計測自動制御学会東北支部 第 257 回研究集会 (2010.5.31)
資料番号 257-9

倒立二輪ロボット制御のモデルベース設計組み込みシステム

Embedded System of Model Based Design for Wheeled Inverted Pendulum

○阿部倫太郎 *，佐藤宏明 **，恒川佳隆 **，長田洋 **

○ ABE Rintaro*, SATO Hiroaki**, TSUNEKAWA Yositaka**, OSADA Hiroshi**

* 岩手大学大学院,** 岩手大学

*Iwate University graduate school,**Iwate University

キーワード： モデルベース設計 (Model Based Design)，倒立二輪ロボット (Two Wheel Inverted Pendulum Robot)，LEGO MINDSTORMS NXT，組み込みシステム (Embedded System)，MATLAB/Simulink

連絡先： 〒 020-8551　盛岡市上田 4-3-5 岩手大学大学院工学研究科電気電子工学専攻　恒川・佐藤研究室
阿部倫太郎，E-mail: t2309005@iwate-u.ac.jp
佐藤宏明，Tel.: (019)621-6392，Fax.: (019)621-6392，E-mail: hsato@iwate-u.ac.jp

---

1. はじめに

現在，ほとんどの製品は組み込み制御によって動作している．組込み制御とは，産業機器や家電製品などに内臓されるマイクロコンピュータ (以下マイコンとする) によって特定機能を実現することを言い，パーソナルコンピュータなどの汎用的なコンピュータとは異なり，要求される機能や性能が特化されている点が特徴である．組込み制御が搭載されている身近なシステムとして，デジタルカメラ，携帯電話，自動車などをあげることができる．

製品の便利性や品質の向上に伴いひとつの製品に数十個のマイコンが使用され，マイコンに取り込むプログラムも複雑化している．そのため製品を設計する工程においてソフトウェアの占めるコストの，または時間の割合は増化している．ソフトウェアをいかに再利用し，いかに透明化するかが課題となる．そこで注目されているのがモデルベース開発である．

モデルベース開発とは，モデルを使用しシステムを構築し，そのモデルシステムをソフトウェアでコード自動生成することによりソフトウェアを生成する手法である．これにより再利用しやすく設計仕様との高い整合性をもったソフトウェアが出来る．

本稿では Matlab/Simulink を用いたモデルベース設計に基づく組み込み制御による倒立二輪ロボットの倒立制御と走行制御の検討について述べる．

2. 組み込みシステム

本研究で用いられる組み込みシステムは LEGO 社製の LEGO MINDSTORMS NXT である．コンピュータブロックである NXT にはフラッシュメモリが内蔵されており，ここにダウンロードされたプログラムによって自律行動をさせることができる．CPU に 32bit の ARM7 と 8bit の AVR を搭載し，USB または Bluetooth で外部の

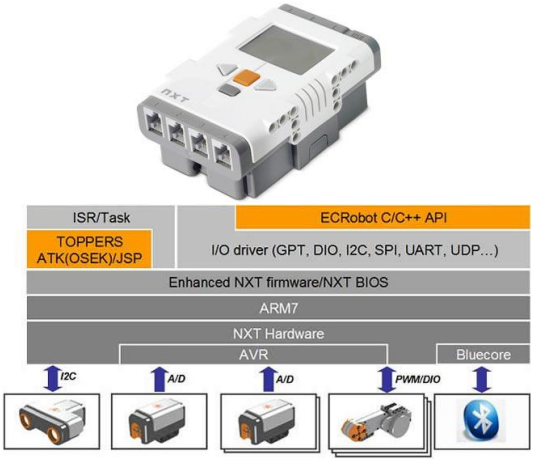

Fig. 1　LEGO MINDSTORMS NXT

PC や他の NXT と通信することができる．NXT には回転角度を検出できるサーボモータやタッチセンサ，超音波センサ，光センサ，HiTechnic 社製ジャイロセンサなどが接続できる．NXT の概観図を Fig.1 に示す．

3.　NXTway-GS 倒立二輪ロボット

3.1　倒立二輪ロボットのモデリング

NXTway-GS は NXT による倒立二輪ロボットで，NXT の左右にサーボモータ，足元に光センサ，背面にジャイロセンサを搭載したものとした．NXTway-GS の概観図を Fig.2 に示す．

Fig. 2　NXTway-GS 図

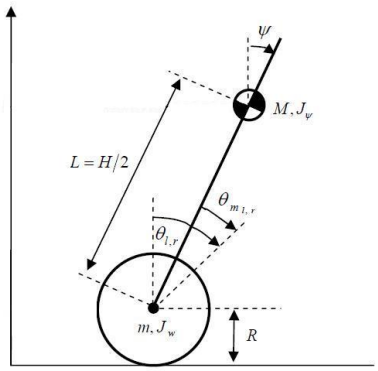

Fig. 3　倒立二輪ロボットのモデリング

3.2　倒立二輪ロボットの運動方程式

NXTway-GS の物理パラメータは Table 1 であり，これに基づき運動方程式を導出する，ここで倒立二輪ロボットのモデリングを Fig.3 のように定める．左右車輪の回転角度を $\theta_{l,r}$，車輪の平均回転角度を θ，左右 DC モータの回転角度を $\theta_{m_{l,r}}$，車体の傾斜角度を ψ，車体の平面回転角度を ϕ とすると運動方程式は式 (1) から (3) のようになる．

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

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

$$[(1/2)(W^2)+J_\phi+(W^2/R^2)(J_w+n^2J_m) + ML^2\sin^2\psi]\ddot{\phi} + 2ML^2\dot{\psi}\dot{\phi}\sin\psi\cos\psi = F_\phi \tag{3}$$

車体の傾斜角度が十分小さいとして $\psi \to 0$ の極限をとり $\sin\psi \to \psi, \cos\psi \to 1$ とし，ψ^2 等 2 次の項を無視する，運動方程式の線形化を行うと運動方程式 (1) から (3) は次の式 (4) から (6) となる．

$$\left[(2m+M)R^2+2J_w+2n^2J_m\right]\ddot{\theta} + (MLR\cos\phi - 2n^2J_m)\ddot{\psi} = F_\theta \tag{4}$$

Table 1　NXTway-GS の物理パラメータ

g=9.98	[m/sec^2]	重力加速度
m=0.03	[Kg]	車輪質量
R=0.04	[m]	車輪半径
J_w=mR^2/2	[Kgm^2]	車輪慣性モーメント
M=0.6	[Kg]	車体質量
W=0.14	[m]	車体幅
D=0.04	[m]	車体奥行き
H=0.144	[m]	車体高さ
L=H/2	[m]	車輪中心から車体重心距離
J_ψ=ML^2/3	[Kgm^2]	車体慣性モーメント
J_ϕ=M(W^2+D^2)/12	[Kgm^2]	車体慣性モーメント
J_m=1×10^{-5}	[Kgm^2]	モータ慣性モーメント
R_m=6.69	[Ω]	モータ抵抗
K_b=0.468	[Vsec/rad]	モータ逆起電力定数
K_t=0.317	[Nm/A]	モータトルク定数
n=1		ギヤレシオ
f_m=0.0022		車体とモータ間の摩擦係数
f_W=0		車輪と路面間の摩擦係数

$$(MLR - 2n^2J_m)\ddot{\theta} + (ML^2 + J_\psi + 2n^2J_m)\psi - MgL\psi = F_\psi \quad (5)$$

$$[(1/2)(mW^2) + J_\phi + (W^2/R^2)(J_w + n^2J_m) + ML^2\sin^2\psi]\ddot{\phi} = F_\phi \quad (6)$$

3.3　倒立二輪ロボットの状態方程式

運動方程式 (4) から (6) より状態量を x_1, 入力を u として現代制御理論における倒立二輪ロボットの状態方程式を求めると

$$x_1 = A_1x_1 + B_1u \quad (7)$$

が得られる，ここで，$x_1 = [\theta, \psi, \dot{\theta}, \dot{\psi}]^T$, $u = [v_l, v_r]^T$, x^T は x の転置，$v_{l,r}$ は DC モータ電圧であり，行列 A，B は以下のようになる．

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_1(3,2) & A_1(3,3) & A_1(3,4) \\ 0 & A_1(4,2) & A_1(4,3) & A_1(4,4) \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_1(3) & B_1(4) \\ B_1(3) & B_1(4) \end{bmatrix}$$

$A_1(3,2) = -gMLE(1,2)/\det(E)$
$A_1(4,2) = gMLE(1,1)/\det(E)$
$A_1(3,3) = -2[(\beta+f\_w)E(2,2)+\beta E(1,2)]/\det(E)$
$A_1(4,3) = 2[(\beta+f\_w)E(1,2)+\beta E(2,2)]/\det(E)$
$A_1(3,4) = 2\beta[E(2,2) + E(1,2)]/\det(E)$
$A_1(4,4) = -2\beta[E(1,1) + E(1,2)]/\det(E)$
$B_1(3) = \alpha[E(2,2) + E(1,2)]/\det(E)$
$B_1(4) = -\alpha[E(1,1) + E(1,2)]/\det(E)$
$\det(E) = E(1,1)E(2,2) = E(1,2)$
$E = (1,1) = (2m + M)R^2 + 2J_w + 2n^2J_m$
$E = (1,2) = E(2,1) = MLR - 2n^2J_m$
$E = (2,2) = ML^2 + J_\psi + 2n^2J_m$
$\alpha = nK_t/R_m$
$\beta = (nK_tK_b/R_m) + f_m$

4.　モデルベース設計によるモデリング

4.1　Matlab/Simulink でのモデリング

運動方程式, 状態方程式をもとに Matlab/Simulink で NXTway-GS をモデルリングした．Matlab/Simulink はブロック線図記述によるモデリング/シミュレーションソフトウェアで，モデルにおける接続線 (信号線) がデータの流れを示し，各ブロックは演算や処理のまとまりを意味する．また，モデルから Real-Time Workshop Embedded Coder という C コード生成ツールを用いることでコードを自動生成することができる．モデリング/シミュレーションの例を Fig.4 に示す．Fig.4 は 2 階線形微分方程式を Matlab/Simu-link で表現したものと，単位ステップ入力に対する応答のシミュレーション結果である．

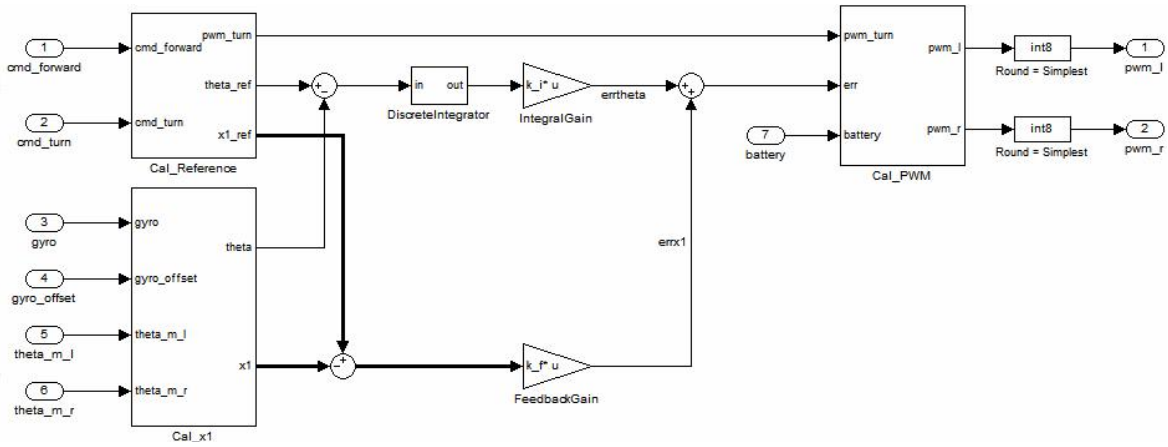

Fig. 5　倒立制御部のブロック図

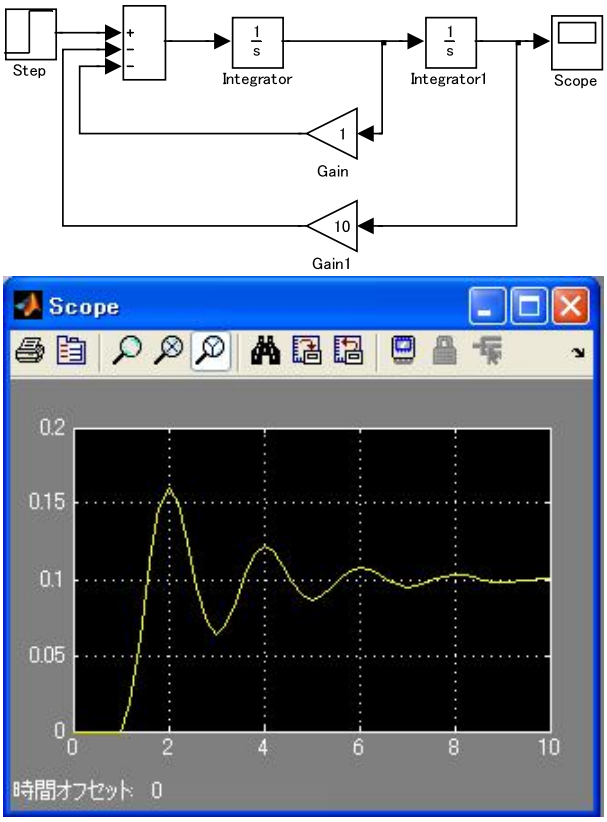

Fig. 4　モデリング/シミュレーション例

4.2　二輪型倒立振子の倒立制御

Fig.5 は倒立制御の Simulink モデルの一部分で，Cal_x1 ブロックは倒立二輪ロボットの状態量 (車輪の平均回転角度 θ，車体の傾斜角度 ψ，車輪の平均回転角速度 $\dot{\theta}$，車体の傾斜角速度 $\dot{\psi}$) の演算部となり，Cal_Reference ブロックは倒立制御の目標値の演算部で $\psi = 0, \dot{\psi} = 0$ としている．k_i^*u と表記されたブロックは，PID 制御の I 制御係数に相当するサーボ制御用積分フィードバック係数の乗算処理ブロックである，k_f^*u と表記されたブロックは，PID 制御の PD 制御係数に相当する状態フィードバック係数の乗算処理ブロックである．

4.3　倒立二輪ロボットの走行制御

NXTway-GS の自律走行制御では白地に黒のラインをトレースする．初めは NXTway-GS はラインの進行方向に向かって左のエッヂを走行し，センサ値が白色のときは右折，黒色のときは左折するようなアルゴリズムを考えた．この様子を Fig.6 に示す．NXTway-GS の光センサは Fig.7

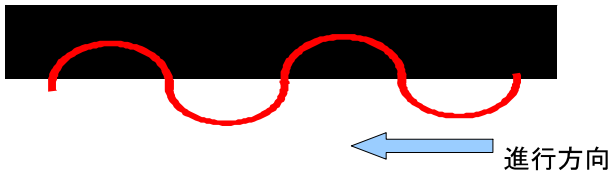

Fig. 6　走行ライン

のようなセンサから照射されたスポット光の反射値でラインの識別を行っている．スポット光がライン上，ライン外にあればそれぞれ黒色，白色と判断できるが，ラインの境界が光センサのス

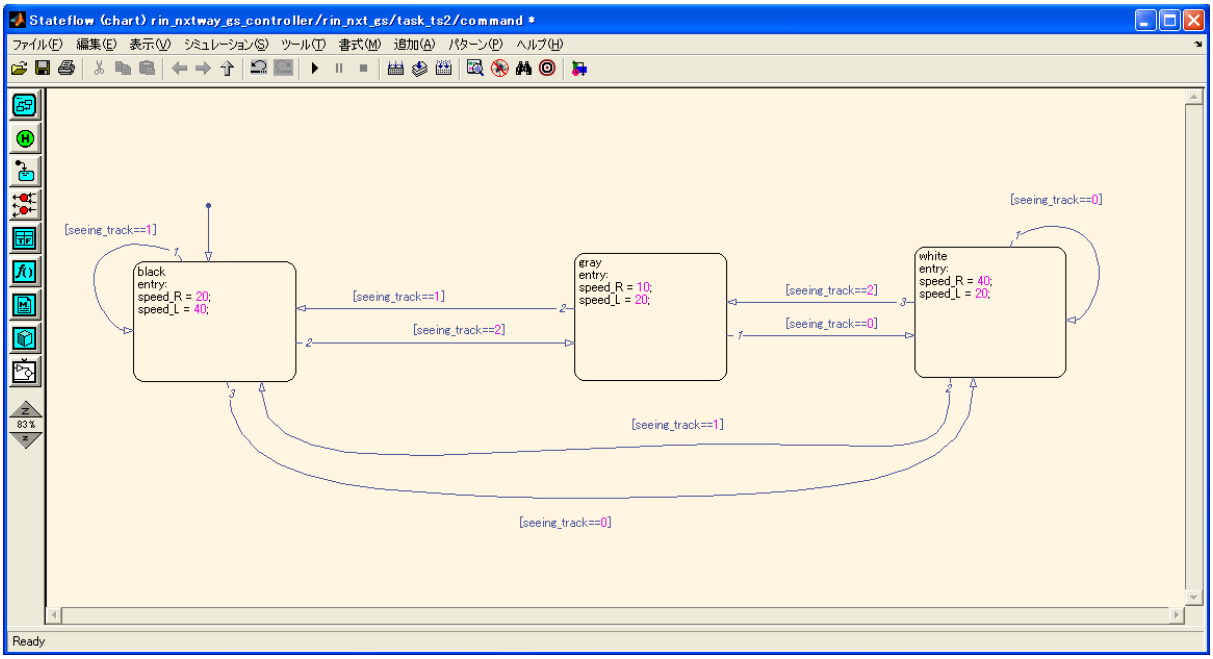

Fig. 10　stateflow ブロックによる記述

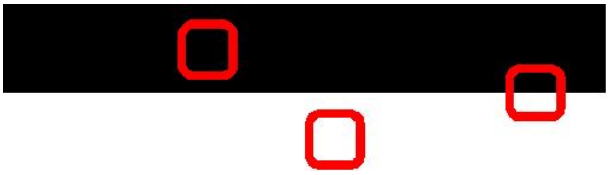

Fig. 7　光センサの色識別

```
if(センサ値が白色のとき){
        /*右折*/}
else if(センサ値が黒色のとき){
        /*左折*/}
else (センサ値が灰色のとき){
        /*左折*/}
```

Fig. 9　状態遷移の擬似プログラム

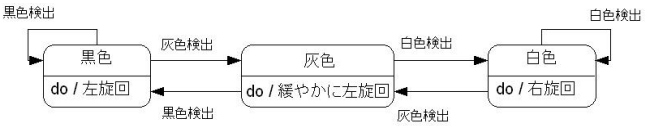

Fig. 8　状態遷移図

ポット光の中心にあると黒色の値と白色の値の平均値が返ってくることになり黒色でも白色でもない中間色 (灰色) と判断される．ライン対して NXT way-GS が直角に近く進入してもラインを突き抜けることがないようにするためには，完全に黒色と判断してから左折するのではなくラインの境界から左折しなければならない．そこで NXTway-GS の走行制御の基本動作アルゴリズムを Fig.8 の状態遷移図および Fig.9 の疑似プログラムのようにした．Fig.10 は Matlab/Simulink で使用できる Stateflow ブロックを使い Fig.8 の状態遷移図を記述したものである．Stateflow ブロックは状態遷移図と制御フロー図の組み合わせをベースとしたイベントドリブンシステムをモデル化し，シミュレーションするためのツールである．Fig.10 において“ seeing_track==1 ”とは黒色検出を表しており，同様に“ ==2 ”は灰色，“ ==0 ”は白色の検出を表す．

5.　倒立制御と走行制御のマルチタスク化

NXTway-GS の走行と倒立の制御を Fig.11 の擬似プログラムのように一つの周期で処理する

```
while(1)
{
    /*倒立制御*/
    /*走行制御*/
}
```

Fig. 11　倒立制御と走行制御の擬似プログラム

ようにした場合 走行制御が複雑になり処理に時間がかかってしまうと安定して倒立制御ができないと考えた．そこで倒立制御と走行制御のそれぞれを一つの周期で処理するのではなく，別々のタスク（マルチタスク）で処理する方法を考えた．Fig.12 の ExpFcnCallsScheduler ブロックを用いてそれぞれの制御をタスク化した，Fig.13 の task_1 が倒立制御，task_2 が走行制御となっている．このときの倒立制御の周期が 4[ms]，走行制御の周期が 20[ms] となるように設定した．

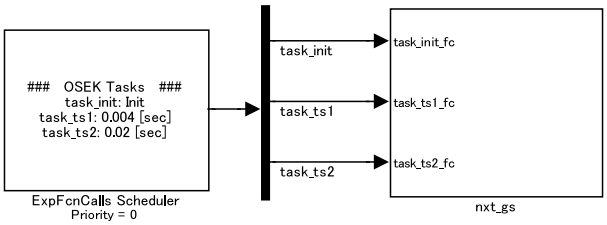

Fig. 12　ExpFcnCallsScheeduler ブロックとマルチタスクの呼び出し

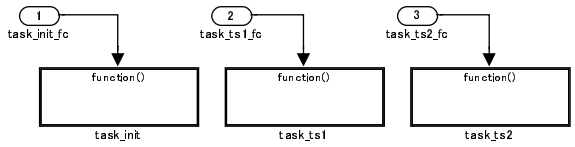

Fig. 13　nxt_gs ブロック内の倒立制御と走行制御のタスク

また，Fig.14 に Matlab/Simulink によるシミュレーション結果を示す．Fig.14 は 200 × 200[cm] の仮想空間の中心に置かれた NXTway-GS の重心位置の軌跡である．起動後は倒立制御のみを行い一定時間後に走行制御が行われていることが確認できる．

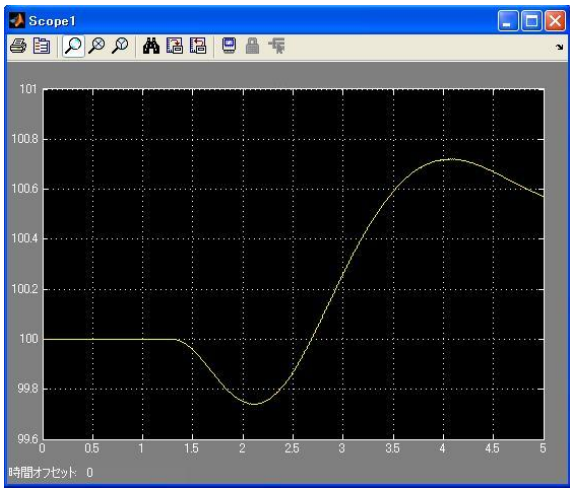

Fig. 14　NXTway-GS の重心位置

6.　まとめ

本稿では Matlab/Simulink を用いたモデルベース設計による組み込み制御の検討を行った．NXT を倒立二輪ロボット NXTway-GS として Simulink モデルで表現し，倒立制御と走行制御の検討を行い，シミュレーションを行った．今後は仮想空間のラインの作成，Real-Time Workshop Embedded Coder で C コードを自動生成し，そのプログラムを NXT に実装し実験を行う予定である．

参考文献

1) ET ロボコン実行委員会: ロボットレースによる組み込み技術者養成講座，毎日コミュニケーションズ (2008)

2) 大庭 慎一郎: 入門 LEGO MINDSTORMS NXT，ソフトバンククリエイティブ (2006)

3) 山 本 順 久:NXTway-GS　(Self-Balancing Two-Wheeled Robot), Controller Design, http://www.mathworks.com/matlabcentral /fileexchange/19147(2009)