

K-018

# 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 Automatic Generation of Real-Time Code for Remote Control System

古賀 雅伸†  
Masanobu Koga

矢野 健太郎†  
Kentarou Yano

## 1. はじめに

制御系の設計は、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。ロボットや倒立振子等の制御を行う際には、リアルタイムプログラム (RT プログラム) が必要となる。RT プログラムのリアルタイム処理を行う部分の作成は、リアルタイム OS の API や制約について理解した上で作成する必要がある。一般に RT プログラムを作成する環境を提供するリアルタイム OS は高額であるため、フリーのリアルタイム OS である RT-Linux[1] を利用するケースが増えている。RT-Linux 上の場合、通常のプログラムとは異なりカーネル空間で動作するプログラムとして作成する必要がある。このため、RT プログラムはシミュレーションプログラムなどの非 RT プログラムと比べて作成が容易ではない。

一般に、シミュレーションプログラムを作成する環境 (コンピュータや言語) は RT プログラムを作成する環境と異なるため、RT プログラムはシミュレーションプログラムと独立に作成することが多く、多大な労力を要する。

本研究では RT プログラムを、非 RT プログラムと同様に簡単に作成できるように、RT プログラム作成フレームワーク MK-Task を開発した。また、シミュレーションプログラムで繰り返される計算内容とリアルタイム制御プログラムで一定時間 (サンプリング周期) 毎に実行される計算内容が類似するという性質に着目して、シミュレーションプログラムからリアルタイム処理コードを自動生成するシステムを開発した。このシステムを用いることにより、RT プログラムを新たに作成しなくてもシミュレーションからシームレスに制御実験を行うことができる。自動生成の元となるシミュレーションプログラムの作成には MATX[2] を用いる。

## 2. RT プログラム作成フレームワーク

### 2.1 行列計算関数

制御系の計算では行列が多用されるが、RT-Linux の制約によりユーザ空間用に開発された行列計算ライブラリを使用することはできない。そこで、カーネル空間で使用できる行列計算関数を作成した。作成した関数は、`kmalloc()` と `kfree()` を用いて動的にメモリ確保・解放を行い、任意の大きさの行列の四則演算を行うことができる。

### 2.2 ローカル制御・リモート制御兼用 GUI

MK-Task は分散オブジェクトを利用することで、遠隔制御機能を提供する。リアルタイム制御システムで提供される GUI を図 1 に示す。この GUI は

- 入出力モジュールのロード、アンロード
- リアルタイムタスクのロード、アンロード
- データの表示
- プロパティの表示・変更
- データの保存

など、リアルタイム制御を行うための操作を簡単なマウス操作とキーボード入力のみで行える。

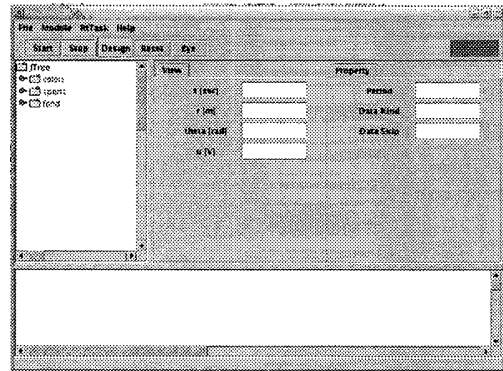


図 1: リアルタイム制御用 GUI の起動画面

図 2 に作成したリアルタイム制御システムのクラス図を示す。

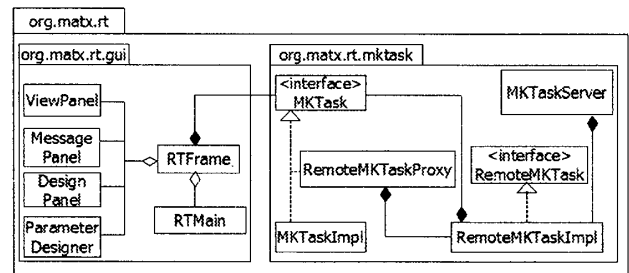


図 2: リアルタイム制御システムのクラス図

図 2 に示すクラス図では、リアルタイム制御用の GUI を提供する RTFrame クラスが MKTask インターフェイスをフィールドとして持っている。ローカルマシンでの制御を行う時は MKTaskImpl クラス、遠隔制御を行う時は RemoteMKTaskImpl クラスを用いる。

## 3. リアルタイム処理コード自動生成

開発したシステムは、シミュレーションプログラムからリアルタイム処理コード生成に必要な情報を抽出し、XML ベースのファイルに保存する。そして、XML ファ

†九州工業大学

イルからリアルタイム処理コードを生成する。このように処理を分散することにより、シミュレーションプログラムに依存しない汎用性の高いシステムとなっている。自動生成を行う際の処理の流れを図3に示す。

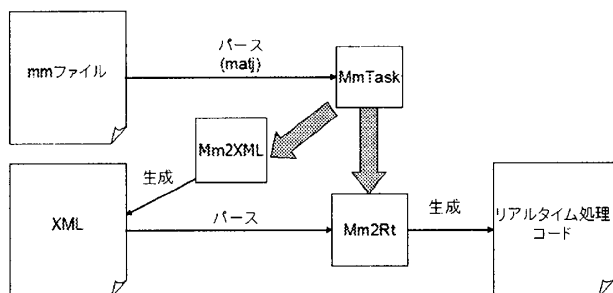


図3: 自動生成の流れ

MmTaskはMaTXのソースファイルであるmmファイルをMaTJ[3]を用いてパースし、得られた情報を保持するクラスである。Mm2XMLクラスはMmTaskオブジェクトの保持する情報をXMLファイルに保存する。そして、Mm2RtクラスはMmTaskオブジェクト又は、XMLファイルから自動生成に必要な情報を取得し、リアルタイム処理コードを生成する。

#### 4. 倒立振子の振り上げ制御実験

本研究で開発したシステム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[4]を行った。

図4に本研究で開発した遠隔制御システムを示す。実験装置はアンプを通してサーバマシンと接続されている。サーバマシンはネットワークを介して複数のクライアントマシンと接続されており、クライアントマシンはサーバマシン経由で実験装置を共有している。

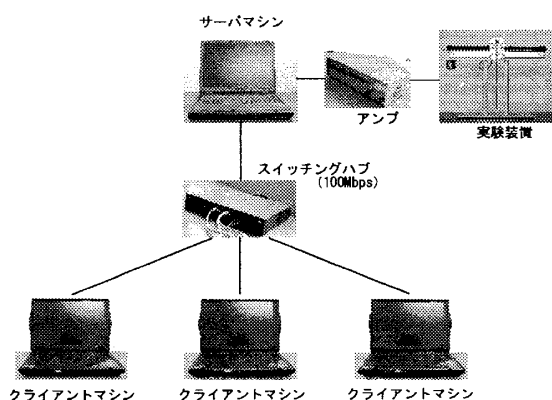


図4: 遠隔制御システム

なお、今回作成したリアルタイム処理コード自動生成システムにより、振り上げ制御を行うプログラムの雛形部分と安定化制御部分を作成した。

実験結果を、図5に示す。図5で上が台車の動き、下が振子の動きを表している。図5より、実験開始から約

4秒後に振子の角度が0rad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。

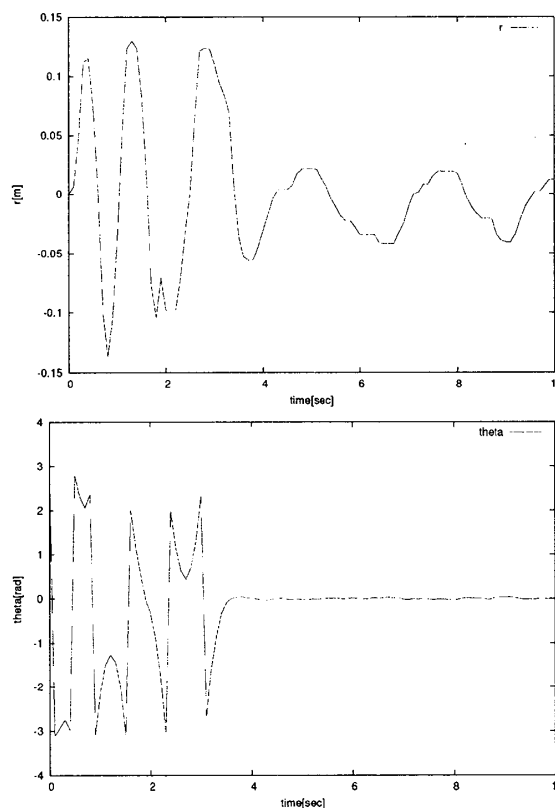


図5: 実験結果

#### 5. まとめ

本研究ではRTプログラム作成フレームワークの開発を行った。また、シミュレーションプログラムからリアルタイム処理コードを自動生成するシステムを開発し、倒立振子の振り上げ制御を行い有用性を確認した。今回開発したシステムを用いることで、シミュレーションからシームレスに制御実験を行うことが可能となる。

#### 参考文献

- [1] 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.
- [2] 古賀雅伸. 制御・数値解析のためのMaTX. 東京電気大学出版, 2000.
- [3] 石田聡. プログラミング言語のオブジェクトモデル化に基づく数値計算言語のJavaによる実装, 2004. 2003年度修士学位論文.
- [4] K.Furuta Astrom. *Swinging up a pendulum by energy control; Automatica vol 36 (2000)*. 2000.