

新HILS「CRAMAS-X」の開発

Development of New HILS “CRAMAS-X”

丸山 晃
Akira MARUYAMA

前田 勉
Tsutomu MAEDA

木本 恭平
Kyohei KIMOTO

川島 英嗣
Hidetsugu KAWASHIMA

要旨

自動車用の電子制御装置（以下、ECU）開発における評価では、開発期間短縮やコスト削減を目的に、シミュレータが活用されている。このシミュレータをHILS（Hardware-In-the-Loop Simulator）と呼び、当社では、独自開発したHILS「CRAMAS」を活用し、実機を必要としない評価を実現している。

近年の自動車制御は、複数 ECU が連携する制御が増加しており、その評価を行うシステム評価環境の構築には、設備投資の増大や必要機材の準備に時間を要するなどの課題がある。

また、自動車用機能安全規格（ISO26262）が制定され、制御開発に使用するツールの信頼性は、ユーザ自身が証明しなければならず、その評価用機材の準備や工数が膨大となる。

当社では新 HILS「CRAMAS-X」を開発し、これらの課題を解決するための機能を実現したので、ここに紹介する。

Abstract

As for the evaluation for the development of Electronics Control Unit (hereinafter “ECU”), a simulator is used to aim to reduce the development period and cost. This simulator is called HILS (Hardware-In-the-Loop Simulator), and we realize the evaluation without actual vehicle by utilizing originally developed HILS, “CRAMAS.”

The recent vehicle control in which several ECUs cooperate have been increasing. As for the establishment of the system evaluation environment for the evaluation, there are problems such as the increase of equipment investment and taking time for preparation of required equipment.

As automotive functional safety standard (ISO26262) has been established, user must prove the reliability of tools used for the control development by himself / herself. As the result, it takes huge man-hour to prepare evaluation equipment and evaluate the tools.

DENSO TEN developed a new HILS, “CRAMAS-X,” and realized the function to solve the above problems. We introduce this “CRAMAS-X” in this paper.

1. はじめに

自動車用の電子制御装置（以下、ECU）開発における評価では、開発期間短縮やコスト削減を目的に、シミュレータが活用されており、制御対象を模擬するシミュレータと ECU を接続して評価を

実施している。

シミュレータは、制御対象の物理現象を数式化した制御対象モデルを搭載し、ECU からの信号を元に制御対象モデルを計算、その結果である車両状態を電気信号に変換して、ECU に出力することでシミュレーションを行う。このシミュレーショ

ンを行うシミュレータを HILS (Hardware-In-the-Loop Simulator) と呼ぶ (図 1)。当社では HILS 「CRAMAS」を開発し、主にパワートレーン系 ECU で実機を必要としない評価を実現している。

従来の自動車制御では、各 ECU が個別に制御を行っていたが、近年は複数 ECU が連携する制御が増加しており、その評価を行うシステム評価環境は、連携制御に関わる全ての ECU とそれらの制御対象を模擬するために、複数の HILS が必要となり、次の二つの課題があった。

課題①：評価対象外の ECU 準備が困難

課題②：複数の HILS 準備による設備投資増大

また、自動車の電気／電子部品と、マイコンに搭載されるソフトウェアの開発において、自動車用機能安全規格 (ISO26262) が制定され、開発に使用するツールの信頼性についても規定されている。パワートレーン系 ECU は、その機能から安全性要求レベル (ASIL) が高くなり、ツールの信頼性はユーザ自身が確認する必要があるため、次の課題がある。

課題③：ツールが正しく動作することを証明するのに時間がかかる

当社では、これらの課題を解決するために、新しい HILS 「CRAMAS-X」を開発した。この稿ではその機能実現方法について述べる。

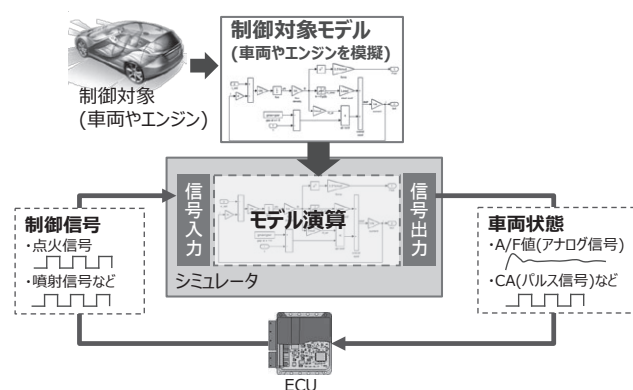


図 1 Hardware-In-the-Loop Simulator

2. 課題への対応

前章で述べた課題を解決するため、「CRAMAS-X」で実現すべき要件を次のように定義した。

課題①：評価対象外の ECU 準備が困難

⇒要件① ECU の模擬

ECU に搭載されるソフトウェアそのもの（以下、ECU ソフト）を、HILS に搭載する。

課題②：複数の HILS 準備による設備投資増大

⇒要件②マルチコアの駆使

複数の ECU と制御対象を HILS1 台で模擬するために、マルチコア CPU を採用する。

課題③：ツールが正しく動作することを証明するのに時間がかかる

⇒要件③ I/O 故障有無の自己診断

FPGA (Field-Programmable Gate Array) に実装した診断ロジックが各端子の故障有無を確認する。

要件①②③の詳細について、次以降の章で述べる。

3. ECUの模擬

ECU ソフトを HILS に搭載するため、ECU ソフトの実行に必要な機能 (OS、レジスタなど) を抽出し、開発した。この章では、最も難易度が高い「割込処理」の模擬を紹介する。

3.1 ECUの割込処理

HILS は一般的には周期処理だけで、割込処理は存在しない。一方、エンジン ECU には最適なタイミングで点火するために、周期処理以外に二種類の割込処理が存在し、それらの模擬が必要となる。

- ・ クランク角処理（以下、CA処理）
クランク角センサ入力をもとに実行されるハードウェア割込処理
- ・ 点火処理
点火タイミングを算出する処理の演算結果をもとに実行されるソフトウェア割込処理

3.2 「割込処理」模擬

HILSの演算周期内に実行されるECUの周期処理、割込処理をまとめ、一つの周期処理として実行することで、「割込処理」を含めたECUソフトの実行順序を実際のECUと同じ順序とした。HILSの周期処理の最初に、演算周期内の全処理をリストアップ後、「処理リスト（実行順序）」を作成し、「処理リスト」の順番に従って処理を実行することで、HILSでの割込処理を実現している（図2）。

しかし、リスト作成後に点火処理の実行タイミング（処理の実行順序）が変更される場合を考慮しなければ、ECUの実行順序と差異が発生してしまうため、処理リストに実行時刻の情報を持たせ、実行時刻の変更がある場合に「処理リスト」を更新することで、ECUと同じ実行順序を実現した（図3）。

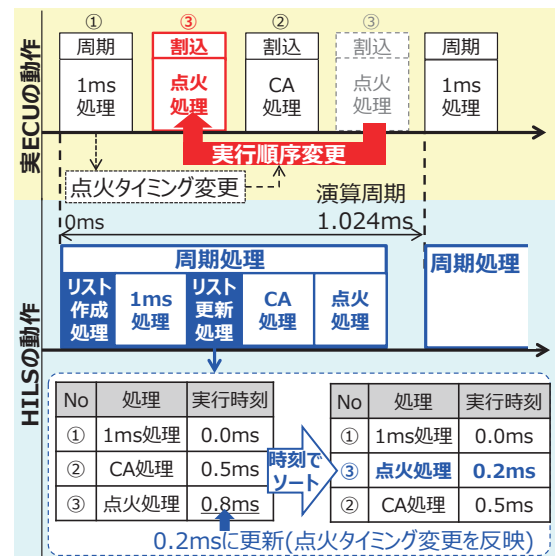


図3 「処理リスト」の更新

3.3 効果

HILSでエンジンとエンジンECUを模擬したシミュレーションを実行し、割込処理の「模擬なし」、「模擬あり」でエンジン回転数を比較した。

「模擬あり」では、実ECUとの比較でエンジン始動時の挙動が改善され、ECU模擬精度が向上していることを確認した（図4）。

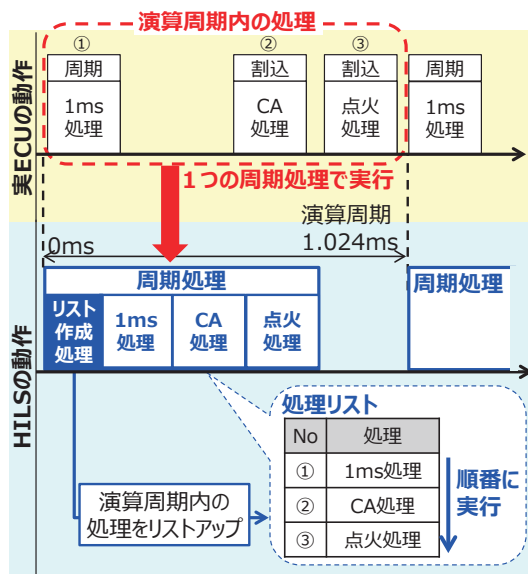


図2 「処理リスト」の作成

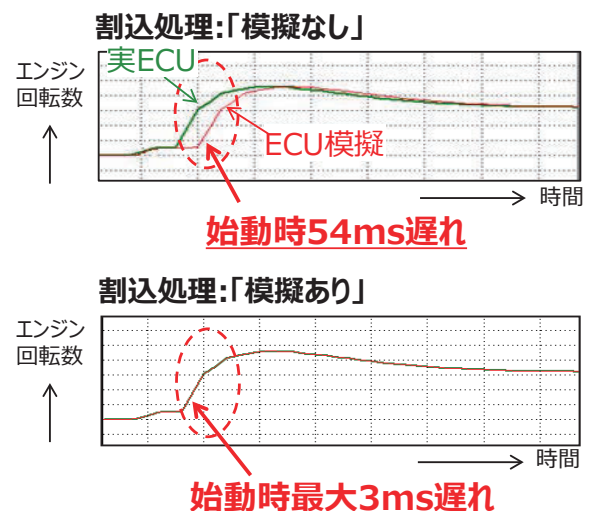


図4 「割込処理」模擬無／有のエンジン挙動比較

4. マルチコアの駆使

「CRAMAS-X」は、産業用 PC でも使用可能な高性能マルチコア CPU を採用している。HILS で演算するモデル（制御対象モデル、ECU モデル）をマルチコア CPU の各コアに割り当て、演算負荷を軽減することを狙いとしている。

ただし、HILS に搭載するモデルを暫定的にコア配置した場合、モデルの組合せや実行する評価パターンによっては処理負荷が増大し、処理抜けが発生する場合がある（図5）。そのため、モデルの組合せや実行する評価パターンに応じて、各コアへのモデル配置の最適化が必要となる。

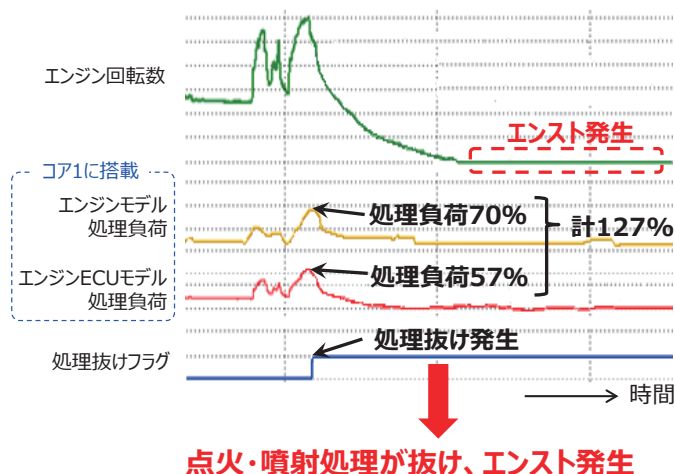


図5 処理負荷増大による処理抜けの発生

4.1 「処理完了判定式」によるコア配置

処理抜けを防ぐには、各モデルが演算周期内に処理を完了する必要があるため、演算周期内での処理完了可否を周期内の処理時間の総和から判定する「処理完了判定式（以下、判定式）」を考案した（図6）。シミュレーションを実行し、処理抜けが発生した場合、処理抜け発生時の各モデルの処理時間を計測し、その時間と判定式からコア配置を自動的に決定する手法を実現した。

例:モデルA、Bを同一コアに配置

	演算周期	処理時間
モデルA	1.0ms	0.50ms
モデルB	2.5ms	1.25ms

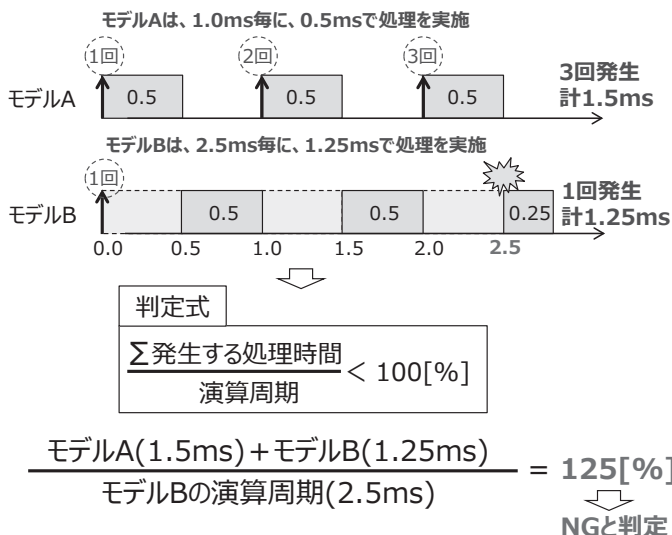


図 6 処理完了判定式

4.2 効果

処理抜けが発生した際に、判定式を用いたコア再配置を実施、再度シミュレーションを実行することで、処理抜けが発生しなくなることを確認した(図7)。このように、モデルの組合せや実行する評価パターンに応じた、モデルのコア配置を可能にした。

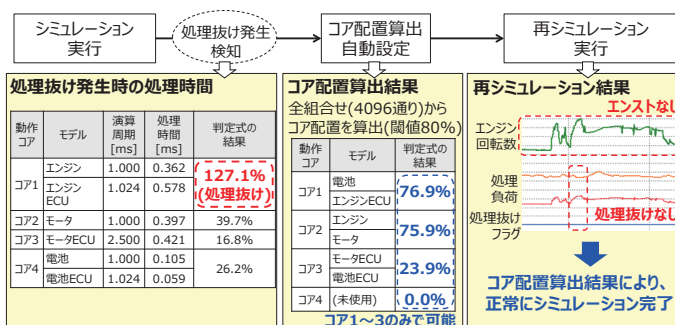


図7 コア配置算出による再シミュレーション

5. I/O故障有無の自己診断

HILS 評価を実施する前に、I/O の故障有無を確認する場合、従来はオシロスコープなどの計測器を準備し、手作業および目視で実施していた。

「CRAMAS-X」は、I/O ボードに搭載された FPGA に自己診断ロジックを実装し、HILS 自身が各端子の故障有無を特定する。

5.1 FPGAロジックによる故障診断

故障診断機能は、HILS ユーザがホスト PC の GUI (Graphical User Interface) で診断開始を指示し、FPGA ロジックが診断を実行する。

アナログ系・デジタル系の汎用信号は、本来機能の回路とは別に、診断用の回路を 1ch 内蔵しており、故障診断時は、本来機能の回路と診断回路を内部で接続し、FPGA ロジックで端子切り替えを行いながら計測・判定を行う (図 8)。

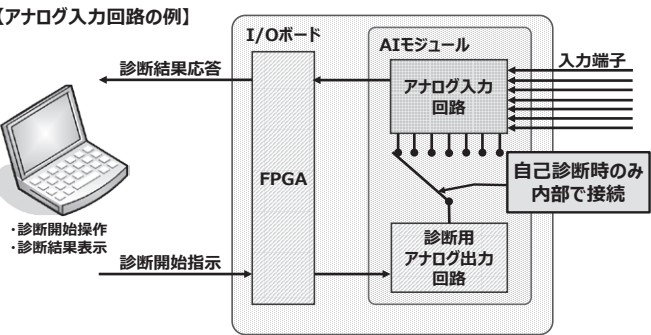


図 8 故障診断

- ・アナログ系回路の判定
診断回路の入出力電圧値と、本来機能の回路の入出力電圧値の一致を確認する。
- ・デジタル系回路の判定
診断回路の入出力論理と、本来機能の回路の入出力論理の一致を確認する。

診断結果は、故障箇所を視覚的に伝えるため、GUI に異常箇所を端子ごとに表示する (図 9)。

Result	Name	1Ch	2Ch	3Ch	4Ch
OK	AI	OK	OK	OK	OK
OK	AO	OK	OK	OK	OK
NG	DI	OK	NG	OK	OK
NG	DO	OK	OK	NG	NG
OK	PI	OK	OK	OK	OK
OK	PO	OK	OK	OK	OK
OK	PSC	OK	OK		

図 9 診断結果

5.2 効果

従来の手作業による故障確認は、数時間を要していたが、FPGA ロジックによる故障診断機能は、診断開始の指示から数 10 秒で診断できることを確認した。

GUI 操作だけで故障診断が実施でき、ツールの信頼性を確認する作業の効率化を可能にした。

6. おわりに

新 HILS「CRAMAS-X」の開発により、評価対象外の ECU レス、および HILS 1 台でのシステム評価環境構築が可能になった。今後の自動車制御は、コネクティッド・自動運転・電動化が進み、ますます膨大化、複雑化する。制御開発を効率化するために、ツールへの期待・要望はさらに大きくなっていくと考える。今後も制御開発の変化に追従した機能開発を進めていく。

- ・CRAMASは、株式会社デンソーテンの登録商標です。

参考文献

1) 木本恭平:「新 HILS (CRAMAS-X) の開発」, 自動車技術会関西支部ニュース第 53 号 ,pp.8-9, [2018]

2) 福岡亘二ほか:「CRAMAS-VF の開発」, 富士通テン技報 59 号 ,Vol.31 No.1,pp.15-20, [2014]

筆者紹介



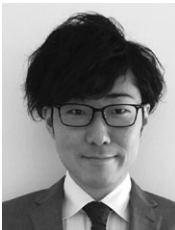
丸山 晃
まるやま あきら

AE 技術本部
制御ソフト技術部



前田 勉
まえだ つとむ

AE 技術本部
制御ソフト技術部



木本 恭平
きもと きょうへい

AE 技術本部
制御ソフト技術部



川島 英嗣
かわしま ひでつぐ

AE 技術本部
制御ソフト技術部