

RTAI-Labシステムの構築

The construction of RTIA-Lab system

笠原 雅人, 笠原 拓也*¹

Masato KASAHARA, Takuya KASAHARA*¹

1 まえがき

現在普及している汎用のオペレーティングシステム (OS) は CPU を使える時間が通常, 不安定となっている. 一定のサンプリング間隔で計測と制御などを行う場合, データの読み込み, 書き込みにおいて時間ごとの割り込み処理を行わなければならない. 時間割り込みを行うにあたり, CPU を使える時間が不安定であるため, 読み飛ばしなどの不具合を生じることがある. この点から現在使用している OS の変更が必要となる.

CPU の使用時間が不安定であることに関して, リアルタイム OS を用いると CPU を使う時間に対し調整が行えるため, 一定間隔でデータの読み込み書き込み等の動作が保障される. リアルタイム OS として, RTAI(the RealTime Application Interface)-Lab を用いることとする. RTAI-Lab とは, リアルタイムソフトウェアと制御システム開発のためのツールであり, RTAI リアルタイム LinuxOS 上で, Scilab/Scicos で開発されたプログラムを実行することができるものであり, 計測と制御を行ううえで有効な OS である. しかしながら, LinuxOS 上のシステムであるため, リアルタイム OS を導入するにはコンピュータの環境に合わせ設定を行う必要があり, 試行錯誤的な面も多く存在する.

本報告では, リアルタイム OS の導入のための環境を具体的に示し, リアルタイム OS 環境下での実行例示すことを目的とする. 実行例を示すことで, リアルタイム OS の動作確認を行い, 実用性, 有効性を確認する.

2 環境

リアルタイム OS を導入するための環境としては相性が悪い場合があるため, 検討する必要がある. 今回, システムを導入する仕様は, つぎのように与える.

2.1 ハードウェア

主なハードウェア環境はこのようになっている.

- DELL OPTIPLEX 745
 - CPU Intel Core 2 Duo 1.86GHz
 - メインメモリ 2GB
- NATIONAL INSTRUMENTS 社製 SC-2075
 - BNC アナログ入力ポート 8ポート
 - BNC アナログ出力ポート 2ポート
 - BNC ユーザ定義入力ポート 2ポート
 - デジタル I/O ポート 8ポート

2.2 ソフトウェア

環境を整えるために用いたソフトウェア [1] は, 以下の通りである.

- OS Fedora8 x86_64
レッドハットが支援するコミュニティ「Fedora Project」によって開発されている, RPM 系 Linux ディストリビューション. Fedora については, 多くのバージョンが出ているが, 今回はハードウェアに最も適していた Fedora8 を選んでいる.
- Mesa Library[1], EFLTK Library[2]
OpenGL 互換な 3D グラフィックスライブラリである. 用意されている関数群も, ほとんど他家 OpenGL と互換性がある. UNIX 系システム上でも 3D グラフィックスを高速に動作させることができるライブラリである. また, X11 に対するライブラリである.

*¹ 平成 20 年 電子制御工学科卒業 (現筑波大学)

- Rtai-3.6.2[3]
Linux カーネルに適用されるパッチによるパッケージである。優先順位のつけられた、ハードリアルタイムタスクを実行することができるサブカーネルをはめ込む。RTAI-Lab は、MESA Library と EFLTK Library という 2 つのライブラリに依存しており、Rtai のカーネルのコンパイルする前にインストールする必要がある。
- Linux-2.6.23.14
Linux カーネルである。
- Comedi-0.7.76
Control and Measurement Interface の略で、制御と計測のためのインターフェースである。アナログ-デジタル変換器、デジタル-アナログ変換器、デジタル入出力、カウンターとタイマーを含む多種多様なデータ取得プラグインボードのためのドライバをサポートしている。今回は、NATIONAL INSTRUMENTS 社製の SC-2075 を使用しているため、Comedi に含まれている、in_pcmio というドライバを用いる。
- Comedilib-0.8.1
Comedi を使うためのライブラリである。
- Scilab-4.1.2
フランスの INRIA と ENPC で開発された数値計算システムである。数値計算機能以外に、信号処理、行列や多項式の数式処理、関数のグラフィック表示なども充実している。また Scicos により、ブロック線図を作成することでシミュレーションが実行可能である。
- Qt-4.4.3
Qrtailab をインストールするために、インストールしなくてはならない。Qt とは、ノルウェーのトロールテック社によって開発された C++ 言語で書かれた GUI ツールキットである。
- Qwt-5.1.1
Qrtailab をインストールするために、インストールしなくてはならない。Qwt とは、Qt GUI アプリケーションフレームワークへのグラフィックスの拡張をするものである。
- Qrtailab-0.1.6
RTAI-Lab に実装されているデジタルオシ

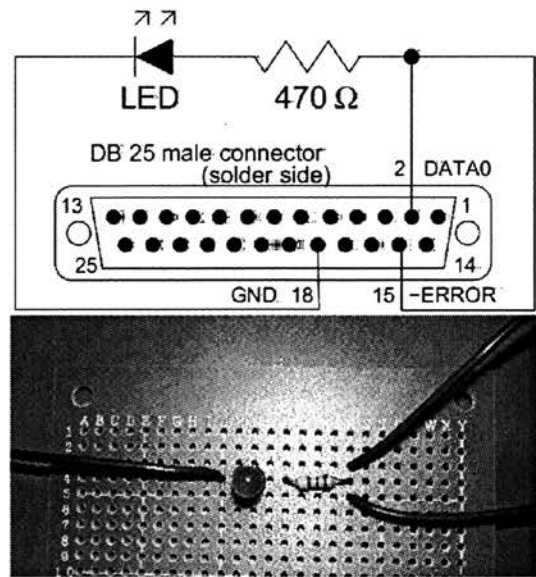


図1 作成した LED 点灯回路

ロスコープのような働きをするソフトウェア Xrtailab を、32bitOS の場合使うことが可能であるが、今回使用している 64bitOS の場合 Xrtailab を使うことが出来ないため、同様なソフトウェアの Qrtailab を別にインストールする必要がある。

システムはハードウェアに依存する点が多いため、ソフトウェアをインストールするには適切な設定が必要となる。設定方法に関しては、参考文献 [4] に従って行っている。

3 実行例

3.1 プリントポートによる LED 点灯

実行例 1 として、プリントポートを用いた LED 点灯により確認を行った。scicos で作成したプログラムを使い、プリントポートから 0-3V のパルスを出力し、作成した回路に入力する。作成した回路を図 1、Scicos のブロックを図 2 に示す。PC 上で表示されている信号と同タイミングで回路の LED が点灯していることを確認した。

3.2 ローパスフィルタの測定

実行例 2 としてローパスフィルタの周波数特性の測定を行った。ゲインと位相差について周波数に対しまとめ、ボード線図としてまとめる。今回は、図 3 に示すような抵抗 482 Ω 、コンデンサ 330 μF を用いて、カットオフ周波数 1Hz のローパスフィルタを作成した。入力信号 V_{in} として正弦波信号を与え、フィルタ回路の出力信号 V_{out} を取

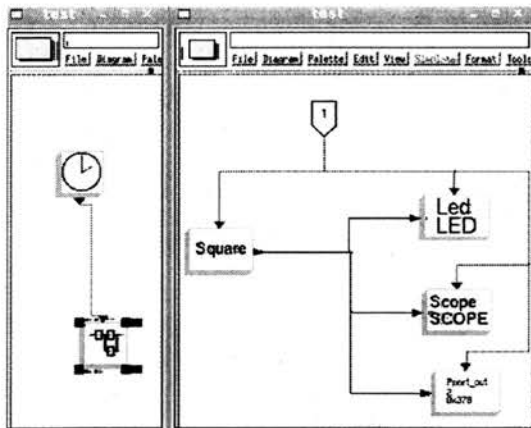


図2 実行例 3.1 のブロック線図

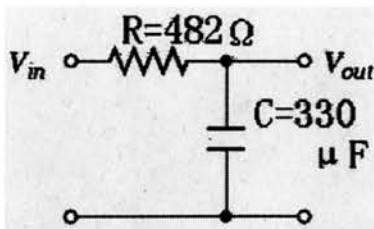


図3 RC ローパスフィルタ回路図

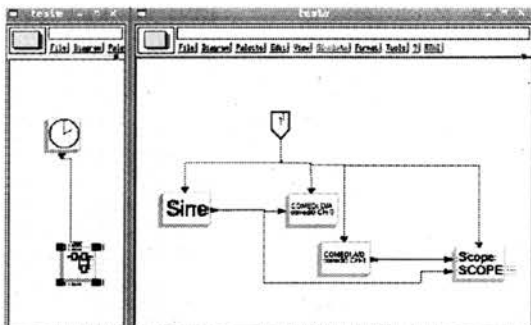


図4 実験 3.2 ブロック線図

り込む、Scicos による作成したブロック線図を図 4 に示す。また、図 5 のプロットは、PC から各周波数の正弦波 V_{in} を SC-2075 へ出力し、フィルタ回路を通して SC-2075、PC に入力された波形 V_{out} に基づきゲイン、位相差を求めたものである。

3.3 ARX モデルによるシステム同定

実行例 3 として、3.2 で用いた RC 回路のシステム同定を行った。正規乱数で与えられる二値信号 (0-5V) を入力とし、入出力の測定で得たデータを基に、ARX モデルでのシステム同定を行った。図 6 に Scicos に二値信号を与え出力信号を測定するブロック線図を示す。システム同定を行うに当たり、入出力データをそのまま使うのではなく、前処理として、まずアウトライアの除去のため、初めの 200 点を省く。また、残りの 800 点のデータの

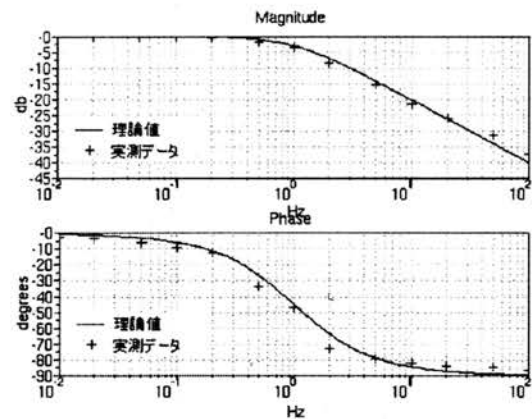


図5 カットオフ周波数 1Hz ローパスフィルタのボード線図

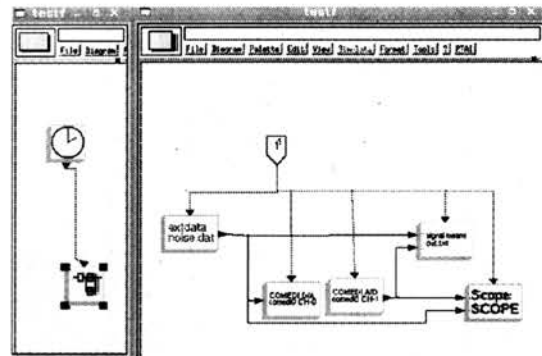


図6 実行例 3 のブロック線図

うち前半 400 点を同定用、後半 400 点を検証用と分け、同定用データにはトレンドの除去を行い平均値を 0 とする。

システム同定より得た伝達関数は、以下の通りである。

$$G(z) = \frac{0.4148 + 0.002z}{-0.5801 + z} \quad (1)$$

このモデルが適切であるかどうか検証するために、今回はステップ応答の比較と検証用データによる出力データの比較を行った。

図 7 にフィルタ回路のステップ応答を示す。実線がシステム同定により得たモデルのステップ応答、破線が実測より得た応答である。多少の誤差があるものの、ほぼ同じ応答をしているといえる。

図 8 にフィルタ回路の検証用データとの比較を示す。実線はシステム同定により得たモデルに、検証用の入力データを入力し、その出力をグラフにしたもの、破線は検証用の出力データによるものである。初めの部分にズレがあるが、過渡応答部分であると考えられるので、無視することができる。初めの部分が無視すると、その他の部分はほぼ一致しているといえる。

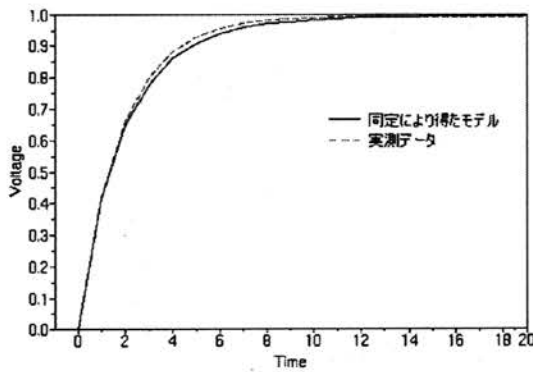


図7 ステップ応答比較

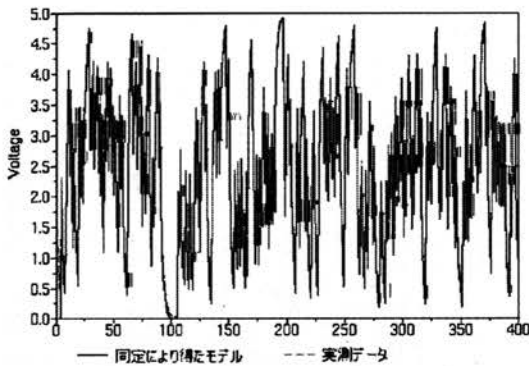


図8 モデリング妥当性の検証

参考文献

- [1] <http://www.mesa3d.org/>
- [2] <https://ede.svn.sourceforge.net/svnroot/ede/trunk/efitk>
- [3] <https://www.rtai.org/index.php>
- [4] Roberto Bucher, "RTAI-Lab tutorial : Scilab,Comedi,and real-time control"

小山工業高等専門学校 電子制御工学科
E-mail : kasahara@oyama-ct.ac.jp

「受理年月日 2009年9月30日」

4 おわりに

本報告では、RTAI-Lab システムを構成する上でのハードウェア、ソフトウェアの構成を示した。本システムを構成するに当たり、ハードウェアに依存する点が大きいため、ソフトウェアのバージョンなど適切に設定する必要がある。また、本システムの動作確認を3つの実行例により確認した。

今回、実行例を行うに前に、サンプリング周期を変化させ、システムの動作範囲に関して確認を行っている。サンプリング周期を $10\mu\text{s}$ に設定すると、コンピュータがフリーズしてしまう現象が起こった。この現象はコンピュータが処理に追いついていけなくなっていると考えられる。今回のシステムについての動作確認では、データの処理などの負荷をかけていない状態で行っているため、計算などの処理を付加する場合には $100\mu\text{s}$ 程度のサンプリングが限界であると考えられる。

サンプリングの確認および実行例により、RTAI-Lab システムの構築ができ、本システムは計測と制御を行う上で、十分に実用性、有効性を確認できた。