

車載電子モジュール用プラットフォームの開発 (第1報)

1 モデルベース開発による台車型倒立振り制御

大賀 誠, 山本 晃, 古本浩章, 宮野忠文, 長谷川浩治, 問山清和

Development of the innovative platform for the electronic module using vehicle applications (1st report)
Control system of the inverted pendulum using model based development method

OHGA Makoto, YAMAMOTO Akira, FURUMOTO Hiroaki, MIYANO Tadafumi, HASEGAWA Koji
and TOIYAMA Kiyokazu

The MBD (Model Based Development) is a de facto standard development method in the automotive industry. This paper provides a control system of the Inverted Pendulum adopting the method. It makes several design processes efficiently, and can reduce the working time in the coding and validation processes. In order to simulate the designs, it is converted the plant of the Inverted Pendulum into the model, and prepared the controller model based on the modern control theory. After the simulation and validation, almost the program code which realized the control algorithm is generated automatically from the controller model by using the CAE software. In result, it became possible that the system can be completed with a minimum of the working time in the every design and validation processes.

キーワード: モデルベース開発 倒立振り制御 現代制御理論 自動コード生成 MATLAB/Simulink

1 緒 言

環境や安全, 快適といった自動車の高付加価値化により, 車載の電子モジュールに搭載される組込みソフトウェアの開発規模は増加の一途をたどり, 車両全体で数千万行に到達するとの推計も報告されている¹⁾。大規模化するソフトウェアは複数名で開発されるが(プロジェクト開発), メンバー間の意思疎通不足等で生じる問題のため, 開発効率やソフトウェア品質に不具合が発生するケースが多い。このような問題を解決するため, 自動車業界においてはモデルベース開発 (Model Based Development) の導入が進められている。これは開発対象をモデル化し, それを拠り所として設計・検証を進める手法で, 開発関係者間の意思疎通が良好となる。

本報告ではこの手法を活用し, シンプルな構造でありながらも高度な制御技術が要求される, 台車型倒立振り制御を実現する。この手法による設計・検証を実際の工程に沿って開発する。コーディングを主とするコードベースの開発ではなく, モデル (設計書) ベースで開発したシステムが, 必要最低限のコーディング及び検証で倒立制御可能であったことを報告する。

2 モデルベース開発とは

モデルベース開発とは開発対象となるシステム (制御対象及びコントローラ) を記述したモデルを仕様として定義し, そのモデルを拠り所とした設計・検証で開発を

進める手法と定義される²⁾。制御系設計においては図1に示すようなVサイクル開発と一体として表わされることが多い³⁾。図の左側が設計工程を, そして右側が検証工程を表わす。モデルは通常, 制御対象, コントローラともにグラフィカルに表現され, 各工程においては必要に応じてモデルと実機を置き換えて作業を進めていく。

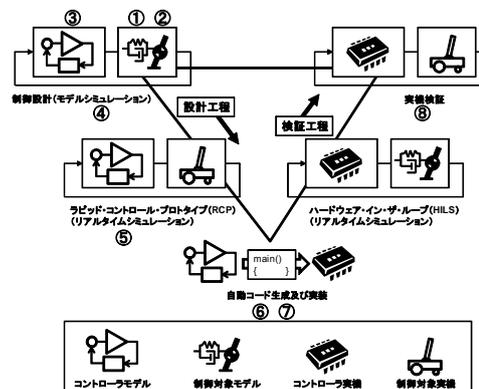


図1 Vサイクル開発とモデルベース開発

開発に使用した CAE ツールは MATLAB/Simulink である。これはモデルベース開発に先進的に取り組んでいる自動車業界で標準的に採用されているツールである。今回の台車型倒立振り制御の開発においては, 設計工程を中心に報告を行い, 具体的には次に示す工程で開発を進める (図1中番号と対応)。なお, 図中のハードウェア・イン・ザ・ループに関してはシミュレーション環境の制約から今回は実施しない。

- ①制御対象モデリング
- ②パラメータ推定
- ③コントローラモデリング (制御系設計)
- ④モデルシミュレーション
- ⑤リアルタイムシミュレーション
- ⑥コントローラモデルの離散化と自動コード生成
- ⑦実装
- ⑧動作検証

3 モデリング

3.1 台車型倒立振子のハードウェア

台車型倒立振子とは台車上に進行方向前後 1 自由度の回転軸を介して配置した棒を、台車を動かして倒立制御するシステムで、手の上で棒を立てる遊びを模している。

倒立振子制御の台車として、今回は株式会社北斗電子から販売されている PUPPY を改造して使用した^{4), 5)}。本来 PUPPY は 2 輪倒立振子制御学習キットであるが、前輪を取り付けて台車とした。車輪角度を計測するエンコーダも実装されている。マイコンは PUPPY に装着可能な評価ボードを使用し、CPU はルネサスエレクトロニクス株式会社製 SH7125 (SH2-Tiny) を選定した。振子は 250mm × 10mm × 3mm のアルミ製で重量は 25g。1 周 4000 パルス分解能のロータリーエンコーダを介して台車に取り付けた。ハードウェアの外観とブロック図を図 2 に示す。

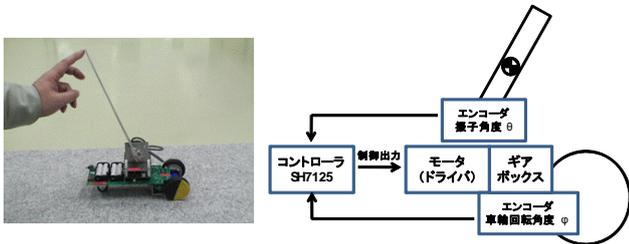


図 2 台車型倒立振子の外観とブロック図

3.2 制御対象モデリング

制御対象となる台車型倒立振子のモデリングを行う。図 3 のようにパラメータを設定した。これらのパラメータを基にラグランジュの運動方程式を解くことにより式(1)の状態方程式を得る。

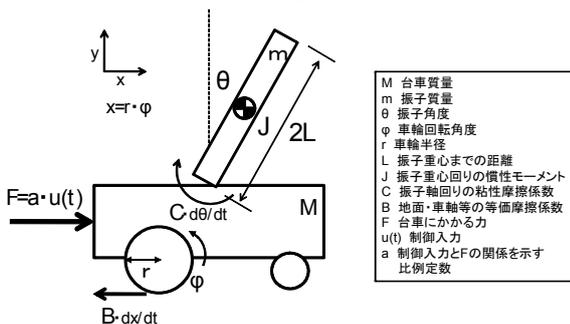


図 3 パラメータの設定

$$\frac{d}{dt} \begin{pmatrix} \theta \\ \varphi \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & 0 & a_2 & a_3 \\ a_4 & 0 & a_5 & a_6 \end{pmatrix} \begin{pmatrix} \theta \\ \varphi \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ b_1 \\ b_2 \end{pmatrix} u(t) \quad \dots (1)$$

ここで、各パラメータは、

$$a_1 = m \cdot g \cdot L \cdot r \cdot (M + m) / \alpha_0$$

$$a_2 = -C \cdot r \cdot (M + m) / \alpha_0$$

$$a_3 = B \cdot r^2 \cdot M \cdot L / \alpha_0$$

$$a_4 = -m^2 \cdot g \cdot L^2 / \alpha_0$$

$$a_5 = C \cdot m \cdot L / \alpha_0$$

$$a_6 = -C \cdot r \cdot (M + m) / \alpha_0$$

$$b_1 = -m \cdot L \cdot r \cdot a / \alpha_0$$

$$b_2 = a \cdot (J + m \cdot L^2) / \alpha_0$$

$$\alpha_0 = \{(J + m \cdot L^2)(M + m) - m^2 \cdot L^2\} \cdot r$$

で示される。また g は重力加速度である。今後、この状態方程式を制御対象のモデルとして扱っていく。なお、状態変数は次の式(2)で示される。

$$\begin{pmatrix} \theta & \varphi & \dot{\theta} & \dot{\varphi} \end{pmatrix}^T \quad \dots (2)$$

3.3 パラメータ推定

式(1)の状態方程式には未知のパラメータが含まれている。台車に関しては地面・車輪等の等価摩擦係数 B、制御入力と F の関係を示す比例定数 a、振り子に関しては重心回り慣性モーメント J、振り子軸回りの粘性摩擦係数 C である。これらのパラメータは計測器等で求めることは難しい。

そこでパラメータを推定するための実験を行い、その結果と MATLAB/Simulink によるシミュレーション結果を比較することで値を決定した。例えば台車の場合は実際に走らせ、また振り子の場合は自由振子運動させ実験データを計測した。シミュレーションに用いた運動方程式を式(3)、(4)に示す。

台車の運動方程式

$$M \cdot r \cdot \ddot{\varphi} = a \cdot u(t) - B \cdot r \cdot \dot{\varphi} \quad \dots (3)$$

自由振子運動の方程式

$$\ddot{\theta} = (-C / (m \cdot L^2 + J)) \dot{\theta} - (m \cdot g \cdot L \cdot \sin \theta) / (m \cdot L^2 + J) \quad \dots (4)$$

3.4 コントローラモデリング

制御設計によりコントローラをモデリングする。今回は現代制御理論を採用し、多入力制御系を扱うことが可能な状態フィードバック制御で設計する^{6) ~ 8)}。制御入力は振り子の角度 θ と車輪回転角度 φ をそれぞれ

れエンコーダで計測して用い、また制御出力（操作量）はモータの制御電圧とする。

状態フィードバック制御における状態変数は全て計測可能でなければならない。しかし、今回の設計では振り角速度、車輪回転角速度といった微分項は計測が不可能である。そこで計測可能な値を用いて制御対象の状態変数を推定する手法であるオブザーバを導入し、2つの微分項の状態推定量を求めた。

一般的に今回の制御系のような閉ループ系においてシステムの動特性は極配置で設計するが、操作量の大きさやパラメータ変動への感度等調整が困難なケースが多い。そこで適正な応答の速度と操作量の大きさを決定するために、2次形式評価関数である最適レギュレータを導入して動特性の設計を行った。

状態フィードバック制御は本来制御対象を安定化するだけであり、出力（この場合車輪回転角度 φ ）を任意の参照値に一致させることは出来ない。そこで出力を参照値 φ_0 に追従させるための制御系（サーボ系）を設計した。

図4にこの工程で設計した制御ブロック図を、図5(a)にMATLAB/Simulinkで記述したコントローラのモデル、図5(b)にオブザーバを含んだ制御対象のモデルを示す。これまでの工程によりシミュレーション可能なコントローラと制御対象モデルの作成が完了した。

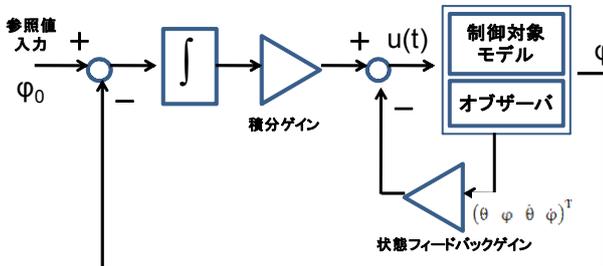
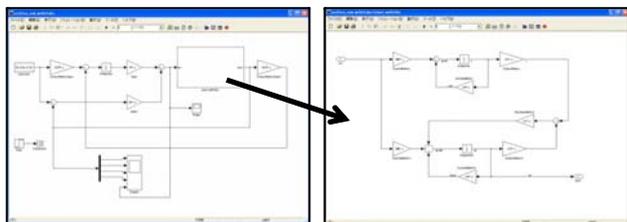


図4 制御ブロック図



(a)コントローラモデル (b)オブザーバを含む制御対象モデル

図5 MATLAB/Simulink モデル

4 シミュレーション

4.1 モデルシミュレーション

図5のモデルを用い、モデルシミュレーションで設

計検証を行う。 φ_0 を 2π （車輪1回転分）として動作開始から5秒間のシミュレーションを実施した。図6にシミュレーション結果を示す。図は上から振り角角度 θ 、車輪回転角度 φ 、制御量 $u(t)$ で、時間の経過に伴う各パラメータの変化を表示する。この結果から θ は0に、そして φ は 2π に収束することが確認できる。よってシミュレーションにより倒立振り制御が実現可能なことが示された。

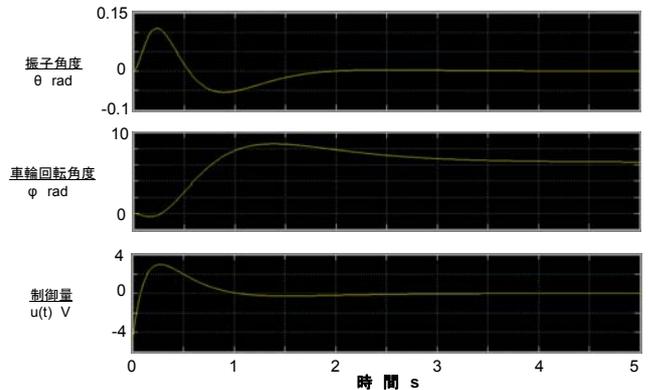


図6 モデルシミュレーション結果

4.2 リアルタイムシミュレーション

モデルシミュレーションの結果を踏まえ、制御対象モデルを台車及び振りの実機ハードウェアと置き換えて、リアルタイムシミュレーションを行う。このシミュレーションはコントローラモデルで実機を制御することで、実時間での設計検証が可能となる。この工程は一般的にラピッド・コントロール・プロトタイプ（以下RCPと記す）と呼ばれる。図7にRCPのイメージを示す。RCPはMATLAB/Simulinkのオプション機能であるxPC Targetを活用して実施する。

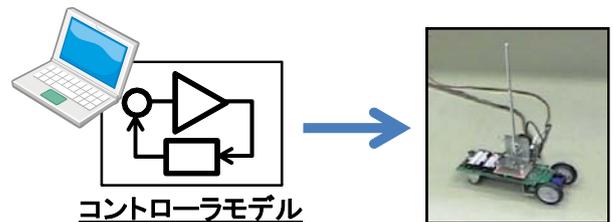


図7 RCPのイメージ

RCPの結果、積分ゲイン及び状態フィードバックゲインを若干調整するだけで、倒立振り制御を実現することが出来た。この後、コントローラモデルを制御周期5msで離散化して離散系の設計においてもRCPを行い同様の動作を確認した。RCPにおいてはMATLAB/Simulink上でコントローラモデルを動作させ、I/Oボードを介して実機へ制御信号を供給する。

5 実装及び検証

RCP によるリアルタイムシミュレーションにより実機動作によるコントローラモデルの検証が完了した。この離散化されたコントローラモデルからC言語のソースコードをMATLAB/Simulinkにより自動的に生成する。得られるコードは倒立振り制御のアルゴリズムに関する部分で、I/O 制御や割込み等に関する部分は含まれない。これらの自動生成されない部分は、使用するマイコンの仕様に合わせてプログラマがコードを作成した。自動生成されたコードは、変更することなくマイコンのプログラム上でそのまま使用した。その結果、I/O (ロータリーエンコーダ入力や PWM 出力等) や時間割込みの仕様を合わせるだけで、倒立振り制御を実現することが可能であった。その動作の様子を図8に示す。完成した台車型倒立振り制御において、マイコンのプログラムコード上でのデバッグはI/O 制御部分のみで、振り制御のアルゴリズムに関する部分ではデバッグする必要はなかった。

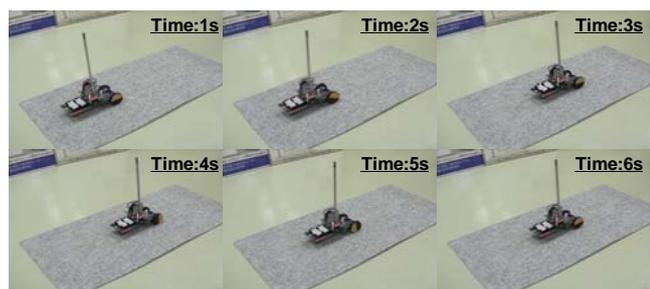


図8 完成した台車型倒立振り制御

6 結 言

組込みソフトウェアの開発効率及びプログラム品質向上が期待されるモデルベース開発を導入し、この開発手法における一連の設計工程を実践することで台車型倒立振り制御を実現した。まず初めに制御対象及びコントローラのモデリングを行い、モデルシミュレーションによる設計検証を行った。設計したコントローラモデルを用いて実機を制御する、リアルタイムシミュレーションにて制御系設計の検証を行った後、離散

化したコントローラモデルからプログラムを自動コード生成してマイコンへ実装した。その結果、必要最低限のコーディングと検証で倒立振り制御を実現することが出来た。

設計の根幹部分である制御アルゴリズムを自動コード生成することで、開発工数の削減が可能となる。また、設計変更はモデルの修正から自動コード生成という開発パスとなり、コードを直接変更する機会を極力少なくする開発プロセスを構築することが出来る。プログラマが作成するコードを最低限に抑えるということは、作りこまれるバグの量を抑え込むことを意味し、品質に与える影響は大きい。

このようにモデルベース (設計書ベース) の設計を組み込みソフトウェア開発に導入することは、開発関係者間の意思疎通を良好とし、設計上流工程での不具合混入を防ぐ効果も期待でき、開発効率及びプログラム品質の向上に寄与することができる。

今回の開発で得られた基盤技術を基に、例えばシステム同定を取り入れたモデリング手法の高度化、アクチュエータやセンサ、制御手法等の更なるモデル化を継続する。加えてシミュレーション環境の充実を図るとともに、車載電子モジュールを中心とした組み込みソフトウェア開発サポート環境の構築を目指す。また、今後この台車型倒立振り制御システムは、モデルベース開発による組み込みソフトウェア開発手法の紹介や技術移転のために活用する。

文 献

- 1) 広島県：ひろしまカーエレクトロニクス戦略, 2008
- 2) <http://monoist.atmarkit.co.jp/fembedded/articles/mbd/mbda.html>
- 3) 城所：インターフェース 10 (2007) 96
- 4) <http://www.hokutodenshi.co.jp/>
- 5) PUPPY 取扱い説明書 株式会社北斗電子(2005)
- 6) 島田：モーションコントロール オーム社 (2004)
- 7) 天野：やさしいシステム制御工学 森北出版社(2008)
- 8) 野波：MATLAB による制御理論の基礎 東京電機大学出版局(1998)