

2 J-9 シーケンス制御プログラムの自動生成

田中 誠一郎 田中 博明
株式会社 東芝 システム・ソフトウェア技術研究所

1. はじめに

現在、自動制御されている製造ラインでは多くのプログラマブルコントローラ(以下PCと略す)が使われている。PC用のプログラミング言語としてはラダー図が広く使われている。ラダー図はリレーで制御が行われていた時代のリレー回路図に改良を加えたものである。しかし、ラダー図は理解しにくいと開発や保守が困難という問題点がある。そこで我々は、一定の動作を行う製造ラインをターゲットとし、その動作仕様であるタイムチャートからラダー図で記述されたシーケンス制御プログラムを自動生成するツールを試作したのでこれについて報告する^[1]。

2. シーケンス制御プログラム開発の問題点と解決方針

従来、シーケンス制御プログラムを開発するとき、プログラマは、タイムチャートやデシジョンテーブル、フローチャートなどで記述された動作仕様をもとにラダー図を用いてプログラムを組んでいた。しかし、ラダー図を用いてシーケンス制御プログラムを開発するときには次のような問題点がある^[2]。

- ①シーケンスの動きが全て接点とコイルで表現されるので理解しにくい。
- ②仕様とシーケンスの対応がわかりづらい。

このような問題点から開発や保守が困難になってくる。そこでこれらの問題点に対して次の解決方法を考えた。

- ①動作仕様からシーケンス制御プログラムを自動生成する。
- ②保守を容易にするため、生成するシーケンス制御プログラムの構成を理解しやすくする。

以下、この解決方法の詳細について述べる。

3. シーケンス制御プログラムの自動生成

3.1 自動生成における制約

今回のシーケンス制御プログラムの自動生成は、次の制約のもとに行う。

- ①対象システム内の各制御対象の始動や停止の順序はあらかじめ決められているものとする。
- ②制御対象は、ON/OFFの信号で制御できるものまたはパルスモータ。
- ③センサーは、ON/OFFの2状態により制御対象の動作を検出する。
- ④マシン異常を検出した場合の異常処理では、制御対象の動作をその場で停止させる。
- ⑤制御対象の動作方法は次の2つとする。
 - ・センサーや押しボタンなどの入力による制御
 - ・タイマによる遅延制御

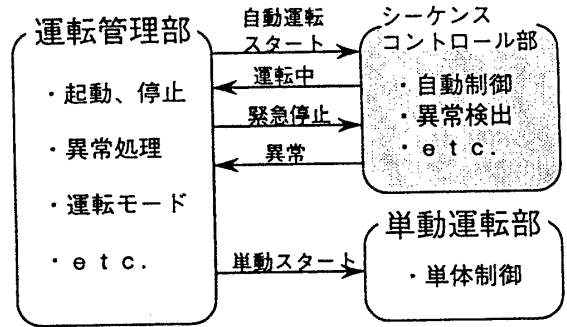


図1. シーケンス制御プログラムの構成

これらの制約は、エレベータのように押されるボタンの種類により動作の順序が変わるようなシステムを対象とするには問題となるが、今回ターゲットとするあらかじめ動作の順序が決まっている製造ラインの正常な処理には十分と考えられる。

また、シーケンス制御プログラムの構成を図1に示すが、自動生成の対象はそのうちのシーケンスコントロール部のみとする。これは他の部分についてはある程度部品として作っておいたものを再利用することが可能であるのに対し、シーケンスコントロール部は制御対象が類似したものであっても他の設備との関係やマシンの配置の違いにより対象ごとに設計されるからである。

3.2 動作仕様

従来、シーケンス制御の動作仕様としてタイムチャートやデシジョンテーブル、フローチャートなどが使われている。自動生成を行うにあたり、我々は動作仕様として次の理由でタイムチャートを採用した。

- ①動作の順序や制御対象間の相互の動作の関係を実時間で記述できる。
- ②視覚的に制御対象の動作を理解しやすい。
- ③表記法を工夫することによりインターロックや異常検出条件を表現することができる。
- ④機械設計、開発の段階で既にタイムチャートが作られていることが多い。

3.3 タイムチャートの記述方法

動作仕様であるタイムチャートからラダー図への変換を計算機により機械的に行えるようにするため、また、インターロックや異常検出条件を記述できるようにするため形式的なタイムチャートの記述方法を提案した。図2にタイムチャートの例を示す。

タイムチャートの各行(以下ボックスと称す)には次の4種類を用意した。

- 1) タイマボックス: タイマによる遅延制御を記述
- 2) フラグボックス: ユーザがフラグとして使用
- 3) 2状態ボックス: ON/OFFで制御する対象の動作を記述
- 4) n状態ボックス: パルスモータの動作を記述

"Automatic Generation of Sequential Control Programs" by S.Tanaka and H.Tanaka (Systems & Software Engineering Laboratory, TOSHIBA Corporation)

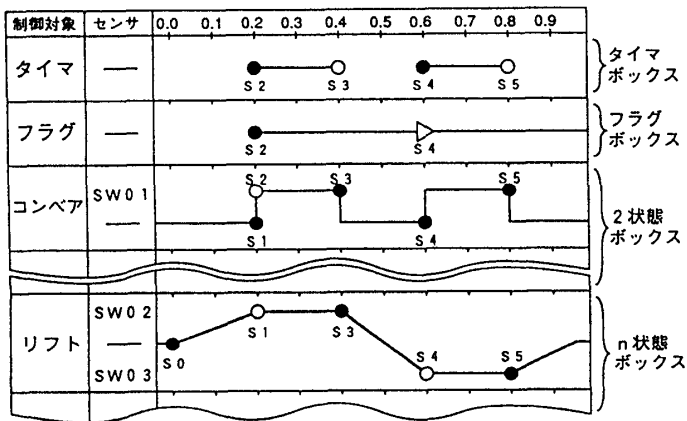


図2. タイムチャートの一例

これらのボックスに記述された制御対象の動作やタイマ、フラグを関係づけるために次の4つの記号を用意した。

- 1) ○ イベント : シーケンスを推移させる条件
- 2) ● アクション : 制御対象の動作
- 3) ▷ コンディション : シーケンスを推移させるための前提条件
- 4) ◁ タイムアウト : 他の動作と同期を取るための時間的制約条件

通常の動作仕様は、イベントとアクションの組み合わせにより記述し、インターロックや異常検出条件をコンディションとタイムアウトで記述する。アクションに対しこれらの条件が複数あった場合、全て論理積として扱う(論理和は今回扱わない)。また、これらの記号にシリアル番号を付加することにより、同一時刻に並列して動作している無関係な事象を区別するようにしている。

3.4 シーケンスコントロール部の構成

自動生成するシーケンスコントロール部は、次の4つから構成する。

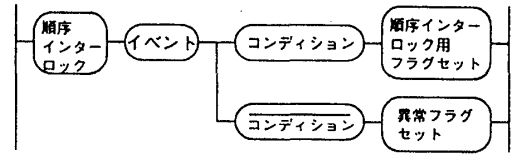
- 1) 初期設定部 : タイマの設定時間やパルスモータの制御データなどの初期値を設定する部分。
- 2) リセット部 : サイクリックに行われる制御のために途中で使用されたフラグをリセットする部分。
- 3) イベント & 異常検出部 : センサーやタイマからの入力によりイベントや異常を検出する部分。
- 4) アクション部 : 検出されたイベントに対応した動作を実行する命令を出力する部分。

このように構成を明確に分けることにより理解しやすいプログラムとなる。また、このような構成をとることにより異常が検出された場合、アクション部全体を読みとばすことにより制御対象の動作をその場で停止させることが容易にできる。これは人命や設備の損害を最小におさえることを目的とし、その後の異常処理は運転管理部で行うようにする。

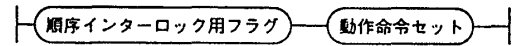
3.5 タイムチャート→ラダー図変換方法

3.5.1 順序インターロック

タイムチャートからラダー図へ変換する際、関連する全ての事象に対し順序インターロックをかける。つまり、タイムチャート上の異なる時刻にある2つの事象において、後の事象の生起する前提条件として前の事象が生起



(a) イベント & 異常検出部



(b) アクション部

図3. 変換用テンプレート

しているというものを自動的に与える。これは次の理由による。

- ①異なる時刻に同じ条件で異なる動作を行う場合、その条件を明確に区別することを機械的に行う。
- ②各事象に対し1つずつフラグを用意し、タイムチャートとラダー図の対応をわかりやすくし保守を容易にする。

これを実現するため、イベント & 異常検出部では1つ前の事象に対応したフラグが立っていることを暗黙の前提条件とし、アクション部ではその事象のフラグの立ち上がりにより動作命令を出力するようにする。

ただし、まったく無関係に並行して動作しているものについてはこの限りでない。

3.5.2 変換用テンプレート

タイムチャートからラダー図へ変換するとき、ラダー図の変換用テンプレートを用いる。図3にその一部を示す(簡単にするため、タイマに関する部分を省略している)。各事象ごとにこのテンプレートの各要素をタイムチャートの情報をもとに自動的に埋めてラダー図を生成する。各要素には、a接点やb接点、変化検出接点、セットコイル、リセットコイルなどの組み合わせが入る。これらの組み合わせは、記号の種類と記号のあるボックスの種類およびボックス内での位置により決定する。

4. おわりに

製造ラインなどで使われているPC用のシーケンス制御プログラムをその動作仕様であるタイムチャートから自動生成する方法について述べた。これについては、タイムチャートエディタおよびタイムチャート→ラダー図変換ツールを開発したので、実際に試行評価していく予定である。今後の課題としては、そのときどきの条件により動作の内容や順序を変更するようなシステムに対応できるように動作仕様に条件分岐やループ制御といった制御構造を記述できるようにすることである。

参考文献

- [1] 田中ほか：制御系ソフト自動生成システム，情報処理学会第83回ソフトウェア工学研究会，1992
- [2] 電気学会：シーケンス制御工学，オーム社，1988