

モデルベース開発のためのエンジンモデルの ECU 組込み技術

Technology for Embedding Engine Models in ECU for Model-based Development

赤 崎 修 介*
Shusuke AKAZAKI
清 水 治 喜**
Haruki SHIMIZU

久 保 知 也*
Tomoya KUBO
山 崎 文 誉***
Fumitake YAMAZAKI

千 田 耕 平*
Kohei CHIDA

要 旨

内燃機関の制御では、モデルベース開発のV字プロセスにおいて上位レベルから下位レベルまでの制御対象モデルに統一性を持つことが重要である。そこで上位レベルの一次元エンジンモデルと中位レベルの平均値エンジンモデルにパラメータ適合を用いて一定の誤差内になるよう関連性を持たせた。そしてHardware-in-the-Loop Simulationにて使用する平均値エンジンモデルをそのままECUに組込むための環境を構築した。モデルは数式シミュレーションツールによって記述され、Cコードに変換される。このコードをもとにECU搭載用のCコードに自動で変換するツールを開発した。変換されたCコードをCPU評価ボードで実際に動作させ、エンジンのデバイス制御の演算周期である10 ms以内に動作可能であることを確認した。これにより、モデルベース開発の上位レベルのモデルと中位レベルでのモデルを関連づけ、下位レベルでのECUへのモデル実装も可能となり、V字プロセスのエンジンモデルの再利用性を向上することができた。

ABSTRACT

Consistency in the control subject model from the top level to the lower level of the V-process used in model-based development is an important factor when designing internal combustion engine control. For this reason, parameter calibration was used to provide relevance between the 1D engine model in the top level and the mean value engine model in the middle level so that the deviation is within a certain level. In addition, an environment was constructed for embedding the mean value engine model used by Hardware-in-the-Loop Simulation as is in an ECU. The model is described using a formula simulator and converted to C code. A tool was developed that automatically converts this code into C code for mounting in an ECU. The converted C code was actually operated using a CPU evaluation board, and was confirmed to operate in 10 ms or less, which is the calculation cycle for engine device control. This made it possible to link the models in the top and middle levels of model-based development and also enabled model mounting to an ECU in the lower level, which increased the reusability of V-process engine models.

1. ま え が き

近年、ダウンサイジング過給エンジンの技術やLow Pressure Loop (LPL)-EGR⁽¹⁾、希薄燃焼の研究など、熱効率の向上を中心とした燃費向上技術が注目されている。

高効率エンジンには専用のセンサやデバイスおよびソ

フトウェアが必要であるが、一方で排気処理のための触媒改良などシステム全体が複雑になってきている。これまで単体ベンチテストでのマッピングなど比較的単純な作業によって適合が計られてきたが、量産へ向けてはパラメータ設定に多大な時間を要するのが現状である。最適化ツールやシミュレーションの進化により効率の良いデータサン

* 四輪 R&D センター

** 株式会社両毛システムズ

*** サイバネットシステム株式会社

リング法や応答曲面などを用いた誤差最小化のための最適化による対応が試みられている⁽²⁾。また、各国の排ガス規制も厳しくなることが予想され、燃費と排ガス規制対応のために空燃比制御やEGR制御の精度向上など、エンジン制御に対する要求が増加することが懸念されている。

一方、モデルベース開発の推進から、検証用のツールとしてHardware-in-the-Loop Simulation(HILS)⁽³⁾が普及してきている。制御システムの設計では、HILSと実際のECUを用いた制御ロジックの構築が試みられている。モデルベース開発の効果を向上するには、モデルの再利用性を高めることで、モデルの改良進化を促すことが望ましい。そこで、HILS用に構築した制御系設計および検証用に適合され、精度を保証された平均値エンジンモデル⁽⁴⁾をそのまま制御用ECUへ実装することで、実際のエンジンの状態予測器としてECU内にて再利用することを検討した。これまでは、CPUの演算性能が不足するため、エンジンの一部をモデル化して組込む例⁽⁵⁾はあったが、吸気系から燃焼および排気系まで含めて実行した例は少ない。今回用いた平均値エンジンモデルは、直噴過給エンジンを数式シミュレーションツールで作成され、あらかじめ数式モデルでの最適化を実施した。その後、ECUへの搭載を検証するため、市販のCPU評価ボードを使って実行し、演算時間や演算結果を比較した。

2. HILS と ECU 組込みモデルの共通化と課題

2.1. モデルベース開発用エンジン制御 HILS

Fig. 1 に示すようにモデルベース開発の中間レベルにおける機能設計時に Model-in-the-Loop Simulation(MILS)またはHILSを用いて機能要求を満たす制御ロジックを設計する。制御対象モデルとしてエンジンモデルを用いているが、モデルは実際のエンジンと定常状態の挙動が一致するようにパラメータを適合される。そして過渡状態の動きについては、定常と同様に再現されていることが望ましい。機能設計時のMILSと検証時のHILSで共通のモデルを用いれば、設計時と検証時に違いは生じない。ただし、

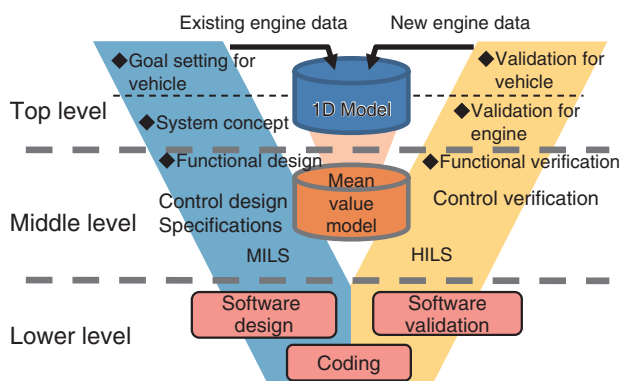


Fig. 1 Standard V-process

MILSでのモデルは実時間実行の必要はないが、HILSにおいては実際のECUを用いるため、実時間実行の必要がある。

2.2. ECU 組込みのための目標設定

HILS上で実時間実行可能なモデルは、比較的簡易な演算で構成されているので、ECUでの演算も可能であると考えた。そこで、モデルを再利用して観測の難しいポイントの温度や圧力などを予測するために使用できるか検討した。実エンジンとは異なり、ECU内のモデルは測定に必要なセンサを自由に搭載可能である。今回は燃焼モデルによるトルクの演算に注目して開発した。駆動力制御からの要求ではトルク演算誤差が全開トルクの5%である10 Nm以内であったので、これを満足できるようにモデルによるトルク演算誤差も、10 Nm以内とした。制御周期は、スロットルなどのデバイス制御の周期が10 msであることから10 ms以内とした。以下に開発目標を示す。

- (1) 制御系設計時や制御性の検証時に使用するHILS用平均値エンジンモデルを吸気から排気まで含めたフルモデルでECUに実装可能とする。
- (2) 数式モデルから自動生成するCコードをECUに搭載可能なコードへ自動修正するツールを開発する。
- (3) エンジンモデルのECU演算周期を10 ms以内であること。トルク演算誤差は10 Nm以内とする。

3. 平均値エンジンモデルと実機適合

3.1. 組込み用のエンジンモデル

これまでの段階的システム同定⁽⁴⁾のエンジンモデルのパラメータ適合の流れをFig. 2に示す。V字フローの上位レベルから中位レベル、下位レベルといった各レベルごとに使用するツールが違うため、各モデルの設定パラメータをシミュレーション結果が一致するように適合する構成とした。開発初期には現行のエンジンを用いて一次元モデルを適合し、次に平均値エンジンモデルへとステップを踏んで適合していく。そして最終段階では、モデルを制御ECUへ実装するという開発ステップである。開発エンジンの改良とともに、このステップを繰り返すことでシステム開発レベルも向上していく。

Fig. 3に今回開発したエンジンモデルを示す。ダウンサイジング過給直噴ガソリンエンジン1.5 L 4気筒を数式処理機能をもつモデリング・シミュレーションツールのMapleSimを用いて平均値モデルで作成した。モデルの部品構成としては、実際のエンジン部品の組み合わせとなっている。これは部品の構成や諸元変更の際にモデルの変更が最小限となるようにモジュール化を考慮し配置した。これにより部品の仕様変更に対応する時は、部品の特性の合わせ込み結果を部品単位の単体計測データやパラメータ変

更にておこなうことができる。MapleSim の部品コンポーネントは、標準で用意されているものがあるが、燃焼モデルなど標準で準備されていないものは、数式化してカスタムコンポーネントにて作成した。カスタムコンポーネントで作成した LPL-EGR と燃焼モデルを説明する。

3.2. LPL-EGR モデル

LPL-EGR は排気ガスの導入遅れが反映されていない場合 HILS での制御系設計と実エンジンの過渡状態での挙動が不一致となり課題となる。そこで、EGR ガスの遅れをパラメータ設定で実際の挙動が再現可能かどうかシミュレーションにて確認した。Fig. 4 に MapleSim による PC 上でのシミュレーション結果を示す。この結果は

LPL-EGR バルブの開度制御でステップ応答をおこなったものである。EGR_rate は、吸気管に入る EGR ガスの割合を示している。LPL-EGR バルブが閉じる際に吸気管の入り口にあるスロットルが開いているときの rate 1 と閉じているときの rate 2 を比較する。スロットルを開いている rate 1 では、即座に EGR ガスは応答しているが、スロットルが閉じているときの rate 2 は、吸気管前のインタークーラなどのチャンバ内でのガス交換時間が大きくなり LPL-EGR ガスが完全になくなるまでに 2 s という長い時間が必要であることがわかる。この応答のむだ時間と応答遅れが実測の場合とシミュレーションと同じ 2 s であることから LPL-EGR モデルとしてガスの遅れを表現できていると判断した。

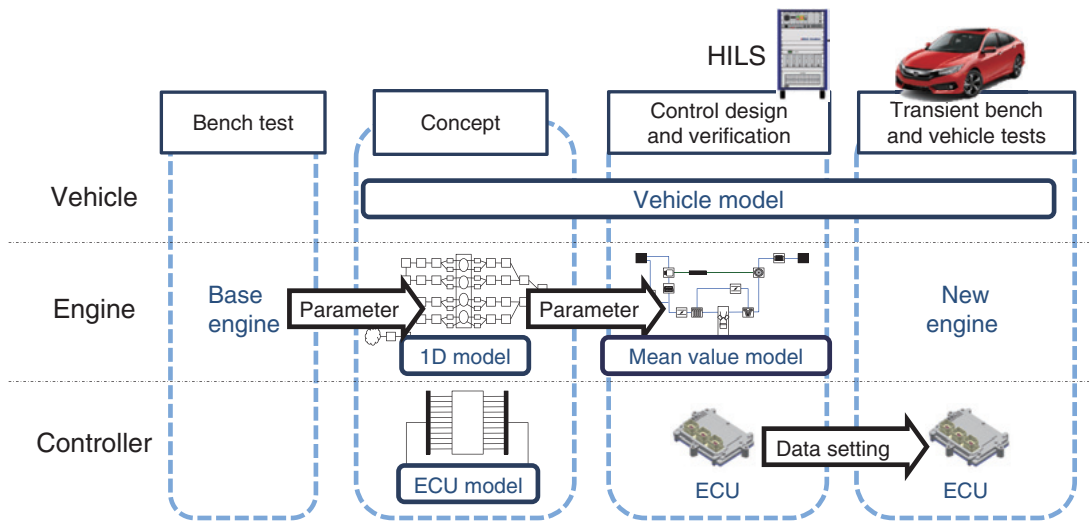


Fig. 2 Model parameter identification flow for engine plant model

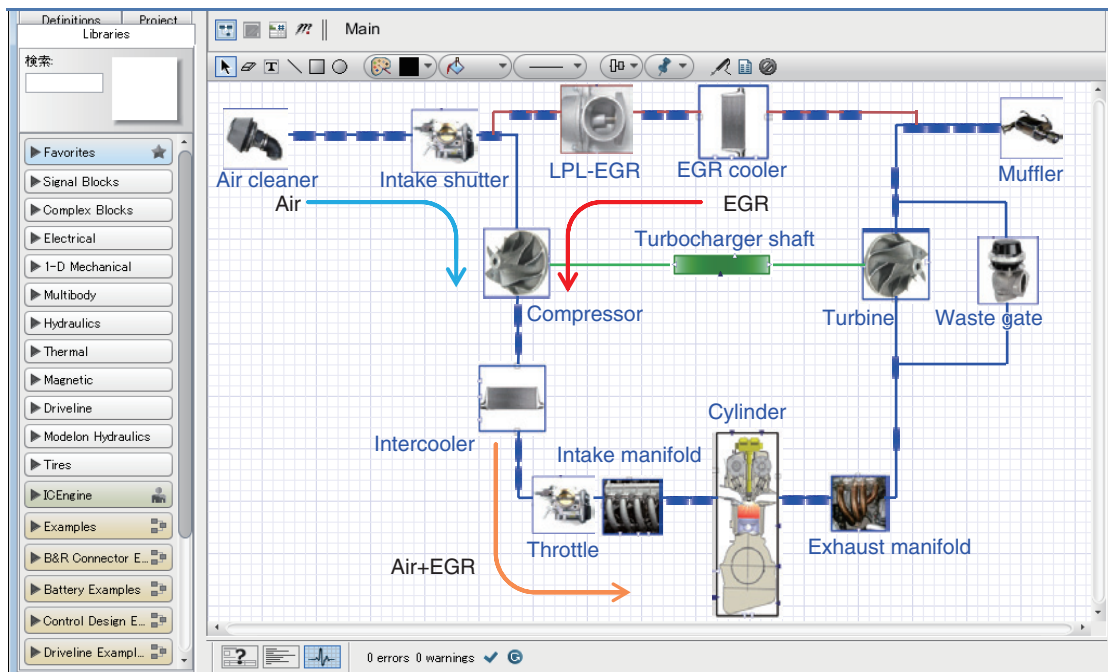


Fig. 3 Mean value engine model with turbocharger and LPL-EGR

3.3. エンジン燃焼モデル

通常の燃焼解析装置などの燃焼演算では Wiebe 関数が用いられることが多いが、クランク角 0.5 deg から 1.0 deg 程度のステップによる演算回数が必要なため、HILS に採用するためのリアルタイム実行が容易ではない。そこで、熱発生のパターンをあらかじめ実測したデータの代表点で設定できるようにして計算式の簡単化をおこなった。HILS で使用するモデルでは、この設定値を実エンジンと一致するように回転数と吸入空気量、点火時期等のエンジン制御パラメータの関数とした。エンジンでの測定は、基準のキスラー社製センサ (305B) と AVL 社製燃焼解析装置 Indicom を用いた。

Fig. 5 に示すように熱発生量を基に赤いポイントが設定した代表点を表す。演算開始の点、熱発生開始位置とピーク値の 3 点に、二階微分値のピーク位置 2 点を加え、さらに質量燃焼割合 5 % 位置と燃焼終了位置付近の三角で示す 2 点、トータルで 7 点とした。

この各ポイントを直線近似式で補完計算をおこないクランク角度に対する熱発生パターンを求めた。これと行程容積から燃焼期間中の筒内圧を演算する。

式 (1) に示すのは、筒内圧と行程容積、比熱比から熱発生を算出する式である。この式 (1) を変形し筒内圧の式 (2) を求めた。

$$\frac{d}{d\theta} Q(\theta) = \frac{1}{\kappa - 1} \cdot V(\theta) \frac{d}{d\theta} \cdot P(\theta) + \frac{\kappa}{\kappa - 1} \cdot P(\theta) \cdot \frac{d}{d\theta} V(\theta) \quad (1)$$

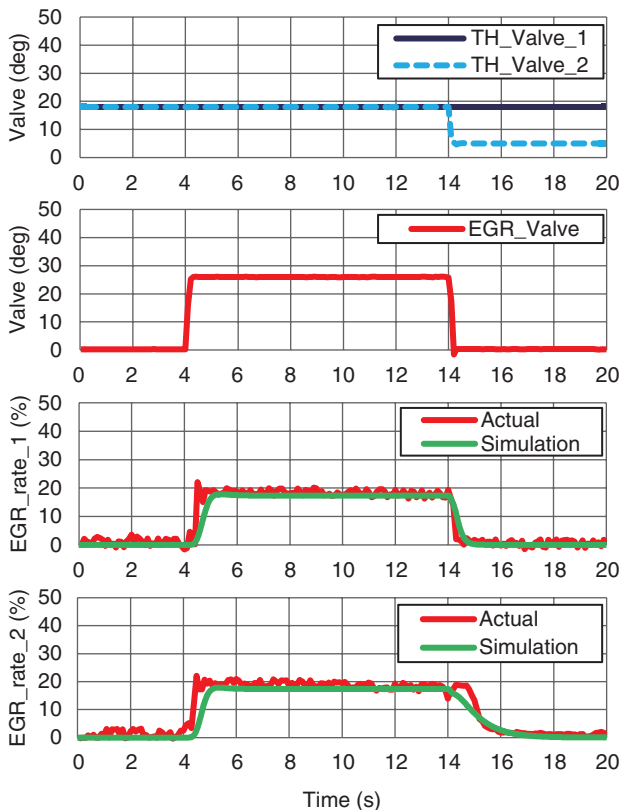


Fig. 4 LPL-EGR simulation results

上記、筒内圧と熱発生量の関係式のパラメータ

$V(\theta)$: 容積 (cm^3)

$P(\theta)$: 圧力 (kPa)

$Q(\theta)$: 熱量 (J)

κ : 比熱比

$$P(\theta) = \left\{ \int_0^\theta \left[\frac{d}{d\theta'} Q(\theta') \right] \cdot (\kappa - 1) \cdot V(\theta') \cdot d\theta' + \frac{P_{ini}}{V(0)^{-\kappa}} \right\} \cdot V(\theta)^{-\kappa} \quad (2)$$

ただし

θ' : クランク角度 0 から θ までの積分値

P_{ini} : 初期圧力

IMEP を式 (3) に NMEP を式 (4) に PMEP を式 (5) にて記す。

$$IMEP = \frac{1}{V_d} \int_{V_{-360}}^{V_{360}} P \cdot dV \quad (3)$$

$$NMEP = \frac{1}{V_d} \int_{V_{-180}}^{V_{180}} P \cdot dV \quad (4)$$

$$PMEP = IMEP - NMEP \quad (5)$$

V : Volume (m^3)

dV : Derivative of Volume

V_d : $V_{max} - V_{min}$ (m^3)

P : Pressure (kPa)

IMEP: Indicated Mean Effective Pressure

NMEP: Net Indicated Mean Effective Pressure

PMEP: Pumping Indicated Mean Effective Pressure

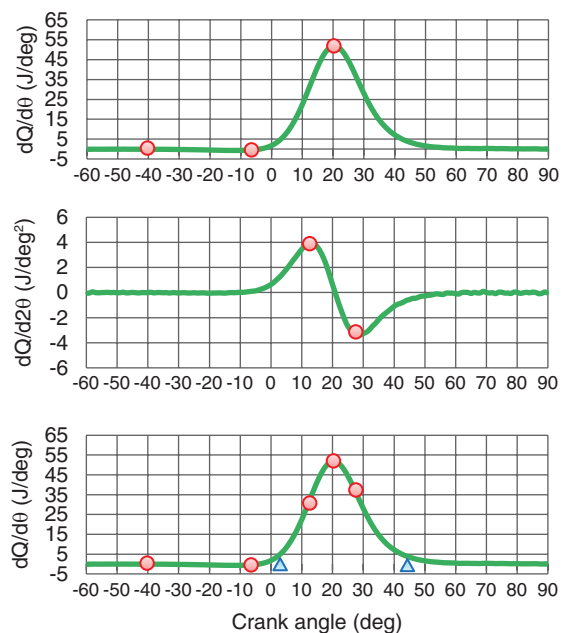


Fig. 5 Heat release rate of combustion and setting points for model calibration

燃焼期間中の NMEP は式 (4) の通り、筒内圧力を行程容積において積分することで算出している。次にポンピングロス (PMEP) は、トルクの演算に対する影響が比較的小さいことから吸気マニホールド内の圧力と回転数によるマップから実測の数値を検索して簡易的に求めた。以上により IMEP の演算時間を短縮化することができた。

3.4. 一次元モデルとの比較

一次元モデルは実際のエンジンと適合する事例⁽⁶⁾からみられるように、定常から過渡において一般に利用されている。実エンジンの計測結果と一次元モデルおよび平均値エンジンモデルのパラメータを一致するように適合させた。

平均値エンジンモデルへの適合は、前記燃焼モデル 7 点の設定値をトルク誤差が最小になるようにした。

Fig. 6 に計測したエンジンのトルクに対して二つのシミュレーションについての適合結果を比較する。この結果から二つのシミュレーションは、低出力時から高出力時にかけて線形であり一次元モデルと平均値エンジンモデルのトルク演算値の誤差は、10 Nm 以内であった。この結果から ECU に組み込むモデルとして問題ないと判断した。

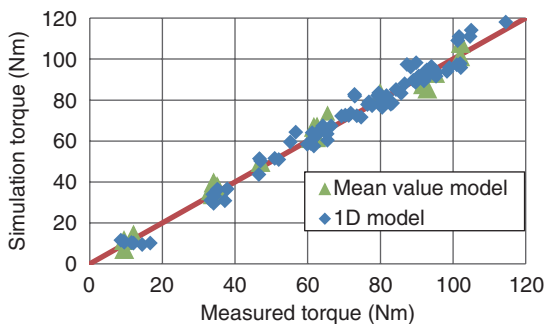


Fig. 6 Comparison of simulation results

4. ECU 組込みツールの開発

4.1. エンジンモデルのコード変換

これまで紹介したモデルは MapleSim にて記述されている。MapleSim の自動 C コード生成機能によりモデルから C コードを生成する。この C コードは、MATLAB の Simulink 用と一般コードの二通りが選択可能であるが、今後のツールのバージョンアップなどによる影響を可能な限り小さくするため、一般コードを採用した。次にこれを ECU 組込みのためにプログラム作成ルールに則った組込み用のプログラム形式へと変換する必要がある。それをターゲット CPU のコンパイラを使って HEX ファイルを作成してダウンロードすることで組込みは完了する。

今後の開発時に随時更新されるさまざまなエンジンのモ

デルにタイムリーに対応するため、組込み用の C コードに自動変換するツールを開発し作業効率を向上した。

Fig. 7 に開発したコード変換ツールの実行画面を示す。このように簡単な画面上で対象ファイルを選択するだけで実行することができる。開発ツール本体の開発は継続的なバージョンアップを考慮し MATLAB にて作成した。コード変換ツールの機能は以下である。

(1) 変数の定義などプログラム作成ルールの適用

変数名変更、型名変更、コメント文の追加、リミット処理、ゼロ割防止処理、キャスト演算子の型宣言、関数のプロトタイプ宣言、変数の追加

(2) 処理負荷軽減

倍精度を単精度へ変更、浮動小数点での割り算を専用関数へ変更

(3) 既存ライブラリの適用

ヘッダファイルの読み込み処理

このツールによる C コードの変換に要する時間は約 11 分で、コードサイズは 500 k バイトであった。

4.2. CPU ボードでの動作確認

Fig. 8 に動作確認をおこなった評価ボードの外観を示

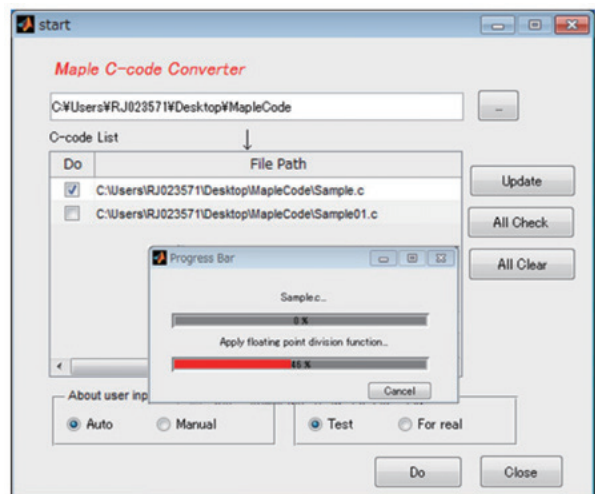


Fig. 7 File setting screen of conversion tool

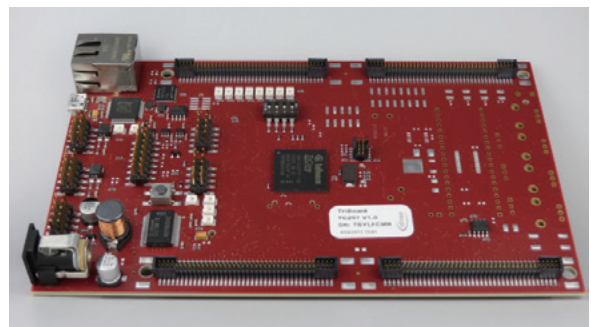


Fig. 8 Appearance of CPU evaluation board

す。評価ボードは、マルチコア CPU でクロック周波数を任意に設定できるため、インフィニオン社製の TC297 評価ボードを選択した。この評価ボードにプログラムを実装して演算時間を計測した。このボードの性能は、ROM と RAM の容量および処理能力で比較して現在の量産型 ECU のほぼ 2 倍である。

これは、今後の組込みモデルの拡大に対応できるようあらかじめ余裕をもって設定した。今回の時間測定では、通常のエンジン制御やその他の演算部は搭載していないので、割り込み等の影響は考慮していない。

Fig. 9 にオイラー法とルンゲクッタ法による演算時間の測定結果を示す。プラントモデルは ECU で演算される際に、離散化⁽⁷⁾された信号を用いるが、その離散化の手法として代表的なオイラー法とルンゲクッタ法を比較した。CPU のクロック周期は、標準の 200 MHz を想定した。ECU 上でのエンジンモデルの動作は可能な限り周期は短いことが望まれるが、吸入空気量によるトルクの制御周期は 10 ms であることから、それに十分に間に合えばよい。比較結果では演算量の多いルンゲクッタ法にて実行時間が 2.5 ms であった。計測時のエンジン運転条件は、LA4 モードでの車両運転条件である。

一方で離散化には、信号入力や演算値が前回演算時と今回演算時にステップ状に変化する特徴がある。このため差分が大きくなるとオーバーフローなど演算処理が不安定になりやすい。特にオイラー法による簡易的な離散化処理では、演算周期が延びると内部演算結果の差分が大きくなり、演算に段差が生じるため短周期が要求される。しかし全体の演算回数が増えるため、エンジン回転数が高くなると点火時期や燃料噴射などイベント割り込みの増加で、10 ms 周期の演算終了に時間が掛かってしまう。今回の結果では、ルンゲクッタ法でも、演算時間は 2 倍程度延びたものの目標時間内であり、安定した動作を得ることができたので、ルンゲクッタ法を選択した。

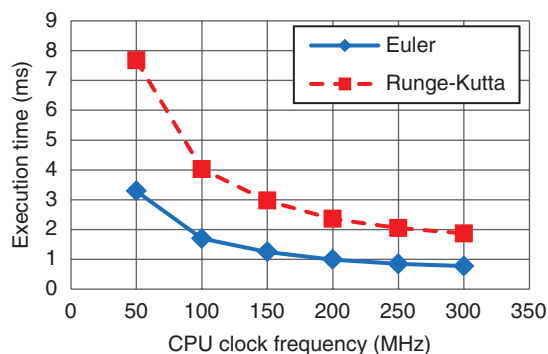


Fig. 9 Comparison of execution time with Euler's and Runge-Kutta methods

Table 1 Performance comparison between CPU and HILS

Execution condition	Evaluation board	HILS
Data length	16 bit	32 bit
Control cycle [ms]	10	0.1

5. HILS とモデル組込み ECU の演算結果の比較

エンジンモデルを HILS に搭載した演算結果と ECU を模擬した評価ボードに実装した場合の演算結果の差異を検証した。検証にて走行した運転パターンは、これまでと同じ LA4 モードである。演算結果の比較には、HILS と評価ボードを CAN にて同時にモニタリングして 100 ms で計測した。

使用した HILS と ECU の演算性能の差を Table 1 に示す。

データ長が 16 ビットと 32 ビットで違うのと演算周期が 10 ms と 0.1 ms と異なる。このためモニタリングした演算結果に有効桁数の違いによる桁落ちやモニタの更新周期の違いがあり、単純に差を求めると細かなノイズのような誤差波形が観測される。

Fig. 10 にトルク演算値を LA4 モードの 2 山目について比較した結果を示す。細かなノイズのような誤差が発生しているが、これは先に述べた CPU 性能の違いのためである。

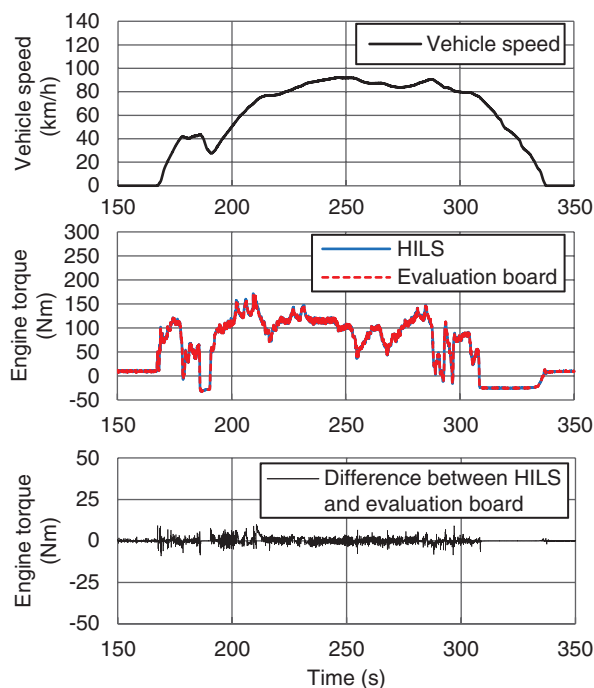


Fig. 10 Comparison of calculation results in LA4 mode

これまで述べたように HILS に搭載した平均値エンジンモデルを ECU に組込むことで、HILS による制御系設計時と動作検証時そして実際のエンジンの制御時のエンジンモデルを統一化することができた。直噴インジェクター一体式の筒内圧センサ⁽⁸⁾など、燃焼解析装置と同等の高精度な燃焼演算ができるようになってきている。今後、こうしたセンサと組み合わせることで、エンジン燃焼モデルを改良しながら、詳細な筒内情報を用いたリーンバーン制御や、エンジン振動の抑制などにも応用が期待できる。

6. まとめ

制御設計や検証時のエンジンモデルと同じものを ECU に組込む環境を構築した。これにより、V 字プロセスの上流から、下流のエンジンモデルの誤差や不具合が、修正された場合に他プロセスでの不具合も反映される。手戻りが減りプロセス全体の効率化が期待できる。

下記にその内容をまとめる。

- (1) 制御系設計時や制御性の検証時に使用する HILS 用平均値エンジンモデルを吸気から排気まで含めたフルモデルで ECU に実装可能とした。
- (2) MapleSim から自動生成される C コードを ECU に搭載可能なコードへ自動で変換するツールを開発した。
- (3) CPU 評価ボードを用いてエンジンモデルの演算実行時間が目標の 10 ms 以下の 2.5 ms であった。HILS とのトルク演算誤差は最大出力値の 5 % 以内の 10 Nm 以下であった。

参考文献

- (1) Siokos, K., Koli, R., Prucka, R., Schwanke, J., Miersch, J.: Assessment of Cooled Low Pressure EGR in a Turbocharged Direct Injection Gasoline Engine, SAE Int. J. Engines 8(4), p. 1535-1543, (2015)
- (2) 梶洋隆, 喜多一: エンジン制御パラメータの実験ベース進化的多目的最適化: YAMAHA MOTOR TECHNICAL REVIEW, No.43, (2007)
- (3) 小堂智史, 藤本茂希, 小野高志, 鈴木哲也: パワープラント系制御開発用ドライビングシミュレータ, Honda R&D Technical Review, Vol.26, No.2, p.111-118
- (4) Guzzella, L., Onder, C.: Introduction to Modeling and Control of Internal Combustion Engine Systems, Springer, (2010)
- (5) 藤本茂希, 大高義行, 小堂智史: MBD 実現に向けたエンジンプラントモデルと段階的システム同定環境の構築, Honda R&D Technical Review, Vol.24, No.2, p.120-127
- (6) 窪山達也, 山田敏生, 森吉泰生, 大瀧康宏, 畑村耕一,

森川弘二: 過給ガソリン機関の 1D シミュレーションモデルの構築, 自動車技術会 2015 年春季大会予稿集, No.3-15S, p.63-67, (2015)

- (7) 山本裕: サンプル値制御理論の回顧と展望, 計測と制御, Vol.40, No.1, p.76-82, (2001)
- (8) 赤崎修介, 井上泰輔, 鈴木哲也, 林貴之, 高奥淳司: 直噴インジェクター一体型筒内圧センサの改良とフィードバック型エンジントルク制御手法の開発, 自動車技術会論文集, Vol.47, No.1 p.17-22, (2016)

■ 著 者 ■



赤崎 修 介



久保 知 也



千田 耕 平



清水 治 喜



山崎 文 誉