

システム創成プロジェクトII(倒立振子の安定化制御)

平成 26 年 5 月 26 日

倒立振子の安定化制御実験における Jamox の使用方法について説明する。

1 Jamox の利用

1.1 Eclipse の起動

1. メニュー一覧から「アプリケーション」→「アクセサリ」→「端末」メニューを選択、または、ショートカット「Ctrl+Alt+T」で端末を起動する。
2. 端末内のコマンドライン上で以下のように入力し、Eclipse を起動する。図 1 に Eclipse の起動画面を示す。
 - (a) 「`sudo -s`」と入力する。
 - (b) パスワードを入力する。
 - (c) 「`eclipse`」と入力する。

コマンドライン上での入力 —————

```
project@ubuntu:~$ sudo -s
[sudo] password for project: ←ここにパスワードを入力
root@ubuntu:~# eclipse
```

1.2 Jamox の起動

1. Eclipse ウィンドウの左側にある「Package Explorer」ビューの「org.mklab.jamox.rcp」を右クリックし、メニューを表示する。「Run As」→「Eclipse Application」を選択し、Jamox を起動する。図 2 に Jamox を起動する様子を示す。
2. 「/home/project/git/realtime/lecture」をワークスペースに設定する。

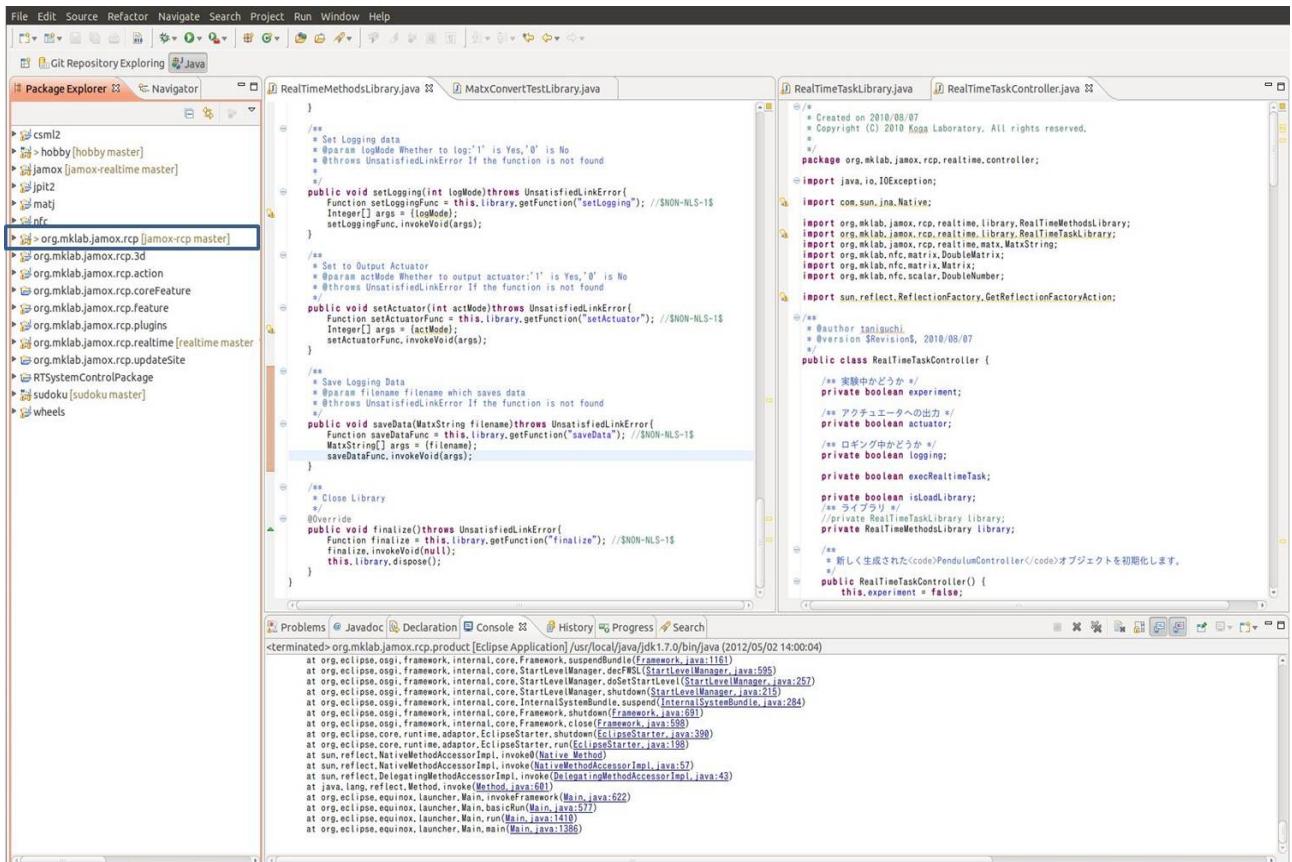


図 1: Eclipse の起動画面

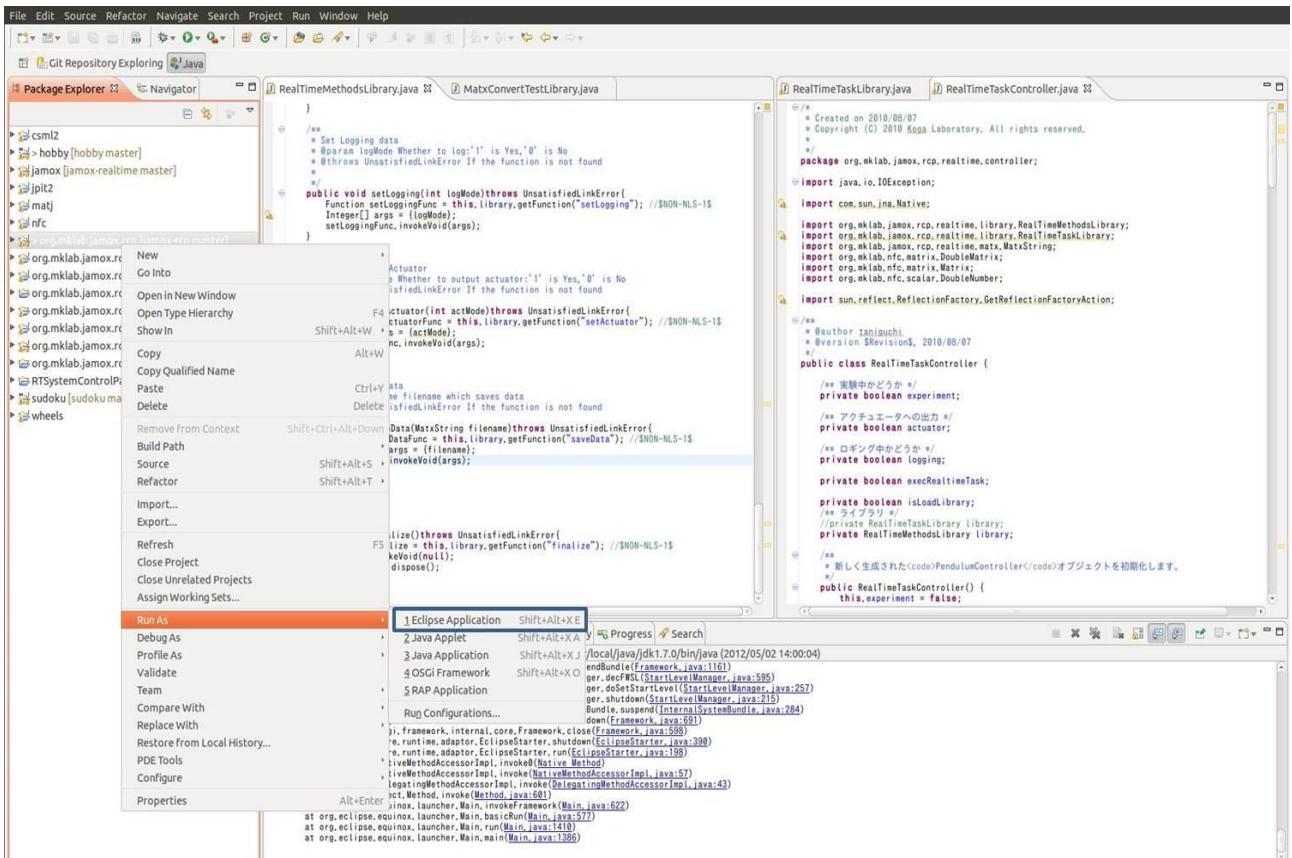


図 2: Eclipse から Jamox を起動する様子

1.3 jamox ファイルの読み込み

Jamox の起動後の画面を図 3 に示す。Jamox ウィンドウの左側にある「ファイルエクスプローラー」にはワークスペース内にある jamox ファイル (*.jamox) が表示される。ファイルをダブルクリックすると、ファイルが読み込まれキャンバスビューにブロック線図が表示される。

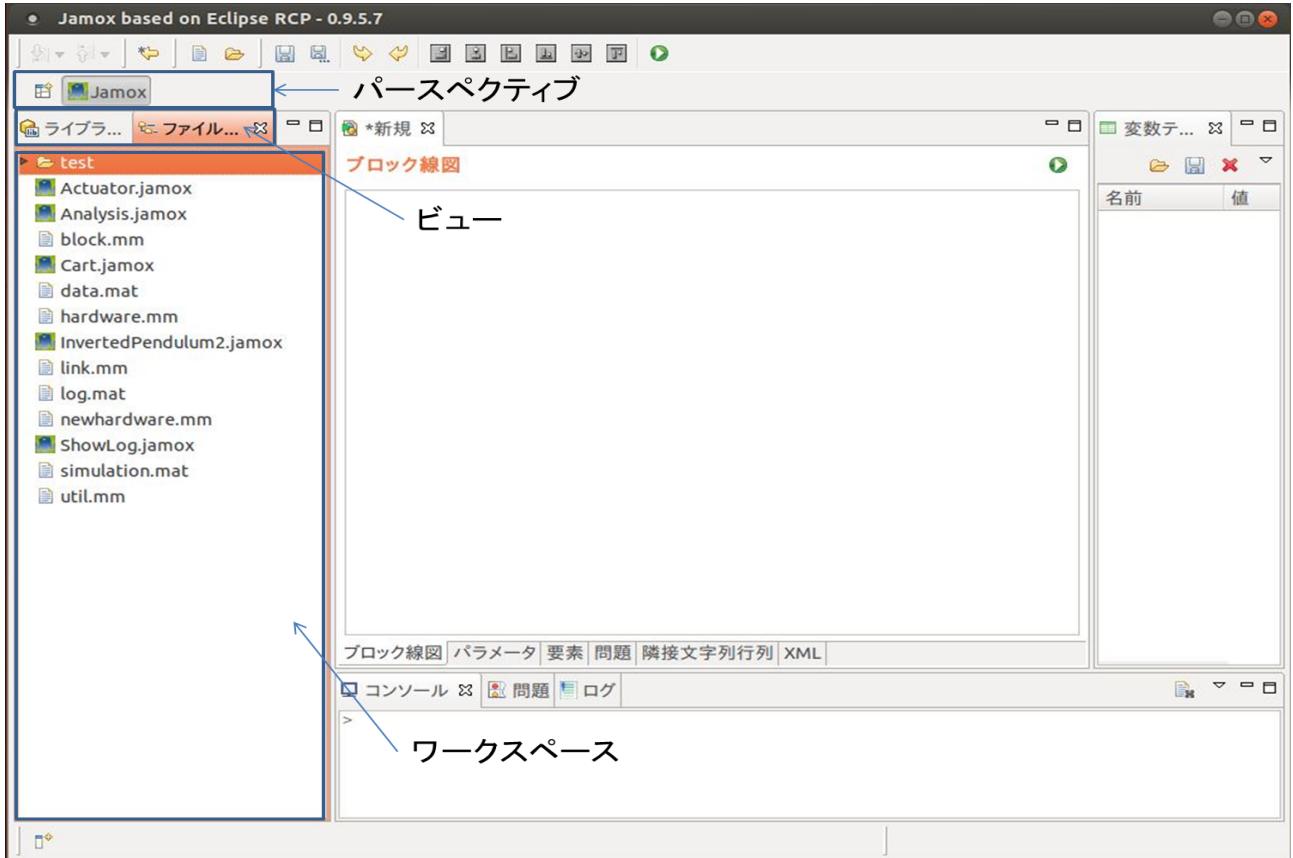


図 3: Jamox の起動画面

2 制御系の開発

2.1 パラメータの同定

2.1.1 パラメータの同定で使用するブロック線図

パラメータの同定で使用するブロック線図について説明する。

1. パラメータ a の同定

パラメータ a を同定する際はワークスペース内にある Actuator.jamox を使用する。図 4 にブロック線図を示す。アクチュエータに入力する電圧の値を変更するには Tuner ブロックを編集する。ブロックをダブルクリック、または、右クリックから「パラメータ設定」を選択し、パラメータ設定ダイアログを開く。そして「システムの設定」タブを選択し、変数 `outputValue` の値を変更する。図 5 に変更画面を示す。

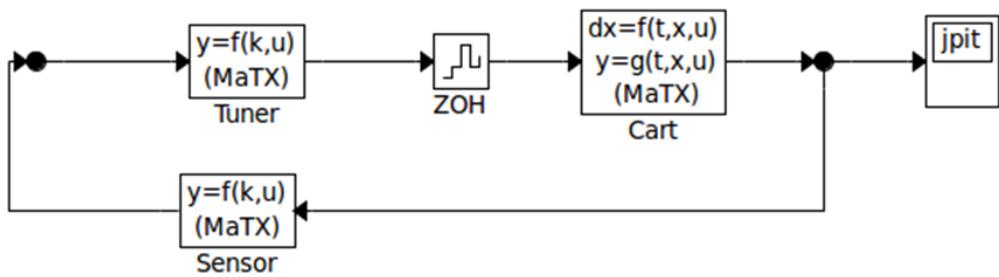


図 4: パラメータ a の同定で使用するブロック線図

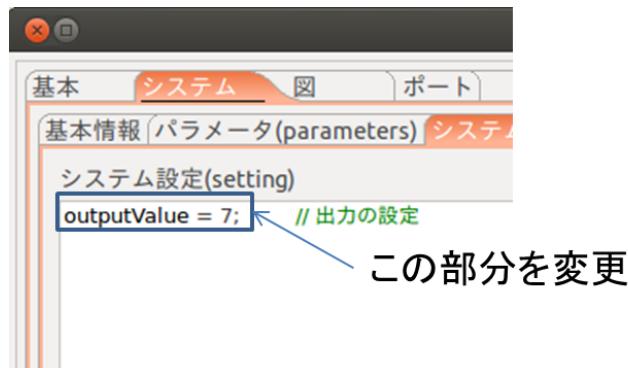


図 5: Tuner ブロックの編集

2. パラメータ M と f の同定

パラメータ M と f を同定する際は、ワークスペース内にある Actuator.jamox を使用する。図 4 にブロック線図を示す。アクチュエータに入力する電圧の値を変更するには Tuner ブロックを編集する。

3. パラメータ J と c の同定

パラメータ J と c を同定する際は、ワークスペース内にある FreePendulum.jamox を使用する。図 6 にブロック線図を示す。

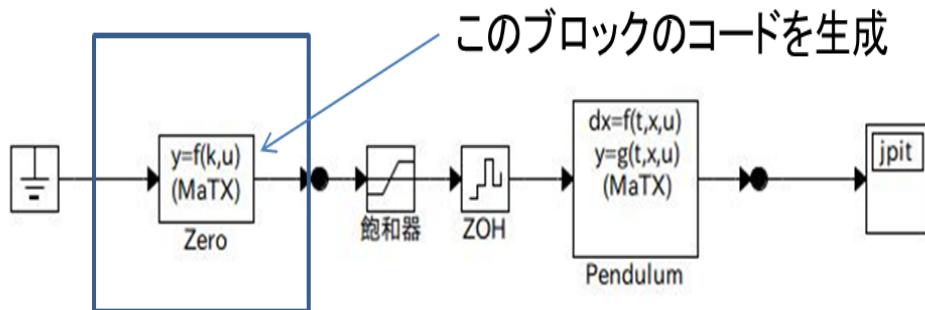


図 6: パラメータ J と c の同定で使用するブロック線図

2.1.2 同定実験

- 必要に応じて Tuner 等のブロックのパラメータを変更する。
- 図 7 に示すように、左上にあるアイコンをクリックすることでパースペクティブの候補を表示し、図 8 に示すように「other...」を選択する。

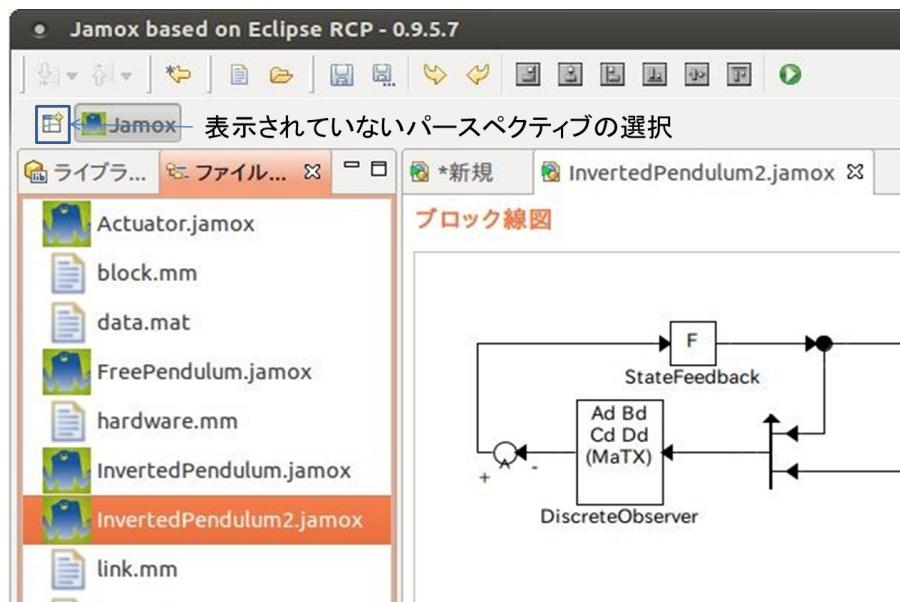


図 7: パースペクティブ選択のアイコン

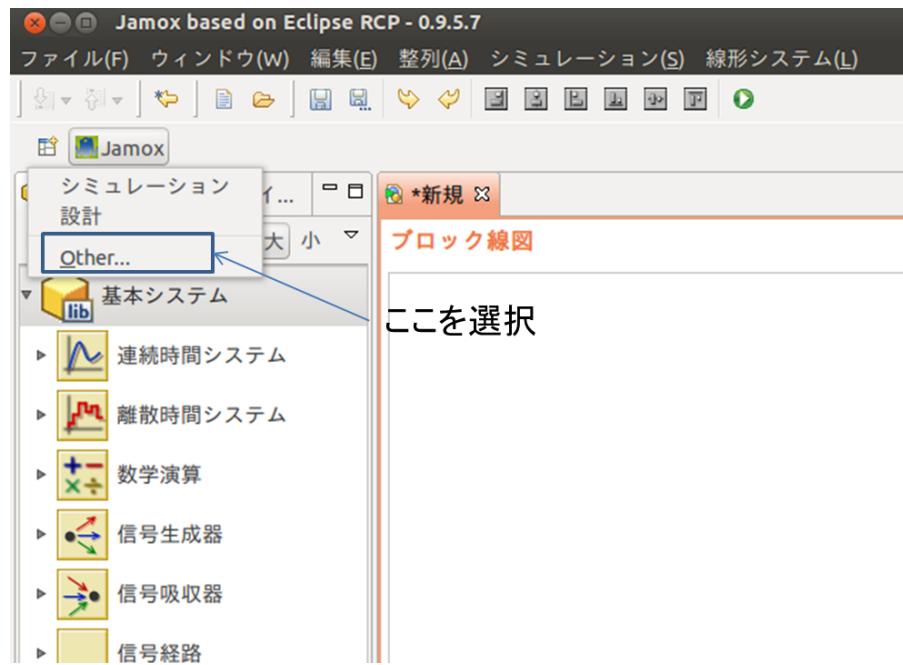


図 8: パースペクティブの選択

図 9 に示すようなパースペクティブの一覧が表示されるので、その中から「RealTime」を選択する。RealTime パースペクティブに変更すると、図 10 に示すような画面が表示される。



図 9: パースペクティブの変更

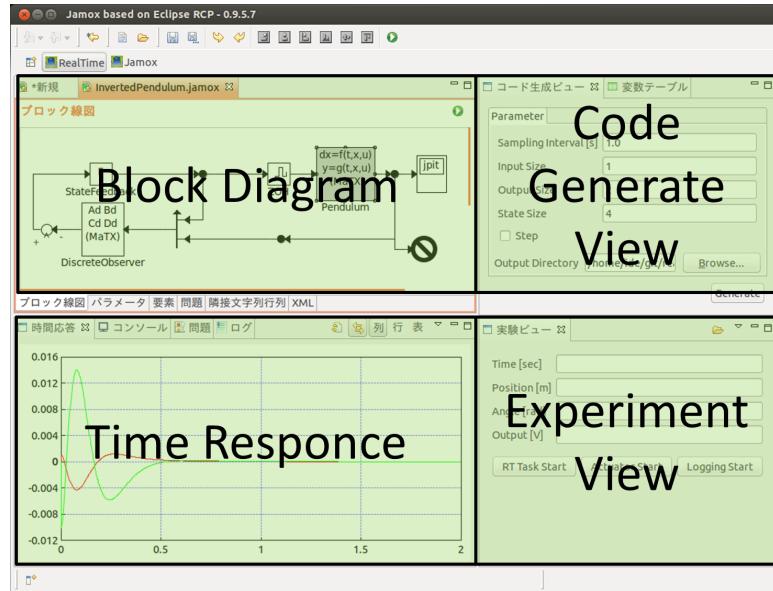


図 10: リアルタイムパースペクティブ

3. 同定用のブロックを選択し、コード生成ビュー (Jamox ウィンドウの右上) にある「Generate」ボタンをクリックすることでリアルタイム制御実験用のコードを生成する。

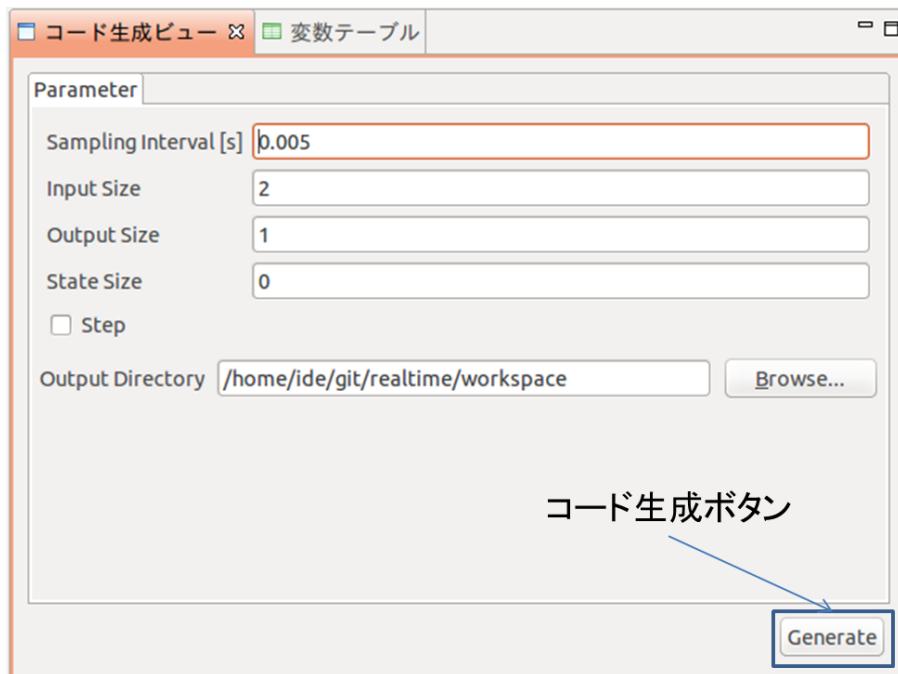


図 11: コード生成ビュー

4. 実験ビューの右上にあるフォルダアイコンをクリックし、フォルダ選択ダイアログを表示する。ダイアログから「/home/project/git/realtime/workspace/」以下にある生成したコード libExperiment*.so を選択する。(例：01/01 の AM11:00 の場合 libExperiment01011000.so)

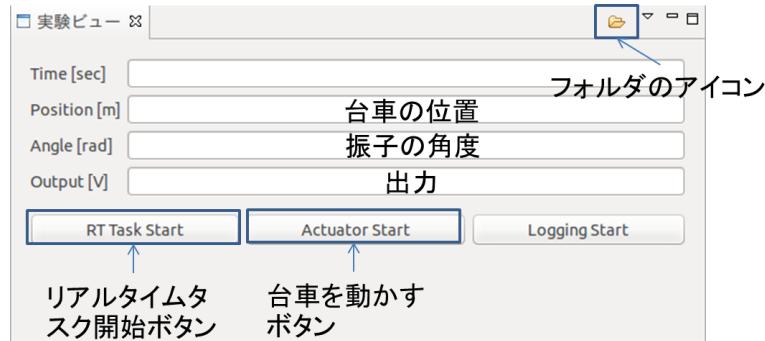


図 12: 生成コードの選択

5. ドライバの左側にあるアクチュエータ(モータ)電源のスイッチ SW1 が OFF であることを確認し、右側にある主電源のスイッチを ON にする。なお、実験を行わないときは必ずアクチュエータ(モータ)電源を OFF にしておく。ON の状態にしておくと勝手に動き出す恐れがある。

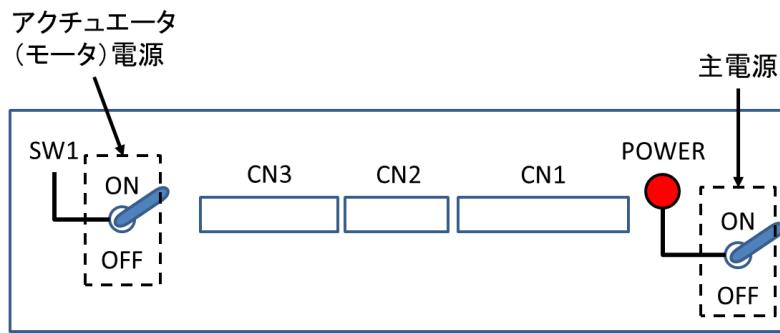


図 13: ドライバ

6. 実験ビューにある「RT Task Start」ボタンをクリックすると、台車の位置と振子の角度が実験ビューに表示され、サンプリング周期毎に更新され続ける。

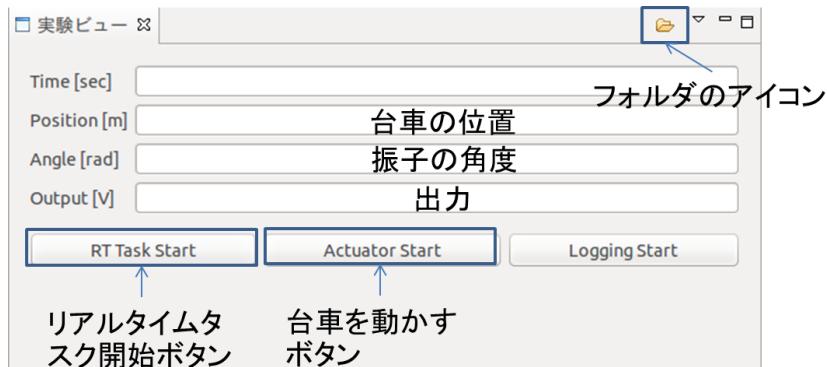


図 14: リアルタイム制御の開始

7. 台車の位置、振子の角度、アクチュエータへの電圧が適切な値であることを確認し、ドライバの左側にあるアクチュエータ(モータ)電源のスイッチ SW1 を ON にする。

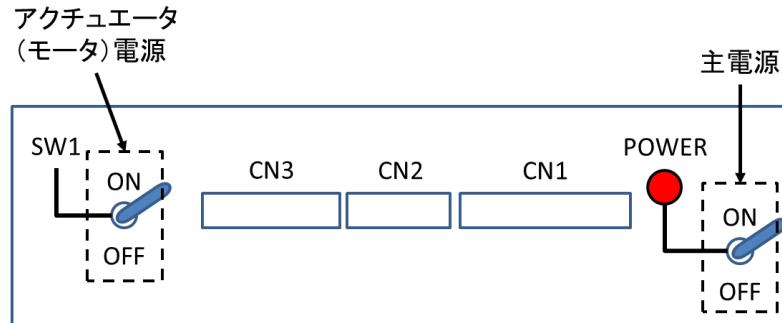


図 15: ドライバ

8. 実験ビューにある「Actuator Start」ボタンをクリックすると、台車が動き始める。同時に台車の位置、振子の角度、アクチュエータへの電圧の値の記録が開始する。

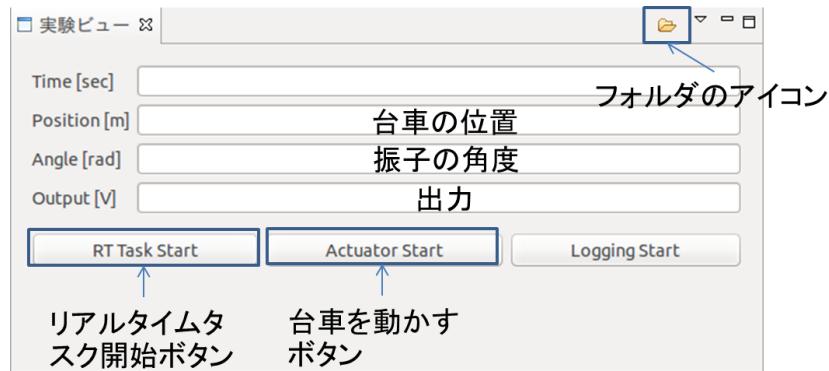


図 16: アクチュエータへの出力開始

9. 実験ビューにある「Actuator Stop」ボタンをクリックすると、台車が停止する。同時に台車の位置、振子の角度、アクチュエータへの電圧の値の記録が終了する。



図 17: アクチュエータ出力の終了

10. 実験ビューにある「RT Task Stop」ボタンをクリックすると、実験ビューに表示されている台車の位置と振子の角度の値の更新が終了する。同時に記録された値は Jamox 内の変数 `actual` として保存される。



図 18: リアルタイム制御の終了

11. ドライバの左側にあるアクチュエータ(モータ)電源のスイッチ SW1 を OFF にする。

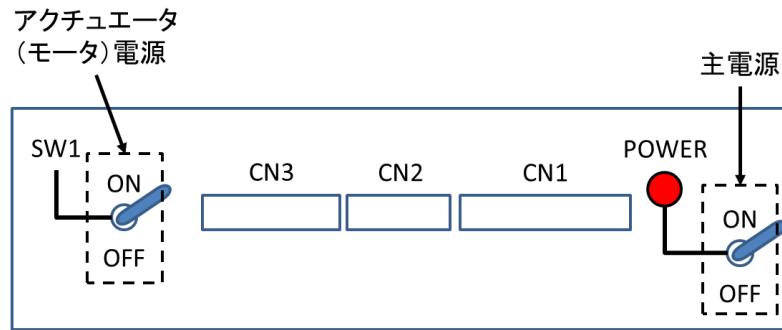


図 19: ドライバ

12. パースペクティブを「Realtime」から「Jamox」へ変更する。

13. 測定データのグラフを描画するため、ワークスペース内にある ShowData.jamox を読み込む。図 20 にブロック線図を示す。

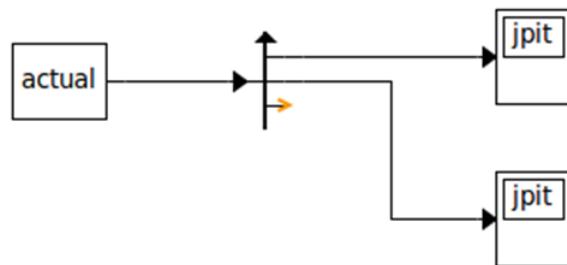


図 20: 測定データのグラフを描画するためのブロック線図

14. シミュレーション開始ボタンをクリックすると、測定データのグラフが描画される。

2.2 システムの特性解析

システムの特性解析(安定性、可制御性、可観測性)を行う。まず、ワークスペース内にある Analysis.jamox を読み込み、倒立振子を表す Pendulum ブロックのパラメータを同定したパラメータに変更する。図 21 にブロック線図を示す。

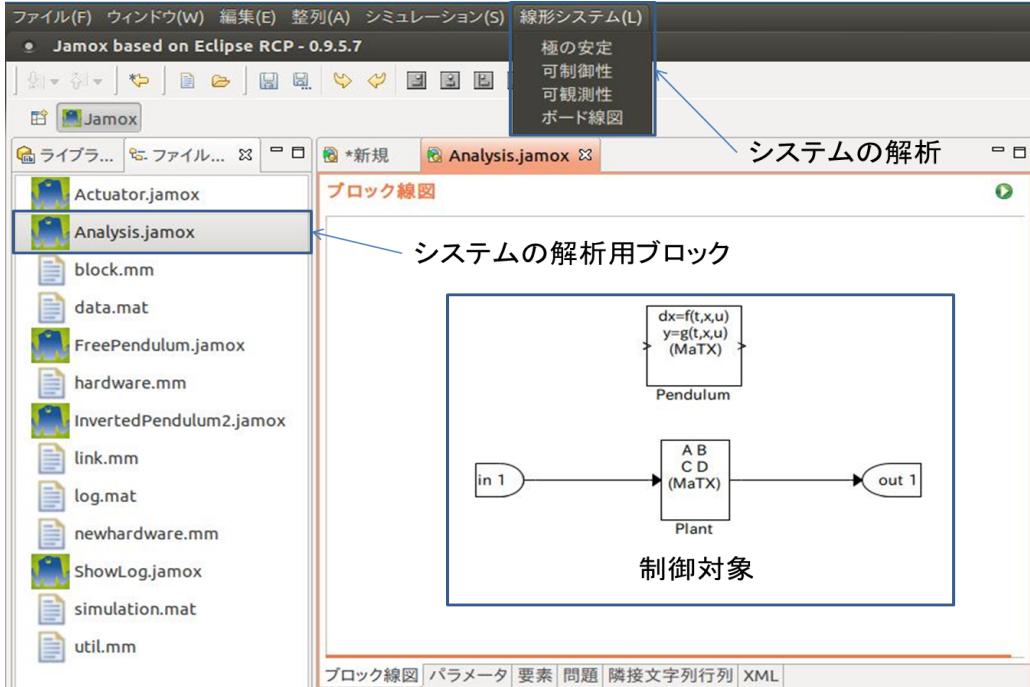


図 21: システムの特性解析

ブロックのパラメータを設定するには、そのブロックをダブルクリック、または、右クリックから「パラメータ設定」を選択し、パラメータ設定ダイアログを開く。そして、「システム設定」タブを選択する。パラメータの設定の様子を図 22 に示す。

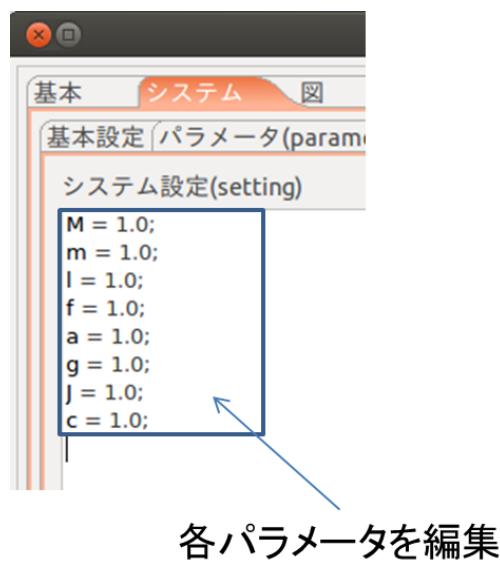


図 22: Pendulum ブロックのパラメータの編集

2.2.1 安定性

メニューバーにある「線形システム」メニューから「システムの極」を選択すると、In ブロックから Out ブロックまでのシステムの極がコンソールに表示され、変数テーブルに pole という名前で登録される。図 23 に実行例を示す。

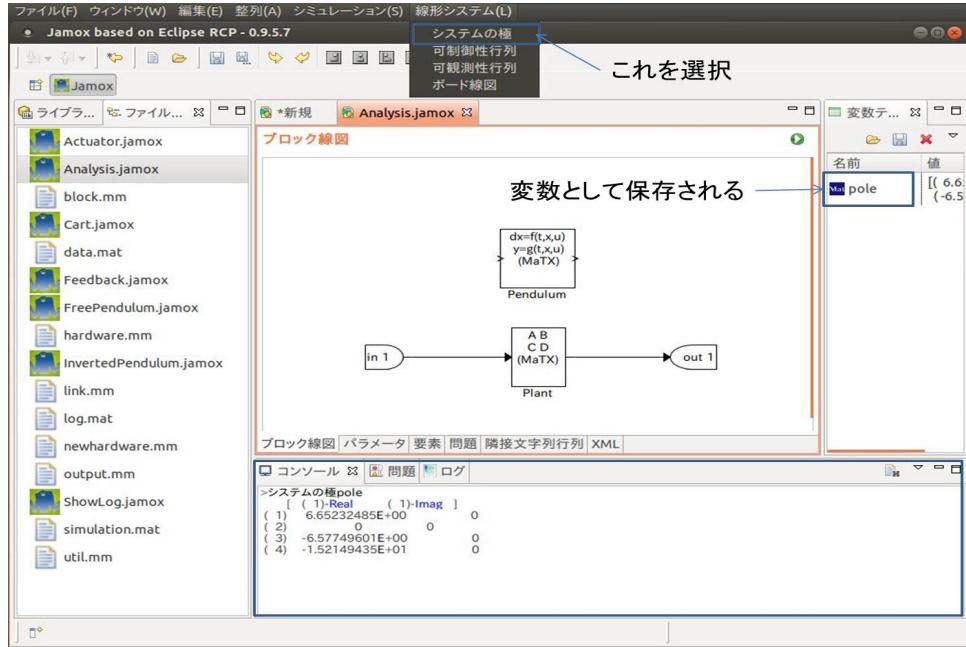


図 23: 安定性の解析

2.2.2 可制御性

メニューバーにある「線形システム」メニューから「可制御性行列」を選択すると、In ブロックから Out ブロックまでのシステムの可制御性行列がコンソールに表示され、変数テーブルに Nc という名前で登録される。図 24 に実行例を示す。

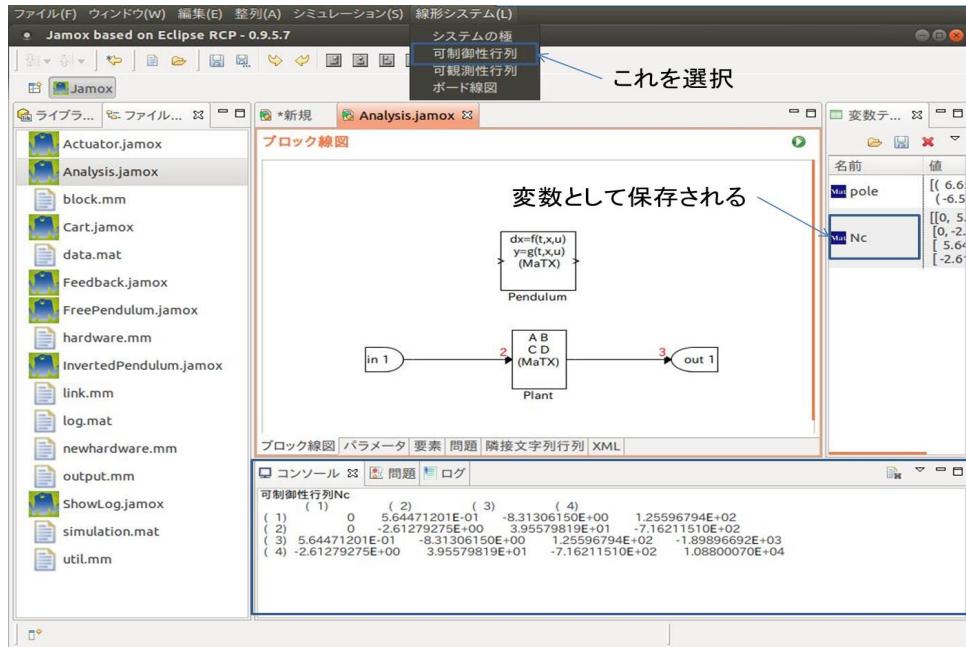


図 24: 可制御性の解析

2.2.3 可観測性

メニューバーにある「線形システム」メニューから「可観測性行列」を選択すると、In ブロックから Out ブロックまでのシステムの可観測性行列がコンソールに表示され、変数テーブルに No という名前で登録される。図 25 に実行例を示す。

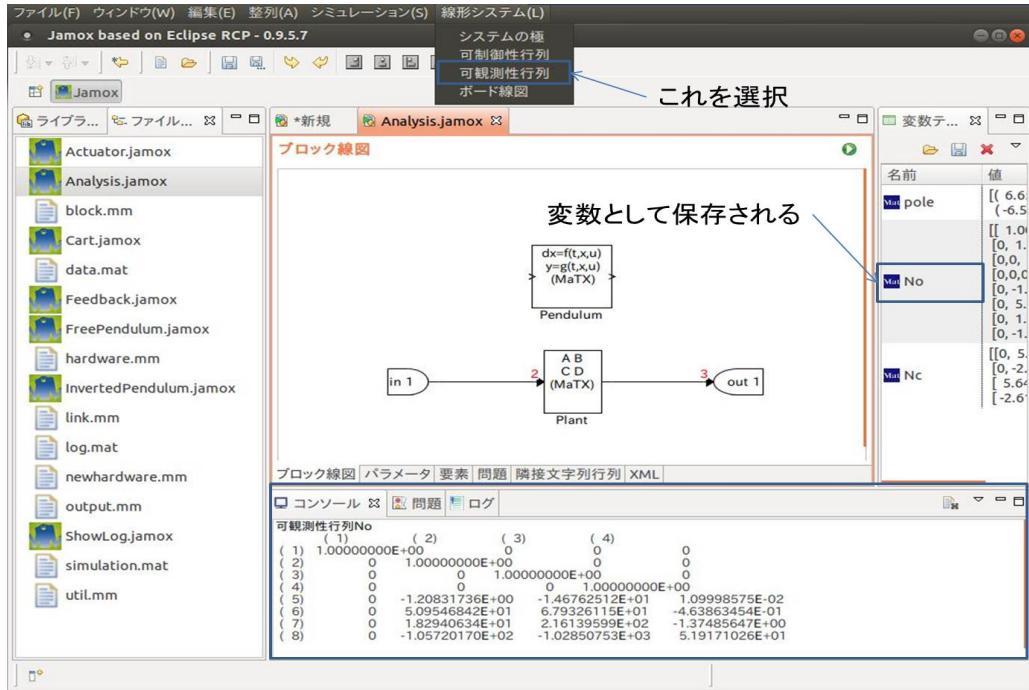


図 25: 可観測性の解析

2.3 コントローラの設計

コントローラー (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) を設計する。まず、ワークスペース内にある InvertedPendulum.jamox を読み込み、倒立振子を表す Pendulum ブロック内のパラメータを同定したパラメータに変更する。図 26 にブロック線図を示す。

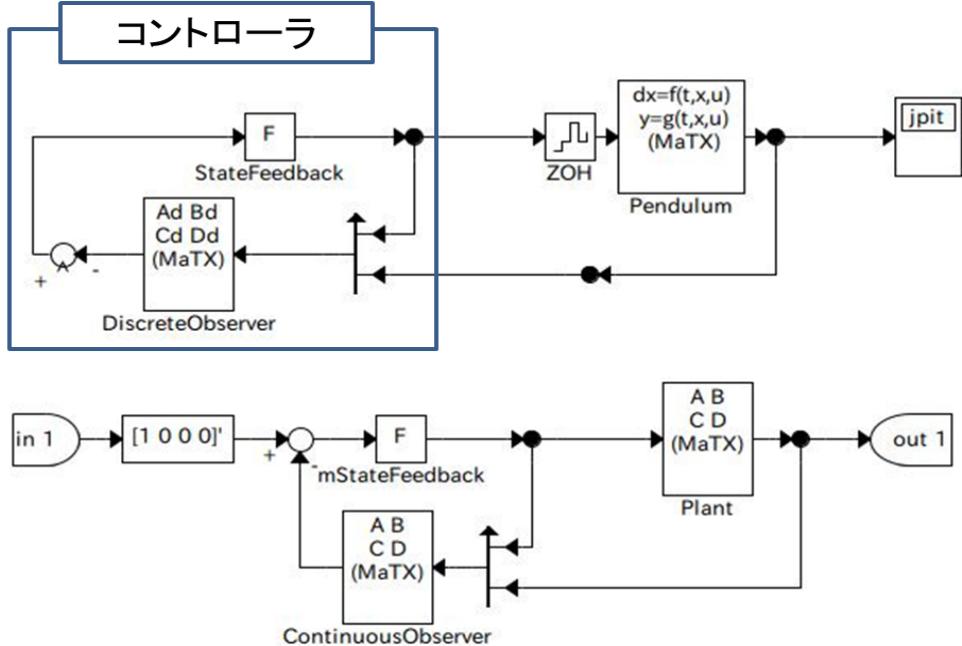


図 26: InvertedPendulum.jamox

2.3.1 LQ 最適制御の設計

mStateFeedback ブロックをダブルクリック、または、右クリックから「パラメータ設定」を選択し、パラメータダイアログを開く。そして、「システム設定」タブの変数 Q と R に評価関数の重み行列 Q と R の値をそれぞれ設定する。図 27 に編集の様子を示す。

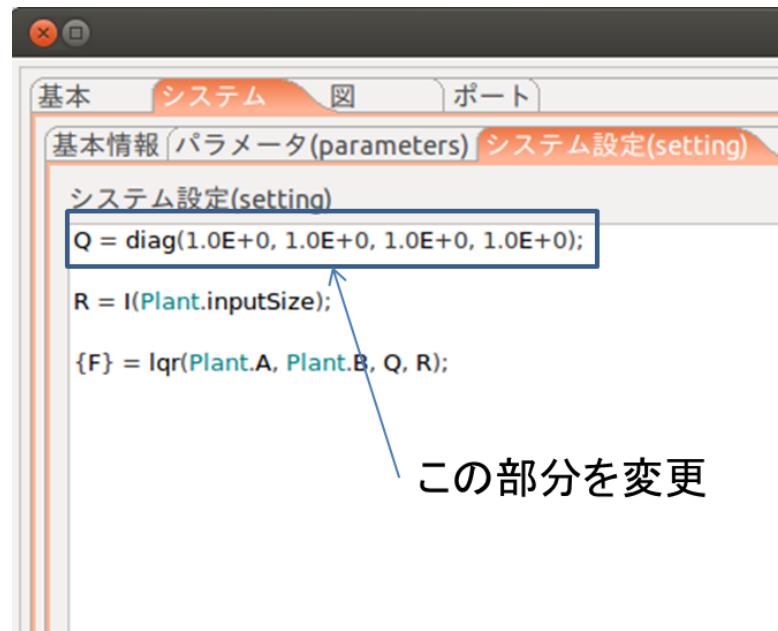


図 27: mStateFeedback の編集

2.3.2 連続時間オブザーバの設計

ContinuousObserver ブロックをダブルクリック、または、右クリックから「パラメータ設定」を選択し、パラメータダイアログを開く。そして、「システム設定」タブの変数 pole にオブザーバの極を設定する。図 28 に編集の様子を示す。

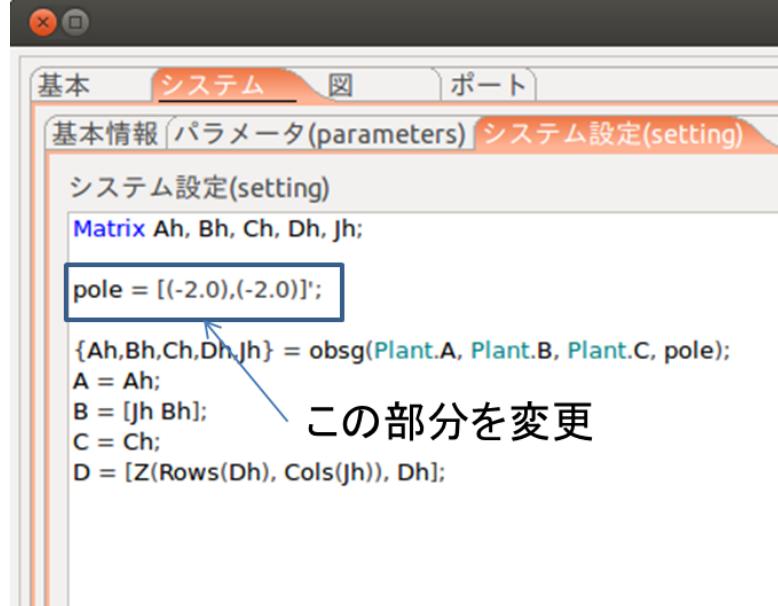


図 28: ContinuousObserver の編集

2.4 シミュレーション

ツールバーにある「再生ボタン」をクリックする、あるいは、メニューバーにある「シミュレーション」メニューから「シミュレーション開始」を選択することでシミュレーションを行うことができる。「シミュレーション」メニューから「シミュレーションパラメータの設定」を選択することで、シミュレーションの設定を変更できる。図 29 にシミュレーションに関するウィンドウの説明を示す。

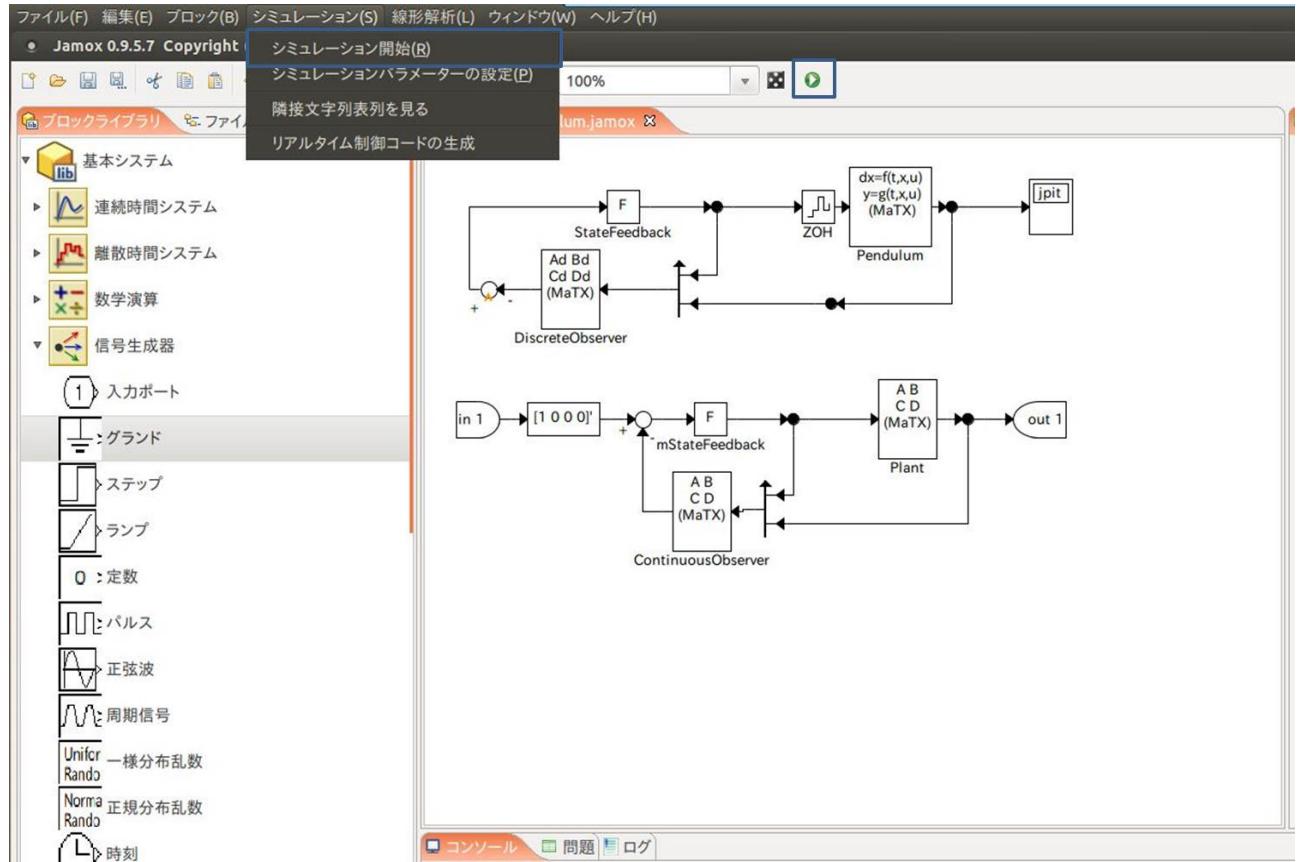


図 29: シミュレーションの実行方法

シミュレーションによりシステムの応答を確認したら、ブロック線図をファイルに保存しておこう。

2.5 制御実験

2.5.1 RealTime パースペクティブへの変更

図 30 に示すように、左上にあるアイコンをクリックすることでパースペクティブの候補を表示し、図 31 に示すように「other」を選択する。すると、図 32 に示すその他のパースペクティブの候補が表示されるので、「RealTime」を選択する。RealTime パースペクティブに変更されると、図 33 のようなレイアウトの画面が表示される。

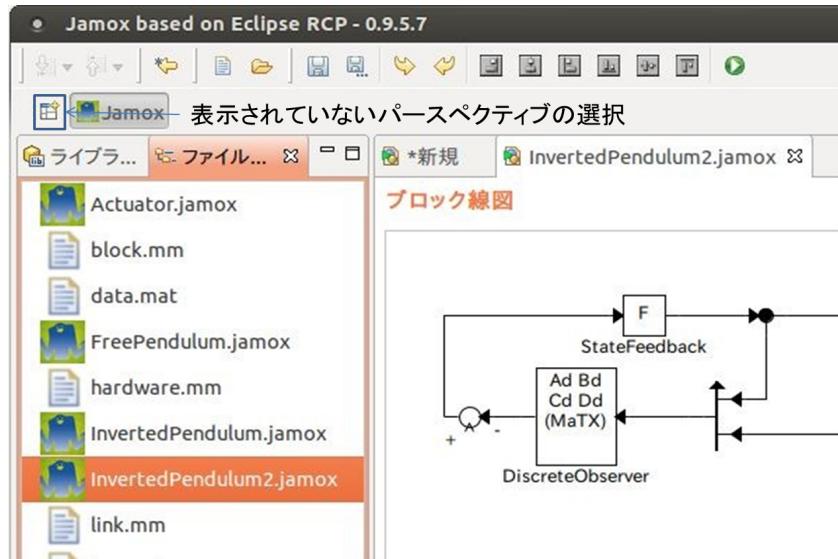


図 30: パースペクティブ選択のアイコン

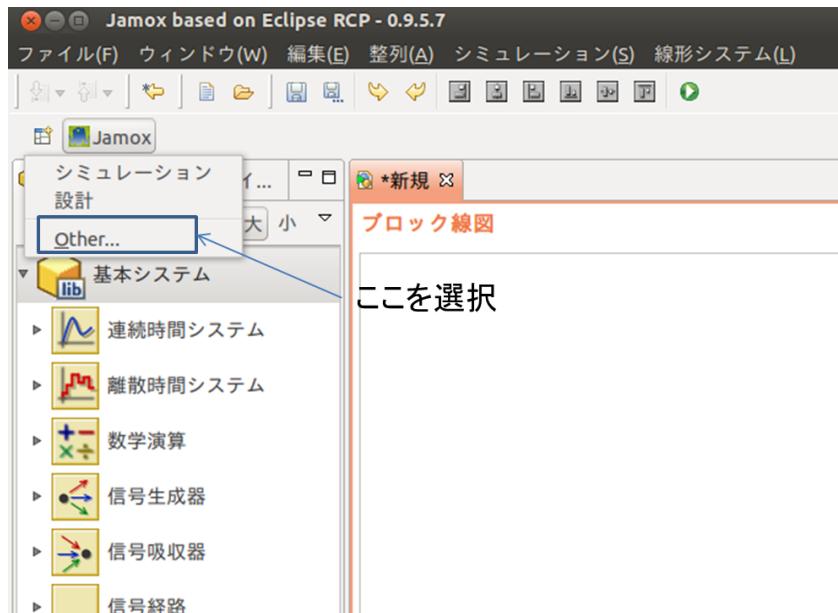


図 31: パースペクティブの選択



図 32: パースペクティブの変更

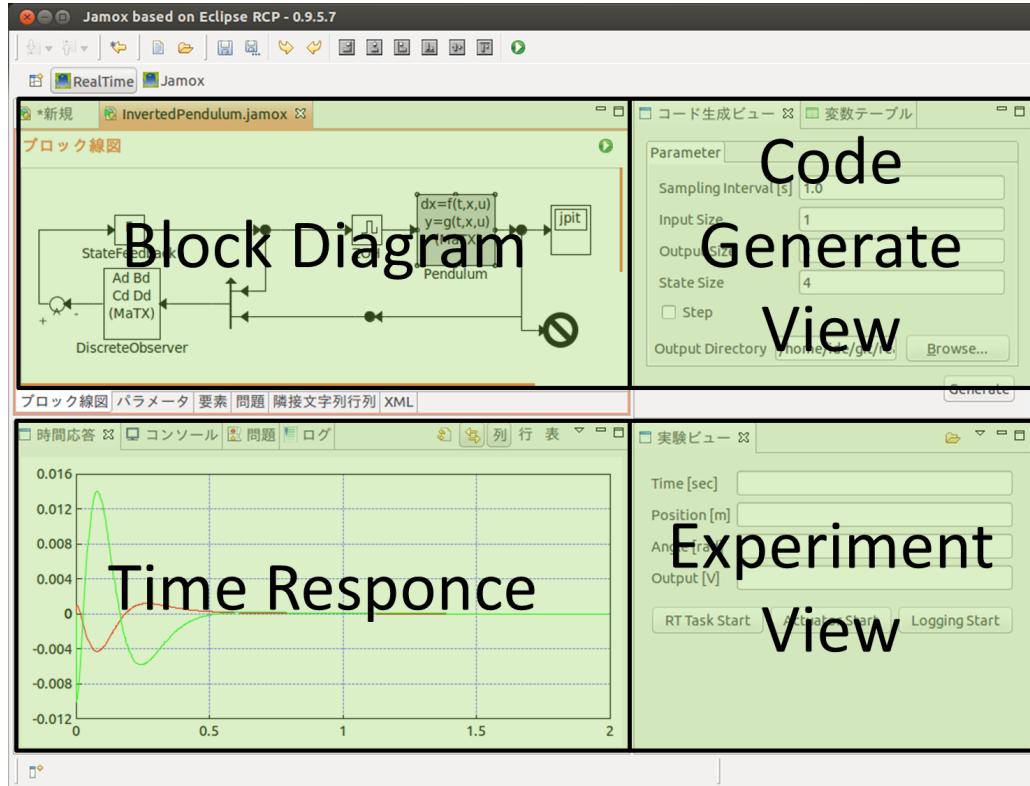


図 33: リアルタイムパースペクティブ

2.5.2 コード生成

リアルタイム制御実験ではワークスペース内にある InvertedPendulum.jamox を使用する。リアルタイム制御用コードの生成手順を以下に説明する。

- 図 26 のブロック線図の左側にあるコントローラの部分をドラッグして選択し、右クリックで表示されるメニューから「サブシステム化」を選択する。サブシステム化後のブロック線図を図 34 に示す。(注: サブシステム化したままのブロック線図を保存すると再度開くことができなくなるので注意。パラメータを変更した場合はサブシステム化する前に保存を行う。)

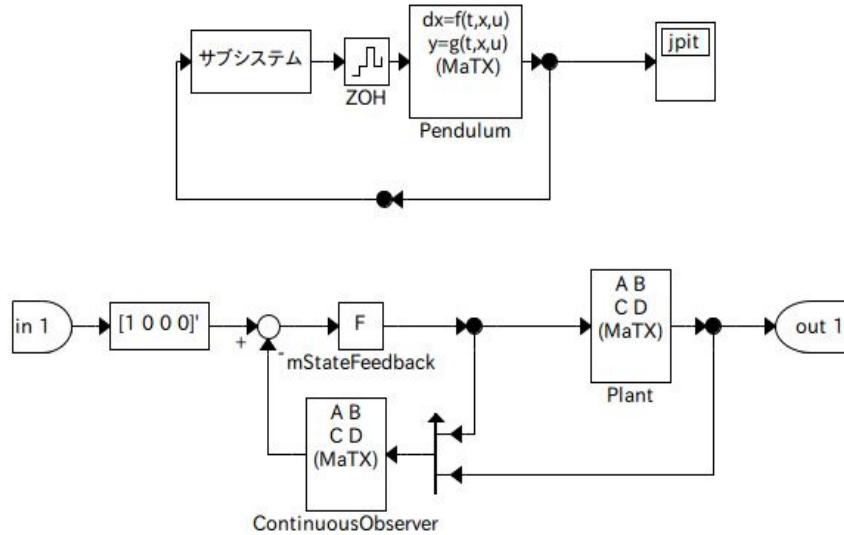


図 34: サブシステム化後のブロック線図

- コード生成を行うブロック（ここではサブシステムブロック）をクリックして選択し、コード生成ビューの「Generate」ボタンをクリックする。生成されるコードの保存先ディレクトリはデフォルトではワークスペースと同じである。保存先を変更する必要がある場合は「Browses...」ボタンをクリックしてディレクトリを変更する。

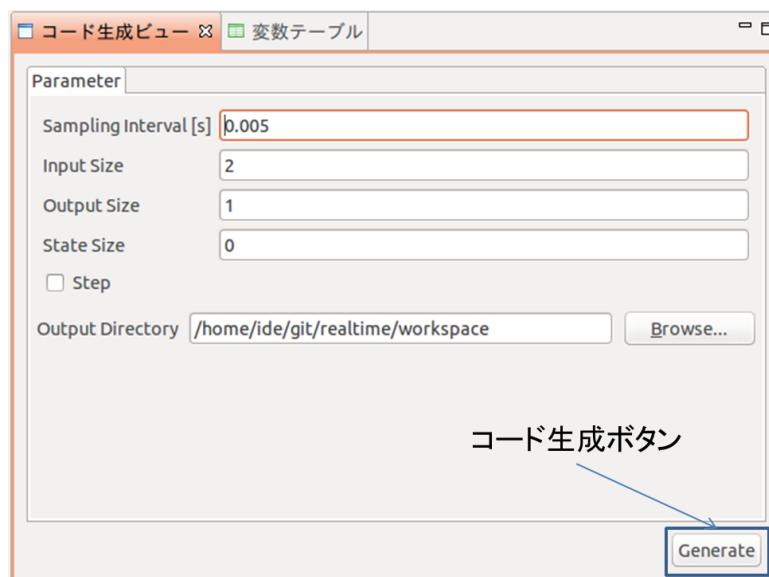


図 35: コード生成ビュー

2.5.3 リアルタイム制御実験

リアルタイム制御実験を行う手順を説明する。図 45 に実験を行う際の画面を示す。

1. 実験ビューの右上にあるフォルダアイコンをクリックし、フォルダ選択ダイアログを表示する。ダイアログから「/home/project/git/realtime/workspace/」以下にある生成したコード libExperiment*.so を選択する。(例：01/01 の AM11:00 の場合 libExperiment01011000.so)

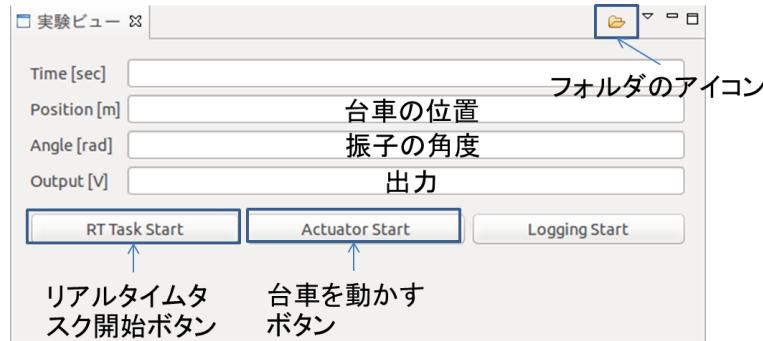


図 36: 生成コードの選択

2. ドライバの左側にあるアクチュエータ(モータ)電源のスイッチ SW1 が OFF であることを確認し、右側にある主電源のスイッチを ON にする。なお、実験を行わないときは必ずアクチュエータ(モータ)電源を OFF にしておく。ON の状態にしておくと勝手に動き出す恐れがある。

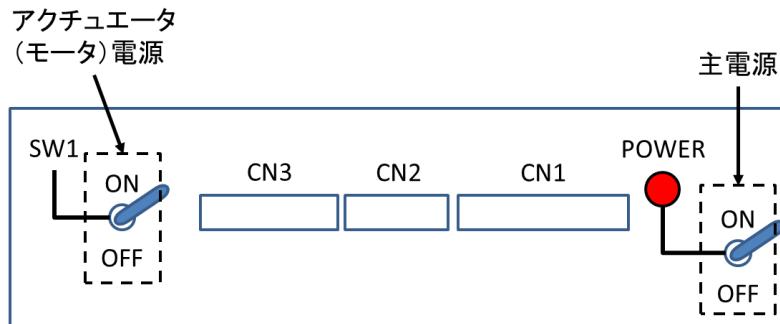


図 37: ドライバ

3. 実験ビューにある「RT Task Start」ボタンをクリックすると、台車の位置と振子の角度が実験ビューに表示され、サンプリング周期毎に更新され続ける。

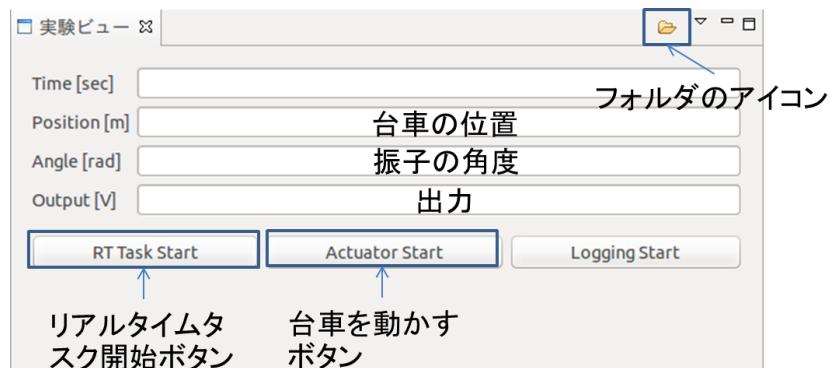


図 38: リアルタイム制御の開始

4. 台車の位置、振子の角度、アクチュエータへの電圧が適切な値であることを確認し、ドライバの左側にあるアクチュエータ(モータ)電源のスイッチ SW1 を ON にする。

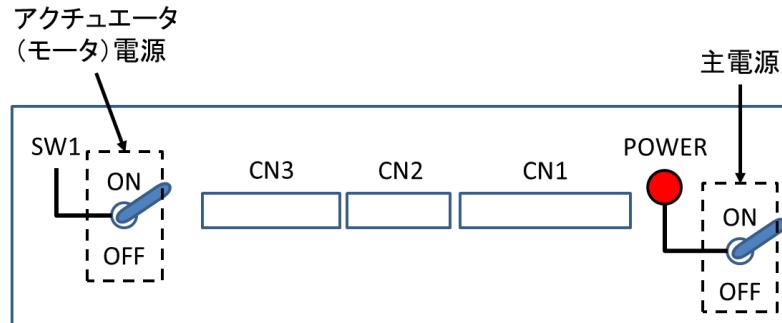


図 39: ドライバ

5. 実験ビューにある「Actuator Start」ボタンをクリックすると、台車が動き始める。同時に台車の位置、振子の角度、アクチュエータへの電圧の値の記録が開始する。

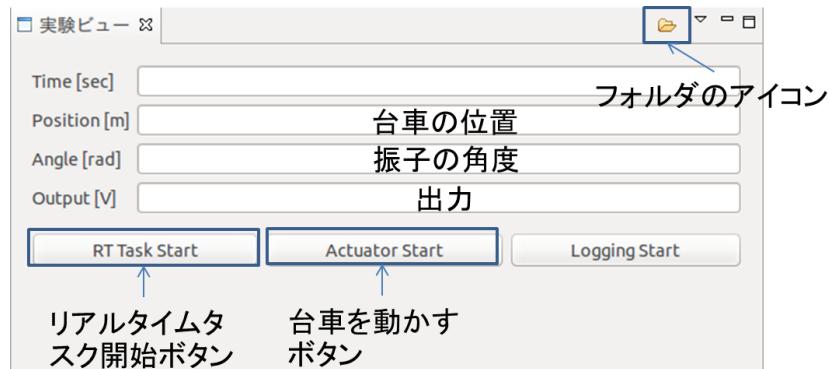


図 40: アクチュエータへの出力開始

6. 実験ビューにある「Actuator Stop」ボタンをクリックすると、台車が停止する。同時に台車の位置、振子の角度、アクチュエータへの電圧の値の記録が終了する。



図 41: アクチュエータ出力の終了

7. 実験ビューにある「RT Task Stop」ボタンをクリックすると、実験ビューに表示されている台車の位置と振子の角度の値の更新が終了する。同時に記録された値は Jamox 内の変数 actual として保存される。



図 42: リアルタイム制御の終了

8. ドライバの左側にあるアクチュエータ(モータ)電源のスイッチ SW1 を OFF にする。

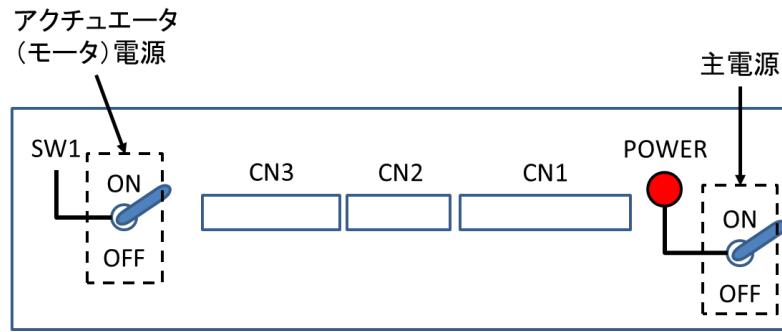


図 43: ドライバ

9. パースペクティブを「Realtime」から「Jamox」へ変更する。
10. 測定データのグラフを描画するため、ワークスペース内にある ShowData.jamox を読み込む。図 44 にブロック線図を示す。

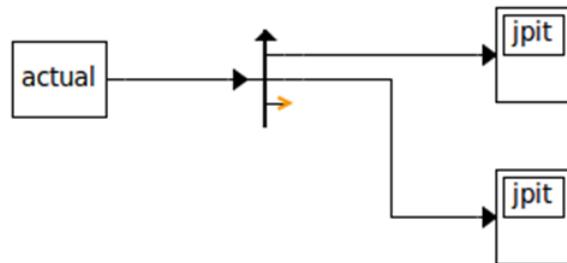


図 44: 測定データのグラフを描画するためのブロック線図

11. シミュレーション開始ボタンをクリックすると、測定データのグラフが描画される。

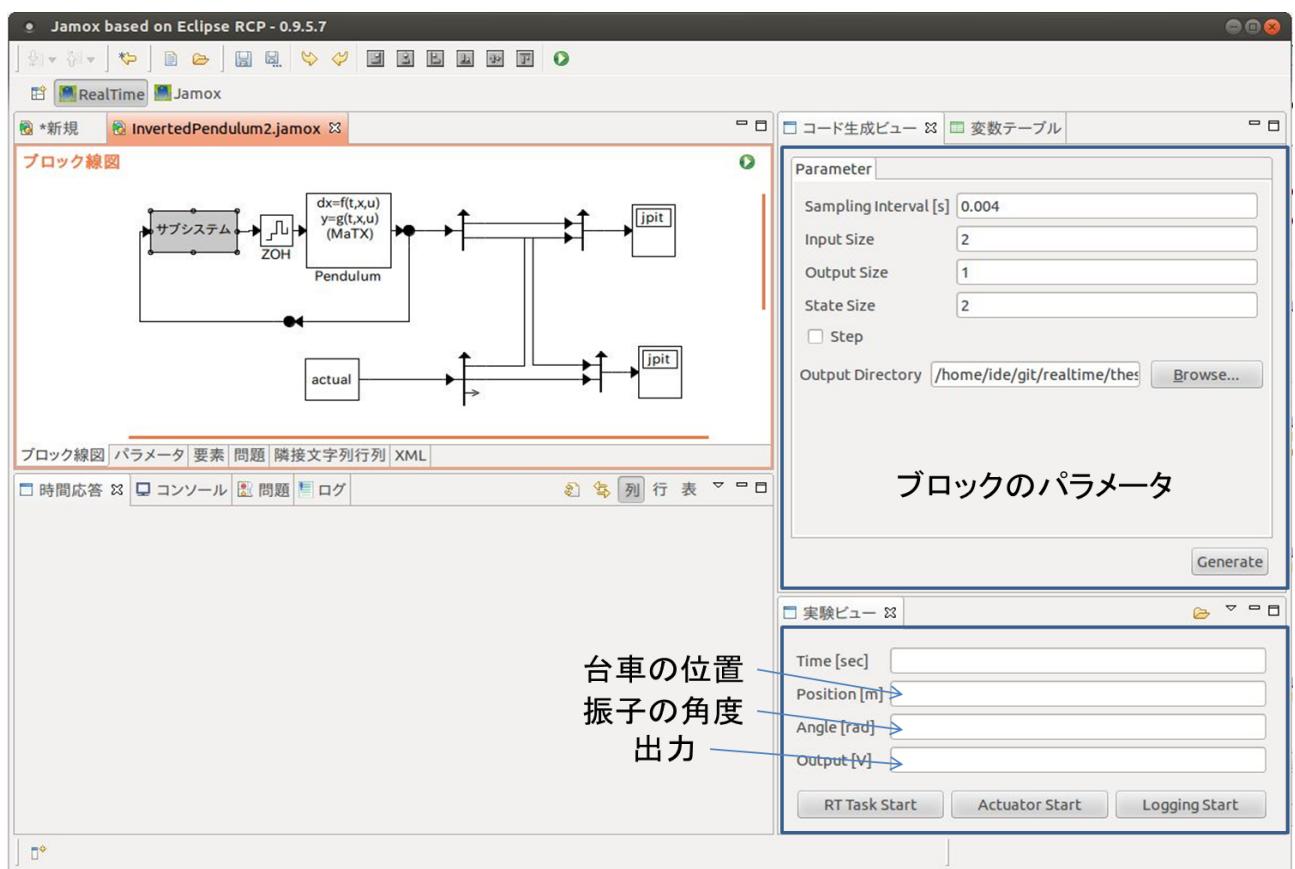


図 45: 実験を行う際の画面