

# Jamox を用いた倒立振子による 制御系モデルベース開発手法の教育

井手敬也 古賀雅伸（九州工業大学） 矢野健太郎（福岡工業大学短期大学部）

## Education of Model-Based Development of Control System by Inverted Pendulum Using Jamox

\*T.Ide, M.Koga (Kyushu Institute of Technology) and K.Yano (Fukuoka Insutitute of Technology)

**Abstract**— In this paper, developed tool be able to perform process in modeling, analysis, simulation, experiment and so on seamlessly in single one. Developed tool be able to experiment with appropriate block diagram In all process, and experience Model-Based Development for control system by using developed tool and learning materials.

**Key Words:** real-time, jamox, inverted pendulum, M<sub>A</sub>TX

### 1 はじめに

近年、制御工学では、制御系ソフトウェアによる演習と講義を組み合わせた教育手法が展開されている。この教育手法は、講義で学んだ内容を演習で応用させることで制御工学に対する理解を促進させ、教育効果を高めることを目的としている。

制御系のモデリングやシミュレーションでは Simulink / MATLAB<sup>1, 2)</sup>, Xcos / Scicos<sup>3)</sup>, Jamox / M<sub>A</sub>TX<sup>7, 4)</sup>などの GUI 形式のツールが使用されている。こういったツールでは、制御対象のモデリングや解析、コントローラ的设计、シミュレーションといった部分をサポートしている。しかし、制御工学では実機を用いた制御実験までを通して行うことが理想であり、制御実験を行うことが可能なツールは多くはない。

Simulink / MATLAB では Simulink Coder<sup>5)</sup>を用いることで Simulink / MATLAB で作成したコードやブロックから c 言語のコードの生成が可能であり、生成されたコードはシミュレーションの高速化やラピッドプロトタイピング、HIL(hardware-in-the-loop)<sup>6)</sup>テストなど、リアルタイムと非リアルタイムのアプリケーションに利用できる。また、M<sub>A</sub>TX ではリアルタイム機能を実装した R<sub>T</sub>M<sub>A</sub>TX<sup>4)</sup>を用いたコマンドラインベースのツールを使用して制御実験を行うことができる。Fig. 1 に M<sub>A</sub>TX を使用したコマンドラインベースのツールを示す。

Simulink / MATLAB では Simulink Coder で生成したコードを使用して制御実験を行う場合、制御対象と通信を行う設定には専門的な知識が必要であり、制御実験の環境を構築するには多くの時間が必要と成ります。一方、M<sub>A</sub>TX を用いたツールではコマンドラインベースのツールであるため、操作はほとんどキーボードで行うため操作性が低く、モデリングから制御実験まですべて数値や数式で表されるため制御対象やコントローラの関連性が非常にわかりづらい。

そこで本研究では、ブロック線図からリアルタイム制御コードを生成できる機能をモデリング・シミュレーションツールである Jamox に追加し、倒立振子を用いた MDB<sup>7)</sup>手法に基づいた制御工学教育を行うことのできるシステムと、教材を提案する。MDB 手法の導入



```
ide@ubuntu: ~/git/RTSystemControlPackage
ide@ubuntu: ~/Dropbox... x ide@ubuntu: ~/git/RTSys... x

<<倒立振子のシミュレーション>>

台車の初期位置 [m]      : r0 = 0 ----> 0
台車の目標値 [m]       : ref = 0 ----> 0.1
振子の初期角度 [deg]   : th0 = 10 ----> 8
シミュレーション時間 [sec] : tf = 3 ----> 5
```

Fig. 1: コマンドラインベースのツール

により、制御系開発プロセスの各フェーズでブロック線図を用いるので、制御対象とコントローラの関連性がわかりやすく、各フェーズ間の移行も容易になり理解を深めることができる。また、GUI 形式による操作性の向上も期待できる。制御実験環境の構築は USB メモリ内に構築できるよう OS や制御実験に必要なパッケージ、ツールをインストールしたイメージファイルを作成することで複数の環境を容易に構築できる。

### 2 開発したシステム

#### 2.1 システムの構成

本研究で開発したシステムの構成を Fig. 2 に示す。本システムではモデリングから制御実験を一貫して行うためリアルタイム処理が可能な OS が必要となる。そこでリアルタイム処理を可能とするために、リアルタイム制御パッケージ<sup>8)</sup>をインストールした Ubuntu<sup>9)</sup>を用いて実験環境を構築する。

制御実験では Jamox で生成したブロック線図から M<sub>A</sub>TX のコードを生成し、生成したコードから R<sub>T</sub>M<sub>A</sub>TX

を用いて C 言語のコードに変換する。この時、 $\text{MaTX}$  のコードでハードウェアとの入出力処理を記述したファイルも同時に変換する。そして C 言語のコードを共有ライブラリに変換する。Jamox では生成した共有ライブラリを読み込みリアルタイム制御を行う。共有ライブラリの読み込みには JNA<sup>10)</sup> を使用することで C 言語の関数を Java から使用することができる。

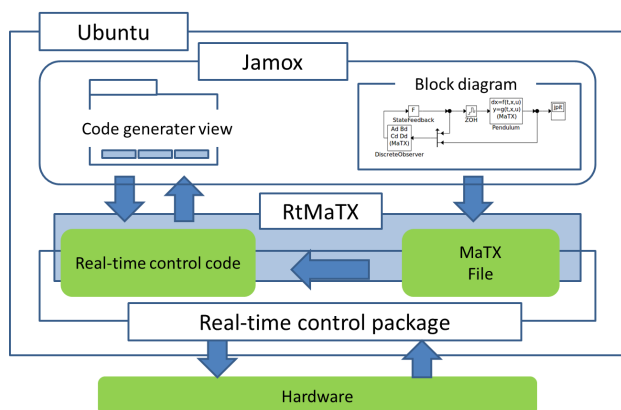


Fig. 2: システム制御学習システムの構成

## 2.2 システムの特徴

本システムは以下の特徴を持ち、制御工学教育を効率的に行うことができる。

- リアルタイム制御環境を提供  
リアルタイム制御パッケージによるリアルタイム制御環境が提供される。
- 既存の環境に影響を与えないシステム環境の構築  
USB メモリ内にカスタマイズした OS をインストールして、USB メモリから起動するため、PC 内にある既存の環境に影響を与えることなく環境を構築できる。また、カスタマイズした OS のイメージを複数の USB メモリに入れるだけで容易に複数の環境を構築することも可能である。
- ブロック線図を用いた学習システム

Fig. 3 に示すように制御実験の各フェーズにおいてブロック線図を用いるために、制御系やコントローラの関連性がわかりやすく、制御実験の流れを把握しやすい。

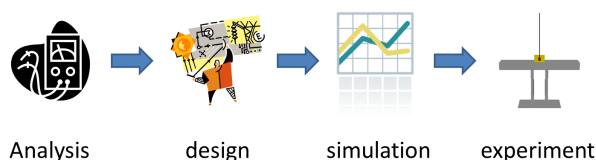


Fig. 3: 制御実験のプロセス

## 2.3 リアルタイム制御用コードの生成

Fig. 4 に制御系ブロック線図の例を示す。制御対象やコントローラのブロックでは数値計算言語  $\text{MaTX}$  で記述された定数行列と線形システムのブロックが使用できる。コードの生成では、一つのブロックの場合、ブロッ

ク内に記述されたコードがそのまま出力される。複数のブロックをコード生成する場合は、複数のブロックを一つにまとめたサブシステムブロックを使用する。サブシステムブロックでは各ブロックの接続関係を表す隣接行列から状態と出力を計算する式がコードとして出力される。生成されるコードにはリアルタイム処理に必要なコードも含まれる。

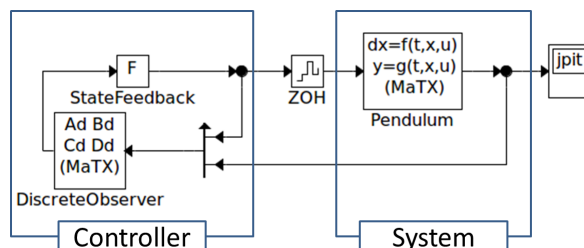


Fig. 4: 制御系のブロック線図

## 3 開発したシステムを用いた制御実験

本システムを倒立振子の安定化制御を行う学生実験意適用した。

### 3.1 倒立振子系

実験で使用する倒立振子を Fig. 5 に示す。

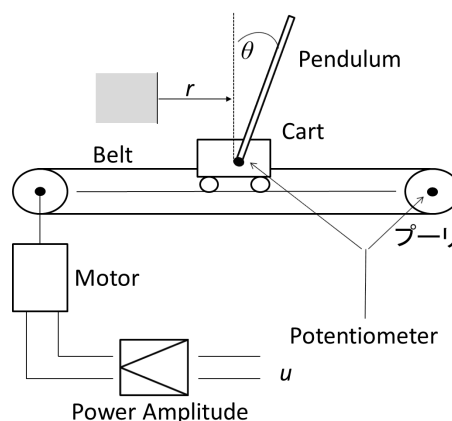


Fig. 5: 倒立振子系

台車と振り子に関する運動方程式は

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ K^{-1} \begin{bmatrix} -f\dot{r} + m\dot{l}\sin\theta \cdot \ddot{\theta} + au \\ mgl\sin\theta - c\dot{\theta} \end{bmatrix} \end{bmatrix}$$

$$K = \begin{bmatrix} M + m & ml\cos\theta \\ ml\cos\theta & J + ml^2 \end{bmatrix}$$

となる。なお倒立振子の物理定数の一覧を表??に示す。倒立振子をモデリングしたブロックを Fig. 6 に示す。

### 3.2 物理定数の同定

倒立振子の物理定数を専用のブロック線図を用いて同定する。Fig. ??では台車に一定の出力を与えて、得られたステップ応答から台車の物理定数を同定するためのブロック線図を示す。

Table 1: 物理定数の一覧

$m$	[kg]	振子の質量
$l$	[m]	回転軸・重心間距離
$M$	[kg]	台車の質量
$f$	[kg/s]	摩擦係数
$J$	[kgm <sup>2</sup> ]	重心周り慣性モーメント
$c$	[kgm <sup>2</sup> /s]	回転軸摩擦係数
$g$	[m/s <sup>2</sup> ]	重力加速度

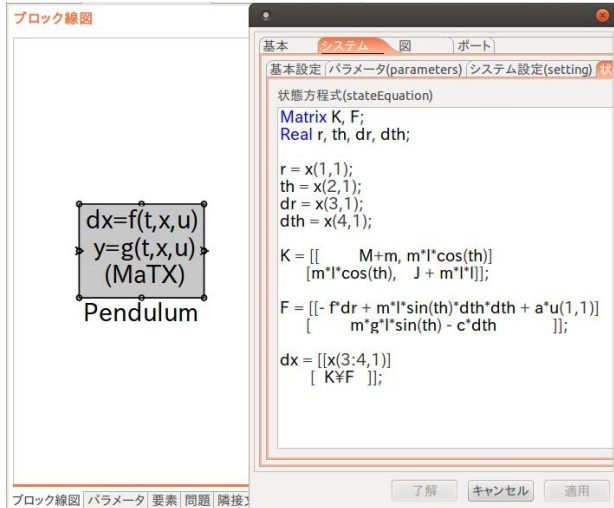


Fig. 6: 倒立振子のブロック

ブロック線図からリアルタイム制御用のコードを生成し、リアルタイム処理を行う。Gain ブロック内に記述されている出力に関する定数を変更することで台車与える出力を変更することができる。また、このブロックでシミュレーションを行うことで実際の結果とシミュレーションの結果を比較することもできる。

### 3.3 システムの解析

解析には Fig. 8 に示すブロックを使用する。

Plant ブロックは倒立振子の線形近似モデルであり、Pendulum ブロックのパラメータを参照している。Fig. 8 に示すようにメニューバーから目的とするメニューを選ぶとシステムの極、可制御性行列、可観測性行列が Fig. 9 に示すようにコンソールに出力される。

解析結果は本システム内に変数として保存されるので、得られた結果を用いて、コンソール上で様々な解析を行うことができる。

### 3.4 コントローラの設計

コントローラを LQ 最適制御による状態フィードバックと連続時間オブザーバを用いて設計する。状態フィードバックの設計では、評価関数の重み行列を調整し、オ

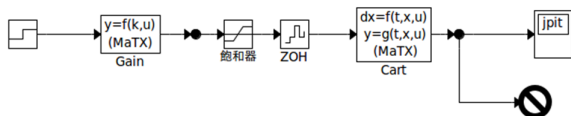


Fig. 7: 台車の物理定数を同定するためのブロック線図

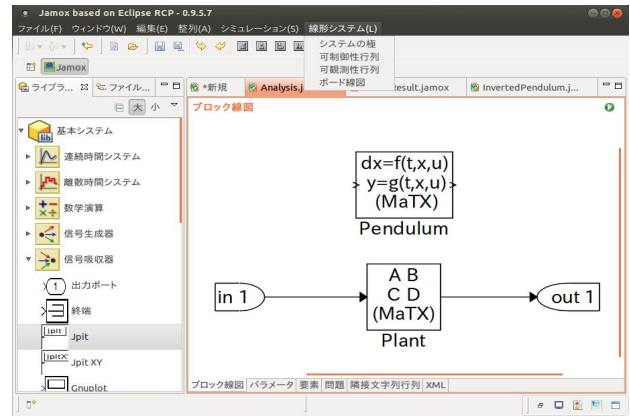


Fig. 8: システムの解析に使用するブロック線図

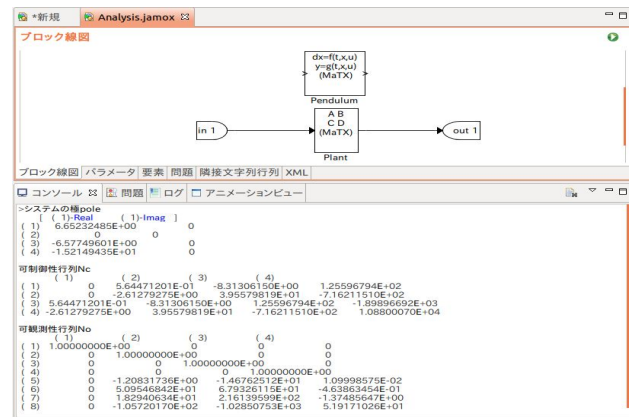


Fig. 9: コンソール画面

ブザーバの設計ではオブザーバの極を調整する。Fig. 10 に設計パラメータを調整する画面を示す。コントローラを含む倒立振子系のブロック線図を Fig. 11 に示す。



Fig. 10: 状態フィードバックと連続時間オブザーバの調整

### 3.5 シミュレーション

シミュレーション画面を Fig. 12 に示す。Fig. 11 に示すブロック線図を用いてシミュレーションを行う。シミュレーションでは振子と台車の状態が JpIt ブロックへ入力されグラフ描画される。連続時間オブザーバや状態フィードバックのブロック内にある極や重み行列を調整しながらシミュレーションを繰り返す。

### 3.6 リアルタイム制御実験

リアルタイム制御実験では Fig. 11 のブロック線図を用いて行う。Fig. 13 に実行画面を示す。以下に実験の手

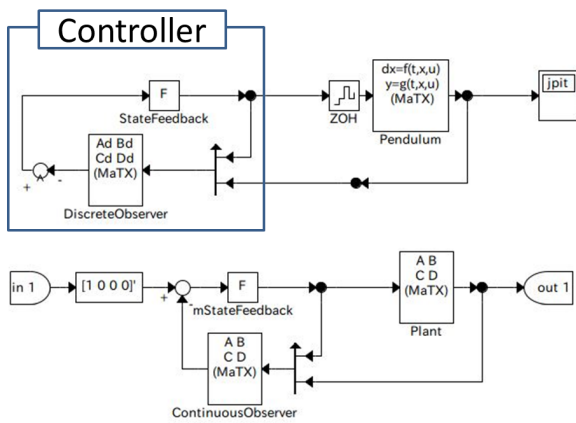


Fig. 11: 制御系全体を表すブロック線図

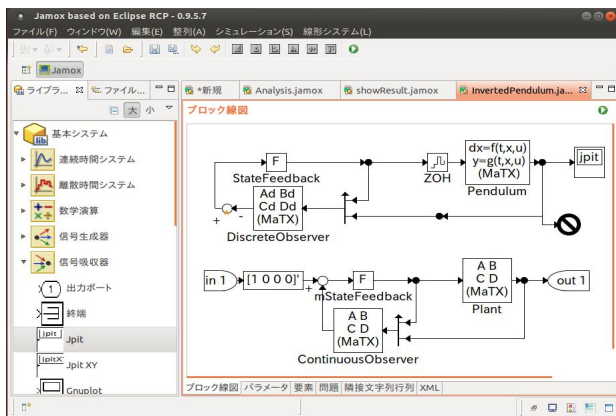


Fig. 12: シミュレーション画面

順を示す.

1. コード生成ビューにおいてコントローラ部分のブロックを選択し、コード生成に続けて共有ライブラリを作成する。
2. リアルタイム制御のための共有ライブラリ (libExperiment\*.so) を読み込む。
3. 実験ビューの「RT Task Start」ボタンをクリックしリアルタイムタスクを開始する。リアルタイム

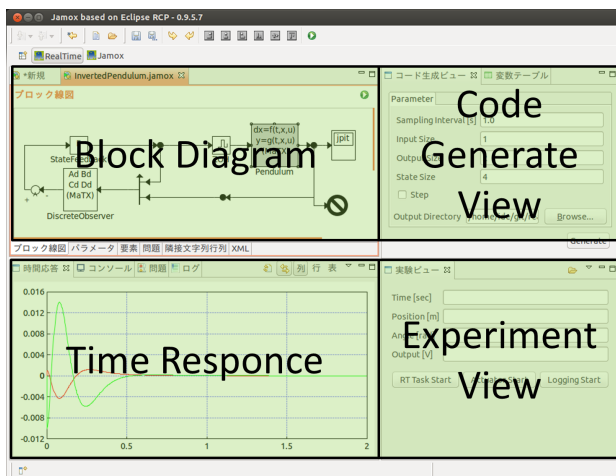


Fig. 13: リアルタイム制御実験用 Jamox

タスクが開始すると台車の位置と振子の角度が時々刻々表示される。

4. 「Actuator Start」ボタンをクリックし、モータへの出力を開始する。
5. モータへの出力を停止するには「Actuator Stop」ボタンをクリックする。
6. リアルタイムタスクを終了するには「RT Task Stop」ボタンをクリックする。その後実験データが変数テーブルに *actual* という変数名で保存される。

### 3.7 実験結果の表示

リアルタイム制御実験が終了すると実験中の倒立振子の台車の位置と振子の角度、台車への入力電圧が変数に保存される。変数に保存されたデータを Fig. ?? に示す実験結果参照用のブロック線図を使用して見ることができる。Fig. 15, 16 に示すように容易にシミュレーションと比較を行うことができる。

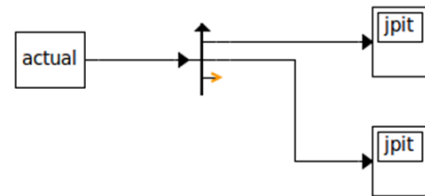


Fig. 14: 実験結果の表示を行うブロック線図

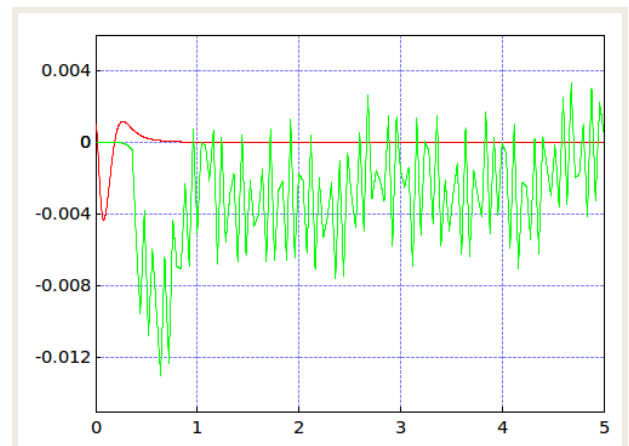


Fig. 15: 台車のシミュレーションと実験結果の比較



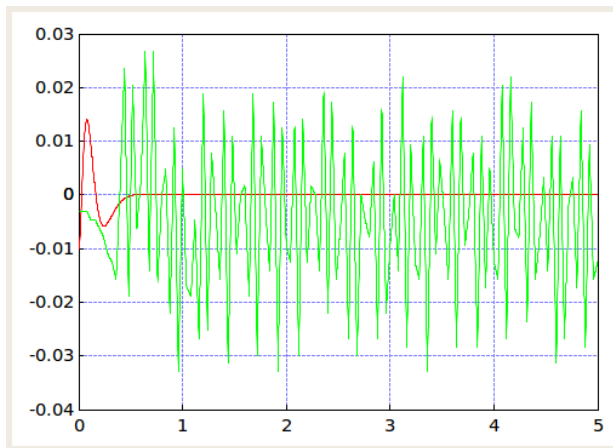


Fig. 16: 振子のシミュレーションと実験結果の比較

#### 4 評価

本システムは2012年前期から九州工業大学情報工学部システム創成情報工学科のシステム創成プロジェクトの制御実験において実際に学部3年生に利用されている。学生は本システムを含むリアルタイム処理が可能なOSがインストールされたUSBメモリと倒立振子を用いて安定化制御を目的に実験を行った。学生には制御実験のテキストと本システムのマニュアルを参考にして実験を行う。今回は2013年度の前期に行われた制御実験を受講した学生23名を対象にアンケートを行った。表25段階評価のアンケート結果を示す。点数は平均であり、5点満点で数字が大きいほど評価が高いことを表す。

Table 2: アンケート結果

アンケート項目	学生
ブロック線図の使い方はわかりやすい	4.00
物理定数の測定方法はわかりやすい	3.69
システムの特性格解析の方法はわかりやすい	3.65
動作実験の方法はわかりやすい	3.09
実験結果の表示方法はわかりやすい	3.26
各ブロック線図は使いやすい	3.64
物理定数は測定しやすい	3.77
システムの特性格解析で使った機能は使いやすい	3.64
リアルタイム制御機能は使いやすい	3.50
実験は行いやすい	3.52

すべての項目について概ね良好な評価となっている。マニュアルでの制御実験の方法や実験結果の表示方法については若干評価が低い。これについては制御実験に関する手順についてより詳細に記述する必要があると考えられる。またシステム面での制御実験に関する評価も若干低い。これは制御実験で使用するUIの部分の改善が必要であると考えられる。

#### 5 まとめ

本研究では、ブロック線図からリアルタイム制御コードを生成できる機能をモデリング・シミュレーションツールであるJamoxに追加し、MDB手法に基づいた制御工学教育を行うことのできるシステムを提案した。MDB

の導入により制御系開発プロセスの各フェーズの移行にブロック線図を用いるので制御対象とコントローラの関係がわかりやすくなり、GUI形式により操作性が向上した。また、学生実験への導入を行い、アンケートをもとにその評価を行った。

#### 参考文献

- 1) The MathWorks - Simulink. <http://www.mathworks.co.jp/products/simulink/>.
- 2) The MathWorks - Matlab. <http://www.mathworks.co.jp/products/matlab/>.
- 3) Scicos homepage. <http://www.scicos.org/>.
- 4) 古賀雅伸. Linux・Windows できる MATLAB による数値計算. 東京電機大学出版, 2000.
- 5) The MathWorks - Simulink Coder. <http://www.mathworks.co.jp/products/simulink-coder/>.
- 6) Hil テストアーキテクチャ. <http://www.ni.com/white-paper/10343/ja/>.
- 7) MATLAB /Simulink によるモデルベース開発の導入. <http://www.exmotion.co.jp/solution/auto.html/>.
- 8) 古賀雅伸, 岸田和也. Linux のリアルタイム制御パッケージを用いた制御教育環境構築システムの開発. 第5回情報科学技術フォーラム, 2006.
- 9) Ubuntu. <http://www.ubuntu.com/>.
- 10) Jna. <https://github.com/twall/jna/>.