックスで表現できる。これと等価なマクロなボックスとして、Fig.6-4(b)に示す容量ボックス(capacity box)を定義する。

容量ボックスは、規定された個数までのマークを受け入れることができる。可点弧への寄与は、ボックスの容量を N 、ボックス内のマーク数を n とすると、

$$\left\{ \begin{array}{l} \text{入力側トランジションの可点弧の必要条件} \quad n < N \\ \text{出力側トランジションの可点弧の必要条件} \quad n > 0 \end{array} \right.$$

である。また、入力側トランジションが点弧するとボックス内にマークが1つ増え、出力側トランジションが点弧するとボックス内のマークが1つ減る。

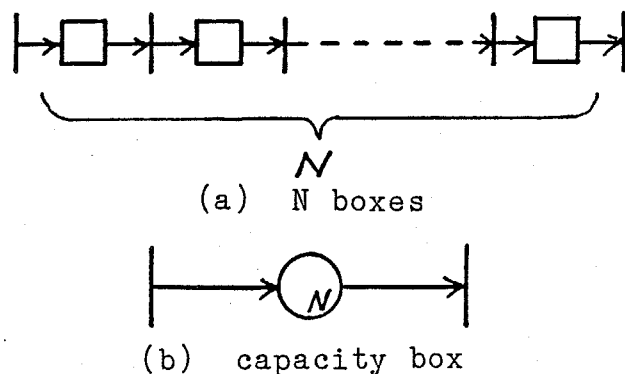


Fig.6-4 容量ボックス

6.3.3 パレタイジングボックス

生産工程において、いくつかのワークピースを1枚のパレットに載せたりマガジンに収め、それを単位として処理をする場合がある。また、バッチ処理などにおいては、所定の個数に達するまで待ちその個数になると作業を開始する。このような動作はFig.6-5(a)のグラフで表現できるが、これと等価なマクロなボックスとして、Fig.6-5(b)に示すパレタイジングボックス(palettizing box)を定義する。

パレタイジングボックスは、ボックス内のマーク数が規定数量未満のときのみその入力側トランジションの点弧を許し、点弧するとボックス内にマークが1つ増える。ボックス内のマーク数が規定数量に達すると出力側トランジションの点弧を許し、点弧するとボックス内すべてのマークは消滅する。

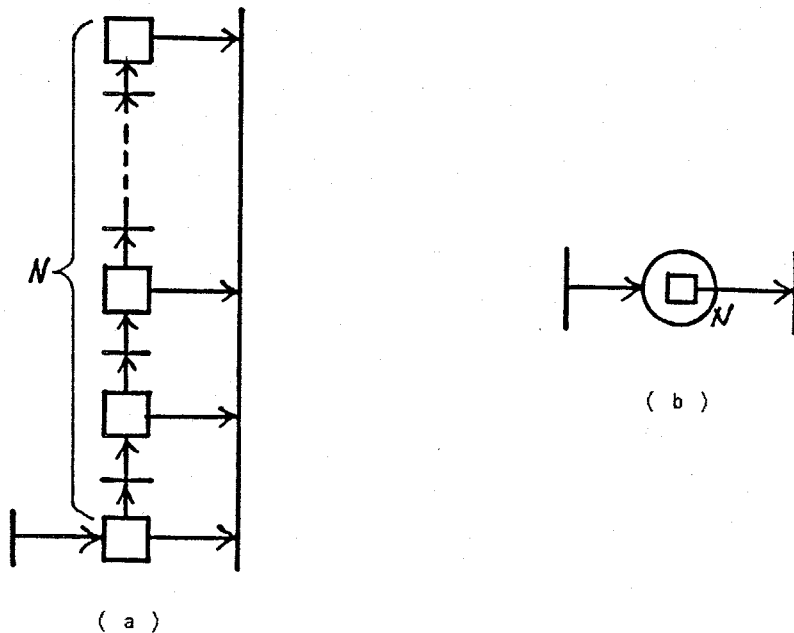


Fig.6-5 パレタイジングボックス

6.3.4 デパレタイジングボックス

上で述べたパレタイジング動作とは逆に、パレットやマガジンに収められているワークピース、あるいはバッチ処理終了後のワークピース1つずつ取り出すという動作もある。これはFig.6-6(a)に示すグラフで表現できるが、等価なマクロなボックスとして、Fig.6-6(b)に示すデパレタイジングボックス(deparettizing box)を定義する。

デパレタイジングボックスは、ボックス内にマークが1つもないときのみその入力側トランジションの点弧を許し、点弧するとボックス内に規定数量のマークが生ずる。また、ボックス内にマークがあるときのみ出力側トランジションの点弧を許し、点弧するとマークが1つずつ減る。

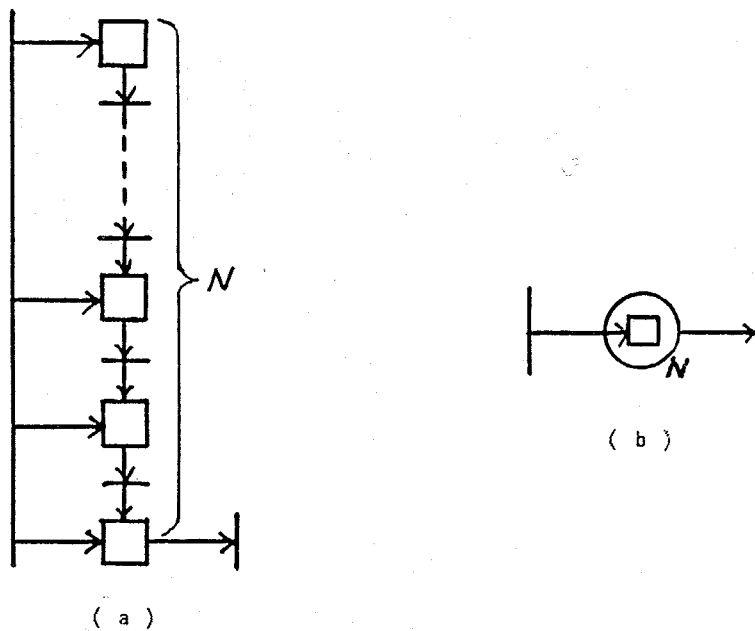


Fig.6-6 デパレタイジングボックス

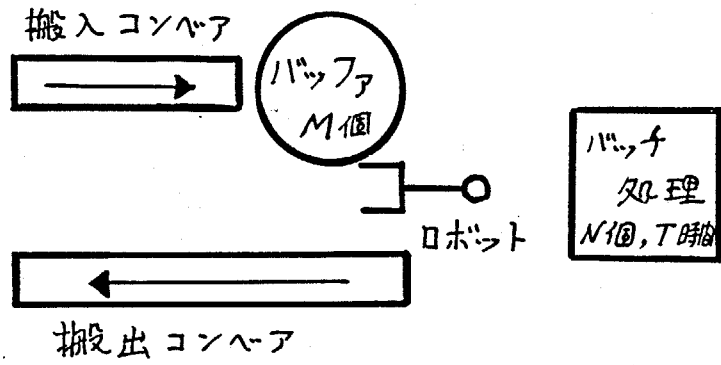
6.3.5 表現例

時間およびマクロなボックスの導入されたMFGによる、システムの表現例を示す。

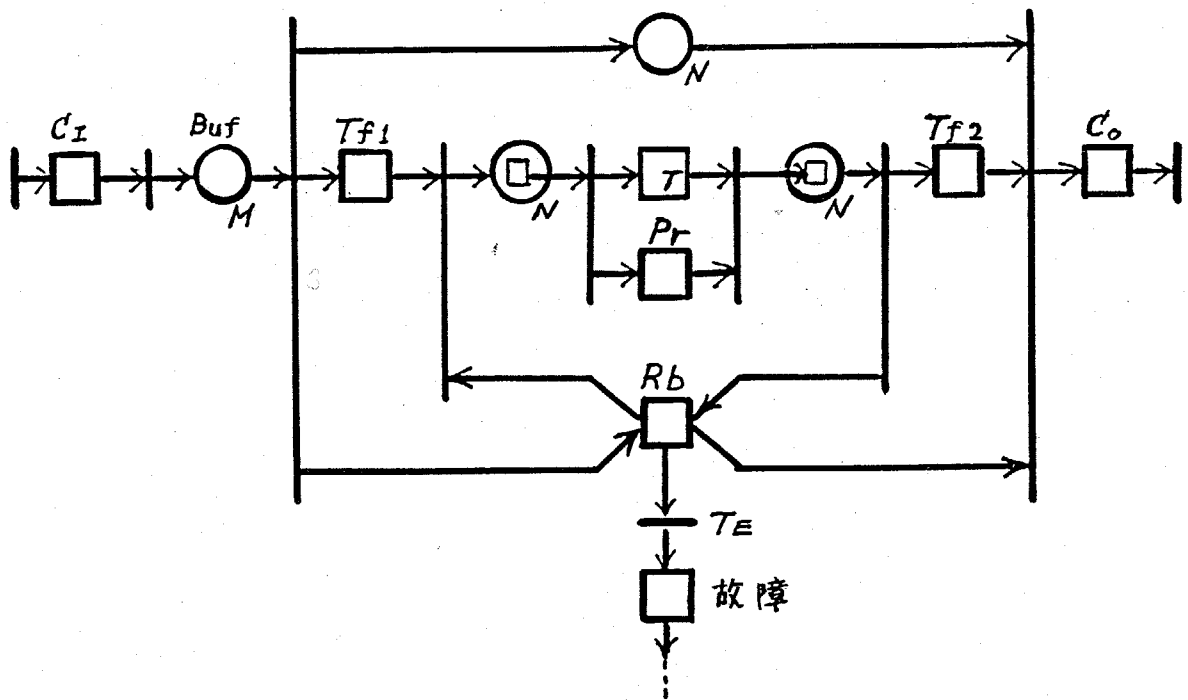
Fig.6-7(a)は、搬入コンベア、搬出コンベア、ロボット、容量M個のバッファ、N個のワークピースを一括してT時間の処理を行なうバッチ処理装置で構成されているシステムであり、その動作仕様は、

- ① 搬入コンベアによりワークピースが容量M個のバッファに搬入される
- ② ロボットがバッファから処理機に1つずつワークピースをロードする
- ③ 処理機はN個のワークピースがセットされると処理を開始する
- ④ T時間後に処理が終了する
- ⑤ ロボットが処理済のワークピースを1つずつとりだし搬出コンベアに載せる
- ⑥ 搬出コンベアがワークピースを搬出する
- ⑦ ロボットがT_e時間以上たっても作業を完了しない場合は故障と判断するというものである。

これを、MFGで表現するとFig.6-7(b)となる。



(a)



(b)

Fig.6-7 時間およびマクロを含むMFGの例

6.4 シミュレータ

上で述べたMFGによりモデル化したシステムをシミュレートするために、パーソナルコンピュータを用いたシミュレータを作成した。

このシミュレータは、Fig.6-8 に示す形式によりシステムを表わすMFGが入力される。このMFGに対し、単位時間をきざみとして時刻を定め、時間およびマクロなボックスを導入したMFGの規則にしたがって、各時刻ごとに可点弧なトランジションを点弧しマーキングを遷移させていく。その原理は5.4.3と同様である。ただし、時間の割り当てられていないボックスやトランジションについては、マークの移動に要する時間は理想的に0とするので、同一時刻においてMFGのタイムシーケンスが複数段進むこともある。

シミュレーションの結果として、Fig.6-9 のようにトランジションが点弧しマーキングの遷移が生じた時刻ごとに、その時点で点弧したトランジションとその結果のマーキング（容量のあるボックスについてはマーク数も含む）がプリントアウトされる。また、いくつかの特定のボックスに注目しその時系列的な挙動を知りたい、たとえばバッファにたまるワークピースの数が時間とともにどのように変化するかを求めるといった場合、Fig.6-10に示すように注目するボックス内の各時刻におけるマーク数の経過として出力できる。

そのほか、システムの生産能力や各機器の稼働率の算出に必要なデータとして、シミュレーションを行なった時間内の各トランジションの点弧回数（すなわちマークの通過数）、および各時間ボックスにおける有効マーク存在時間率がFig.6-11のように出力される。ここで有効マーク存在時間率とは、マークが時間ボックスにより時間的拘束を受けていた（機器の稼働に相当する）時間の総和の全時間に対する比を表わすものである。

CONNECTION

	BOX NO.
T 1	: 1
T 2	: -1 2
T 3	: -2 3
T 4	: -3 4 66
T 5	: -4 13 -66
T 6	: 5
T 7	: -5 6
T 8	: -6 7
T 9	: -7 8 66
T10	: -8 13 -66
T11	: 9
T12	: -9 10
T13	: -10 11
T14	: -11 12 66
T15	: -12 13 -66
T16	: -13 14
T17	: -14 15
T18	: -15 16
T19	: -16 17
T20	: -17 18
T21	: -18 19
T22	: -19 20
T23	: -20 21
T24	: -21 22
T25	: -22 23
T26	: -23 24
T27	: -24 25
T28	: -25 26
T29	: -26 27
T30	: -27 28
T31	: -28 29
T32	: -29 30
T33	: -30 31
T34	: -31 32
T35	: -32 33 66
T36	: -33 34 -66
T37	: -34 35
T38	: -35 36 66

T39	: -36 37 -66
T40	: -37 38
T41	: -38 39 40 74
T42	: -39 41
T43	: -40 41
T44	: -41 42 67
T45	: -42 43 -67
T46	: -43 44
T47	: -44 45 67 68 -73
T48	: -45 46 49 -67
T49	: -46 47
T50	: -47 48
T51	: -49 50
T52	: -48 -50 51 67
T53	: -51 -67
T54	: -44 52 67 -69 70
T55	: -52 53 56 -67
T56	: -53 54
T57	: -54 55
T58	: -56 57
T59	: -55 -57 58 67
T60	: -58 -67
T61	: -44 59 67 -71 72
T62	: -59 60 63 -67
T63	: -60 61
T64	: -61 62
T65	: -63 64
T66	: -62 -64 65 67
T67	: -65 -67
T68	: -68 69
T69	: -70 71
T70	: -72 73
T71	: -74 75
T72	: -75
T74	: 76
T75	: -76 77
T76	: -77

GATE CONDITION

	BOX NO.
T 4	: -13
T 9	: -13
T14	: -13
T16	: 77
T35	: -34
T38	: -37
T41	: -75
T44	: -43
T47	: -48 -49
T54	: -55 -56
T61	: -62 -63
T74	: -77

Fig.6-8 シミュレータへの入力

BOX SPECIFICATION

B 1 (T= 46),	B 2 (T= 46),	B 3 (C= 2),	B 4 (T= 1),	B 5 (T= 46),
B 6 (T= 46),	B 7 (C= 2),	B 8 (T= 1),	B 9 (T= 46),	B10 (T= 46),
B11 (C= 2),	B12 (T= 1),	B14 (T= 12),	B15 (T= 12),	B16 (T= 12),
B17 (T= 12),	B18 (T= 12),	B19 (T= 12),	B20 (T= 12),	B21 (T= 12),
B22 (T= 12),	B23 (T= 12),	B24 (T= 12),	B25 (T= 12),	B26 (T= 12),
B27 (T= 12),	B28 (T= 12),	B29 (T= 12),	B30 (T= 12),	B31 (T= 12),
B33 (T= 1),	B35 (C= 39),	B36 (T= 1),	B39 (T= 6),	B40 (T= 13),
B41 (C= 2),	B42 (T= 1),	B44 (C= 23),	B45 (T= 1),	B46 (P= 24),
B47 (T=300),	B48 (D= 24),	B50 (C= 23),	B51 (T= 1),	B52 (T= 1),
B53 (P= 24),	B54 (T=300),	B55 (D= 24),	B57 (C= 23),	B58 (T= 1),
B59 (T= 1),	B60 (P= 24),	B61 (T=300),	B62 (D= 24),	B64 (C= 23),
B65 (T= 1),	B68 (P= 24),	B69 (D= 24),	B70 (P= 24),	B71 (D= 24),
B72 (P= 24),	B73 (D= 24),	B74 (P= 37),	B75 (T= 73),	B76 (T= 11),
B77 (T= 1),				

6.5 実例

上記のシミュレータを用いて、実在の非連続生産システムのシミュレーションを行ったので、その結果を示す。

6.5.1 対象とした非連続生産システム

対象とした非連続生産システムは、Fig.6-12に示すように大きく分けて A,B,C,D,E の5つの段階を経て製品を製造する。

ワークピースはAに搬入され加工を受けた後、Bでの加工を経て容量が2のBの出口の置場に置かれる。A1, A2, A3 および B1, B2, B3 はそれぞれ同一の加工機であり、加工に要する時間は A, Bともに46分である。加工機 A, B からなるステーションは3組ありそれぞれ独立に稼働するが、同一の品種を扱っている。

Bの出口にワークピースが到着すると、ロボット1がそれを加工ラインCの入口に搬送する。ただし、Cの入口の容量は1であるので、Cの入口に前のワークピースがあるうちはロボットは搬送作業には移行しない。

加工ラインCは全部で18段の工程からなり、ワークピースは12分のタクトタイムでライン内をシフトしてゆく。18段の工程を終了すると、Cの出口の置場に到達する。

Cの出口に到着したワークピースは、ロボット1により容量が40のバッファ1に搬送される。ただし、バッファが満杯の場合には、ロボットは搬送作業に移行しない。さらに、バッファ1内にワークピースがありかつ容量2の加工機Dの入口が受け入れ可能なとき、ロボット1はワークピースをバッファ1からDの入口に搬送する。

加工機Dは、素材として受け入れた1つのワークピースを2つに分割したのち処理を行ない、受け入れてから6分後に1つ目そして13分後に2つ目を容量2のDの出口の置場に排出する。なお、この機械は実働8時間毎に1時間の保守作業が行なわれる。

Dの出口にワークピースが置かれると、ロボット2が容量24のバッファ2に搬送する。ただし、バッファが満杯の場合にはロボットは搬送作業に移行しない。

装置E1, E2, E3は同一の機械で、それぞれ24個のワークピースをまとめて一括して300分の処理を行なうバッチ処理装置である。ワークピースはロボット2によりバッファ2から搬入されるが、その順序は初めの24個はE1, 次の24個はE2, その次の24個はE3, 以後この繰り返しである。Eは24個のワークピースを受け入れると、処理を開始する。処理終了後、処理済のワークピースはロボット2により1つずつ搬出される。処理済のワークピースが全て搬出された時点で、次の新しいワークピースの受け入れが可能となる。

この生産システムをモデル化したMFGを、Fig.6-13に示す。

6.5.2. シミュレーション (1)

この生産システムは、無休止連続操作を行なう。つまり、システムの最上流である加工機 A には、常にワークピースが供給されている。しかし、システムの途中にある加工機 D は、上で述べたように実働 8 時間毎に 1 時間の保守作業が行なわれ、この間は D での加工作業は行なわれない。この影響として、加工機 D の上流側では保守作業中に未処理のワークピースが滞留し、また下流側では再開時に処理済のワークピースが一時的にたまる可能性がある。これを吸収させるために上流側ではバッファ 1 が、下流側ではバッファ 2 が用意してある。また装置 E では他の機器に比べ処理時間の長いバッチ処理を行なっているため、その入口側で待ちが生ずることがあり、バッファ 2 はこの吸収にも利用されている。

そこで、主としてバッファ 1 やバッファ 2 の必要な容量を求めるため、および E1, E2, E3 における挙動を確認することを目的としてシミュレーションを行なった。この結果を用いて求めた線図を Fig.6-14 に示す。

これより、平常時において

(1) バッファ 1 と加工機 D の入口には合せて最大 5 個のワークピースがたまるが、次の保守作業が始まるまでには解消される

(2) バッファ 2 にはワークピースがたまらない

(3) バッチ処理装置 E では平均して約 50 分の空き時間が生じてる

などがわかる。

またその他の装置の稼働率を算出すると、加工機 A, B が 100% であるのに対し、C や D では 80% 前後となっていることもわかった。

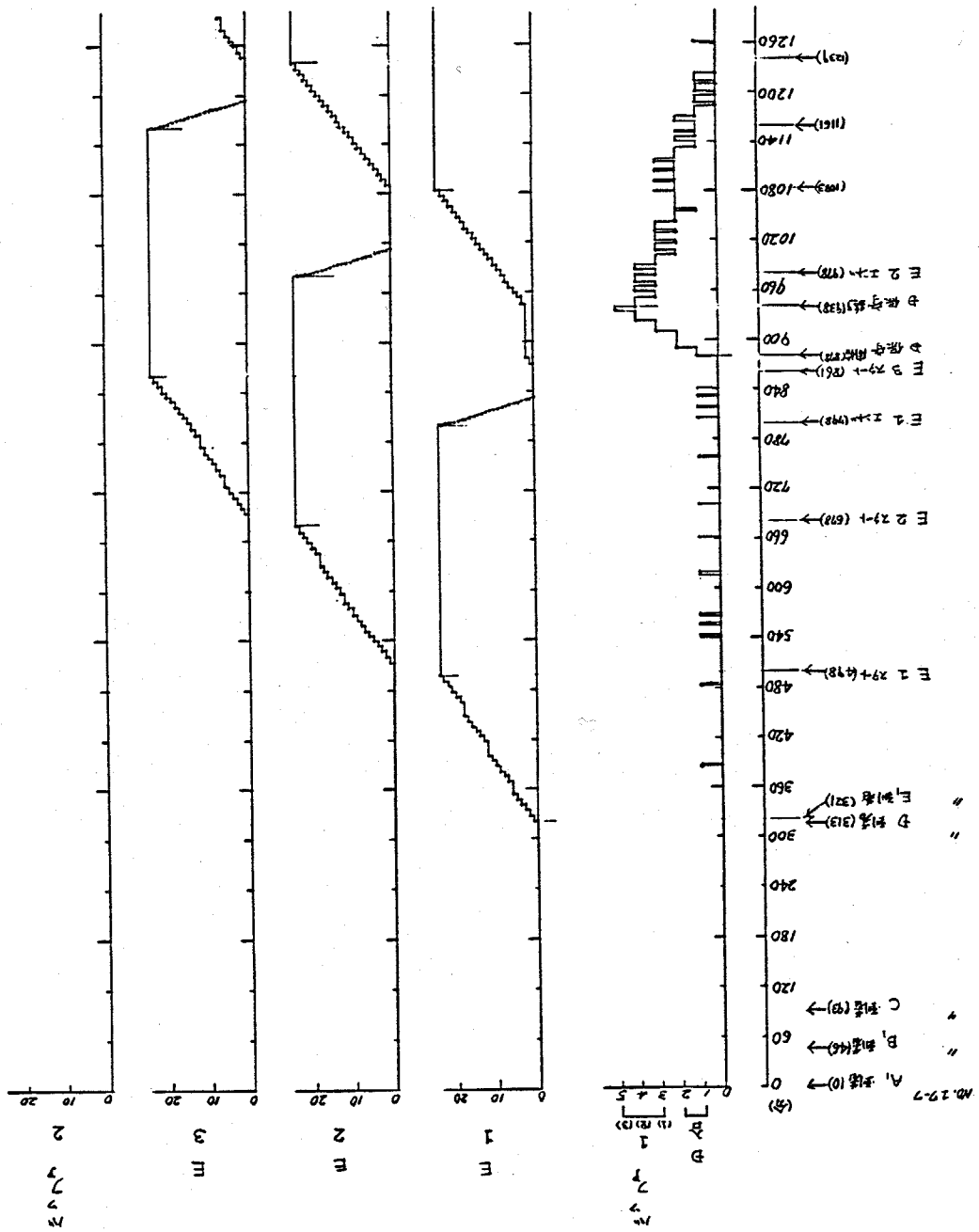


Fig. 6-14 シュローターの結果による線図

6.5.3 シミュレーション (II)

前節のシミュレーション結果より、この生産システムの現状として、加工機 A, B は製造能力を最大限に利用しているが、その他の装置にはまだ余裕があることがわかった。また装置 C, D, E については、現在その最大定格速度以下で使用しており、スピードアップが可能とされている。そこで、加工機 A, B からなるステーションを 1 組増設し、さらにその他の装置の速度を向上することにより、システム全体として増産できることが見込める。増産にともなうシステムの設計変更にはいろいろな案があるが、それらの妥当性を検討する場合においてもシミュレーションは有効である。

変更案のうち最も実現しやすいものとして、

- (1) 加工機 A, B からなるステーションを 1 組増設する
- (2) 加工ライン C のタクトタイムを 12 分から 11 分に短縮する
- (3) 加工機 D の加工時間を (6 分, 13 分) から (5 分, 10 分) に短縮する
- (4) 装置 E の処理時間を 300 分から 250 分に短縮する

という案にもとづき、Fig.6-13 の MFG を修正し Fig.6-15 とし (修正した部分は下線を付してある)、再びシミュレーションを行なった。6.5.2 で問題とされた項目について、この結果より求めた線図を Fig.6-16 に示す。

ライン拡張案

- C ----- 9分 12分 → 11分
- D ----- 入口側 13分 → 10分
出口側 6.5分 → 5分
- E ----- 混在存し (混水貯器)
容量 24
処理時間 300分 → 250分

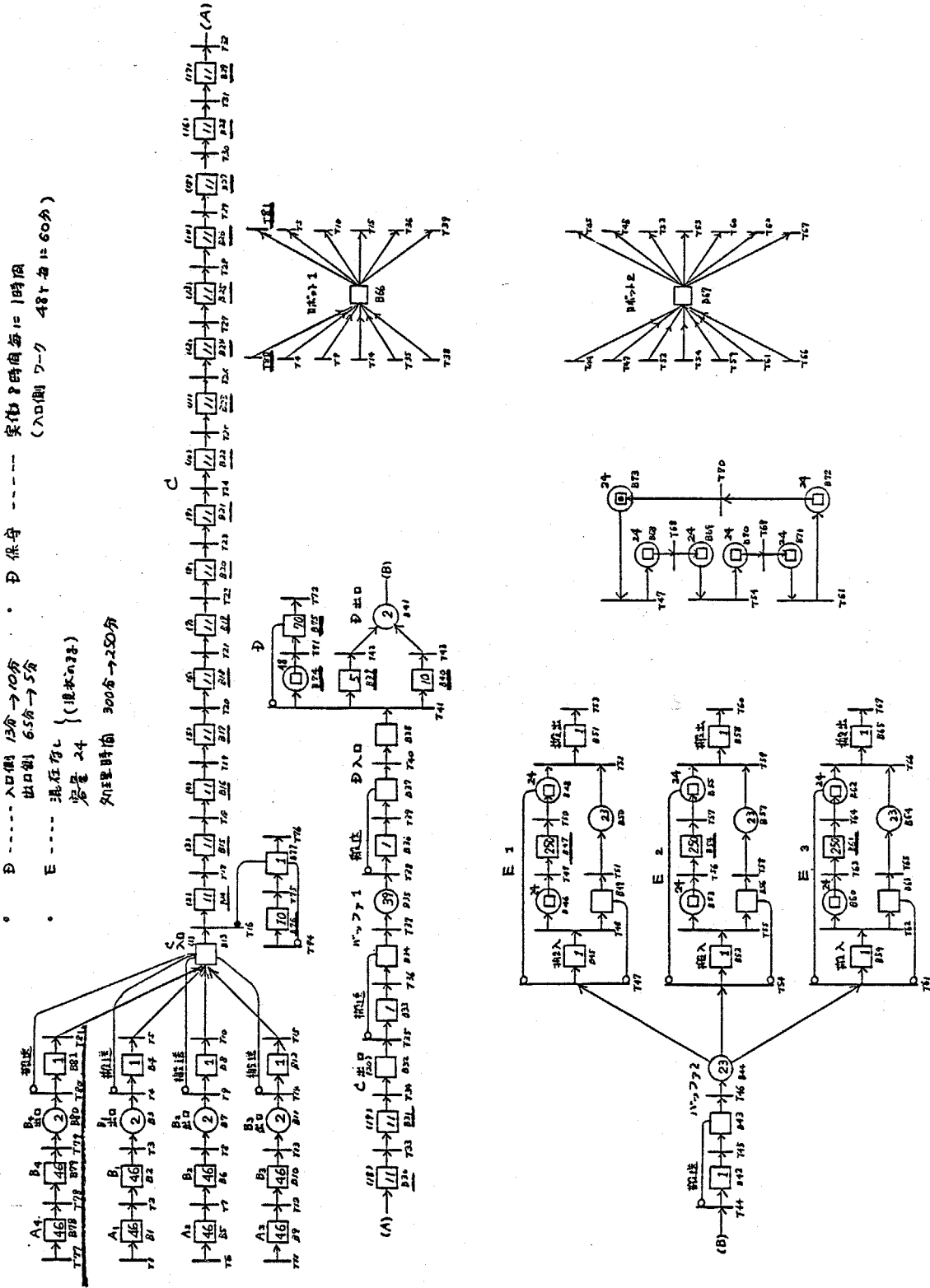


Fig. 6-15 変更された M F G

ライン拡張案 シミュレーション結果

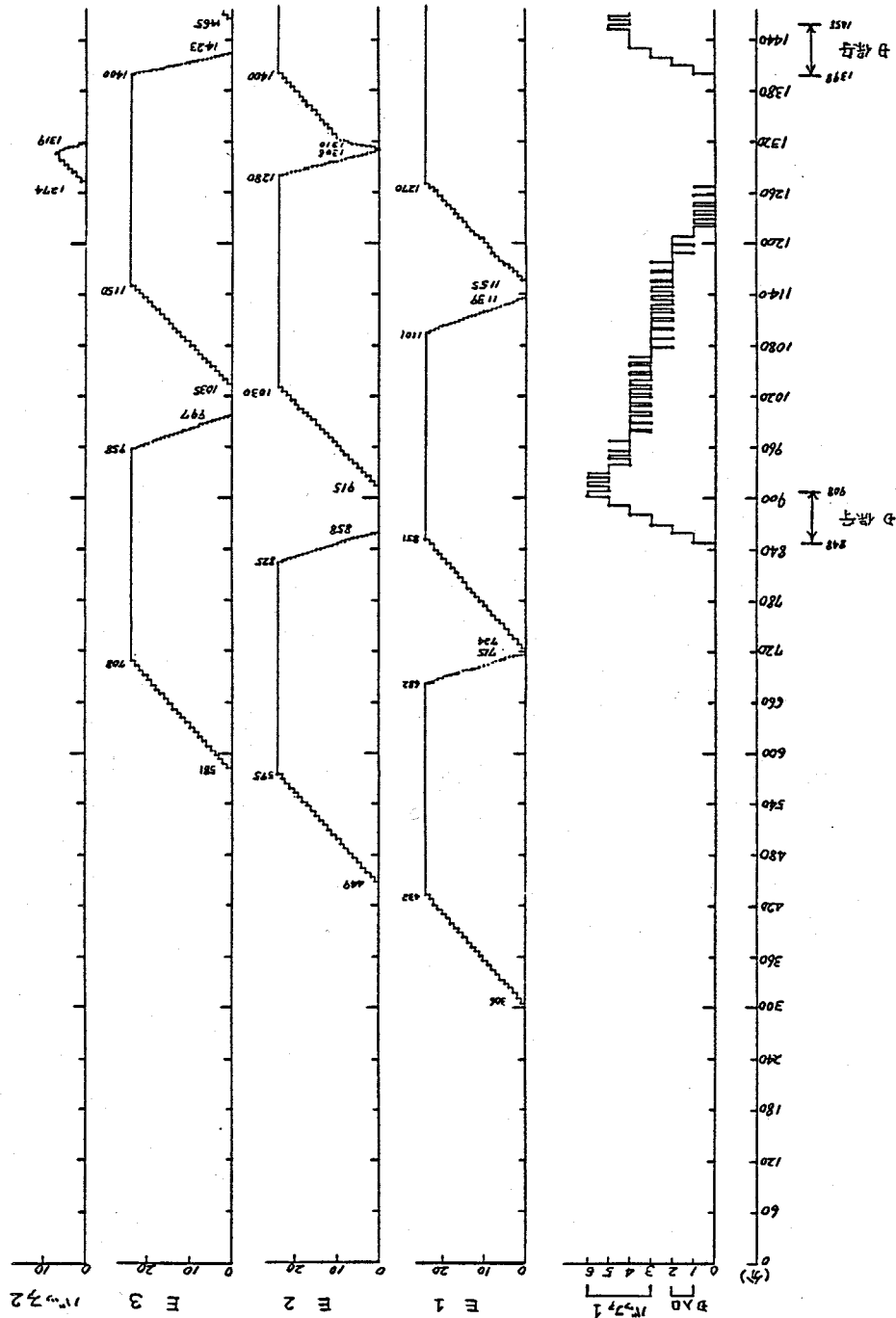


Fig.6-16 変更後のシミュレーションによる線図

これよりわかることは、

(1) バッファ1と加工機Dの入口にたまるワークピースの数が5個から6個に増すが、次の保守作業が始まるまでには解消され問題がない。ただし、解消されるまでに要する時間は長くなっている

(2) バッファ2に最大7個のワークピースがたまるようになった

(3) 装置Eでは空き時間がわずかになった。バッファ2にワークピースがたまるようになったのはこの影響である。しかし、たまったワークピースは後に影響を残すことなく解消され問題がない

などである。

また各機器の稼働率は、A、Bは100%と最大稼働のままであり、C、Dは95%前後に改善されている。生産能力についても、変更前は1時間当たり3.9個であったのに対し、変更後は1時間当たり5.2個と1.33倍に向上している。

以上より、システムは不都合なく動作し所期の増産が達成されることがわかり、この案の妥当性が確認できた。また、その際必要となるバッファの容量も得ることができた。

6.6 今後の課題

MFGに時間の概念とマクロなボックスを導入することにより、非連続生産システムのシミュレーションをMFGにより行なえることを確認した。ここで対象としたシステムは単品種しか流れないシステムであったが、多品種少量生産にともない同一システム内を複数の品種が混在して流れる場合が増加している。このようなシステムでは、品種によりワークピースの流れ方や機械における加工時間が変化する。このような混流へ対応することが今後の課題である。

第 7 章 非連続生産システムをモデル化する M F G の自動生成の基礎

7.1 M F G の自動生成

非連続生産システムを M F G により制御あるいはシミュレートするためには、システムを M F G で記述する必要がある。しかし、複雑な仕様のシステムの場合には、この作業は手間のかかるものとなる。そこで、与えられた仕様によりそのシステムをモデル化する M F G を、自動的に生成することが望まれる。M F G は 2.2 で述べたように、ペトリネットの概念に「安全」という性質と「信号の入出力機能」の表現能力をもたせたものである。M F G の自動生成を考えるにあたり、まずはその原点のペトリネットに戻り検討し、そのうえで M F G に変換を行なうことにする。本章では、この基礎的なアルゴリズムを提案する。

7.2 取り扱う問題

並列、非同期動作が行なわれる非連続システムをペトリネットで表現するにあたり、システム構成要素の動作や処理を状況 (condition) に割当てるか事象 (event) に割当てるかが問題になる。これは、事象のとらえ方、すなわち primitive event としてとらえるか non-primitive event とするかに依存するが、どちらの場合も等価である。

7-1) これらを事象としてとらえると、システムの動作順序仕様は事象を表現しているトランジションの点弧系列とみなすことができる。

ここではまず、システム構成要素の動作順序仕様を事象に対応するトランジションの点弧系列とみなし、この点弧系列を繰り返し発生するペトリネットの生成アルゴリ

ズムを提案する。ただし、点弧系列は次の条件を満たすものとする。

(1) 系列の中に、同一のトランジションが繰り返し出現してよい。

(2) 並列動作区間がある場合、同時に進行する各系列は独立であるものとする。すなわち、同時進行する各系列内での点弧順序は一定であるが、その間のシステム全体としての点弧順序には規定がなく、並列な系列はそれぞれ他の系列に干渉されることなく点弧が行なわれる。そのため、同一のトランジションが独立に進行するいくつかの系列にあってはならない。

なお、生成するペトリネットは、セルフープおよび多重アークをもたないものとする。

7.3 重複接続行列

ペトリネットの表記法の一つに、トランジションに対応する行とプレースに対応する列で構成される接続行列 C がある。セルフープおよび多重アークをもたないペトリネットにおいては、接続行列の (i, j) 要素 c_{ij} を、

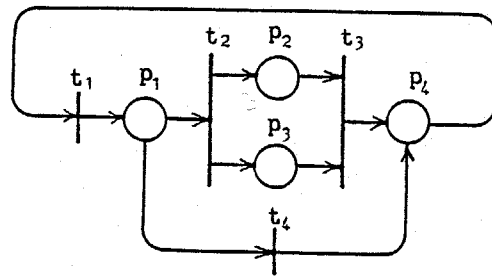
$$c_{ij} = \begin{cases} 1: \text{トランジション } t_i \text{ の出力側にプレース } p_j \text{ が接続されている} \\ -1: t_i \text{ の入力側に } p_j \text{ が接続されている} \\ 0: t_i \text{ と } p_j \text{ は接続されていない} \end{cases}$$

と定めることにより、グラフを表現することができる。

本アルゴリズムではこの接続行列の他に、あるトランジションに関する行が重複して出現することを許す行列 C' を導入する。 C' の k 行および l 行が共に t_i に関する行であるとき、 c'_{kj} と c'_{lj} はどちらも c_{ij} に等しいものとする。なお、 C' を重複接続行列とよぶことにする。Fig.7-1に例を示す。(図中では、0は省略して空白としてある。)

7.4 基本経路の作成

与えられた点弧系列を発生するペトリネットを生成するにあたり、まず、トランジションをブレースとアークにより接続し所要の点弧系列を可能とする基本経路を作成する。本アルゴリズムでは、以下の手順により重複接続行列の各要素を決めることで行なう。なお、説明のために対応するペトリネットグラフも付記する。



ペトリネットの例

	P ₁	P ₂	P ₃	P ₄
t ₁	1			-1
t ₂	-1	1	1	
t ₃		-1	-1	1
t ₄	-1			1

接続行列

	P ₁	P ₂	P ₃	P ₄
t ₁	1			-1
t ₂	-1	1	1	
t ₃		-1	-1	1
t ₄	-1			1

重複接続行列

Fig.7-1 接続行列と重複接続行列

7.4.1 基本手順

経路の作成において、重複するトランジションおよび並列区間のない場合が基本となる。

$$t_1 - t_2 - t_3 - t_4$$

なる点弧系列を例として、この基本手順を示す。

(1) 点弧系列を重複行列の行に付す。 [Fig.7-2(a)]

(2) 第1列に $*p_1$ (*が付いているブレースは、初期マーキングでトークンを一つ持たせる) を付し、(1,1) 要素を -1 とする。これは、最初に点弧すべきトランジションの入力側に、初期マーキングにおいてトークンを一つ持つブレースを接続したことになる。

第2列に p_2 を付し、(1,2) 要素を 1 とする。すなわち、 t_1 の出力側ブレースとして p_2 を接続した。 [Fig.7-2(b)]

(3) 第2行に移り、上の行で要素が 1 であった列の成分を -1 とする。この例では (2,2) 要素。これは、前に点弧するトランジションの出力ブレースを、このトランジションの入力ブレースとして接続したことになる。

そして、第3列に p_3 を付し (2,3) 要素を 1 とする。 [Fig.7-2(c)]

(4) 第3行も (3) と同様にして要素を決める。

(5) 最終行では -1 とする要素は (3) と同様に決めるが、1 とする要素は最初の行で -1 となっている列を選ぶ。これは、最後に点弧するトランジションの出力ブレースを最初に点弧するトランジションの入力ブレースとすることであり、これによりマーキングを初期状態に戻すことができる。 [Fig.7-2(d)]

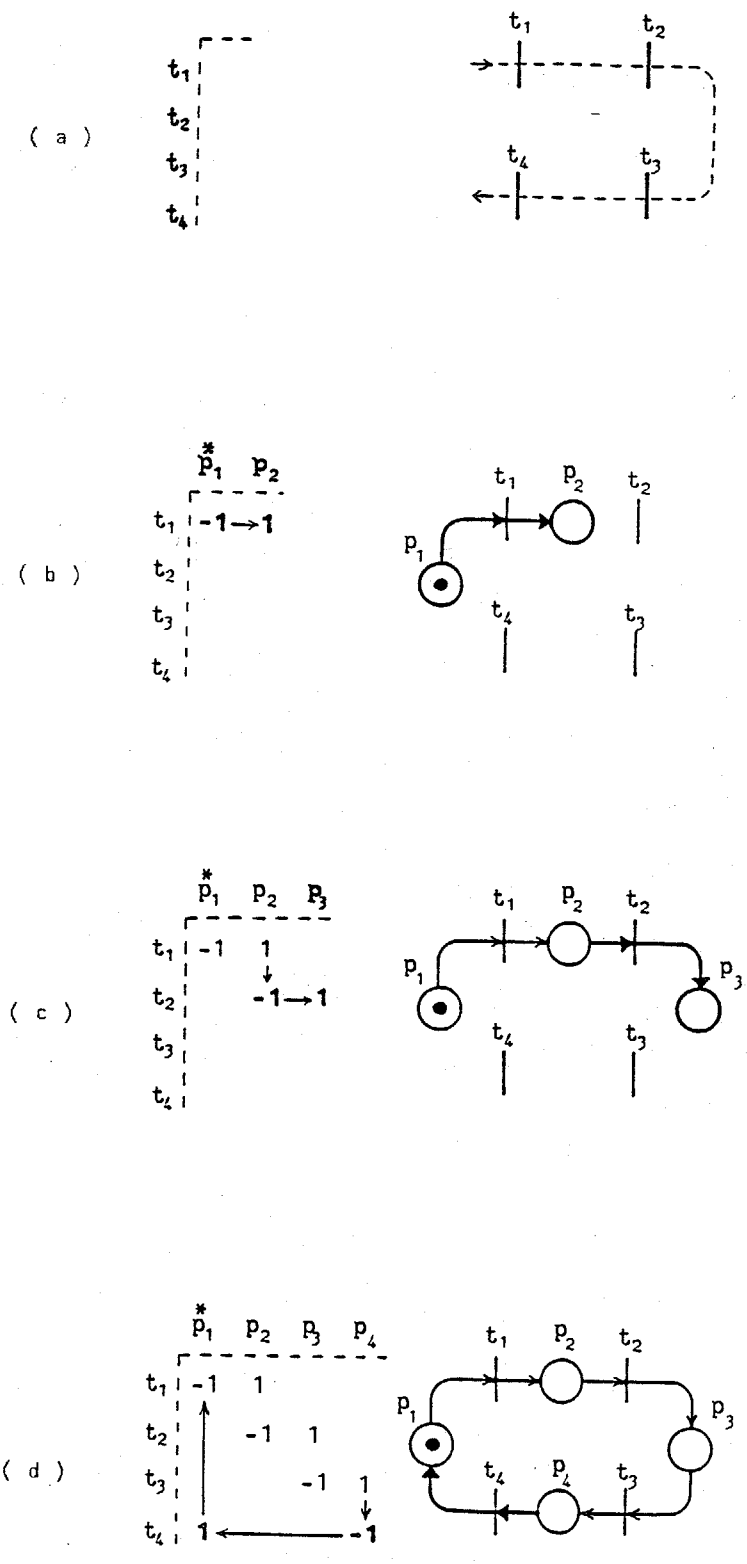


Fig.7-2 基本手順

7.4.2 重複の取り扱い (I)

たとえば

$$t_1 - t_2 - t_3 - t_4 - t_2 - \dots$$

のように、同一のトランジションが重複して出現する場合は、つぎのように取り扱う。

この例では、まず第1, 2, 3行を基本手順により決定しておく。ただし、第2行および第5行はどちらも t_2 に関する行であるので、第2行の各要素が決まった時点で第5行にも同じ内容を記入する。 [Fig.7-3(a)]

そのため、第4行ではその次の行 (第5行) がすでに決まっている。そこで、上の行の1の下に-1を、下の行の-1の上に1を記入する。 [Fig.7-3(b)] これは、前に点弧するトランジションの出力ブレースと後に点弧するトランジションの入力ブレースが決まっている時、両者をこのトランジションにより接続したことになる。

[Fig.7-3(c)]

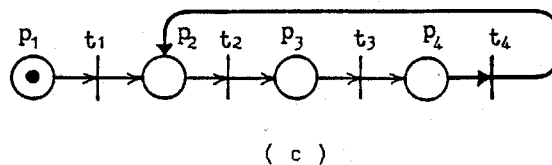
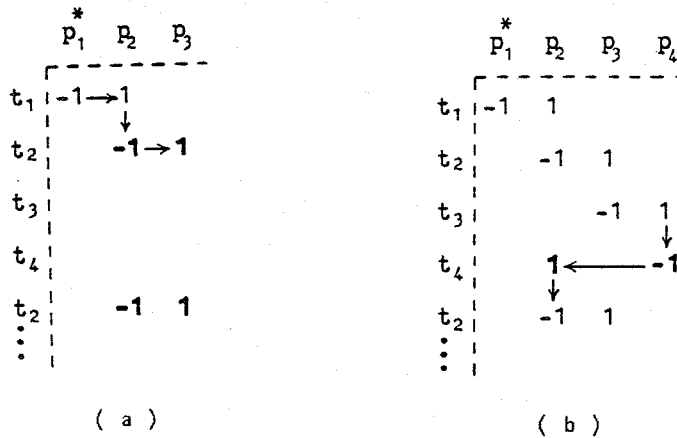


Fig.7-3 重複の取り扱い (I)

7.4.3 重複の取り扱い (II)

また、

$$t_1 - t_2 - t_2 - \dots$$

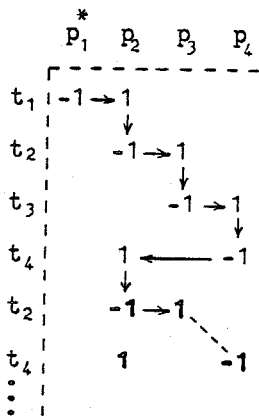
のように同じトランジションが連続して現れる場合、あるいは

$$t_1 - t_2 - t_3 - t_4 - t_2 - t_4 - \dots$$

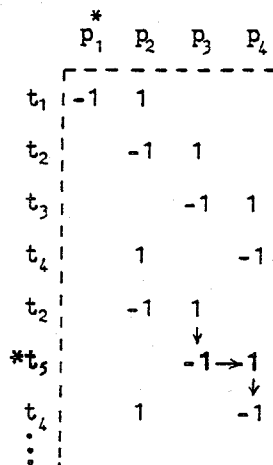
のように2つ以上のトランジションが重複して出現する場合などは、重複接続行列において連続した2行があらかじめ決まってしまうことがある。そして、この2行が経路を構成しない、すなわち上の行の1の下に-1がない場合が問題となる。 [Fig. 7-4(a)]

このような場合、経路を構成するためにこの2行の間にダミーのトランジションを挿入する。(ダミーのトランジションには、トランジション名に*を付してある。) このダミーのトランジションに関する行において、上の行の1の下-1を、下の行の-1の上に1を記入する。 [Fig.7-4(b)] これは、前に点弧するトランジションの出力ブレースとその次に点弧するトランジションの入力ブレースが異なっている時に、この2つのブレースをダミーのトランジションを介して接続したことになる。

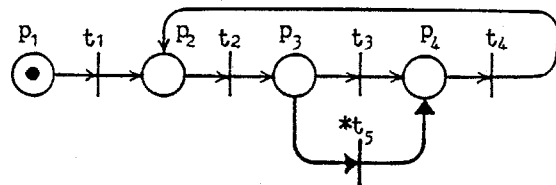
[Fig.7-4(c)]



(a)



(b)

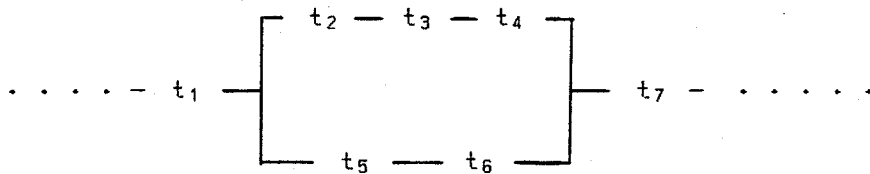


(c)

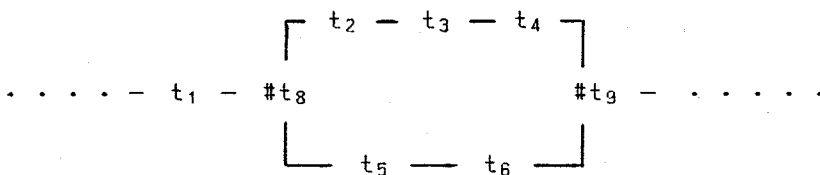
Fig.7-4 重複の取り扱い (II)

7.4.4 並列区間の取り扱い

点弧系列において、



のように並列区間がある場合、まず並列開始および並列終了を示すトランジション（トランジション名に#を付して区別する）を挿入する。すなわち、



とする。そして、並列して進行する各系列

$$\#t_8 - t_2 - t_3 - t_4 - \#t_9$$

および

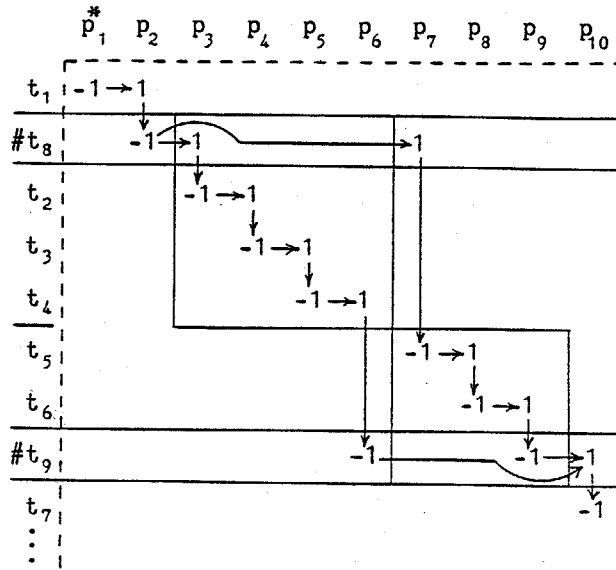
$$\#t_8 - t_5 - t_6 - \#t_9$$

に対し、それぞれ 7.4.1~7.4.3 で述べた手順により経路を作成する。なお、重複接続行列上では並列区間がわかるように、列に

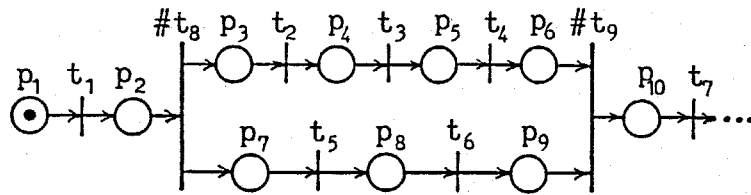
$$\dots t_1 \ \#t_8 \ t_2 \ t_3 \ t_4 \ | \ t_5 \ t_6 \ \#t_9 \ t_8 \ \dots$$

と付しておく。Fig.7-5に例を示す。

以上 7.4.1~7.4.4 で示した手順により、与えられた点弧系列を可能とする基本経路が得られる。しかし、トランジションの重複出現により複数の出力トランジションをもつブレースがあり、ここで生じる競合に関しては何ら規定していないので、この段階のペトリネットはまだ必要条件にすぎない。



(a)



(b)

Fig.7-5 並列区間の取り扱い

7.5 競合の調停

上で述べた手順により作られた基本経路において、与えられた点弧系列のみを発生させるために、以下の手順により競合を発見し、これを調停するグラフを付加する。

(1) いま作られた経路において、1つのブレースがその出力側で複数のトランジションに接続されている時、これらのトランジションが競合する。そこで、まず重複接続行列より重複する行を取り除き通常の接続行列を求め、-1なる要素を複数個持っている列を探す。これが、複数個の出力トランジションを持つブレースである。

[Fig.7-6(a)(b)] もしなければ、調停のためのグラフを付加する必要はない。

(2) 再び重複接続行列に戻り、(1)で探し求めた列において上から順に-1なる要素を探し出し、その行が対応しているトランジションを書き出す。このトランジション列が、競合するトランジション間の点弧系列である。 [Fig.7-6(a)]

(3) 求められた競合するトランジション間の点弧系列のそれぞれに対して、その点弧系列を規定するグラフを作成する。その際の手順は、この点弧系列を新たな問題として7.3と同様な手順に従う。そして、それを競合するトランジション間に付加する。 [Fig.7-6(d)] これにより、競合していたトランジションの点弧はここで付加したグラフに依存してその順序が決定されることになる。

(4) 競合するトランジションの点弧系列において重複がある場合、(3)で付加するグラフにおいて競合が生じることがある。そこで、付加するグラフそれぞれについて、(1)の手順に従って競合を発見する。もしあれば、(2),(3),(4)を競合がなくなるまで繰り返す。

競合の調停を行なうグラフの付加により、生成されたペトリネットの点弧系列は指定されたもののみ限定される。よって、以上のアルゴリズムにより作られたペトリネットは、与えられた仕様を満たすペトリネットの解の1つである。

7.6 実行例

以上のアルゴリズムを計算機に実行させ、点弧系列を入力しその点弧系列を発生するペトリネットを生成した。この生成過程をFig.7-7 に示し、また、そのペトリネットグラフをFig.7-8 に示す。

さらに、検証のために本アルゴリズムで生成されたペトリネットのシミュレーションを行なった。その結果がFig.7-9 であり、仕様を満たしていることが確認できる。

(INPUT = FIRING SEQUENCE)

T1 - T2 - T3 - [(T4 - T5 - T4)(T6 - T7)] - T2 - T8 - T3 - T9

(STEP 1)

	*									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
T1	-1	1								
T2		-1	1							
T3			-1	1						
NT10				-1	1		1			
T4					-1	1				
T5					1	-1				
T4					-1	1				
T6							-1	1		
T7								-1	1	
NT11		1				-1			-1	
T2		-1	1							
T8				-1						1
*T12			1							-1
T3			-1	1						
T9	1			-1						

Conflicting transitions and their firing sequence

T3, T8, T3;
T10, T9,
T5, T11,

(STEP 2)

	*										*		*		*	
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
T1	-1	1														
T2		-1	1													
T3			-1	1							-1	1				
NT10				-1	1		1						-1	1		
T4					-1	1										
T5					1	-1									-1	1
T4					-1	1										
T6							-1	1								
T7								-1	1							
NT11		1				-1			-1						1	-1
T2		-1	1													
T8				-1							1	1	-1			
*T12			1								-1					
T3			-1	1							-1	1				
T9	1			-1										1	-1	
*T13												1	-1			

Conflicting transitions and their firing sequence

T8, T13,

(STEP 3)

	*										*		*		*		*		
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	
T1	-1	1																	
T2		-1	1																
T3			-1	1							-1	1							
NT10				-1	1		1						-1	1					
T4					-1	1													
T5					1	-1									-1	1			
T4					-1	1													
T6							-1	1											
T7								-1	1										
NT11		1				-1			-1						1	-1			
T2		-1	1																
T8				-1							1	1	-1			-1	1		
*T12			1								-1								
T3			-1	1							-1	1							
T9	1			-1										1	-1				
*T13											1	-1				1	-1		

(CONNECTION)

	PLACE NO.			
T1	-1	2		
T2	-2	3		
T3	-3	4	-11	12
T4	-5	6		
T5	5	-6	-15	16
T6	-7	8		
T7	-8	9		
T8	-3	10	11	-12
T9	1	-4	13	-14
NT10	-4	5	7	-13
NT11	2	-6	-9	15
*T12	3	-10		
*T13	11	-12	17	-18

Fig.7-7 例題

(SIMULATION)

	Firing	Marking
(0)		P 1(1) P13(1) P15(1) P17(1) P11(1) P13(1) P15(1) P17(1)
(1) T 1,	 P 2(1) P11(1) P13(1) P15(1) P17(1)
(2) T 2,	 P 3(1) P11(1) P13(1) P15(1) P17(1)
(3) T 3,	 P 4(1) P12(1) P13(1) P15(1) P17(1)
(4) #T10,	 P 5(1) P 7(1) P12(1) P14(1) P15(1) P17(1)
(5) T 4, T 6,	 P 6(1) P 8(1) P12(1) P14(1) P15(1) P17(1)
(6) T 5, T 7,	 P 5(1) P 9(1) P12(1) P14(1) P16(1) P17(1)
(7) T 4,	 P 6(1) P 9(1) P12(1) P14(1) P16(1) P17(1)
(8) #T11,	 P 2(1) P12(1) P14(1) P15(1) P17(1)
(9) T 2,	 P 3(1) P12(1) P14(1) P15(1) P17(1)
(10) T 8,	 P10(1) P11(1) P14(1) P15(1) P18(1)
(11) *T12,	 P 3(1) P11(1) P14(1) P15(1) P18(1)
(12) T 3,	 P 4(1) P12(1) P14(1) P15(1) P18(1)
(13) T 9,*T13,		P 1(1) P11(1) P13(1) P15(1) P17(1)

Fig.7-9 シミュレーション結果

7.7 MFGへの変換

このアルゴリズムで生成されたペトリネットにおいて、システム構成要素の動作は non-primitive event としてトランジションで表現されている。一方 MFG では、システム構成要素の動作は状況としてボックスで表現され、その前後に動作の開始と終了という primitive event を表わすトランジションが接続される。MFG の規則より、あるボックスの入力側トランジションは1度点弧すると、そのボックスの出力側トランジションが点弧するまで次の点弧はできない。よって、ここで生成されたペトリネットにおいてシステム構成要素の動作を表わしているトランジションを、その動作を表わすボックスとその前後に接続されるトランジションとで置き換えることができる。

また、このアルゴリズムでは点弧系列に対応する各経路に1つのトークンしか置かないため、実行時にブレース内に2つ以上のトークンが入ることはなく、ブレースをボックスに置き換えてもトランジションの点弧条件に変化はない。

以上より、本アルゴリズムで生成されたペトリネットにおいて、

(1) システム構成要素の動作を表わしているトランジションを、その動作を表わすボックスとその前後に接続される動作の開始・終了を示すトランジションで置き換える

(2) ブレースをボックスに置き換える

ことにより、MFG に変換することができる。

7.8 今後の課題

本章では、システムをモデル化する MFG の自動生成の基礎として、与えられた動作順序仕様をトランジションの点弧系列とみなして、それを発生するペトリネット

を生成するアルゴリズムを提案し、これが M F G に変換できることを示した。

このアルゴリズムにより、与えられた問題の解の 1 つを得ることができる。しかし、経路の作成や競合の調停のためのグラフの付加の規則が比較的簡単なものであり、また M F G への変換も機械的であるため、得られた解は最小の系とは限らない。この点の改良が課題の 1 つである。

その他、現在は基本経路上には 1 つ（並列区間では各並列基本経路に 1 つずつ）のトークンしかないがこの数を増やす、すなわちパイプライン処理への対応、あるいは単一の点弧系列のみではなく複数の点弧系列の共存、独立であると限定している並列経路に関する制約の解除などについても、今後検討していかななくてはならない。

第 8 章 結 言

8.1 本研究のまとめ

本研究では、機械工業などの非連続生産システムを対象として、体系的な制御技術、およびその設計や解析に必要なシミュレーション手法の確立をめざしてきた。

対象である非連続生産システムは、「事象」と「状況」の交互の接続で活動している非連続システムの1つとしてとらえることができる。その性質として非同期性、順序性、並列性、競合性および階層性をもち、また特異な現象として追突や続行不能などをおこすことがある。非連続生産システムの構成要素は操作、結合、分割、集合、分離、分岐、合流、容量に分類でき、その状態としては渋滞、停留の区別ができる。このような非連続生産システムを、的確にわかりやすく表現するためには表現手法が重要であるが、その候補の1つにペトリネットがある。

ペトリネットの概念を非連続生産システムの制御や設計・解析に導入する際、非連続生産システム独自の性質の表現能力を付加する必要があり、『安全性』と『信号の入出力機能の表現力』をもたせた新しいグラフとして、マーク流れ線図(MFG)を提案した。

MFGは、状況を表わすボックス、事象を表わすトランジション、状況と事象の接続関係を表わす方向枝、ボックスの中におかれ状況が保持されていることを示すマーク、ボックスあるいは外部の機器からトランジションに接続され点弧の許可/抑止条件を与えるゲート信号枝、およびボックスから外部の機器へマークの有無に応じて状態信号を出力する出力信号枝より構成されている。このMFGについて定義を行ない、

その規則、数式による記述、行列による記述を示し、基本的な諸性質について述べた。

MFGにより非連続生産システムを表現する場合、システムの制御の記述も含めるために、工程や作業を状況として取り扱う。まず、工程や作業を表わすボックスを動作の順序関係に従って接続し、つぎにシステム構成要素である各機器の「使用状況」に対応するボックスを、その関与する範囲に接続する。そして、受け入れ容量などの制御条件の規定を行なうと、システムのコーディネータに相当するグラフとなる。

さらに詳細な機器制御まで表現する場合は、各工程や作業を対象機器の「単位動作」に分解し、必要に応じて信号枝を機器との間に接続して「動作命令」や「動作完了信号」の授受を記述する。このMFGは、システムの制御用プログラムとして用いることができる。

このようにして非連続生産システムを表現しているMFGは、コントローラも含めたシステムのモデルであり、このモデルと実際のシステム構成要素の動作を一致させることが制御につながる。これを実現するために、MFGによりプログラムされ、システム構成要素の機器と「信号の授受」を行ないながらMFGを実行する制御装置として、「MFGにもとづいたプログラマブルコントローラ」をマイクロプロセッサを用いて試作した。

また、このコントローラへのプログラムを容易に行なうために、プログラムの開発やデバッグを支援するプログラムローダを、パーソナルコンピュータにより作成した。これは、MFGをライトペンによりCRT上に描くことでプログラミングを行ない、そのグラフ上で「トランジションの点弧」や「マークの移動」を表示しながらシミュレーションを行なうことで、プログラムの検証を行なうものである。完成したMFGは、通信回線によりMFGコントローラに転送され、プログラムのローディングが完

了する。コントローラが制御を実行している間、この装置はそのマーキングをCRT上のMFGグラフに表示し、モニタとして機能する。

これらの装置を用いて、生産ラインの模型の制御実験を行なった。その結果、仕様どおりの制御が実現できた。この実験は小規模なものであるが、原理的には大規模な実際のシステムと同じものであり、MFGにより非連続生産システムの制御が適切に行なえることを確認した。

非連続生産システムの設計の際に重要な問題となる「動作状況」や「効率」は、システムの実態に即した「定量的な挙動」を知ることによって得られるが、システムの性質により解析的に求めることは困難であり、シミュレーションが必要とされる。制御との一貫性を考慮し、MFGによる定量的なシミュレーションを提案した。

そのために、MFGに時間の概念として「時間ボックス」、「時限トランジション」を導入し、また表現の繁雑さを避けるためのマクロなボックスとして「容量ボックス」、「パレタイジングボックス」、「デパレタイジングボックス」を定義した。これらにとづいて非連続生産システムを定量的にシミュレートし、「マーキングの経過」や「トランジションの点弧回数の総和」、「有効マーク存在時間率」という解析に必要なデータが得られるシミュレータを作成した。これを用いて、実在の非連続生産システムのシミュレーションを行ない、さらにこのデータをもとに立案した増産計画の妥当性についても考察し、その結果を示した。この例より、非連続生産システムのシミュレーション手法としてMFGが有効であることを確認した。

非連続生産システムをMFGにより制御あるいはシミュレートするためには、システムをMFGで記述する必要があるが、複雑な仕様の場合この作業は手間のかかるものであり、自動生成が望まれる。

そこで、非連続生産システムでの作業仕様を点弧系列とみなし、その点弧系列をもつペトリネットを生成するアルゴリズムを提案した。このアルゴリズムは、接続行列を拡張した重複接続行列を用いて「経路の生成」と「競合の調停」を行ない、仕様を満たすペトリネットを生成するものであり、その実行例を示した。さらに、ここで生成されたペトリネットをMFGに変換できることを示し、非連続生産システムをモデル化するMFGの自動生成の基礎とした。

8.2 今後の課題

今後の課題として、以下の事項があげられる。

(1) MFGによる非連続生産システムの制御において、本研究では1レベルのMFGにより制御プログラムを記述し制御を実現したが、システム構造のもつ階層性に着目した階層的な制御に対応できるように、その方式の拡張を検討していかなければならない。

(2) MFGによる非連続生産システムのシミュレーションにおいて、本研究では単一の品種しか流れないシステムを対象としたが、最近の多品種少量生産では複数の品種が混在して流れる「混流」がみられる。このようなシステムでは、品種によりワークピースの「流れ方」や機械における「加工時間」が変化する。これに対応するために、

① マークにタグを付し、タグにより「流れの制御」や「データベースからの読み出し」を行なう

あるいは、

② マークとともにデータを流し、トランジションにおいて判定や演算を行なう
など、MFGの拡張を検討中である。なお、これらの拡張は制御にも応用できるものである。

(3) 非連続生産システムをモデル化するMFGの自動生成において、本研究で提案したアルゴリズムにより与えられた問題の解の1つを得ることができが、最小の系とは限らない。場合によっては、この点の改良が望まれる。また、現在は基本経路上には1つ(並列区間では各並列基本経路に1つずつ)のトークンしかないがこの数を増やす、すなわちパイプライン処理への対応、あるいは単一の点弧系列のみではなく複数の点弧系列の共存、独立であると限定している並列経路に関する制約の解除などの点についても検討しなくてはならない。

【 謝 辞 】

本研究を行なうにあたり、終始、問題の提起や御指導をいただきました長谷川健介教授に、心から感謝いたします。

本研究で用いた手法であるマーク流れ線図(MFG)は、1976年にその原型が長谷川教授により提案され、以来、増田良介先生(現 東海大学助教授)および大野秀嶺氏(三菱重工業)をはじめ、卒業研究や修士論文のテーマとして研究に携わった諸先輩により発展してきました。これらの諸氏に敬意を表します。

また、諸問題についての提案や議論をいただきました Z. Banaszak 氏(Broclaw 工大講師)、和田淳君(現 味の素)ほか、MFG研究会の構成員各位に感謝します。

最後になりましたが、研究室におきまして助言をいただきました中野道雄助教授、回路技術の基礎を教えてくださいました畔津明仁先輩(現 応用計測研究所)ほか、研究生生活を共にした長谷川・中野研究室の皆様にお礼を申し上げます。

【参考文献】

第1章

- 1-1) シーケンス制御工学体系化調査専門委員会：シーケンス制御の工学体系化に関する調査研究，電気学会技術報告（II部）第138号，2/4（1982）

第2章

- 2-1) Analysis of Production Schemata by Petri Nets: M. Hack, Master's thesis, Department of Electrical Engineering, MIT, 21/35 (1972)

第3章

- 3-1) 市川，小林：事象駆動型システムの表現と制御，計測と制御，21-10，929/938（1982）
- 3-2) J.L. Peterson : Petri Net Theory and the Modeling of Systems, 80, 205/207, Prentice-Hall (1981)
- [3-3] 長谷川，高橋，増田，大野：非連続システム制御のためのマーク流れ線図の提案，計測自動制御学会論文集，20-2，122/129（1984）
- [3-4] 増田，高橋，長谷川，大野：MFGの挙動解析アルゴリズム，第20回SICE学術講演会予稿集，99/100（1981）

第4章

- [4-1] 長谷川，*高橋：マークフローグラフに基づくプログラマブルコントローラ，電気学会システム・制御研究会資料，SC-83-7，（1983）

第5章

- [5 - 1] Hasegawa and Takahasi : MFG (Mark Flow Graph)-based Programmable Controller and Its Application to FMS , Third International Conference on System Engineering (1984)
- [5 - 2] 長谷川, 増田, *高橋, 大野: MFGのコンフリクトボックスにおけるマークの流れの制御, 第20回SICE学術講演会予稿集, 101/102 (1981)
- [5 - 3] 長谷川, *高橋: マイクロコンピュータを用いたMFG技法による非連続システムのシミュレータ, 第21回SICE学術講演会予稿集, 109/110 (1982)
- [5 - 4] *高橋, 長谷川: MFGに基づいたプログラマブルコントローラ, 第21回SICE学術講演会予稿集, 111/112 (1982)
- [5 - 5] 長谷川, *高橋, Miyagi, 泉: マーク流れ線図による生産ラインのプログラミングとシステム制御, 第23回SICE学術講演会予稿集, 143/144 (1984)

第6章

- [6 - 1] 長谷川, 高橋, 和田, 丹下: 時間付マーク流れ線図による生産ラインの解析, 第22回SICE学術講演会予稿集, 375/376 (1983)
- [6 - 2] 長谷川, *高橋: マーク流れ線図の機能の拡張について (I), 第26回自動制御連合講演会前刷, 191/192 (1983)

第7章

- 7 - 1) J.L. Peterson : Petri Net Theory and the Modeling of Systems,

35/40, Prentice-Hall (1981)

[7 - 2] 高橋, 長谷川, Banaszak : 指定点弧系列をもつペトリネットの合成方法
計測自動制御学会論文集, 21-3, 掲載予定

[7 - 3] Banaszak, 長谷川, *高橋: 非同期動作仕様より Petri Net への変換アル
ゴリズムに関する基礎的考察, 第9回システムシンポジウム講演論文
集, 255/260 (1983)

第8章

[8 - 1] 長谷川, 高橋, Miyagi, 三品: 第26回自動制御連合講演会前刷, 193/194
(1983)