

K\_020

## フレームワークとオブジェクトモデルを用いた 制御系シミュレーション・実験統合環境

### Integrated Environment for Control Simulation and Experiment using Framework and Object Model

矢野 健太郎<sup>†</sup>  
Kentaro Yano

古賀 雅伸<sup>†</sup>  
Masanobu Koga

#### 1. はじめに

制御系の設計プロセスは、モデリング、制御器設計、シミュレーション、制御実験という流れで行われる。制御実験を行う際の制御対象がロボットや倒立振り子である場合、リアルタイム制御が必要となる。そのため、リアルタイム制御を行うにはリアルタイムプログラム (RTプログラム) が必要である。

RTプログラムである制御プログラム (RT制御プログラム) は作成に専門知識が必要である。このため、学生実験などでは制御実験は行われず、シミュレーションまでで終わってしまうケースが多くある。だが、制御実験を行うことで学生の理解を深めることができると報告されている [1]。これより、制御実験を容易に行えるようにすることが求められている。

また、RT制御プログラムの作成・実行と、シミュレーションプログラムの作成・実行は別々のプログラムで行われる。このためシミュレーションプログラムとは個別にRT制御プログラムを作成する必要がある。設計プロセスではシミュレーションと制御実験が繰り返されることが多く、このように作成・実行環境が異なるとこれらの処理を効率的に実行できない。

そこで本研究では、フレームワークとオブジェクトモデルを用いた制御系シミュレーションとリアルタイム制御実験の統合環境を開発する。RT制御プログラムの作成・実行は、RT制御プログラム作成フレームワークによって行われる。そして、オブジェクトモデルを用いたシミュレーションプログラムの変換、Factory Methodパターンを用いたRT制御プログラムの自動生成により、シミュレーションとリアルタイム制御を統合して実行できる。

#### 2. RT制御プログラム作成フレームワーク

本研究では、RT制御プログラムの作成を容易にするためにRT制御プログラム作成フレームワーク ReTiCoF (Real Time Control Framework) [2]を開発した。ReTiCoFはRT-Linux[3]上で動作し、C言語とJava言語を用いて記述されている。分散オブジェクトを用いた遠隔制御も可能であり、プラットフォームに依存せずに制御実験を実行可能である。

ReTiCoFのアーキテクチャーを、図1に示す。図1に示すように、ユーザからのデータの受付・解釈、データの表示、リアルタイムタスクの部分のみを作成すれば、RT制御プログラムを作成することができる。

また、ReTiCoFではユーザインターフェイス部分のホットスポット作成を簡略化し、RTタスクの作成と制

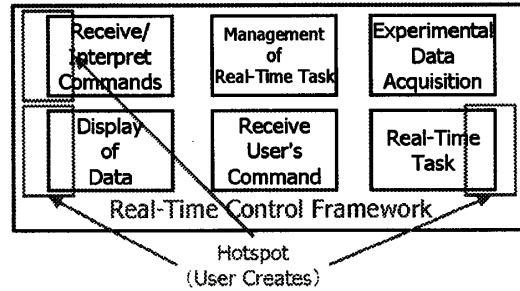


図1: Architecture of the framework

御実験に専念できるようにリアルタイム制御用 GUI を提供している。ReTiCoFによって提供される GUIであるRTWindowの起動画面を図2に示す。

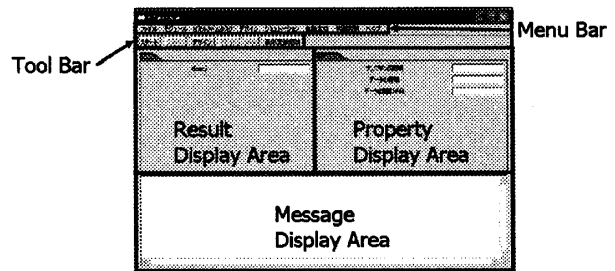


図2: The start-up screen of RTWindow

#### 3. オブジェクトモデルを用いたシミュレーションプログラムの変換

本研究で開発した統合環境では、オブジェクトモデルを用いてシミュレーションプログラムをJavaプログラムに変換することで、シミュレーションとリアルタイム制御の実行環境を統合している。この機能を実現するために、matj[4]を用いてMATX[5]で記述されているプログラムをJavaプログラムに変換することにした。図3にMATXオブジェクトモデルを用いたシミュレーションプログラムの変換の概要を示す。

MATXソースコードの変換を行う際、まずParserがMATXソースコードを受け取る。Parserは構文解析を行い、文法規則に基づいてオブジェクトツリーの構築を行う。上記の処理により、MATXソースコードの情報を持ったオブジェクトツリーを得ることができる。そして、生成

<sup>†</sup>九州工業大学

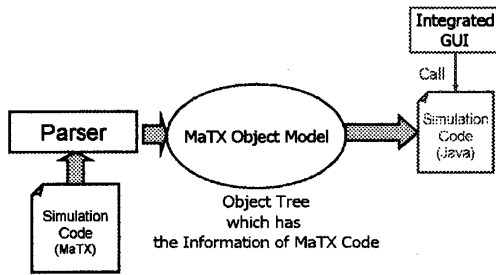


図 3: Transform of simulation program using MaTX Object Model

されたオブジェクトツリーから Java ソースコードの生成を行える。統合環境からその Java プログラムを呼び出すことで、シミュレーションと制御実験の実行を統合できる。

4. RT 制御プログラムの自動生成

本研究で開発した統合環境では, matj に Factory Method パターン [6] を適用し, シミュレーションプログラムから RT 制御プログラムを自動生成する。

Factory Method パターンを用いたリアルタイム処理コードの自動生成の概要を, 図 4 に示す。図 4 の Parser がオブジェクトモデルを作成する部分に Factory Method パターンを適用する。まず, Parser がオブジェクトモデルを作成する際の作成方を Creator で規定する。そして, シミュレーションコードを変換するオブジェクトの作成は SimCreator によって行う。リアルタイム処理コードを生成する際には, リアルタイム処理コードを生成するオブジェクトの作成を RTCreator によって行う。こうすることで, 同じ構造のオブジェクトモデルを用いてシミュレーションプログラムの変換とリアルタイム処理コードの自動生成を行える。また, Creator とオブジェクトツリーに組み込まれるクラス群を作成すれば, 生成するコードの種類を追加することができる。

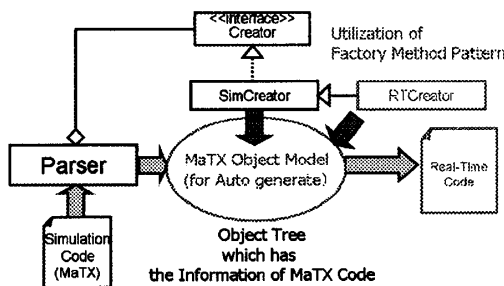


図 4: Automatic generation of Real-Time control program using Factory Method Pattern

5. 倒立振子の振り上げ制御

本研究で開発した統合環境の有用性を確認するために, 倒立振子の振り上げ制御実験 [7] を行った。倒立振子の

振り上げ制御を行うことで, 制御系設計プロセス全体を経験することができる。今回開発した統合環境を用いることにより, シミュレーションプログラムのうち微分方程式を記述した関数と制御器の情報を記述した関数のみを記述するだけで, シミュレーションから制御実験までを実行することができる。

実験結果を図 5 に示す。図 5 で横軸は時間 [sec], 縦軸は振子の角度 [rad] を表している。また, 実験を行った際のサンプリング周期は 5[ms] である。図 5 より, 実験開始から約 4 秒後に振子の角度が 0[rad] に近くなり倒立振子の振り上げ制御に成功していることが分かる。

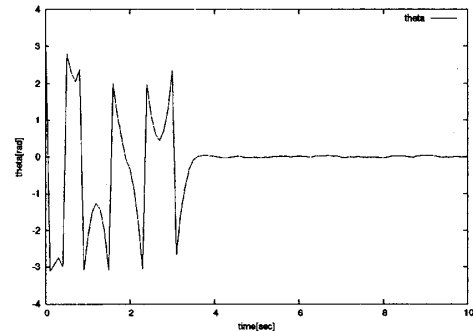


図 5: The experiment result of swinging up a pendulum

6. まとめ

本研究では, シミュレーションからリアルタイム制御実験までをシームレスに実行できる統合環境を開発した。そして, 今回開発した統合環境を用いて倒立振子の振り上げ制御を行い, 有用性を確認した。

参考文献

- [1] 平田研二, 大原伸介, 新銀秀徳. 制御系設計・実験統合環境の構築と学部生実験への適用による考察, 2005. 第5回計測自動制御学会制御部門大会 pp.749-752.
- [2] 古賀雅伸, 矢野健太郎. リアルタイム処理コードを自動生成する遠隔制御システム, 2004. FIT2004 第3回情報科学技術フォーラム, pp.433-434.
- [3] 舟木陸議, 羅正華. Linux リアルタイム計測/制御開発ガイドブック. 秀和システム, 1999.
- [4] 松木毅. オブジェクトモデル化に基づく Java による数値計算エンジンの開発. Master's thesis, 九州工業大学大学院情報工学研究科, 2005.
- [5] 古賀雅伸. 制御・数値解析のための MaTX. 東京電気大学出版, 2000.
- [6] 結城浩. Java 言語で学ぶデザインパターン入門. ソフトバンク, 2001.
- [7] Astrom, K.Furuta. Swinging up a pendulum by energy control. *Automatica*, Vol. 36, pp. 287-295, 2000.