

Scilab/Scicos によるモデルベース開発に関する研究

苫小牧工業高等専門学校 ○ 杉本 大志, 吉村 斎, 阿部 司, 北海道立工業試験場 大村 功

要 旨

近年, 組み込みシステム開発の現場では, モデルベース開発の手法が広く使われるようになってきた. 本研究では, Scilab/Scicos を用いたモデルベース開発を提案する. 適用例として, 状態フィードバック制御による二輪型倒立振子の安定化制御を試みた. プラントのモデリング, 制御系の設計, および制御応答のシミュレーションを行った. また, 設計した制御系を自動的に LEGO Mindstorms NXT に実装し, 安定な実験結果を得る事で, 提案する方法の有用性を確認した.

1. 研究の目的

本研究では, LEGO MINDSTORMS NXT を用いた二輪型倒立振子 NXTway-GS(図 1) を実機として採用し, シミュレーション結果から, NXTway-GS 用の姿勢制御タスクを自動的に生成する事を目的とした. このために, 実機のモデリングと制御系の設計を行い, シミュレーションを行った. そして, シミュレーション結果から得たゲインをタスクに実装し, 実機を動作させた.

山本による先行研究^[1]では, 数値計算システム Matlab/Simulink によってシミュレーションを行った. Matlab は安定性が高いシステムであるが, その反面非常に高価であり, 拡張パッケージの入手にも有償である. 更に, 『数の定義』や演算子のオーバーロードなどができず, 『拡張性』に乏しいという欠点を持っている. 一方, 本研究で用いる Scilab/Scicos は, 無償で入手できるソフトウェアであり, 拡張パッケージも無償で配布されている. また, ユーザが独自に『数の定義』や, 演算子のオーバーロードなどができ, 『拡張性』に優れたシステムと言える.

この『優れた拡張性』という利点を持つ Scilab/Scicos が, モデルベース開発に適した環境であると考え, 本研究ではこれを採用した.

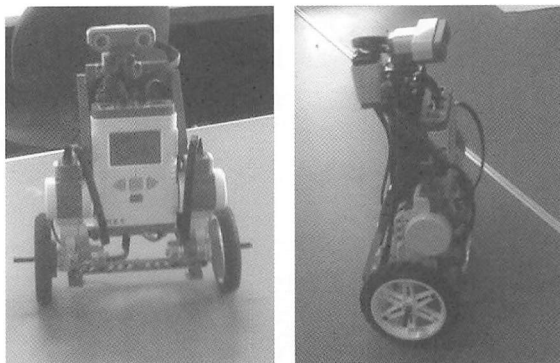


図 1 NXTway-GS

2. システム構成と処理の流れ

本研究で開発したシステムの機能は,

- (1) シミュレータ機能
- (2) 自動コード生成機能
- (3) ビルド・アップロード機能

の 3 つである. 図 2 に, システム構成と処理の流れを示す.

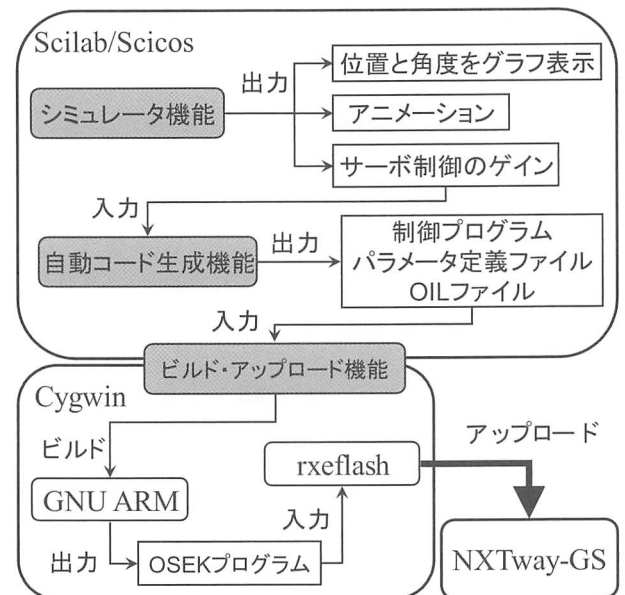


図 2 システム構成と処理の流れ

2.1 シミュレータ機能

プラントである NXTway-GS を, 二輪型倒立振子としてモデリングする. モデルの主要変数を表 1 に示す. ラグランジュ方程式をもとに, 鉛直近傍で線形化した運動方程式を立てると, 次式を得る.

$$\left[(2m + M)R^2 + 2J_w + 2n^2J_m \right] \ddot{\theta} + (MLR - 2n^2J_m) \ddot{\psi} = F_\theta \quad (1)$$

$$(MLR - 2n^2J_m) \ddot{\theta} + (ML^2 + J_\psi + 2n^2J_m) \ddot{\psi} - MgL\psi = F_\psi \quad (2)$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2} (J_w + n^2J_m) \right] \ddot{\phi} = F_\phi \quad (3)$$

以上の運動方程式を $\mathbf{x}_1 = [\theta \ \psi \ \dot{\theta} \ \dot{\psi}]^T$, $\mathbf{x}_2 = [\phi \ \dot{\phi}]^T$, $\mathbf{u} = [v_l \ v_r]^T$, $y = \theta$ とすることで

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u} \quad (4)$$

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u} \quad (5)$$

$$y = \mathbf{C} \mathbf{x}_1 \quad (6)$$

を得る.

この状態方程式を用いて, 状態フィードバックにより制御器を構成した. フィードバックゲインは最適レギュ

レータ法により設定した。構成したサーボ制御系を図3に示す。

表1 変数・定数の定義

定数	意味
g	重力加速度
m	駆動輪1本の質量
R	駆動輪半径
J_w	駆動輪慣性モーメント
M	車体質量
W	車体幅
L	車輪中心から車体重心までの距離
J_ψ	車体慣性モーメント(ピッチ)
J_ϕ	車体慣性モーメント(ヨー)
J_m	DCモータ慣性モーメント
n	ギア比
θ	駆動輪の平均回転角度
ψ	ピッチ角
ϕ	ヨー角
F_θ	θ に対応する一般化力
F_ψ	ψ に対応する一般化力
F_ϕ	ϕ に対応する一般化力

2.2 自動コード生成機能

自動コード生成機能は、

- (1) Cファイル
 - 制御パラメータを定義したファイル
 - バランス制御プログラム
 - タスクを記述したファイル
- (2) OIL(OSEK Implementation Language) ファイル
 - OSEK アプリケーションの設定を記述したファイル
- (3) make ファイル

を生成する。制御パラメータを定義したファイルには、

- シミュレータで用いたローパスフィルタのゲイン
- シミュレーション結果より得たサーボ制御の状態フィードバックゲイン
- PWM制御用バッテリー・ゲインとオフセット値

を記述する。バランス制御プログラムには、Scicosで構成した制御則(図3)を記述する。

2.3 ビルド・アップロード機能

make コマンドによって、OSEK プログラムを GNU ARM を用いてビルドする。そして、rxeflash コマンドによって、OSEK プログラムを NXTway-GS にアップロードする。

3. 検証

3.1 シミュレーションの応答

構成したサーボ制御系を図3に示す。図3の k_f, k_i は、最適レギュレータによって得たフィードバックゲインである。二輪型倒立振子の初期車体傾斜角を $\psi_0 = 4[\text{deg}]$ 、ジャイロ・キャリブレーションに要するむだ時間を $2[\text{msec}]$ とし、シミュレーションを $0 \leq t \leq 10 [\text{sec}]$ において実施した。図4は、システムの応答である。

初期状態から目標値をゼロにする状態の応答では、車体傾斜角がゼロの近傍で僅かに変化している。

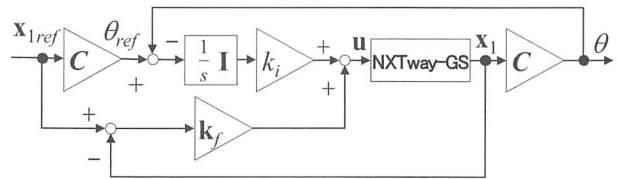


図3 NXTway-GS用サーボ制御器のブロック線図

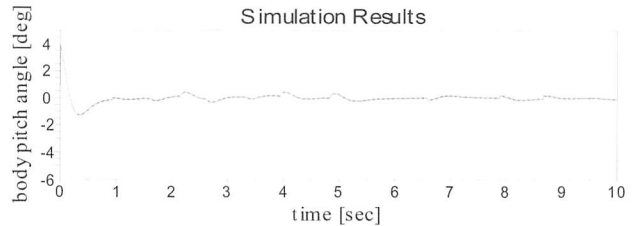


図4 シミュレーションの応答

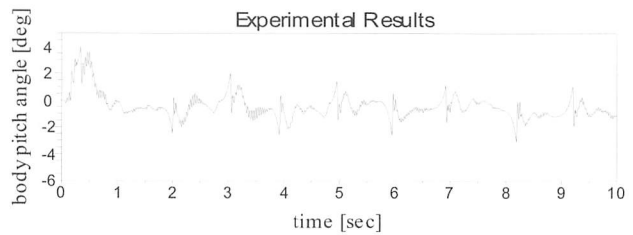


図5 実機の応答

3.2 実機の応答

シミュレーションで得たゲインを実機に実装し、倒立状態を維持させたときの応答を図5に示す。

図5のグラフから、図4のシミュレーションと同様に車体傾斜角が $\pm 2[\text{deg}]$ 以内に抑えられながら目標値のゼロに追従する安定な制御結果を得た。

ここで、図5における車体傾斜角は、ジャイロセンサの出力を数値積分する事により得た。

4. 結論

提案した Scilab/Scicos によるモデルベース開発手法を適用する事で、制御系の設計から実機への実装を容易にし、さらに実機を安定して制御できる結果を得た。

実機においては、シミュレーションで得たチューニングパラメータを実装する事で、姿勢の維持が可能であることを示した。

以上の結果から、提案した Scilab/Scicos によるモデルベース開発手法の有用性を示した。

今後の課題は、

- (1) 実機とシミュレーションとの誤差の最小化
- (2) 3次元CADによる形状モデルの作成とパラメータの計算

を実現する事である。

参考文献

- [1] Y.Yamamoto, "NXTway-GS Model-Based Design - Control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT-", CYBERNET SYSTEMS CO.,LTD. , 2008.