

CRAMAS-VFの開発

Development of CRAMAS-VF

福 岡 亘 二	Koji FUKUOKA
木 本 恭 平	Kyohei KIMOTO
尾 崎 辰 典	Tatsunori OSAKI
森 山 裕	Yuu MORIYAMA



要 旨

近年の自動車制御において、環境規制、先進安全、情報通信等の市場ニーズの多様化に対応するため、制御システムの大規模化と複雑化が急速に進む中、自動車用電子制御装置（以下、ECU）の開発・設計の効率化のためシミュレーション技術の活用が進んでいる。

当社でも独自開発したシミュレータ「CRAMAS（Computer Aided Multi-Analysis System）」（クラマス）をHILS（Hardware in the Loop Simulation）として活用している。

HILSがECU開発の下流工程での活用であるのに対し、近年ではECUが完成する前に制御ソフトの評価が可能なSILS（Software in the Loop Simulation）の活用が進んでいる。今回、このSILS環境を短期間で構築し、制御ソフトを効率的かつ実用的に開発することが可能なフレームワーク「CRAMAS-VF」を当社で開発したので紹介する。

Abstract

In these days, automobile control systems are rapidly increasing in size and being complicated due to various market needs concerning environmental regulations, advanced safety, information communication and other needs. Under this circumstance, simulation technology is widely used for efficient development and designing of electronic control units for automobile (hereinafter, referred to as ECU).

We at FUJITSU TEN also heavily use our original simulator "CRAMAS (Computer Aided Multi-Analysis System)" as HILS (Hardware in the Loop Simulation).

While the HILS is used at a downstream step in ECU development process, the use of SILS (Software in the Loop Simulation) that enables the evaluation of control software before the completion of ECU is now increasing. This paper introduces our newly-developed framework "CRAMAS-VF" that enables creation, within a short period, of the environment for SILS suitable for efficient and practical development of control software.

1

はじめに

近年の自動車制御において、環境規制、先進安全、情報通信等の市場ニーズの多様化に対応するため、制御システムの大規模化と複雑化が急速に進んでいる。一方で開発コストの削減、開発期間の短縮が要求されている。当社では、自動車用の電子制御装置（以下、ECU）の開発・設計の効率化を目的に、独自開発したシミュレータ「CRAMAS（Computer Aided Multi-Analysis System）」（クラマス）を活用している（図1）。CRAMASは、現在広く普及しているHILS（Hardware In the Loop Simulation）といわれるハードウェアを使用した閉ループシミュレーションを実現するためのシミュレータに分類される。

CRAMASの最大の特長は、制御の対象となるエンジンや車両の挙動を数式化したプログラム（以下、プラントモデル）を用いて、リアルタイムに模擬できることである。CRAMASに対象となるECUを接続することにより、車の機能部品がない環境でもECUの開発、評価をすることが可能となり、開発の効率化に貢献しており、エンジン、トランスミッション、ハイブリッドのパワートレイン制御や統合制御、ミリ波レーダなどの先進安全等、幅広い分野で使用されている。また実車検証場面では、危険を伴う試験、故障時のフェールセーフ試験等、現実には発生させにくい試験項目の確実な実行、テストの自動化（終夜無人運転等）により評価期間の短縮に効果がある。



図1 CRAMASの外観
Fig.1 Appearance of CRAMAS

2

モデルベース開発の広がり

開発期間のより一層の短縮を目的に、開発のフロントローディングを行い、アルゴリズム開発段階から制御内容をモデルで表現し、開発を進めるモデルベース開発が広がっている。

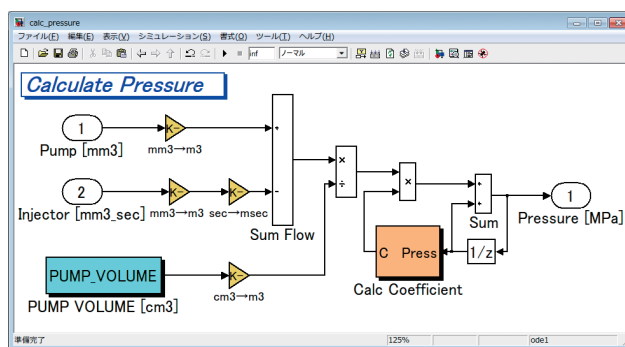


図2 モデルベース開発におけるモデルの例
Fig.2 Example of model in model-based development

図2に示すように、モデルは実行可能な仕様書と呼ばれ、自然言語（日本語による記述）ではなく、物理式等をブロック線図で記述したものである。そして、実際にその制御仕様を実行し、動作結果がモデル上で確認できるため、早期に制御仕様の問題を発見できる。

モデルを活用したECU開発のプロセスを図3に示す。

モデルベース開発を推進するために使用するシミュレーションツールは、従来活用していたHILSのほか、開発対象となる制御ロジックをモデルで表現し、モデルをそのまま使用してシミュレーションを行うMILS（Model In the Loop Simulation）、作成したモデルから自動生成したCコードを用いてシミュレーションを行うSILS（Software In the Loop Simulation）がある。

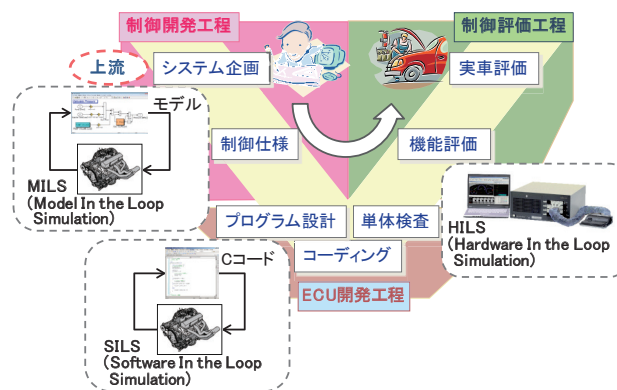


図3 制御開発プロセス（V字プロセス）
Fig.3 Process for development of control (V - shape process)

図3のように上流である制御開発工程のシステム企画や制御仕様検討に使用されるMILS環境を実現するには、制御ロジックをCコードではなくモデルで記述する必要があるが、大規模化、複雑化した制御システムにおける制御機能の数は、膨大であり、従来Cコードで記述されてきた制御全てをモデルで記述および検証を行うには非常に多くの工数がかかる問題がある。また、制御ロジックを効率的かつ実用的に開発するためには制御ソフト全体（制御アプ

リ+ミドルウェア)を評価できることが必要で、そのためにはミドルウェアを正しく模擬できるようマイコンクロック相当([ns]オーダー)で割込み処理をシミュレーションする必要があり、シミュレーション速度が低速になる問題がある。

今回はモデルベース開発推進のため、これらの問題を解決しMILS/SILS環境を簡単に構築可能なフレームワークをCRAMAS-VFとして開発したので紹介する。

3 CRAMAS-VFの概要

「CRAMAS-VF」を用いて構成されたMILSシミュレーション環境の全体構成を図4に示す。

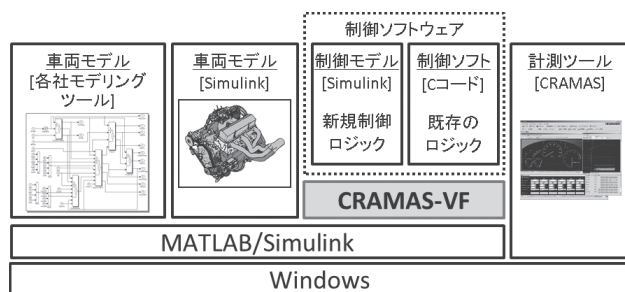


図4 CRAMAS-VFの構成図
Fig.4 CRAMAS-VF block diagram

CRAMAS-VFは、Windows PC上で動作可能であり、ユーザはPC1台ですべての環境を構築可能である。CRAMAS-VFの動作環境を表1に示す。

表1 CRAMAS-VF動作環境
Table 1 CRAMAS-VF operation environment

項目	内容
OS	WindowsXP Professional SP3 32bit Windows7 Professional SP1 32bit Windows7 Professional SP1 64bit
CPU	Intel®Core™2 Duo 2.0GHz 以上
メモリ	2GB以上
HDD	5GB以上

図4に示すように、CRAMAS-VFは、モデリングツールであるMathWorks社のMATLAB®・Simulink®をユーザインターフェースとしており、Simulink上で車両モデルと制御ソフトウェアを接続することでシミュレーション環境を構築する。

Simulinkは、提供されている機能ブロックを並べ、線を結ぶことで、容易にモデル全体を組み立てることができるため、制御開発におけるモデリングツールのデファクトスタンダードとなっている。また、各社のモデリングツールと連携することが可能である。

CRAMAS-VFとCRAMAS GUIアプリケーション（以

下、CRAMASアプリ)を接続することで、アクセル操作等のシミュレータへのデータ入力やモニタ、計測したデータの解析を行うことが可能となる。このCRAMASアプリは、HILS環境で使用するアプリケーションと同一のものをを用いているため、コンテンツ（計測データファイルや入力データファイル等）を共通化することができる。そのため、MILS/SILS環境とHILS環境において、共通のコンテンツを用いた評価が実施可能となる。

自動車の電子制御機能を対象とした機能安全についての国際規格で、自動車向けに適用した規格「ISO26262」が制定され、各メーカーが対応を進めている。モデルベース開発においてもモデル(仕様)と製品に実装する組み込みコードの一致性検証（以下、Back-to-Backテスト）動作が要求項目として盛り込まれており、MILS/SILS環境とHILS環境で共通のコンテンツを活用することで、Back-to-Backテストを効率的に実施することができる。

4 CRAMAS-VFの特長

CRAMAS-VFは、2章で述べた制御開発における課題の解決やシミュレーション環境の拡張性を持たせるために、以下のような特長を持っている。

- 1) 既存制御ソフトウェア（Cコード）の活用
- 2) 割込み駆動処理の模擬機能
- 3) 他社計測ツールとの連携

4.1 既存制御ソフトウェア（Cコード）の活用

CRAMAS-VFは、図5に示すように開発対象となる新規制御モデルと既存のCコードで記述した制御ソフトウェアを同時に動作させながらシミュレーション実行することができる。

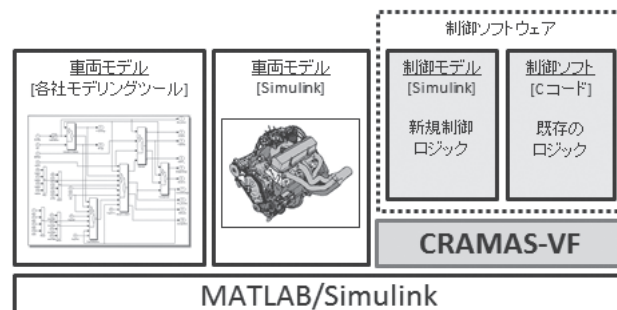


図5 既存制御ソフトウェアの活用
Fig.5 Use of existing control software

開発対象以外の制御に、既存制御ソフトウェアを活用することで、制御ソフトウェアの全てをモデルで記述することなく、短期間で開発環境を立ち上げることができる。

このように、開発対象の新規機能のみをモデルとして開発することができるため、モデルベース開発の課題であった「大規模化、複雑化した、従来Cコードで記述されてきた制御全てをモデルで記述するために、非常に多くの工数がかかる」という問題を解決できた。これにより、モデルベース開発へとスムーズに移行が可能となる。

CRAMAS-VFは、既存制御ソフトウェアを容易に活用するために、図6に示すような構成になっている。

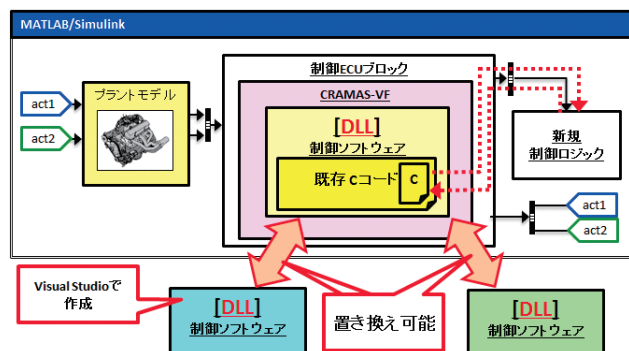


図6 既存ソフトウェアリンク構成図

Fig.6 Block diagram showing linkage with existing software

既存制御ソフトウェア全体は、ユーザーがSimulink上に配置した一つの制御ECUブロックから、まずCRAMAS-VFを呼び出し、そこからDLL (Dynamic Link Library) として呼び出している。既存制御ソフトウェア全体をDLL化し、モデル本体から独立させることで、DLLの置き換えのみで様々な制御ロジックのデバッグが行える。また、DLLはデバッグ機能が充実したIDE (Integrated Development Environment) であるVisual Studio[®]で作成しており、DLL化した既存制御ソフトウェアをVisual Studioのデバッグ機能を使用して、シミュレーションをCコード上のある任意の行で停止させ、そこから1行ずつステップ実行させる等のデバッグが容易に実施できる。この構成により、既存制御ソフトウェアをそのまま流用することが可能となり、新規制御ロジック (モデル) と組み合わせたシミュレーション実行を実現している。

4.2 割り込み駆動処理の模擬機能

Simulink上で動作するCRAMAS-VFの割り込み駆動処理の具体的な構成を図7に示す。

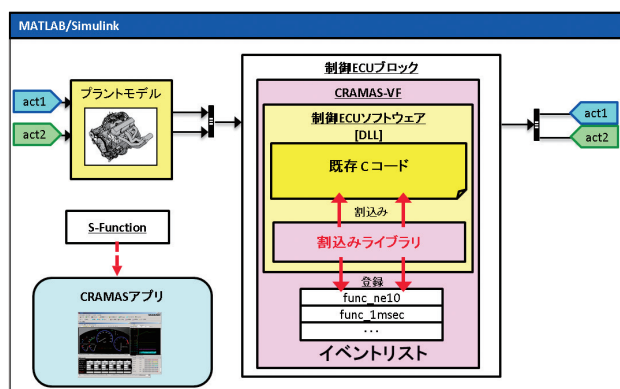


図7 割り込み駆動処理構成

Fig.7 Configuration showing interrupt-driven processing

実際のECUと同じ処理を実現するために、割り込み処理関数をCRAMAS-VFが実行する。正確なタイミングで制御を実施するためには、マイコンエミュレーションのように、シミュレーションをマイコンクロック相当で実行する方法がある。しかし、CRAMAS-VFでは、プラントモデルと制御ソフトウェアを共通の時間間隔で動作させるため、制御ソフトウェアをマイコンクロック相当で動作させるとシミュレーション実行速度が低下する問題がある。そこで、シミュレーション実行速度の低下を抑えるため、割り込み発生タイミングをリスト化 (以下、イベントリスト) し管理する。また、プラントモデルの演算周期と制御ソフトウェアのクロック処理を分離し、プラントモデルとは独立して制御ソフトウェアの実行タイミングを指定できる機能を合わせ持った。(図8参照)

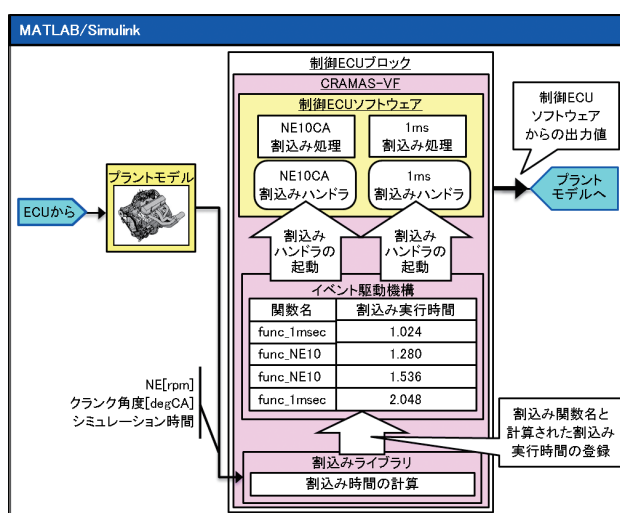


図8 イベント駆動機構

Fig.8 Event-driven function

図8を例に、割り込み処理の流れを具体的に説明する。前提条件として、プラントモデルのシミュレーション演算実

行間隔を入力AD変換処理周期とし（ここでは4[ms]とする）、制御ソフトウェアの処理間隔を割り込み実行最小時間のサンプリング時間として（ここでは125[ns]とする）、シミュレーションした例を示す。

まず、現在時刻（0[ms]とする）におけるエンジン回転数とクランク角度から、制御ソフトウェアの次回1ms時間周期処理時刻（1.024[ms]）におけるクランク角度センサの出力（10度毎のパルス信号）に応じた同期割り込み（以下、10degCAクランク角度同期割り込み）の発生時刻（1.280[ms]）を算出し、CRAMAS-VFのイベントリストへ、割り込み関数名と割り込み実行時刻、次の1ms時間周期処理関数名と実行時刻を登録する。シミュレーション現在時刻が、イベントリストの先頭に登録された時刻（例の場合は1.024[ms]）に到達すれば、CRAMAS-VFによって制御ソフトウェアの1ms時間周期処理が実行される。実行後は制御ソフトウェアの次回1ms時間周期処理の発生時刻（2.048[ms]）を再度算出し、CRAMAS-VFのイベントリストに登録する。同様に10degCAクランク角度同期割り込みも実行される。プラントモデルの計算は、シミュレーション演算実行間隔ごとにMATLABに処理を移し、Simulinkのソルバーによって演算が実行される。また、制御ソフトウェアの割り込みが発生しない期間については次の割り込み発生時刻まで制御ソフトウェアの現在時刻を進めることで効率的に割り込みを実行する。

以上の機能により、正確な割り込み処理実行タイミングと高速なシミュレーション実行速度の両立を実現することができた。

また、従来使用されてきたシミュレータでは、制御ソフトウェアのミドルウェア部分にカスタマイズを実施し、シミュレーション環境を構築していたが、割り込み処理関数をCRAMAS-VFから実行させるようにしたことで、制御ソフトウェアのミドルウェア部分をそのまま使用できるようになっている。そのため、従来ではできなかったミドルウェア部分のデバッグも可能となった。

これらの機能を実装できたことで、実際のECUと同じ演算順序・タイミングで動作する、制御ソフトウェア全体の模擬をすることが可能となり、「MILS/SILS環境において、割り込み駆動が模擬できていないため、実際のECU相当の割り込み処理評価ができない」という課題を解決することができた。

4.3 他社計測ツールとの連携

CRAMAS-VFはCRAMASアプリと連携する以外に、トヨタ自動車(株)において実車評価・ベンチ評価で実績のあるPanel3の計測評価用GUIアプリケーションと連携して動作

することができる。（製品名：SMIL（図9））。

このように、CRAMAS-VFとユーザが使い慣れている評価ツールを連携することで、作業の効率化および、導入が容易になることが期待できる。

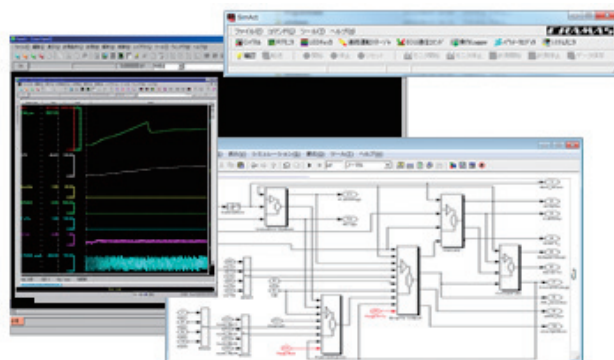


図9 Panel3連携画面

Fig.9 Screen image linked with Panel3

5

まとめ

今回開発したCRAMAS-VFは、当社の量産ECU制御ソフト実装技術とツール開発技術の賜物である。このCRAMAS-VFによって、実際のECU相当の検証が可能なMILS/SILS環境を容易に構築できるようになり、自動車制御におけるモデルベース開発の推進に貢献できた。モデルベース開発を制御開発のプロセスとして導入するにはまだまだ課題も多く、ツールに対する期待・要望は大きい。今後はコンピューティングの進化に追従しながら、このCRAMAS-VFをコア技術としたMILS/SILS分野での製品レベルアップを図るとともに、HILSも含めたCRAMAS製品がモデルベース開発の各工程でツールチェーンとして機能できるよう引き続き開発を進める所存である。

6

謝 辞

最後になりましたが、本開発にあたりご協力いただきましたトヨタ自動車株式会社様、関係各社様に深く感謝いたします。

※CRAMASは、富士通テン株式会社の登録商標です。

※記載した製品名等の固有名詞は、各社の商標または登録商標です。

参考文献

- 1) 藤原靖尚, 伊藤久弘, 魚住晴長, 福岡亘二: 次世代SILSの開発, 自動車技術会 学術講演会前刷集, No.122-12, pp.1~4, 2012
- 2) 福岡亘二: 次世代SILSの開発, 自動車技術会関西支部 ニュース, 自動車技術会関西支部, 42号, pp.2~3, 2013
- 3) 森山裕他: VirtualCRAMAS (SILS) へのISSレス技術の適用, 富士通テン技報52号, Vol.26, No.2, 2008
- 4) 山崎剛他: CRAMASモータボードの開発, 富士通テン技報46号, Vol.23, No.2, 2005

筆者紹介



福岡 亘二
(ふくおか こうじ)
AE技術本部品質監査室



木本 恭平
(きもと きょうへい)
AE技術本部品質監査室



尾崎 辰典
(おさき たつのり)
AE技術本部品質監査室



森山 裕
(もりやま ゆう)
AE技術本部パワトレ開発部
チームリーダー