

ビジュアルな制御用プログラミング手法

慎 ヨン 日, 大西 淳

立命館大学大学院 理工学研究科 総合理工学専攻

525-77 滋賀県草津市野路東 1-1-1

email: {shin,ohnishi}@selab.cs.ritsumeai.ac.jp

本研究は制御用プログラミングに慣れていないユーザによる制御用プログラミングの支援を目標とする。従来の制御用プログラミングの問題点を解決するための支援する方法として高品質な制御用プログラムを効率良く開発する手法を提案する。本手法では、ユーザとシステムとのメッセージのやり取りを対話的に行ない、制御対象システムの定義とその動作の定義による制御用プログラムの生成を手法のキーとする。これらの過程を具体的な例を用いて説明する。

A Visual Developing Method for Sequence Controller Programs

Yongil SHIN, Atsushi OHNISHI

Department of Computer Science, Ritsumeikan University

1-1-1 Noji-Higashi, Kusatsu, Shiga 525-77

The objective of this research is visual support for developing sequence controller programs. There exist several problems in developing programs of sequence controller. We propose a supporting method to develop sequence controller programs to solve these problems. The key is developing programs of sequence controller by an object definition of the sequence controller system and a definition of program behaviors. The behaviors of the supporting method are illustrated with a concrete example.

1 はじめに

今日、制御分野におけるコンピュータの存在は欠かせない。パソコン利用によるオープン化指向、ユーザによる組合せ自在な開発環境など特にパソコンを用いてユーザ主体によるユーザ自身にあったソフトウェア開発環境が研究されつつある。

制御用プログラムの開発では、専門家でないプログラミングが難しい、ミスが生じ易い、プログラムの生産性が低いなど幾つかの問題点を抱えている。これらの問題点を改善し、効率良くプログラムを開発するためには既存のラダー図とか独自の言語ではなく、ビジュアルな高級言語を使った、もっとプログラミングし易い方式が必要となってきた。本研究はこれらの背景に基づいた研究である。

本研究の全体の流れを以下に述べる。

1. 制御用プログラミング支援
2. プログラム動作の検証
3. シーケンサの制御支援

1の段階では、ユーザとシステムとのメッセージのやり取りを対話的に行なうことによって制御対象システムを定義し、制御用プログラムを生成する。2の段階では、プログラムの正しさを検証するために制御用プログラムをシミュレーションしその結果をアニメーションとして出力させる。そして3の段階では、直接シーケンサに制御用プログラムをロードし、制御対象システムをシミュレーションする。このようにパソコン上でユーザと対話的に、そしてアイコンを使いながらアニメーション化していくことによってユーザに使い易いプログラム開発支援の提供を目指している。1の段階を今回の論文で扱っている。

本研究では、制御用プログラムの開発における従来の問題点を改善し、高品質な制御用プログラムを効率良く開発することを目的とする。問題点を改善するために、なるべく従来の文字列や記号を中心としたプログラム言語に代わってアイコン、図形、アニメーションなど人間がより理解しやすいビジュアルな表現を用いてユーザとシステムとのメッセージのやり取りを対話的に行う方針である。さらにこれらのビジュアルな情報を用いて新しい機能をプログラムとして実現するプログラミング支援手法の確立を目標とする。

本研究によって制御用プログラムの開発が効率化され、開発までの時間と費用が軽減できることを期待する。特

に、制御用の特殊な記法に慣れていないプログラマにとっては開発効率の大幅な向上が期待できる。

本稿では次のように構成されている。2章では、従来の制御用プログラミング手順とその問題点及び改善点について述べる。3章では、制御用プログラミング支援手法について述べる。4章では、具体例を挙げながら説明する。そして5章では、考察をのべ、6章では、まとめと今後の課題について述べる。

2 従来の制御用プログラミング手順と問題点

現在の制御はPC (Programmable Controller) を使った制御が主流である。従来のPCのプログラミング手順は、図1のように6段階の手順の通りにプログラミングを行う。制御方法の決定段階では、装置や設備をどのように制御するかのと入力・出力機器の決定、PCの機種、規模などを決定する。入出力番号の割付け段階では、PCと装置との接続関係を決定する。シーケンス制御回路の設計段階では、使用するPCの機能、特徴を十分に生かして設計する。コーディング段階では、シーケンス回路をPCの命令語に変換する。ローディング段階では、プログラムロードからPCへプログラムを書き込む。そしてデバッグ段階では、PC実機によりデバッグを行う。2.1節では、従来の方式の問題点について述べる。2.2節では、その改善点について述べる。2.3節では、支援手法全体の関連について述べる。

2.1 従来の方式の問題点

従来の制御用プログラミング方式は、以下のような幾つかの問題点を抱えている。

1. 専門家でないプログラミングが難しい

PCのプログラミング言語としては、ラダー図方式、ニーモニック方式、フローチャート方式、ロジックシンボル方式、ブール代数方式、タイムチャート方式のようなものがある。このうち、最も普及しているのがラダー図方式とニーモニック方式である。このような、プログラム言語としてのPC用のシーケンス制御回路(ラダー図)や独自の言語(ニーモニック)を使うための専門知識を持っていなければプログラミングが困難である。

2. ミスが生じやすい

プログラマが直接周辺機器を通して入力するため過ちを犯しやすい。

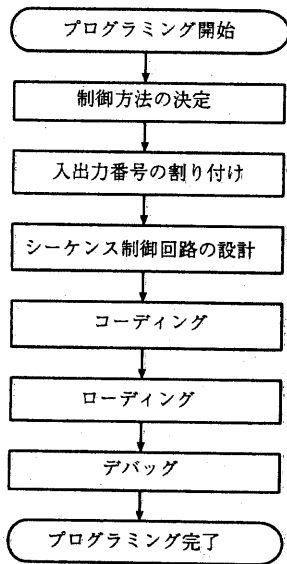


図 1: PC のプログラミング手順

3. プログラムの生産性が低い
プログラミングの自動化がされていないため生産性が低い。

2.2 その改善点

問題点を解決するためにユーザにとって容易なプログラミングを行なうための工夫に重点を置き、下のような幾つかの改善点を考えた。

1、2、3 に対する改善点

- ユーザが理解しやすい部品を用意しておく
- ユーザ自身がアイコン化された部品を組み合わせることで制御対象システムを定義する
- 制御対象システムの動作の定義を行うことによって制御用プログラムを生成する
- アイコン、図形、アニメーションなどビジュアルな表現を用いる

3 に対する改善点

- 部品の組み合わせによって作られた制御対象システムと制御用プログラムを複合部品として扱う

2.3 支援手法全体の関連

本手法の全体の流れを図 2 に示すように 3 段階に分けて説明する。

1. 制御用プログラミング支援
2. プログラム動作の検証
3. シーケンスの制御支援

今回の論文で扱っている 1 の段階は、制御対象システムとそれに対する制御用プログラムを生成するための段階である。ユーザ自身が利用する部品を選択し、そして選択した部品を組合わせていくことによって制御対象システムの定義を行なう。又、制御機器の条件による制御対象となる機械や装置の動作の定義を行なうことにより、制御用プログラムを生成する。2 の段階は、1 の段階での生産物である制御対象システムの定義と制御用プログラムを入力とし、制御用プログラムの正しさを検証するためにシミュレーションを行なう段階である。そしてその結果をアニメーションとして直接ユーザに見せながら検証を行なう。最後に 3 の段階は、1 と 2 の段階の結果を基に、パソコンから直接シーケンスを制御する段階である。このように既存のプログラマブルコントローラを使った制御から開発段階の全てをパソコンを用いた制御に変わっている。これは制御分野での開発の流れとして非常に関心度が高まっていく動向であり、パソコン利用によるオープン化指向、ユーザによる組み合わせ自在な開発環境など特にパソコンを用いたユーザ自身に合ったソフトウェア開発環境が研究されている。

3 制御用プログラミング支援手法

本支援手法は、制御用の特殊な記法に慣れていない人でもプログラミングができるようにビジュアルな表現を用いたユーザとのインタフェース設計を支援する。ユーザとの対話によるプログラミングが行えるようにし、そして目で見て直感的に分かるビジュアル・プログラミング支援を目指して設計した手法である。3.1 節では、制御部品の定義とその分類について述べる。3.2 節では、制御用プログラムの作成支援手法について述べる。3.3 節では、機能を細分化し、各々の機能について述べる。

3.1 制御部品の定義と分類

産業用の自動機械や装置には、いろいろな制御機器が数多く使用されている。本支援手法では、これらの制御

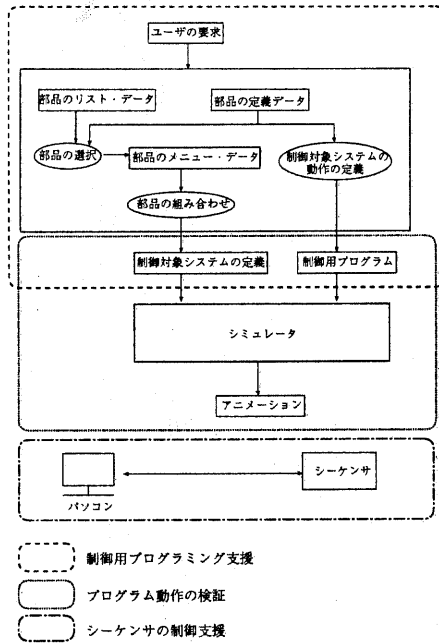


図 2: 支援手法全体の概念

機器や制御対象となる機械と装置を部品とする。そして制御関係によって以下のように二つの種類に分類する。

1. 制御対象となる機械や装置

モータ、シリンダ、電磁弁、ソレノイド、電磁クラッチ

2. 制御機器

- 入力用機器：押ボタンスイッチ、セレクトスイッチ、カムスイッチ、トグルスイッチ、足踏スイッチ
- 制御・演算用機器：制御用リレー、モータタイマ、電子タイマ、電磁タイマ、プログラムタイマ
- 出力用機器：電磁接触器、電磁開閉器、ソリッドステートリレー
- 検出用機器：リミットスイッチ、近接スイッチ、光電スイッチ、圧力スイッチ
- 表示・警報用機器：表示灯、ブザー、ベル、サイレン

一つ一つの部品に対してはその機能、状態、変数名、動作の時系列又は各状態に対する色の変化（機能と状態の時間的な変化や各状態の色の変化を図示することによってユーザに分かりやすく見せる）のような情報をユーザに提供する。

このように各々の部品に対してその定義を正確にすることによってユーザに必要な情報を提供するだけでなく部品を組み合わせた制御対象システムの動作を定義する段階でも各々の部品の条件とそれに対する動作の関係を明確にできる。

3.2 制御用プログラムの作成支援手法

ユーザが部品のリスト一覧から使用可能な部品を選択していく。その過程でユーザは部品の正確な選択のため、部品の定義から利用する部品に関する色々な情報を得ることができる。選択した部品は使用する部品のメニューとして画面にロードされる。次にユーザはロードされた部品を組合わせていく。これによって制御対象システムの定義が表示できる。そして選択した部品の初期状態を設定し、条件に対応する動作の定義を行うことによって制御用プログラムを生成する（図 3）。

3.3 機能の細分化

本制御用プログラミング支援手法は、

1. 部品の選択
2. 選択された部品の組み合わせによる制御対象システムの定義
3. 制御対象システムの動作の定義による制御用プログラムの生成

といった制御用プログラムを生成するための機能を中心に細分化される（図 4）。

3.3.1 部品の選択

制御対象システムを組み合わせるために部品の定義を参照しながら利用する部品を選択する機能である。選択された部品はアイコン化され部品のメニューとして登録される。この機能はさらに以下のように細分化される。

1. 部品のリスト・アップ及び選択
制御部品の一覧を表示することによって、ユーザはその一覧から利用する部品を選択できる。

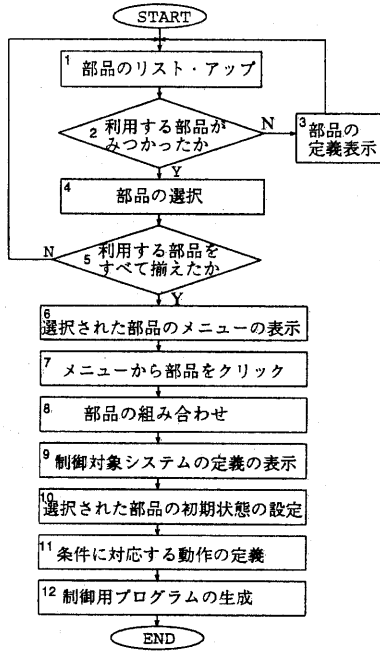


図 3: 制御用プログラムの作成支援手法

2. 部品の定義の表示

各々の制御部品に対してその部品の機能、状態、変数名、動作の時系列、各状態に対する色の変化などの内容をユーザに提供する。この提供により、ユーザは制御対象システムを組み合わせるための制御部品を正しく選択することができる。

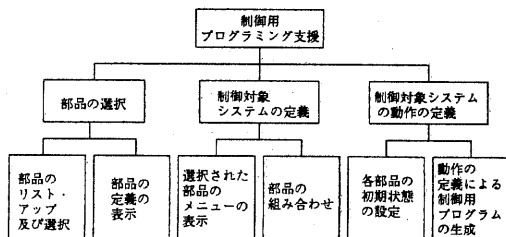


図 4: 機能の細分化

3.3.2 部品の組合せによる制御対象システムの定義

選択し、登録された部品を部品のメニューから部品一つ一つをクリックしたり、ドラッグしたりして制御対象システムを組み合わせていく機能である。そして組み合わせさせた制御対象システムは再利用のために複合部品として扱うべきである。この機能はさらに以下のように細分化される。

1. 選択された部品のメニューの表示

このメニューに制御対象システムを定義するために選択された部品の全てが表示される。

2. 部品の組み合わせ

ユーザはメニューから部品一つ一つをクリックしたりドラッグしながらユーザが設計しているイメージ通りの制御対象システムを組み合わせていく。

3.3.3 動作の定義による制御用プログラムの生成

センサやタイマのような制御機器となる部品とシリンダやモータのような制御対象となる部品の条件とそれに対する動作の定義を行なうことにより、目的とする制御用プログラムを生成する機能である。この機能はさらに以下のように細分化される。

1. 各部品の初期状態の設定

常に機械や装置を動かす前の初期状態と機能を設定しておく。

2. 動作の定義による制御用プログラムの生成

制御機器となる制御部品の条件に対応する制御対象となる制御部品の動作の定義を行なうことにより、制御用プログラムを生成する。

このようにして分解された各々の機能に対するデータの流れを図5に示す。これを入力、処理、出力のように3つの段階に分けて説明すると、入力として部品リストファイルと部品の定義ファイル、そしてユーザのいろんな要求が入力され、6つの機能を処理していくことによって、目的とする制御対象システムの定義と制御用プログラムが得られる。

4 具体例

以上述べてきた手法をシリンダ制御の例題に適用してみる。この例題では1本のシリンダと2本のコンベヤがあり、シリンダの上下にコンベヤが配置される。そして、上下のコンベヤにセンサを配置し、上下のコン

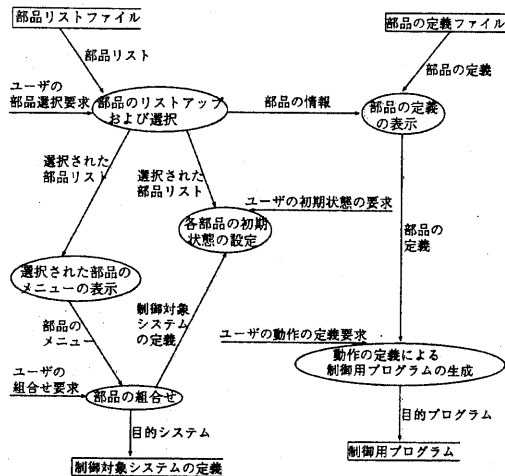


図 5: 制御用プログラムの作成支援 DFD

ベヤーから流れてくる物をセンサが感知すると、シリンダが動き出す仕組みである。リミットスイッチはシリンダの位置を正確に検出するために使われる。この例では、シリンダの一番上と中央と一番下の位置を正確に検出するために3つのリミットスイッチが使われる。以下に利用する部品を述べる。

1. 制御対象となる機械や装置；シリンダ
2. 制御機器；センサ、リミットスイッチ

これらの部品を組合せたものを制御するためのプログラミングを行う。以下にその仮定を述べる。

1. 上と下に物を載せた2本のコンベヤがある
 2. その物がセンサに感知されるとシリンダが動き出す
 3. シリンダは常に中央からスタートし、作業を終えると再び中央に戻ってくる
- 4.1 節では、利用する部品の定義について述べる。4.2 節では、プログラミング過程について述べる。

4.1 利用する部品の定義

センサは、各種の状態や現象を電気信号に変えてとらえるものであり、その数も多い。ここでは、光電スイッチを使う。光電スイッチは、光の透過・遮光、反射を活用した検出スイッチで、非接触式検出スイッチの代表的

なものである。図6に示すように、何らかに感知され、ONになると青い色に、OFFになると赤い色に変わるとする。

リミットスイッチは、物の位置を検出するための接触式の検出スイッチである。図7に示すように何らかに接触され、ONになると青い色に、OFFになると赤い色に変わるとする。

シリンダは、制御対象となる部品であり、センサやリミットスイッチのような制御機器の条件によってその動作が決まる。例えば、図8のように、上のセンサがONになると、中央から上の方に動き出す。そして作業を終えると再び中央に戻ってくる。下のセンサがONになると、中央から下の方に動き出す。そして作業を終えると再び中央に戻ってくる。シリンダの中央と上下の作業をするために停止する2箇所には、リミットスイッチが配置される。

1. センサの定義

- 機能：ON/OFF 機能
- 状態：ON/OFF 状態
- 変数名：sen と表記する
- 各状態に対する色の変化（図6）

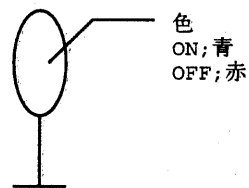


図 6: センサの状態に対する色の変化

2. リミットスイッチの定義

- 機能：ON/OFF 機能
- 状態：ON/OFF 状態
- 変数名：LS と表記する
- 各状態に対する色の変化（図7）

3. シリンダの定義

- 機能：UP/DOWN 機能
UP(up; stop; 作業; down; stop)
DOWN(down; stop; 作業; up; stop)

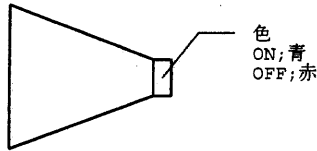


図 7: リミットスイッチの状態に対する色の変化

- 状態: move(up/down)
stop(一番上、中央、一番下)
- 変数名: cyl と表記する。
- 動作の時系列 (図 8)

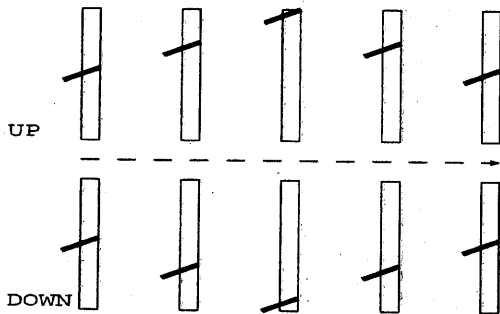


図 8: シリンダの動作の時系列

4.2 プログラミング過程

このプログラミング過程は、ユーザが図 3 の制御用プログラムの作成支援手法に従ってプログラミングを行なうため、図 3 を参照しながら説明する。これから使う番号は図 3 の各手順の番号である。

図 9 の部品の選択画面では、ユーザが部品の定義を参照しながら利用する部品を選択していく過程である (1~5)。例えば、左側の部品のリストからシリンダという文字をクリックすると右側の部品の定義にシリンダの定義がロードされる。又、四角いボタンをクリックするとその部品が選択される。この画面で選択された部品はアイコン化され、部品のメニューとしてロードされる (6)。次にユーザは部品一つ一つを移動しながら制御対象システムを組み合わせていく (7~9)。これらの過程により、図 10 のような制御対象システムの定義が表示できる (9)。そして、選択された部品の初期状態を設

定し (10)、図 11 のように組み合わせられた制御対象システムの動作の定義を行なう (11)。この例では、センサの条件によるシリンダの動作の定義を行なう画面である。

これらの過程により、一つのパターンとしてユーザが図 11 の動作の定義画面のように定義を行なうと、その結果として図 12 のような制御用プログラムが出力される (12)。

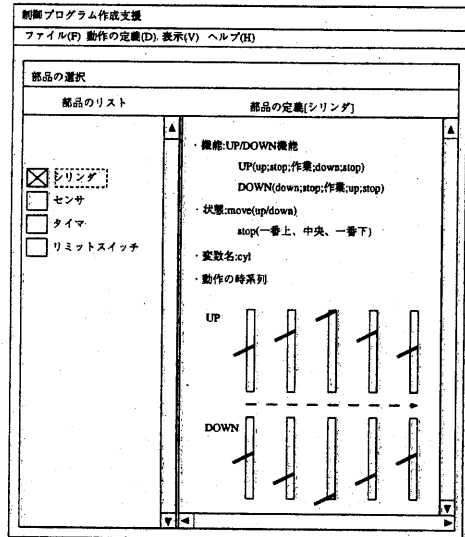


図 9: 部品の選択画面

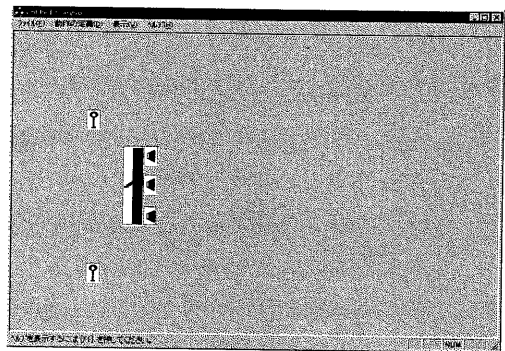


図 10: 制御対象システムの定義画面

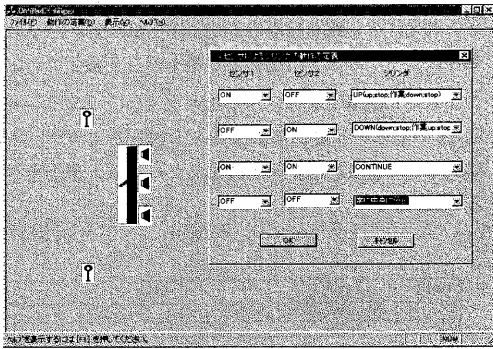


図 11: 動作の定義画面

```

LD      sen1
CALL   &UP
LD      sen2
CALL   &DOWN
END

L0 SUB  UP
   OUT UP_SIG
   LD  LS2
   CJ  L0
   OUT WORK
   TIME 10
L1 OUT DOWN_SIG
   LD  LS1
   CJ  L1

L2 SUB  DOWN
   OUT DOWN_SIG
   LD  LS3
   CJ  L2
   OUT WORK
   TIME 10
L3 OUT UP_SIG
   LD  LS1
   CJ  L3
  
```

図 12: 出力結果ファイル

5 考察

この手法は、制御分野において、従来のプログラミング方式の問題点を改善し、ユーザにとってプログラミングを容易にする手法を目指して開発された。この手法の開発にあたって留意された点は以下の2点である。

1. 部品の組み合わせによって制御対象システムを定義する
2. その制御対象システムの動作の定義を行なうことによって制御用プログラムを生成する

以上に示した2点の項目と支援手法全般に対するの評価を与える。

1. 部品の表示と組み合わせに対する評価
この支援手法では、部品を表示するためのアイテムをアイコン化し、ユーザによる自由自在な移動を可

能にした。そのため、部品の組み合わせによって制御対象システムを定義することができた。又、部品の定義を用意し、ユーザに部品の情報を提供することができた。しかし、最初から部品の大きさが決まっているため、部品ごとの大きさの調節が難しくなった。ユーザが部品一つ一つの大きさを調節することができれば、制御対象システムの見栄えが良くなると思う。

2. 動作の定義に対する評価

部品の分類と部品の定義を明確にしておくことによって制御機器となる部品の条件の全てを明確にすることができた。そして、その条件に対応する制御対象となる部品の動作を定義することができた。今回の例では、制御対象システムの基本動作が多少長い仮定をしたが部品の数を増やすことによってその基本動作を短くできると思う。

6 おわりに

本研究では、従来の制御用プログラミング方式の問題点を改善し、ユーザに使い易いプログラム開発を提供するためのプログラミング手法について提案した。

従来は、PCのようなプログラミング専用機によるプログラム開発であった。しかし、これからはプログラミング専用機ではなく、パソコンを用いたソフトウェア開発に変わっていく動向である。それに基づいた本支援手法によって制御分野での問題点のある程度解決できると考えている。今後は支援環境の構築に加えて

1. 多くの具体例への適用と評価
 2. 制御分野における部品すべての正確な定義
 3. 複合部品としての再利用
- を課題として考えている。

参考文献

- [1] 小野 孝治、塩田 泰仁：「シーケンス制御技術」、産業図書、1994。
- [2] 慎 ヨン 日、大西 淳：「制御分野におけるビジュアル・プログラミング支援」、情報処理学会第54回全国大会講演論文集 6P3、1997.3
- [3] 慎 ヨン 日：「制御分野におけるビジュアル・プログラミング手法」、立命館大学大学院情報システム学専攻修士論文、1997.2