

MATLAB/simulink を用いた倒立モデル Puppy の ためのモデルベース開発教育システム

田中 武*・大村 道郎**・中野 隆司***・有馬 仁志****

(平成23年10月31日受付)

Model base development education system for inverted model Puppy using MATLAB/simulink

Takeshi TANAKA, Michiroh OHMURA, Ryuji NAKANO and Hitoshi ARIMA

(Received Oct. 31, 2011)

Abstract

Hiroshima Institute of Technology has been teaching integrated circuit designs and semiconductor processes since 1989. We made the lecture system about the fabrication of pMOS and CMOS FET using semiconductor processing and testing apparatus in Kitakyushu Science and Research Park for students. The education system was constructed from physical model to demonstration of electronic control using two-wheeler inverted model.

Key Words: physical model, MATLAB, simulink, model based design, electronic control

1. まえがき

近年、集積回路技術のめざましい進歩により、集積回路は、高速化、高性能化、および高集積化される中で、大学における LSI 教育、およびその応用である電子制御技術教育の重要性が指摘されている。本学においても、平成元年度に簡単な集積回路の設計およびプロセス教育のできる「集積回路教育実習システム」が文部省教育装置助成で設置され、その実習内容を報告した¹⁾。その後、平成10年度に日本私立学校振興・共済事業団の「特色のある教育研究の推進」に採択され、集積回路関連設計・プロセス工学教育システムの開発を行ってきた²⁾。

今後の集積回路の高集積化、高密度化、および高性能化に伴い、半導体プロセス技術の発展が目覚ましい。その為、システム設計、回路設計、デバイス設計、半導体プロセス工学までを統合した知識・技術、および、それらの応

用と体験が必要になる。

プリント基板から製作し、システム LSI を設計・実装し、動作確認と同時に、同様な内容を e-learning することにより理解を深め、ものづくりと e-learning システムの両立する高度なシステムの構築するために、北九州学術研究都市の共同研究開発センター³⁾ で用いられている pMOS および CMOSFET プロセスを、システム LSI 関連ものづくり教育に取り込むために初歩的なシステム LSI の作成を試みた⁴⁾。

本研究では、同軸 2 輪倒立モデル（ここでは、(株)北斗電子製 Puppy）の物理モデルから、MATLAB/simulink モデルの構築、dSPACE のラピッドコントロールプロトタイピング、電子制御技術を用いて倒立モデルのモデルベース開発教育システムを構築したので報告する。

2. モデルベース開発教育システムの説明

現在、社会人を対象にしたモデルベース開発教育システ

* 広島工業大学工学部電子情報工学科

** 広島工業大学工学部電気システム工学科

*** (株)北斗電子

**** dSPACE Japan (株)

表1 モデルベース開発教育システムの構築

	1コマ目	2コマ目
1週目	はじめに	倒立モデル (Puppy) のための物理
2週目	マイコンの基礎と、倒立モデル (Puppy) のためのマイコンプログラム	MATLAB/simulink の説明
3週目	倒立モデル (Puppy) のための MATLAB/simulink プログラム	実習

この構築を表1に示す。週に1日2コマで構成を試みた。概要について、以下に説明する。

2-1. 「はじめに」

「はじめに」では、モデルベース開発を必要とする背景、従来の制御ソフトの開発プロセスと課題、モデルベース開発で実現できること、モデルベース開発の効果、dSPACEの特性およびインタフェースについての説明を行う。

2-1-1. モデルベース開発を必要とする背景

(1) カーエレクトロニクス化の進展

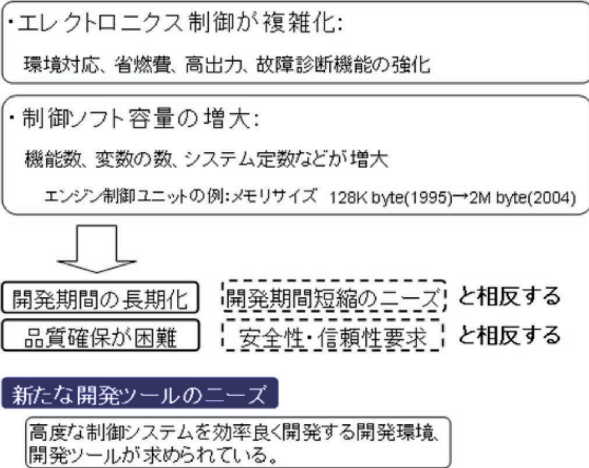


図1 カーエレクトロニクス化の進展

(2) ソフトウェア業界と自動車業界との共存

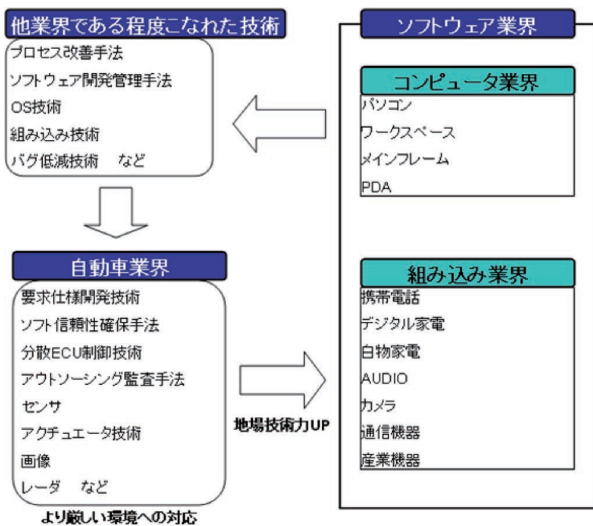


図2 ソフトウェア業界と自動車業界との共存

2-1-2. 従来の制御ソフトのプロセスと課題

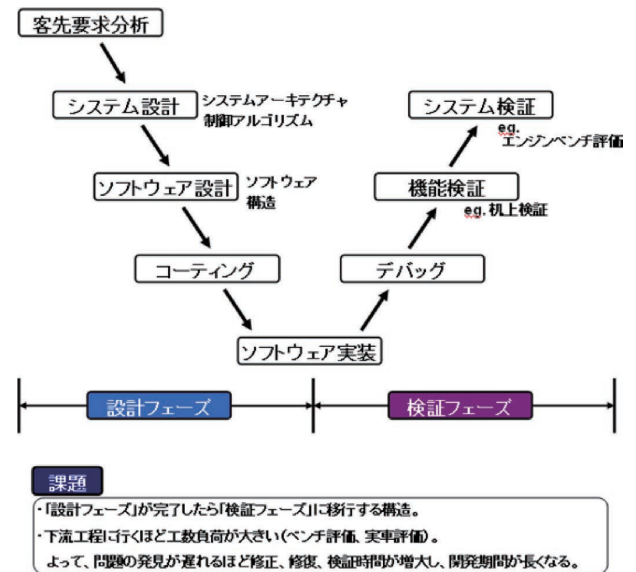


図3 従来の制御ソフトのプロセスと課題

2-1-3. モデルベース開発で実現できること

- 問題の早期発見と早期対応
 - 開発の初期段階から検証ループを細かく回すことが可能。
- 自動コード生成
 - 設計されたモデルを自動で、ソフトウェア (コード) に変換。
 - コミュニケーション、打ち込みミスなどの人間系エラーの発生を防止する。
- 仮想負荷 (バーチャル) による効率的な検証 HILS
 - 単体および複数のシステム評価を仮想負荷を用いて検証。

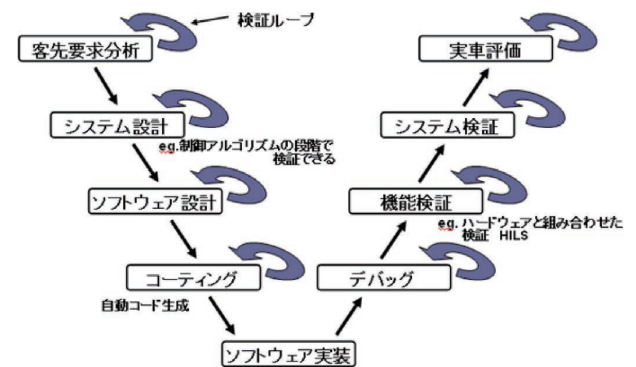


図4 モデルベース開発で実現できること

2-1-4. モデルベース開発の効果

- 効率化のための前提条件
 - モデルベース開発で、開発効率を上げるには、標準化が良い。
 - 最初にモデルを作る時点では、当然、従来の開発より工数がかかる。

- ・モデルやインタフェースの標準化が重要。
- ・条件を網羅したテストシナリオが必要。

(2) モデルベース開発の効果予測

- ・標準化がうまくいったと仮定した場合の効果予測。約30%工数削減。

2-2. 「倒立モデル (Puppy) のための物理」

「倒立モデル (Puppy) のための物理」では、同軸2輪車型倒立振子 (ここでは、(株)北斗電子製 Puppy) の動作原理、Puppy のマイコンのプログラム変更に用いた関数について説明を行う。

構造的に不安定な Puppy を倒立させる制御の状態方程式について述べる。この制御を行うためには、Puppy 本体の傾斜角、角速度、車輪の角速度等から車輪に適切なトルクを求め、そのトルク値を、モータを用いて発生させることにより、倒立振子の倒立状態を維持する。ここではサンプルプログラムでの Puppy の制御モデルを、制御系設計の方法を用いる。以下のように変数とパラメータの記号を定義する。また、Puppy の側面模式図を図5に示す。

- θ : 本体の進行方向の傾斜角
(鉛直軸から進行方向に傾斜したときが正)
- ϕ : 本体から測った車輪の回転角
(前進回転方向が正)
- I : 本体の傾斜方向の慣性モーメント
- L : 車軸から測った本体の重心の距離
- M : 本体の質量
- m : 車輪2つの質量
- J : 車輪の慣性モーメント
- r : 車輪の半径
- D_ϕ : 車輪の回転に伴う速度摩擦係数
- D_θ : 車輪回りに本体が回転するときの速度摩擦係数

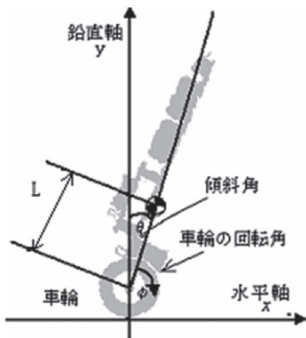


図5 Puppy の側面模式図

2-2-1. 運動エネルギー

剛体の運動エネルギーは、重心回りの回転運動エネルギーと重心の並進運動エネルギーの和で与えられる。車輪

の回転運動エネルギーは車輪の軸回りの慣性モーメントが J なので、

$$T_{Wr} = \frac{1}{2} J (\dot{\phi} + \dot{\theta})^2 \dots\dots (1-1)$$

ここで、回転角速度が ϕ だけではなく θ が加算されているのは、傾斜することによっても車輪が回転するためである。

傾斜方向の回転運動エネルギーは、車軸回りに θ だけ回転するときには本体の重心回りに同じだけ回転していることに注目すると

$$T_{Br} = \frac{1}{2} I \dot{\theta}^2 \dots\dots (1-2)$$

並進運動エネルギーに関しては、車輪の並進運動エネルギーと本体の並進運動エネルギーをそれぞれ求めなければならない。車輪の並進運動エネルギーは、車輪2つの質量が m なので、重心が車軸の中心にあると考えて求めます。車輪の動く速さを v とすると、車輪の並進運動エネルギーは、

$$v = r(\dot{\theta} + \dot{\phi})$$

$$T_{wt} = \frac{1}{2} m v^2 = \frac{1}{2} m r^2 (\dot{\theta}^2 + 2\dot{\theta}\dot{\phi} + \dot{\phi}^2) \dots\dots (1-3)$$

本体の並進運動エネルギーは、

$$T_{Bt} = \frac{1}{2} M V^2$$

$$= \frac{1}{2} M (V_x^2 + V_y^2)$$

$$= \frac{1}{2} M \{ \dot{\phi}_2 r^2 + \dot{\theta}_2 (r^2 + 2rL \cos \theta + L^2) + 2\dot{\phi}\dot{\theta}(r^2 + rL \cos \theta) \} \dots\dots (1-4)$$

ただし、

$$\begin{cases} V_x = \frac{d}{dt} \{ r(\theta + \phi) + L \sin \theta \} \\ V_y = \frac{d}{dt} (L \cos \theta) \end{cases}$$

となる。

運動エネルギーの総和は式(1-1)、式(1-2)、式(1-3)、式(1-4)を足しあわせたものとなる。

$$T = T_{Wr} + T_{Br} + T_{wt} + T_{Bt} \dots\dots (1-5)$$

2-2-2. ポテンシャルエネルギー

ポテンシャルエネルギー U は重心の位置エネルギーなので、

$$U = MgL \cos \theta \dots\dots (1-6)$$

となる。

2-2-3. ラグランジュ関数とラグランジュの運動方程式

運動方程式を書き下すために必要なラグランジュ関数 $L(\phi, \dot{\phi}, \theta, \dot{\theta})$ は、運動エネルギーとポテンシャルエネルギー

から機械的に以下のように求められる。

$$L = T - U \quad \dots\dots(1-7)$$

ラグランジュの運動方程式は、車輪を駆動する軸トルクを τ とすると以下のように与えられる。

$$\begin{cases} \frac{d}{dt} \left(\frac{\partial}{\partial \dot{\phi}} L \right) - \frac{\partial}{\partial \phi} L + D_{\phi} \dot{\phi} = \tau \\ \frac{d}{dt} \left(\frac{\partial}{\partial \dot{\theta}} L \right) - \frac{\partial}{\partial \theta} L + D_{\theta} \dot{\theta} = 0 \end{cases} \quad \dots\dots(1-8)$$

式(1-8)に式(1-7)を代入して計算を進めていくと以下の結果が導き出される。

$$\begin{cases} \left\{ \begin{aligned} &\{(m+M)r^2 + J\} \ddot{\phi} + \{(m+M)r^2 + MrL \cos \theta + J\} \ddot{\theta} \\ &- \dot{\theta}^2 MrL \sin \theta + D_{\phi} \dot{\phi} = \tau \end{aligned} \right. \\ \left\{ \begin{aligned} &\{(m+M)r^2 + MrL \cos \theta + J\} \ddot{\phi} \\ &+ \{(m+M)r^2 + 2MrL \cos \theta + ML^2 + J + I\} \ddot{\theta} \\ &- \dot{\theta}^2 MrL \sin \theta - MgL \sin \theta + D_{\theta} \dot{\theta} = 0 \end{aligned} \right. \end{cases} \quad \dots\dots(1-9)$$

式(1-9)の定数部分を変数変換して見通しを良くする。

$$\begin{cases} a = (m+M)r^2 + J \\ b = MrL \\ c = ML^2 + I \\ u = MgL \end{cases}$$

とおくと、

$$\begin{cases} a \ddot{\phi} + (a+b \cos \theta) \ddot{\theta} - \dot{\theta}^2 b \sin \theta + D_{\phi} \dot{\phi} = \tau \\ (a+b \cos \theta) \ddot{\phi} + (a+2b \cos \theta + c) \ddot{\theta} \\ - \dot{\theta}^2 b \sin \theta - u \sin \theta + D_{\theta} \dot{\theta} = 0 \end{cases} \quad \dots\dots(1-10)$$

と表される。このラグランジュの運動方程式(1-10)を用いて制御の設計を行っている。

2-2-4. Puppy の状態方程式

Puppy が倒立状態にあるとき、 θ と $\dot{\theta}$ は小さな値をとり続ける。前章で導いた運動方程式は、これらの変数に関して非線形な関数を含んでいたが、倒立状態を考えると $\theta=0$ 、 $\dot{\theta}=0$ のまわりで線形近似した式に置き換えても問題が生じないと考えられる。

このことから $\cos \theta \approx 1$ 、 $\sin \theta \approx \theta$ と近似し、 $\dot{\theta}$ 等の非線形項をゼロとみなすと式(1-10)は、

$$\begin{cases} a \ddot{\phi} + (a+b) \ddot{\theta} + D_{\phi} \dot{\phi} = \tau \\ (a+b) \ddot{\phi} + (a+2b+c) \ddot{\theta} - u \theta + D_{\theta} \dot{\theta} = 0 \end{cases} \quad \dots\dots(2-1)$$

と表される。この式を変形すると、

$$\begin{aligned} \ddot{\theta} &= a_{21} \theta + a_{22} \dot{\theta} + a_{23} \dot{\phi} + b_2 \tau \\ \ddot{\phi} &= a_{31} \theta + a_{32} \dot{\theta} + a_{33} \dot{\phi} + b_3 \tau \end{aligned} \quad \dots\dots(2-2)$$

ただし、

$$\begin{aligned} a_{21} &= \frac{au}{ac-b^2} & a_{22} &= \frac{-aD_{\theta}}{ac-b^2} & a_{23} &= \frac{(a+b)D_{\phi}}{ac-b^2} \\ a_{31} &= \frac{-(a+b)}{ac-b^2} & a_{32} &= \frac{(a+b)D_{\theta}}{ac-b^2} & a_{33} &= \frac{-(a+2b+c)D_{\phi}}{ac-b^2} \\ b_2 &= \frac{-(a+b)}{ac-b^2} & b_3 &= \frac{(a+2b+c)}{ac-b^2} \end{aligned}$$

となりこの式を状態方程式の形にすると、

$$\frac{d}{dt} \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \\ \dot{\phi}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ b_3 \end{bmatrix} \tau(t) \quad \dots\dots(2-3)$$

となる。この式(2-3)が以降の制御設計の基礎となる状態方程式となる。

表2 変数とパラメータの記号

車輪半径	r	0.029 [m]
車輪 (2つ分) と車軸の質量	m	0.06 [kg]
本体の質量 (乾電池含む)	M	0.27 [kg]
車輪と車軸の慣性モーメント	J	0.00115 [kg · m ²]
本体の慣性モーメント	I	0.0036973 [kg · m ²]
車軸から本体重心までの距離	L	0.105 [m]
車体の粘性摩擦抵抗	D_{θ}	2.00e-7 [Nms/rad]
車軸の粘性摩擦抵抗	D_{ϕ}	1.00e-4 [Nms/rad]

2-3. 「マイコンの基礎と、倒立モデル (Puppy) のためのマイコンプログラム」

Puppy のマイコンのプログラムは、パソコンを用いて、プログラム内容の変更、コンパイル、マイコンへの書き込みにより変更が可能である。

Puppy の動作を観察しながら、Puppy のプログラムの変更をする必要があるため、プログラム中の簡単な変数の値を変更するのであっても、プログラムの修正時間がかかる。また、修正前と修正後の Puppy の動作の比較は困難である。

この問題を解決するため、制御モデルをリアルタイムで動作させ制御でき、パラメータをオンラインで調節可能な dSPACE 社製のラピッドコントロールタイピングを用いて Puppy のプログラム修正を試みた。また、ラピッドコントロールタイピングを用いるために、Puppy の制御モデルを MATLAB/simulink を用いて記述することを試みた。

この状態方程式を用いて作成したソフトウェアを実装した Puppy の動作を図6に示す。

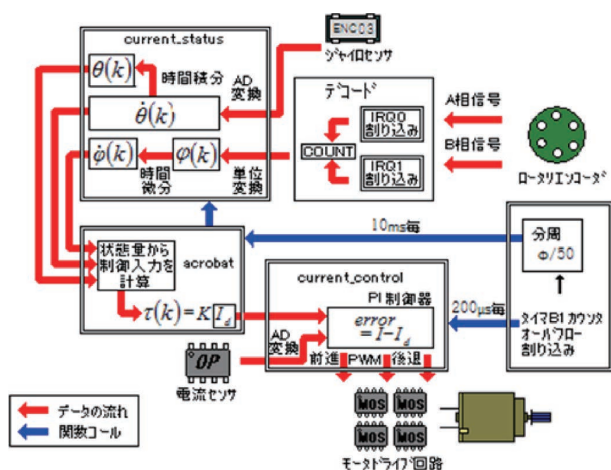


図6 Puppy の動作

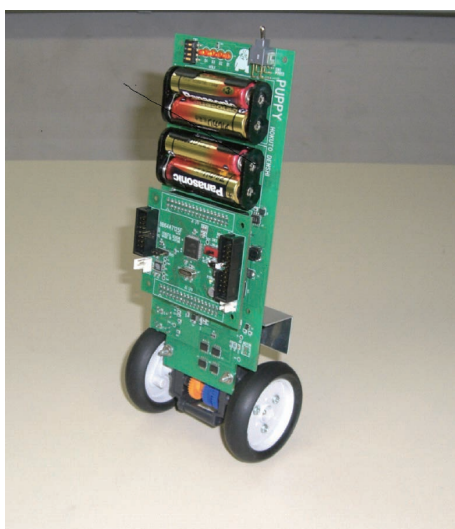


図7 倒立制御中の Puppy

2-3-1. プログラム中の関数

int_irq0

IRQ0端子に接続されたフォトロータリエンコーダA相の出力パルスが割り込み要因になったときに呼び出される関数である。車輪の回転に応じてパルスカウント (Count) を増減する。図の要領で割り込み要因が立ち上がりエッジのときB相がLOのとき、もしくは割り込み要因が立ち下がりエッジのときB相がHIでCountをインクリメント、逆の条件でデクリメントする。割り込み要因を反転させ、次の割り込みに備える。

int_irq1

IRQ1端子に接続されたフォトロータリエンコーダB相の出力パルスが割り込み要因になったときに呼び出される関数。int_irq0 (void) と同様な関数だが Count の増減要因は逆になる。int_irq0, int_irq1 の2つの関数でロータリエンコーダのA相, B相信号をデコードし車輪回転角を検出する。

int_tbt

タイマB1のオーバーフロー割り込みで呼び出される関数。タイマB1のカウンタオーバーフローは $200\mu\text{sec}$ ごとに発生し、関数内でモータ制御関数 control_current を呼び出す。さらに、50回に1回の割合 (10msec 毎) で倒立制御フラグ (Cflag) をセットする。

current_status

現在のPuppyの状態量を観測する関数。Puppyの倒立走行制御には車体傾斜角 θ 、車体傾斜角速度 $\dot{\theta}$ 、車輪回転角速度 ϕ が状態量として必要である。車体傾斜角速度は、傾斜角速度検出回路の出力をAD変換して単位変換する。車輪回転角速度は車輪回転角を表す Count 値車体傾斜角は、車体傾斜角速度を時間積分して求める。車輪回転角速度は、車輪回転角を表す Count を時間微分した後、その結果をSI単位系に変換し、あらかじめ制御の設計で用いた変数 $d\text{ThetaG}(\dot{\theta})$, $d\text{Fai}(\dot{\phi})$ に格納している。また、 ϕ を求めるために $d\text{ThetaG}(\dot{\theta})$ を台形法により時間積分し $\text{ThetaG}(\theta)$ を算出している。この関数は制御周期ごとに呼び出されることを前提に作られているのでそれ以外のタイミングで呼び出されると時間微分、積分が正確に行われないので注意が必要である。

acrobot

倒立走行制御を行うための関数。フィードバックゲインとPuppyの現在の状態量を制御式に代入し、モータのトルク指令値を計算する。Puppy制御説明書で説明したとおり、1ステージ遅れの制御を実施するので、計算結果であるモータ電流指令値 I_d は、次にこの関数が呼び出されたときにモータ制御関数 control_current に渡される。

control_current

モータトルク制御用関数。電流センサ入力のAD変換値をSI単位系に変換し、現在のモータ電流値を表す変数 I に格納します。 I と I_d の差が制御偏差となり、PI制御で制御偏差がすみやかにゼロに収束、つまり電流値が指令値に収束する。

gyro_offset

傾斜角速度検出回路の出力は角速度がゼロのとき、ENC-03Mの内部基準電圧約 (1.2V) となる。本体基板の電源電圧 V_{cc} (安定化 5V) は単三乾電池2本 (約 3V) を昇圧して作られており、電池の消耗具合によっては若干の変動が見られる。このことより、車体傾斜角速度ゼロ時の傾斜角速度検出回路出力のAD変換結果 (GYRO_OFFSET) は必ずしも毎回同じとは限らない。そこで、サンプルプログラムでは電源投入ごとに毎回 GYRO_OFFSET を計測する

ことで、より安定した倒立走行制御を実現している。

wait_m

引数で指定した時間 (msec) だけ待機する。この関数はタイマ B1 のカウンタオーバーフロー割り込みごとに立つフラグ (Cflag) を利用しているため割り込みをマスクした状態で実行すると意図しない無限ループに陥るので、必ずタイマ B1 のカウンタオーバーフロー割り込み許可、CCR の割り込みマスクビットをクリアした状態で使用する。

2-3. 「マイコンの基礎と、倒立モデル (Puppy) のためのマイコンプログラム」

「マイコンの基礎と、倒立モデル (Puppy) のためのマイコンプログラム」では、同軸 2 輪車型倒立振子 (ここでは、(株)北斗電子製 Puppy) の動作原理、Puppy のマイコンのプログラム変更に応じた関数について説明を行う。

構造的に不安定な Puppy を倒立させる制御の状態方程式について述べる。この制御を行うためには、Puppy 本体の傾斜角、角速度、車輪の角速度等から車輪に適切なトルクを求め、そのトルク値を、モータを用いて発生させることにより、倒立振子の倒立状態を維持する。ここではサンプルプログラムでの Puppy の制御モデルを、制御系設計の方法を用いる。

2-4. 「MATLAB/simulink の説明」

初歩的な電子制御技術に関連する教育を行うために、ソフトウェアとして、MATLAB/simulink および dSPACE 製の ControlDesk を用いた。MATLAB/simulink の説明をするにあたって、本学で行っている 3 年次の電子情報工学実験の内容を用いる。

>> は、コマンド入力を示す。

2-4-1. 行列の演算

$$A = \begin{bmatrix} 2 & 2 \\ 4 & 5 \end{bmatrix}$$

を計算して求めると、行列式 A はサラスの方式から、

$$\begin{aligned} A &= 2 \times 5 - 2 \times 4 \\ &= 10 - 8 \\ &= 2 \end{aligned}$$

となる。

この計算を MATLAB を用いて計算する場合、以下のように入力する。

```
>>A=[2 2;4 5];
>>det(A)
```

その結果、

ans=

2

となる。また、

$$B = \begin{bmatrix} 1 & 5 & 6 \\ -2 & -3 & 8 \\ 9 & 4 & 7 \end{bmatrix}$$

を MATLAB を用いて計算すると、以下のようになる。

```
>>B = [1 5 6;-2 -3 8;9 4 7];
```

```
>>det(B)
```

ans=

491

となる。では、次のような回路がある。

まず、実際にそれぞれの電流 I を計算して求める。キルヒホッフの法則から式をたてると、

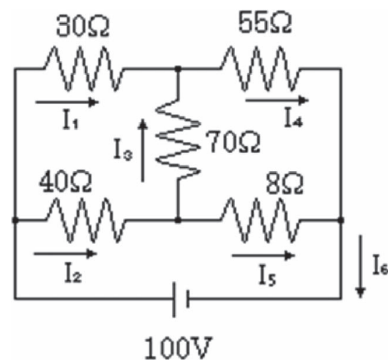


図 8 計算例の回路図

$$\begin{cases} 30I_1 - 40I_2 - 70I_3 = 0 & \cdots \text{I} \\ 50I_4 - 8I_5 - 70I_3 = 0 & \cdots \text{II} \\ 40I_2 + 8I_5 = 100 & \cdots \text{III} \end{cases}$$

となる。また、

$$\begin{cases} I_4 = I_1 + I_3 \\ I_5 = I_2 - I_3 \\ I_6 = I_1 - I_2 \end{cases}$$

であるので、

式 II、III に I_4 、 I_5 を代入すると、

$$\begin{cases} 3I_1 - 4I_2 - 7I_3 = 0 & \cdots \text{I}' \\ 55I_1 - 8I_2 + 133I_3 = 0 & \cdots \text{II}' \\ 12I_2 - 2I_3 = 25 & \cdots \text{III}' \end{cases}$$

$I \times 55 - \text{II}' \times 3$ より、

$$-196I_2 - 784I_3 = 0$$

$$I_2 + 4I_3 = 0 \quad \cdots \text{IV}$$

IV + III' × 2 より,
 $25I_2 = 50$
 $I_2 = 2 \dots\dots V$

V を IV に代入して,
 $I_3 = -0.5 \dots\dots VI$

V, VI を I' に代入して,
 $3I_1 - 4.5 = 0$
 $I_1 = 1.5 \dots\dots VII$

V, VI, VII を I_4, I_5, I_6 に代入して,
 $I_4 = 1.5 - 0.5$
 $= 1.0$

$I_5 = 2 + 0.5$
 $= 2.5$

$I_6 = 1.5 + 2$
 $= 3.5$

よって, $I_1 = 1.5[A], I_2 = 2.0[A], I_3 = -0.5[A], I_4 = 1.0[A], I_5 = 2.5[A], I_6 = 3.5[A]$ が求まる。

実際にはこのような計算が必要となるが MATLAB を用いると次のように求めることができる。

```
>>R=[30 -40 -70;55 -8 133;0 48 -8]
```

```
R=
    30   -40   -70
    55    -8   133
     0    48   -8
```

```
>>inv (R)
```

```
ans=
    0.0161    0.0094    0.0150
   -0.0011    0.0006    0.0200
   -0.0067    0.0037   -0.0050
```

```
>>V=[0;0;100]
```

```
V=
     0
     0
    100
```

```
>>I=inv(R)*V
```

```
I=
    1.5000
    2.0000
   -0.5000
```

このように, $I_1 = 1.5[A], I_2 = 2.0[A], I_3 = -0.5[A]$ を求めることができる。また,

$$\begin{cases} I_4 = I_1 + I_3 \\ I_5 = I_2 - I_3 \\ I_6 = I_1 - I_2 \end{cases}$$

であるので, $I_4 = 1.0[A], I_5 = 2.5, I_6 = 3.5[A]$ も求まる。

2-4-2. Simulink の利用

次のような RL 回路に直流電圧を加えたときの電流を求める。そして simulink モデルを用いてシミュレーションを行う。

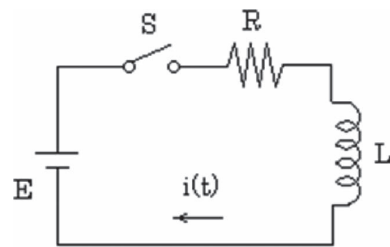


図9 RL回路

インダクタンスからコイルに電流 $i(t)$ を流すとその周辺に儀きが生じる。これにより作られる磁束を $\phi(t)$ とすると,

$$\phi(t) = L \cdot i(t)$$

と表わされる。これにより電圧は,

$$V(t) = \frac{d\phi(t)}{dt}$$

が成り立つ。そして, この式に先ほどの $\phi(t)$ を代入すると,

$$V(t) = L \frac{di(t)}{dt}$$

となり, 電圧はそこから流れる電流 $i(t)$ の時間微分に比例する。

すると回路方程式は,

$$E = R \cdot i(t) + L \frac{di(t)}{dt}$$

となる。よって電流 $i(t)$ は,

$$i(t) = \frac{1}{L} \int (E - Ri(t)) dt$$

となる。この $i(t)$ の式から RL 回路 (図9) を simulink でモデリングすると図10のようになる。その結果を図11に示す。

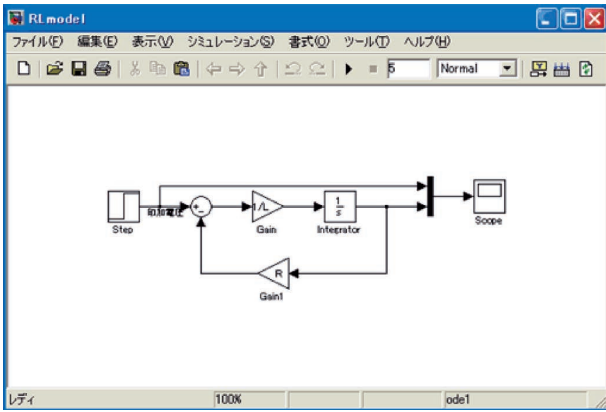


図10 RL回路のsimulinkモデル

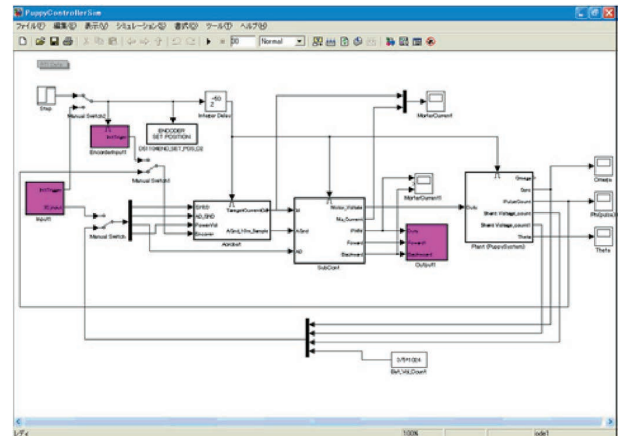


図13 倒立モデル (Puppy) の倒立制御用 simulink モデル

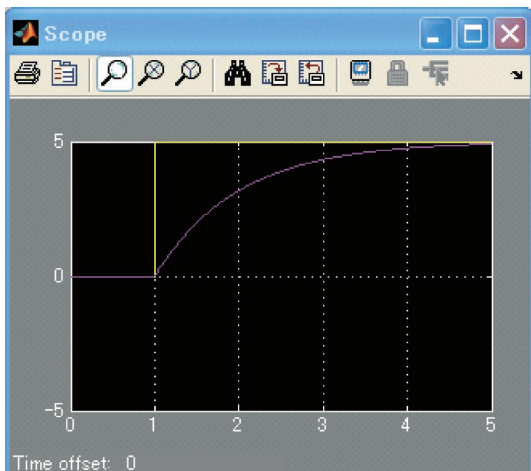


図11 RL回路のsimulinkモデルのシミュレーション結果
(黄：ステップ電圧, 赤紫：出力波形)

2-5. 「倒立モデル (Puppy) のためのMATLAB/simulink プログラム」

2-2-1で述べたようにPuppyの制御モデルは(2-1)式で表される。この状態方程式を、MATLAB/simulinkを用いて図12に示す。さらに周辺のモデルを付加した倒立モデル (Puppy) の倒立制御用 simulink モデルを図13に示す。

この制御モデルでPuppyを動作させることができた。この制御中の動作を調べるために、dSPACE社製のcontroldesk

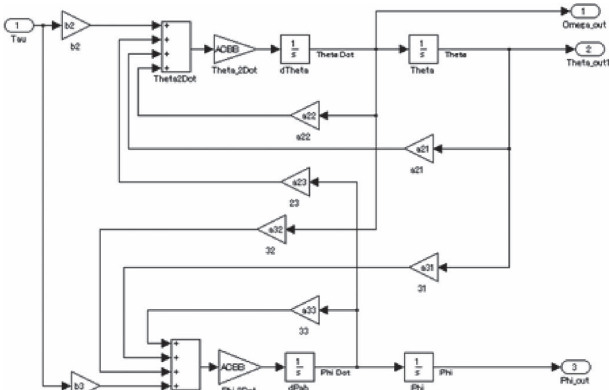


図12 倒立モデル (Puppy) の状態方程式の simulink モデル

を用いた。この controldesk は、Puppy を制御している時、コントローラとPuppyが通信を行い、リアルタイムに変数値を取得し、グラフ表示を行う。さらに、Puppyを実際に制御中に制御パラメータを変更でき、制御性をリアルタイムに観測することができる。(図14)

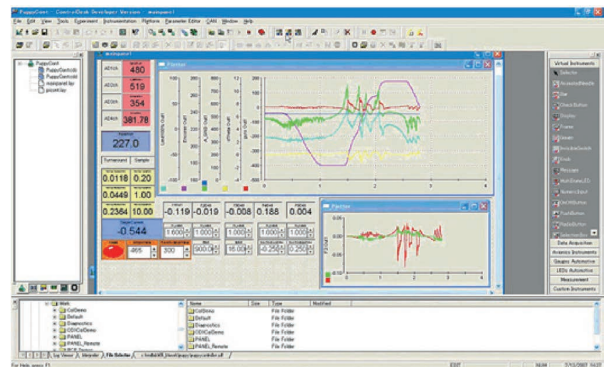


図14 倒立モデル (Puppy) 制御中の controldesk の画面

ピッドコントロールプロトタイピングを用いることにより、プログラムの修正時間が60分の1に短縮することができた。実際にリアルタイム制御しているPuppyを図15に示す。この応用として、Puppyの倒立時における車輪の回転に伴う速度摩擦係数 $D\phi$ のパラメータをPuppyの車輪の下に紙を置くことによって変化させた。この実験によって、条件が変化したときにおいても、パラメータの値をリアルタイムで制御できることを示した。この様子を図16に示す。

以上より、同軸2輪倒立モデルの物理モデルから、MATLAB/simulinkモデルの構築、dSPACEのラピッドコントロールプロトタイピング、電子制御技術を用いてPuppyの実演を行えた。物理モデルから数式化することが出来れば、そこからはソフトウェアにより処理を行うことができた。すなわち、MATLAB/Simulinkのモデルが構築できれば、後はソフトウェアにより電子制御できる可能性が示された。その結果、電子制御したい自然現象の物理モデル化技術の重要性が示された。

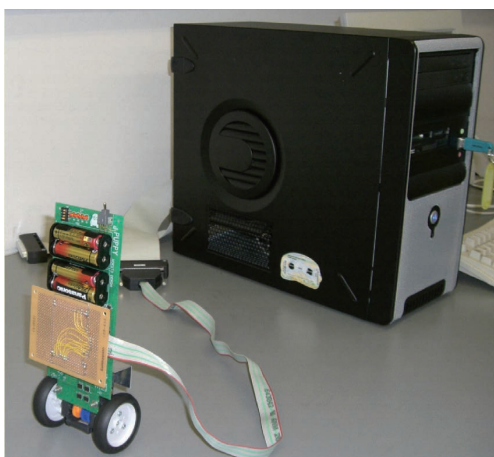


図15 dspace 社製制御ユニットを用いて制御中の Puppy

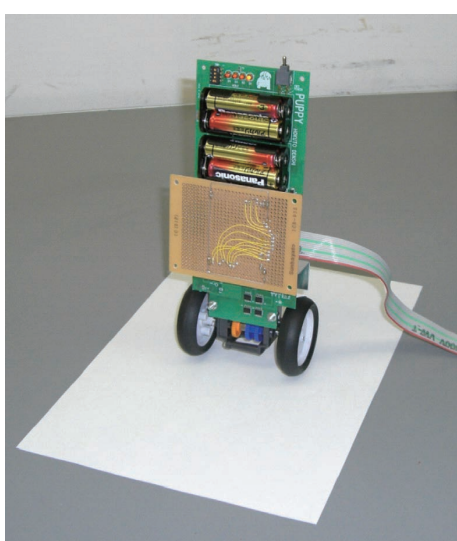


図16 紙を置くことで摩擦係数のパラメータを変化させた状態の Puppy

2-6. 「実習」

「実習」では MATLAB/simulink および controldesk を用いて倒立モデル (Puppy) の制御を行う。

広島工業大学オープンキャンパス2011で倒立モデル (Puppy) について説明した際に用いた制御マニュアルを用いて説明する。

2-6-1. MATLAB R2006b

(1) MATLAB R2006b を起動させると、「Select dSPACE RTI Platform Support」とウインドウが表示されるので、「RTI1104」を選択する。



図17 Select dSPACE RTI Platform Support の画面

(2) MATLAB R2006b から Puppy のプログラムデータを読み込む。コマンドウインドウに「PuppyGo」と入力し Enter で読み込まれる。

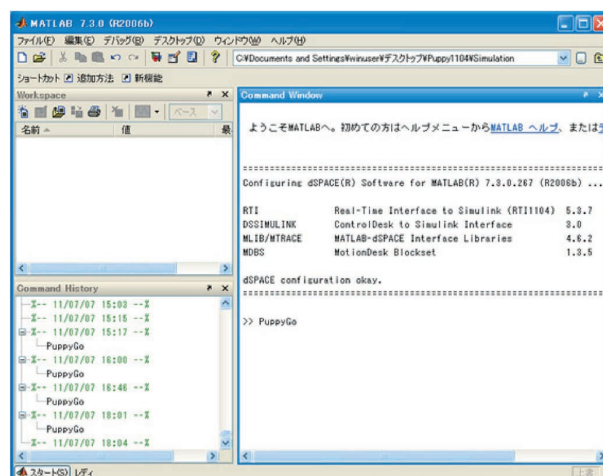


図18 MATLAB R2006b の画面

2-6-2. ControlDesk

(1) ControlDesk を起動させる。

(2) リアルタイムに変数値を取得し、グラフ表示を行うためのファイルを開く。

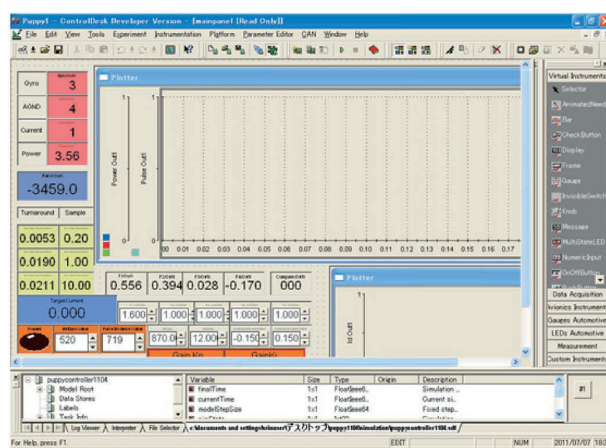


図19 ControlDesk の画面



図20 パソコンに接続した Puppy

2-6-3. Puppy の制御

(1) パソコンに接続した Puppy を持ち上げ、地面から離れた状態でスイッチを ON にし、このときの Gyro の数値を読み取り、スイッチを OFF にする。

(2) 読み取った Gyro の数値を参考にして Gyro/Value 値を調節する。

(3) 実際に Puppy を走らせる。このとき、Puppy が直立し、制止状態を保てていたら成功。Puppy が制止状態を保てず、前進または後進する場合は Gyro/value 値を調節する。

(前進した場合、Gyro/value 値を上げる。後進した場合、Gyro/value 値を下げる。)

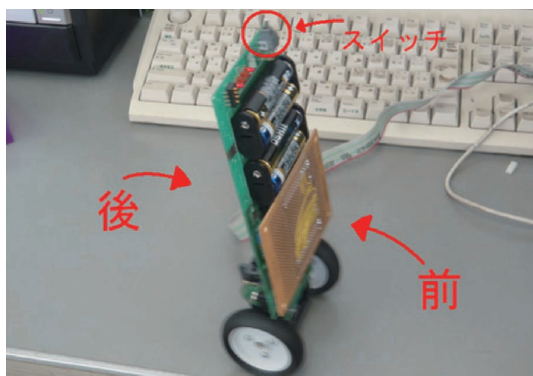


図21 制御中の Puppy

3. まとめ

1) ルネサス製 SH マイコンを用いた倒立制御学習キットの動作を演示し、マイコンプログラムの変更も行った。

2) 物理モデルから、MATLAB/simulink モデルを作成した。その MATLAB/simulink モデルを自動車用電子制御ユニット開発用の dspace 製の制御ユニットを用いて、Puppy の制御を行った。パラメータの変更は、マイコンプログラムの変更と比べ60倍程度高速であることがわかった。

3) Puppy を動作する環境が変化しても、自動車用電子制御ユニット開発用の dSPACE 製の制御ユニットを用いれば、リアルタイムでパラメータの変更を行えることを実証

した。

4) 以上一連の内容を、本学、また高大連携における教育に導入し、受講学生が実際に電子制御を行った。

物理モデルから MATLAB/simulink モデル、Controldesk による電子制御を実際に行うことは、学生にとって容易であるが、内容を理解することは難しいので、今後はより分かりやすいマニュアルを作成し、学生一人一人に考えさせる時間を設けることを試みる。さらに本学 NEXUS 2F CAD 室に MATLAB/simulink が103ライセンス整備されているので“オープン利用”時間に学生の技術のスキルアップのために利用を推進していきたいと思います。

謝 辞

本研究は、「カーエレクトロニクスを題材とした電子情報技術の体験教育支援」の支援を頂いております。dSPACE 製制御ユニットに移植の協力をいただいた城所仁さんに謝意を表します。また、Puppy のマニュアルの利用させていただいた北斗電子(株)に謝意を表します。最後に、本研究論文の図面作成、文章清書等に協力頂いた上本亮君、藤原拓也君に謝意を表します。

文 献

- 1) 川畑敬志, 田中 武, “学部課程におけるマイクロエレクトロニクス実験”, 平成3年度電気・情報関連学会中国支部第42回連合大会講演予稿集, p. 342.
- 2) 田中 武, 水野議一郎, 大村道郎, “基本的な集積回路設計および半導体プロセス工学教育システムの構築”, 応用物理教育26 (2002) 7.
- 3) http://www.ksrp.or.jp/shisetsu/02_center.html.
- 4) 田中 武, 安藤秀幸, 山田明宏, “システム LSI 関連ものづくり教育における IC チップ作成”, 広島工業大学紀要, 2005, pp. 7-12.
- 5) (株)北斗電子, 倒立制御学習キット Puppy.