

# シミュレーションを用いた性質の検証支援に関する考察

2011SE017 藤部貴斗

指導教員：張漢明

## 1 はじめに

文献 [2] によると近年のソフトウェア開発では満足のいく水準まで開発を行うことができる能力に比べてソフトウェアに向けられる要求が過大且つ複雑となっており、特に組込みシステム開発では、テスト後に起こりうる修正とそのコストについての対策を行うために、あらかじめソフトウェアとハードウェアを併せて時間軸に沿った振る舞いを予想しておくことが実装の成功への鍵になるとされている。

本研究の目的は組込みシステムの性質をソフトウェア・ハードウェア協調シミュレーションを用いて検証支援する方法について提案することである。目的を達成するための基本的なアイデアは、組込みシステムの性質を自動的にテストするソフトウェアの構造を提案し、その有用性を確認することであり、本稿では Line Following Robot で上記のような提案を実現し検証を行ったことを示す。Line Following Robot の検証例から、その枠組みについて性質を調査する際の有用性を確認した。

## 2 背景技術

### 2.1 シミュレーションツール

近年の組込み開発において用いられるシミュレーションツールというと、一般的には自動車のモデリングなどで MATLAB/Simulink が広く用いられている。

#### 2.1.1 Crescendo Tool

本研究では組込みシステムの設計支援及び支援を目的としたシミュレーションツールである Crescendo Tool を利用する。

CrescendoTool の概要を図 1 に示す。CrescendoTool は組込み制御ソフトウェア設計支援のためのソフトウェア・ハードウェア協調シミュレーションツールである。図 1 に示した通り、Overture は VDM 記述により離散モデルに対応し、20-sim は連続モデルに対応している。離散モデルと連続モデルには Contract という共有変数が存在し、共有設計パラメータを変更することでモデルのデザインが変更できる。

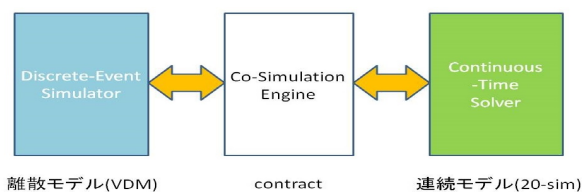


図 1 Crescendo Tool の概要

### 2.1.2 Overture

VDM とその派生言語を使った正確な抽象モデルの利用を支援する、ことが可能なツールであり、Eclipse の操作性を VDM 記述時にも得られる。

### 2.1.3 20-sim

組込みシステムにおけるハードウェアの設計に対応しており、3D アニメーションなどの基本ライブラリを所持している。方程式、ブロック線図、センサ部品などを用いることでより直感的にモデルを作成することができる。

## 2.2 並行システムの性質

文献 [4] によると並行システムで検証可能であり時相論理での記述が可能な性質として、以下の四つがあげられる。

**到達可能性 (Reachability)** システムが初期状態からある状態へ到達可能であることを示す。

**安全性 (Safety)** システムがある条件下で、正当でない状況に陥ることが決して起こらないことを表す。

**活性 (Liveness)** システムがある条件下、でいつか必ず、ある特定の状況が起こることを表す。

**公平性 (Fairness)** システムがある条件下で、ある特定の状況が無限回起こることを表す。

## 3 Line Following Robot の性質分析

### 3.1 Line Following Robot の概要

Line Following Robot はラインを追従する自動走行ロボットである。まず始めに Line Following Robot の本体を図 2 に示す。主要な部品として、IR センサーが 2 つ、車

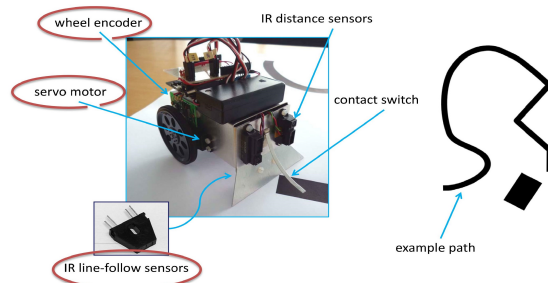


図 2 Line Following Robot

輪が 2 つ、サーボモータや切り替えスイッチがある。コースは図 2 の形状であり、線を探るところからシミュレーションが始まる。

## 3.2 Line Following Robot の性質

Line Following Robot のライン追従の性質について以下のように検討した。

### 1. 強いライン追従

決してラインから外れてはいけないという性質であり、より強い制約が必要とされる。この性質は Line Following Robot の本体がライン上にいることで成り立つものであり、「ロボット本体が必ずライン上にいる」という安全性であると考えることができる。

### 2. 弱いライン追従

ラインから外れることを良しとするが、ロボット本体がラインから外れた際には必ずラインに復帰するという性質である。この性質は「ロボット本体がラインから外れた際に、いつか必ずラインに復帰する」という活性であると考えることができる。

- 復帰の仕方について以下のように検討した。
  - －ラインに復帰する際に「一定時間内にラインに復帰をする」
  - －ラインに復帰する際に「いくら時間がかかってもよいのでいつかラインに復帰する」
  - －復帰するまでにラインを探して「ロボット本体が一回転をしない」
  - －復帰した際に「ロボット本体が逆走状態にならない」

## 4 考察

### 4.1 性質の分類

性質の記述を以下のように分類するのはどうかと考えた。

1. 状態で定義
2. 時間経過の前後の状態で定義
3. ある時間内で満たす性質を定義

#### 【1】状態で定義

そのときの状態が分かるような仕組みを用いることで、その性質が満たされていることを調査する。引数はその時の状態。

#### 【2】単位時間の経過で定義

時間  $t$  と時間  $t+1$  の状態を比べることで得られる情報から性質を調査する。引数は前の状態と後の状態であり、前の状態を記憶しておく必要がある。

#### 【3】時間経過を計測して定義

一定時間ある条件が満たされているか否かを基準に性質を調査する。一定時間中に単位時間ごとに対象の状態を検知し、調査することができる。引数は計測する時間と現在の経過時間である。

## 4.2 検証のための構造

3.2 で検討した Line Following Robot の性質に対し、4.1 で考えた分類【1,2,3】を適用することで検証が可能になることを示す。

- ロボット本体が必ずライン上にいる (安全性)  
ライン上にいることはその時の状態から分かるので、【1】に分類することができる。
- 一定時間内にラインに復帰をする  
「ラインから外れて、復帰するまでにかかった時間」から分かるので【3】に分類できる。
- いくら時間がかかってもよいのでラインに復帰する  
「ラインから外れたこと」と「ラインに復帰したこと」が分かればよいので、【1】に分類できる。
- 復帰するまでに「ロボット本体が一回転をしない」  
ロボット本体が一回転していることは本体の前後の状態を比べることで分かるので、【2】に分類できる。
- 復帰した際に「ロボット本体が逆走状態にならない」  
ロボットが逆走していることは本体の前後の状態を比べ、ベクトルを利用することで分かるので、【2】に分類できる。

## 5 おわりに

性質の記述を分類し、それぞれに性質が調査できるようなソフトウェア構造を準備することで、要求やハードウェアの制約により変化が生まれる可能性のある組込みシステムの性質の検証を行う際に対応がしやすくなると思った。

### 5.1 今後の課題

離散モデル側でのソフトウェア構造の実現にとどまらず、20-sim と連携し 3D アニメーションによる動作の確認を行うこと。そしてシミュレーションで完結するのではなく、実物のハードウェアを用いて Line Following Robot のシステムを実現することが課題として挙げられると考えた。

## 参考文献

- [1] J. Fitzgerald, F. Ishikawa, P. G. Larsen, “Collaborative Modelling and Co-simulation,” <http://research.nii.ac.jp/~f-ishikawa/work/crescendo14/NII.1.Intro-ja.pdf>, 2014.
- [2] J. Fitzgerald, P. G. Larsen, P. Mukherjee, N. Plat, and M. Verhoef, *Validated Designs for Object-oriented Systems*, Springer, 2005.
- [3] S. Wolff, Martin P. Christensen, and P. G. Larsen, *Crescendo Examples Compendium*, Aarhus University, Department of Engineering, 2014.
- [4] 青木利晃, 田原康之, 吉岡信和, SPIN による設計モデル検証, 加藤文明社, 2008.