ハイブリッド自動車の燃費最適化制御器設計 —同時摂動最適化によるモデルフリーアプローチ—

馬場 一郎*・東 俊一*・杉江 俊治*

Controller Design for Optimizing Fuel Consumption of Hybrid Electric Vehicles
—A Model-free Approach Based on Simultaneous Perturbation Stochastic Approximation—

Ichiro BABA*, Shun-ichi AZUMA* and Toshiharu SUGIE*

This paper addresses the benchmark problem of controller design of hybrid electric vehicles, which has been recently proposed by JSAE and SICE. As a solution, we propose a model-free design method based on the simultaneous perturbation stochastic approximation. The proposed method achieves approximately 44 percent improvement compared with the existing controller.

Key Words: hybrid electric vehicle, model-free controller design, stochastic approximation

1. はじめに

近年、環境問題への関心の高まりを背景に、ハイブリッド自動車が大きな注目を集めている。特に、自動車業界では、ハイブリッド自動車の燃費改善に関する研究開発が盛んに行なわれており、ハイブリッドパワートレインの制御も重要な問題の一つとして挙げられる。その一方で、ハイブリッド自動車は、ガソリンエンジンとモータの両方を動力源としているため、その制御は従来の自動車に比べて格段に複雑となっている。

このような問題を産業界と学界が共同して解決するために、ハイブリッド自動車の制御器設計のベンチマーク問題が提案されている^{1)~3)}。このベンチマーク問題では、**Fig. 1**に示されるような、ハイブリッド自動車に対して、燃費性能を最大化する制御器の導出が目標とされている。

本論文では、このベンチマーク問題の検討を行ない、実用的な解を与えることを目的としている。一般に、制御系設計にあたっては、制御対象のモデルが重要な役割を演じるが、その反面、モデル開発には多大な人的・時間的コストを要することが知られている。また、対象への深い知識も要求される。本論文では、このような事情を考慮し、モデルを用いないモデルフリーの設計法によって、このベンチマーク問題の解を与えるを試みる。特に、同時摂動最適化⁴⁾と呼ばれる最適化法をこのベンチマーク問題のために改良した設計アルゴリ

Fig. 1 System to be studied

ズムを提案する。そして、この提案法によって、ベンチマーク問題とともに提供されているサンプル制御器と比較し、約44%もの燃費改善を行なう制御器が得られることを示す。

最後に、つぎの点を注意しておく。著者らの知る限り、現時点において、本ベンチマーク問題に対する解として公表されているものは見当たらない。したがって、本論文では、サンプル制御器との比較によって提案手法の有効性を示している。また、本論文の成果がベンチマーク問題の第一番目の解として、ほかの成果への基準となることが期待される。

2章で、ベンチマーク問題について説明を行なう。3章で、同時摂動最適化を基にした制御器設計の手法について述べ、4章で、実際の制御器を設計することによって提案法の有用性を示す。

表記法 $\mathbf{R}$, $\mathbf{R}_+$ は、それぞれ実数、正の実数の集合を表わす。確率変数 X に対して、 $\mathbb{P}(X = a)$ で $X = a$ となる確率を表わす。一階微分可能な関数 $f: \mathbf{R}^n \rightarrow \mathbf{R}$ の勾配を $\nabla f(x)$ で表わす。すなわち、 $\nabla f(x) := [\frac{\partial f(x)}{\partial x_1} \frac{\partial f(x)}{\partial x_2} \dots \frac{\partial f(x)}{\partial x_n}]^T$ であ

* 京都大学大学院情報学研究科 宇治市五ヶ庄

* Graduate School of Informatics, Kyoto University, Gokasho, Uji

(Received January 15, 2013)

(Revised April 17, 2013)

る. $\delta \in \mathbf{R}_+$ に対し, $\text{sat}_\delta : \mathbf{R}^n \rightarrow \mathbf{R}^n$ は, その第 i 要素が以下のように与えられる飽和関数である.

$$\text{sat}_\delta(x) \text{ の第 } i \text{ 要素} = \begin{cases} \delta & \text{if } \delta < x_i \\ x_i & \text{if } -\delta \leq x_i \leq \delta \\ -\delta & \text{if } x_i < -\delta \end{cases}$$

ここで, $x_i \in \mathbf{R}$ は, $x \in \mathbf{R}^n$ の i 番目の要素である.

2. ベンチマーク問題：ハイブリッド自動車の燃費最適化^{1)~3)}

2.1 対象システム

Fig.1 に示されるような, ドライバと車両で構成されるシステムについて考える. ドライバは, 目標とする車両速度とともに, 要求する駆動力あるいは制動力を決定する. 車両は, エンジンと二つのモータ, およびバッテリーから構成されるパワートレインを搭載したハイブリッド自動車である. また, 車両にはパワートレインの制御を行なう制御器が搭載されている.

Fig.1 のシステムを制御系として捉えと, **Fig. 2** のように表わすことができる. ここで, P は, このシステムの制御対象であり, Fig.1 に示されるドライバと車両を合わせたもの, また, K は制御器である. $u(t) \in \mathbf{R}^7$ は, P への入力であり, その詳細は **Table 1** のとおりである. $y(t) \in \mathbf{R}^{16}$ は, 制御器にフィードバックされる P の出力であり, **Table 2** に示される信号からなる. $\theta(t) \in \mathbf{R}^5$ は, このシステムへの外部入力であり, **Table 3** に示されるように, 目標速度や天候など, この車両の走行データをまとめたものである.

ベンチマーク問題においては, 制御対象 P は, ソフトウェア GT-SUITE⁵⁾ のプログラムとしてあらかじめ与えられている. その一部を **Fig. 3** に例示する. また, K は, Simulink モデルとして記述されるようなものであり, これに対しては, 燃費 17.63 km/L を達成するサンプル制御器 (以下では, これを K_s で表わす) があらかじめ与えられている. 走行データ $\theta(t)$ は, あらかじめ 3 週間分 (80420 秒分) が与えられており, 制御器設計に利用することができる. **Fig. 4** は, 初期時刻から 3000 秒までの $\theta_1(t)$ を表わしている.

Fig.2 のシステムにおいて, 制御器 K を適切に定めれば, MATLAB 上で車両の走行シミュレーションを実行することが可能となっている. シミュレーションを実行することによって, 燃費などの性能指標を実際の数値として得ることができる.

2.2 燃費最適化問題

Fig. 2 のシステムに対し, 燃費はつぎのように定義される.

$$J(P, K, \theta, T) = \frac{\int_0^T |y_1(t)| dt}{\int_0^T |y_2(t)| dt} \quad (1)$$

ここで, $y_1(t), y_2(t) \in \mathbf{R}$ は, それぞれ車両速度と瞬時燃料消費量に対応する. $T \in \mathbf{R}_+$ は燃費を評価する時間である. 右

Fig. 2 Control system

Table 1 Control input $u(t)$

u	Description	Unit
u_1	Target vehicle speed	km/h
u_2	Target throttle-angle of engine	%
u_3	Engine ignition flag (1: Ignition, 0: Stop ignition)	N.A.
u_4	Engine start-up flag (1: Crank up engine, 0: Others)	N.A.
u_5	Torque required to motor 1	Nm
u_6	Power required to motor 2	kw
u_7	Generation power required to battery	w

Table 2 Output $y(t)$

y	Description	Unit
y_1	Actual vehicle speed	km/h
y_2	Instantaneous fuel consumption	L
y_3	Demanded driving power	kw
y_4	Demanded braking power	kw
y_5	Corrected demanded power	w
y_6	Engine speed	rpm
y_7	Engine torque	Nm
y_8	Engine power	kw
y_9	Motor 1 revolution speed	rpm
y_{10}	Motor 1 torque	Nm
y_{11}	Motor 1 power	kw
y_{12}	Motor 2 revolution speed	rpm
y_{13}	Motor 2 torque	Nm
y_{14}	Motor 2 power	kw
y_{15}	Battery open circuit voltage	V
y_{16}	Battery state of charge	%

Table 3 External input $\theta(t)$

θ	Description	Unit
θ_1	Target vehicle speed	km/h
θ_2	Day of week (1: Mon., 2: Tue., 3: Wed., 4: Thu., 5: Fri., 6: Sat., 7: San.)	N.A.
θ_3	Weather number (1: Fine, 2: Cloudy, 3: Rainy and snowy)	N.A.
θ_4	Direction (1: Going, 2: Returning)	N.A.
θ_5	Starting time	o'clock

辺の分母は総燃料消費量 [L] を, 分子は総走行距離 [km] を表わしており, これによって, J が, 燃費 [km/L] となっていることが確認できる.

つぎに, ドライバ満足度と呼ばれる目標速度への追従性は以下のように定義される.

$$g(P, K, \theta, T) = 100 - \int_0^T p(t, P, K, \theta) dt \quad (2)$$

ここで、 $K_s(x)$ は、 K_s にゲイン x_i ($i = 1, 2, \dots, n$) を導入したものを表わす。 $T_d \in (0, 80420]$ は、設計用に燃費とドライバ満足度を評価する時間である。これは、(1) 式の燃費 $J(P, K, \theta, T)$ と (2) 式のドライバ満足度 $g(P, K, \theta, T)$ の値を得るための計算時間が、大きな T に対しては長くなるため、短い計算時間で燃費とドライバ満足度の近似値を得るために導入されている。 $\phi: \mathbf{R} \rightarrow \{0, 100\}$ は、ペナルティ関数

$$\phi(s) = \begin{cases} 100 & \text{if } 90 < s \leq 100 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

であり、これを用いることで、(7) 式では燃費とドライバ満足度を同時に評価している。

つぎに、(6) 式の同時摂動最適化アルゴリズムを、つぎのように飽和関数を導入した形に改良する。

$$x(k+1) = x(k) + \text{sat}_\delta \left(a(k) \begin{bmatrix} \frac{f(x(k)+c(k)r(k)) - f(x(k)-c(k)r(k))}{2c(k)r_1(k)} \\ \vdots \\ \frac{f(x(k)+c(k)r(k)) - f(x(k)-c(k)r(k))}{2c(k)r_n(k)} \end{bmatrix} \right) \quad (9)$$

これによって、 $x(k)$ の更新量となる右辺第二項の大きさを δ に制限し、アルゴリズムを安定に動くようにしている。ただし、 δ は任意に設定される。

このとき、本論文で提案する設計法は、以下のようになる。
モデルフリー設計法

STEP 1: 制御器 K_s の n 個の信号線およびパラメータに、ゲイン x_i ($i = 1, 2, \dots, n$) を導入する。

STEP 2: 設計用の評価関数を定める時間 T_d 、(9) 式の同時摂動最適化アルゴリズムの設定パラメータ $x(0)$ 、 $r(k)$ 、 $a(k)$ 、 $c(k)$ 、 δ および (9) 式を実行する回数 $k_{\max}$ を設定する。 $f(x) := J_d(P, K_s(x), \theta, T_d)$ とした (9) 式のアルゴリズムを $k = 0, 1, \dots, k_{\max} - 1$ に対して実行し、 $x(k_{\max})$ を得る。この際、 $J_d(P, K_s(x(k) + c(k)r(k)), \theta, T_d)$ の値を求めるためのシミュレーションと、 $J_d(P, K_s(x(k) - c(k)r(k)), \theta, T_d)$ の値を求めるためのシミュレーションを MATLAB 上で行なう（つまり、2 回のシミュレーションを実行する）。

STEP 3: 得られた $x(k_{\max})$ に対し、シミュレーションを実行し、燃費 $J(P, K_s(x(k_{\max})), \theta, T_d)$ とドライバ満足度 $g(P, K_s(x(k_{\max})), \theta, T_d)$ をもとめる。燃費 $J(P, K_s(x(k_{\max})), \theta, T_d)$ とドライバ満足度 $g(P, K_s(x(k_{\max})), \theta, T_d)$ について所望の結果が得られていれば、STEP 4 へ移る。そうでなければ STEP 1 に戻る。

STEP 4: シミュレーションを実行し、燃費 $J(P, K_s(x(k_{\max})), \theta, 80420)$ とドライバ満足度 $g(P, K_s(x(k_{\max})), \theta, 80420)$ をもとめる。燃費 $J(P, K_s(x(k_{\max})), \theta, 80420)$ について所望の結果が得られ、ドライバ満足度 $g(P, K_s(x(k_{\max})), \theta, 80420)$ が (4) 式を満たしていれば、 $x(k_{\max})$ を出力して終了する。そうでなければ STEP 1 に戻る。

4. 制御器の設計

3 章で示した設計法を用いて、実際の制御器を設計してみよう。

まず、制御器 K_s の 17 個の信号線およびパラメータに、ゲイン x_i ($i = 1, 2, \dots, 17$) を導入した。このゲインを導入した箇所を Fig. 7～Fig. 16 および、Table 4 の 1 列目に示す。また、Fig. 7～Fig. 16 の階層図を Fig. 5 に示す。

つぎに、 $T_d = 10000$ 、 $x(0) = [1 \ 1 \ \dots \ 1]^T$ 、 $r_i(k)$ の確率分布はベルヌーイ分布

$$\begin{cases} \mathbb{P}(r_i(k) = 1) = 0.5 \\ \mathbb{P}(r_i(k) = -1) = 0.5, \end{cases} \quad (10)$$

$a(k) = 0.2/(k+13)$ 、 $c(k) = 0.02/(k+1)^{\frac{1}{6}}$ 、 $\delta = 0.1$ 、 $k_{\max} = 50$ と設定した。

以上の条件のもとで、3.2 節に示したモデルフリー設計法を実行した結果が、Table 4 に示される $x(50)$ である。これは、CPU: Intel Core i7-3960 (3.30GHz)、RAM: 64.0GB

Table 4 Design parameters

x	Description	$x(0)$	$x(50)$
x_1	Ignition pause time at engine start-up	1.00	1.19
x_2	Minimum value of engine power	1.00	0.76
x_3	Engine speed to define engine start-up	1.00	1.01
x_4	Accelerator pedal position	1.00	2.09
x_5	Proportional gain (1)	1.00	1.70
x_6	Integral gain (1)	1.00	0.98
x_7	Time constant of low pass filter (1)	1.00	1.08
x_8	Torque required to motor 1	1.00	0.87
x_9	Time constant of low pass filter (2)	1.00	0.64
x_{10}	Torque required to motor 1 at engine start-up	1.00	0.68
x_{11}	Engine optimal speed	1.00	0.58
x_{12}	Proportional gain (2)	1.00	0.53
x_{13}	Integral gain (2)	1.00	0.55
x_{14}	Power required to motor 2	1.00	0.90
x_{15}	Target value of state of charge of battery	1.00	0.82
x_{16}	Integral gain (3)	1.00	0.62
x_{17}	Power required to battery	1.00	2.08

Fig. 5 Hierarchy chart of Fig. 7～Fig. 16

Fig. 6 Convergence of objective function $J_d(P, K_s(x(k)), \theta, 10000)$ for design

Table 5 Comparison of results of $K_s(x(0))$ and $K_s(x(50))$

	$K_s(x(0))$	$K_s(x(50))$
Fuel efficiency [km/L] $J(P, K, \theta, 80420)$	17.63	25.41
Driver satisfaction $g(P, K, \theta, 80420)$	100	100
Battery state of charge [%] $y_{16}(80420)$	60.17	61.90

の計算機を用いて約 7 時間の計算で得られたものである。このときの評価関数 $J_d(P, K_s(x), \theta, 10000)$ の更新ごとの変化を Fig. 6 に示したが、評価関数 $J_d(P, K_s(x), \theta, 10000)$ が約 126 に収束していることがわかる。

Table 5 は、ここで得られた制御器 $K_s(x(50))$ を用いたときの、燃費 $J(P, K_s(x(50)), \theta, 80420)$ 、ドライバ満足度 $g(P, K_s(x(50)), \theta, 80420)$ 、バッテリー充電容量の残量を表わしている。ここでは、 $K_s(x(0))$ を用いた場合 (サンプル制御器 K_s に対応) のものも併せて示している。これより、

- 燃費 $J(P, K, \theta, 80420)$ が 44.13% 改善
- ドライバ満足度 $g(P, K, \theta, 80420)$ が (4) 式の制約条件を満たす
- 終端時刻においてバッテリーの過度な放電が生じていないことが確認できる。また、Table 6 に、エンジン、モータ、バッテリーに関する出力 y_i ($i = 6, 7, \dots, 16$) の最大値、最小値、平均値を示したが、 $K_s(x(0))$ と $K_s(x(50))$ の場合で同程度であり、ここで得た制御器は、サンプル制御器 K_s と比較して、パワートレインに過度な負担を与えていないことも見て取れる。このようにして、モデル情報を利用することなく高い性能の制御器が得られる。

最後に、つぎの点を指摘しておく。ここで提案した設計手法は、モデルフリー型であるために、誰でも容易に実行できるという利点をもつが、それと引き換えに燃費改善のメカニズムを明らかにすることは難しい。しかし、設計パラメータ

Table 6 Range and average of signals y_i ($i = 6, 7, \dots, 16$)

(a) Case of $K_s(x(50))$

Signal	Maximum	Minimum	Average
y_6	2.3×10^3	-7.1	530.6
y_7	236.4	-46.9	21.8
y_8	27.8	-5.9	3.1
y_9	3.8×10^3	-5.5×10^3	-372.7
y_{10}	63.1	-153.6	-6.7
y_{11}	36.2	-28.1	-0.1
y_{12}	3.5×10^3	0	878.0
y_{13}	321.9	-208.0	-0.3
y_{14}	26.4	-45.6	-0.6
y_{15}	221.9	0	221.6
y_{16}	63.6	52.5	59.7

(b) Case of $K_s(x(0))$

Signal	Maximum	Minimum	Average
y_6	4.2×10^3	-8.1	937.3
y_7	242.6	-138.6	16.2
y_8	60.0	-30.0	4.4
y_9	7.8×10^3	-5.8×10^3	1.1×10^3
y_{10}	144.1	-153.6	-4.7
y_{11}	34.6	-30.0	-1.9
y_{12}	3.5×10^3	-15.7	873.4
y_{13}	400	-400	11.8
y_{14}	36.6	-50.3	0.8
y_{15}	226.8	0	221.9
y_{16}	87.1	49.9	63.4

x_i ($i = 1, 2, \dots, 17$) の設計前と設計後の値を比較することで、ある程度、燃費改善に寄与する重要なパラメータを特定することができる。実際、Table 4 を見ると、設計パラメータ $x_4, x_5, x_{11} \sim x_{13}, x_{17}$ の値は、設計アルゴリズムの実行前後で変化が大きくなっており、これら 6 個のパラメータが燃費改善において重要な役割を演じていることがわかる。

5. ま と め

本論文では、ハイブリッド自動車の制御器設計のベンチマーク問題を検討し、同時摂動最適化を利用したモデルフリー制御器設計法を提案した。その結果、燃費を約 44% 改善することに成功した。

謝辞 本研究の一部は、科学研究費補助金 若手研究 (A) #24686051 の助成を受けた。記して、謝意を表したい。

参 考 文 献

- 1) 安井裕司：ベンチマーク問題 2：ハイブリッドパワートレインを用いた通勤車両の燃費最適化，自動車技術会学術講演会前刷集，85-11, 5/8 (2011)
- 2) Y. Yasui: JSAE-SICE Benchmark Problem2: Fuel consumption Optimization of Commuter Vehicle Using Hybrid Powertrain, *Proceedings of World congress on Intelligent Control and Automation*, 606/611 (2012)
- 3) JSAE-SICE：自動車のモデリングと制御ベンチマーク問題，http://cig.ees.kyushu-u.ac.jp/benchmark_JSAE_SICE/
- 4) J.C. Spall: Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,

The diagram illustrates the control logic for the accelerator pedal position. It starts with two inputs: EngOptSpd (labeled with a '2' in a circle) and Engine Displacement (labeled with a '1' in a circle). EngOptSpd is multiplied by a gain of 0.10472 and then passed through a saturation block. Engine Displacement is multiplied by a gain of 1497. The outputs of these two paths are summed and then passed through a function block f(u). The output of f(u) is the EngSpd Accelerator Pedal Position, which is fed into a Reverse lookUP Table (BMEP_ENGSpd -> AccPosi). The output of the lookup table is the Accelerator Pedal Position, which is then multiplied by a gain of 1/4. This result is fed back into the saturation block and also passes through another saturation block (Saturation1) to produce the final Accelerator Pedal Position (labeled with a '1' in a circle).

Fig. 11 PID controller (1)

Fig. 12 Motor 1 controller

Fig. 13 Engine optimal speed calculator

Fig. 14 PID controller (2)

Fig. 15 Motor 2 controller

Fig. 16 Battery controller

IEEE Transactions on Automatic Control, **37**-3, 332/341
(1992)

5) Gamma Technologies, Inc.: <http://www.gtisoft.com/>

.....

[著 者 紹 介]

馬 場 一 郎

1987 年生. 2013 年京都大学大学院情報学研究
科システム科学専攻修士課程修了. 2013 年村田機
械 (株) に入社. 現在に至る.

東 俊 一 (正会員)

1976 年生. 2004 年東京工業大学大学院情報理
工学研究科博士後期課程修了. 同年より日本学術
振興会特別研究員, 2005 年京都大学大学院情報学
研究科助手, 2007 年同助教, 2011 年同准教授と
なり現在に至る. この間, 2004~2005 年ジョー
ジア工科大学客員研究員, 2009~2010 年ペンシルバ
ニア大学客員研究員. ハイブリッドシステム制御
論の研究に従事. 博士 (工学). 2005, 2008 年計
測自動制御学会論文賞, 2011 年度同学会制御部門
パイオニア賞などを受賞. IEEE, システム制御情
報学会の会員.

杉 江 俊 治 (正会員・フェロー)

1978 年京都大学大学院修士課程精密工学専攻修
了. 84 年同博士後期課程研究指導認定退学. 78~
80 年日本電信電話公社勤務. 84 年大阪府立大学工
学部助手. 88 年京都大学工学部助教授を経て, 現
在情報学研究科教授. ロバスト制御, システム同
定, 非線形制御などの研究に従事. 工学博士. 91
年システム制御情報学会榎本記念論文賞. 98, 2008
年同学会論文賞. 94, 2000, 2003, 2007, 2008 年
計測自動制御学会論文賞を受賞. 日本鉄鋼協会,
ロボット学会などの会員. IEEE Fellow.

.....