

マルチフィジックス車載システムの サイバーフィジカルMBD

Cyber-physical MBD for Multi-physics Automotive Systems

Sujit S. Phatak DJ McCune

George Saikalis 勝 康夫
Sugure Yasuo

今日の車載組込みシステムは複雑さを増しており、100個以上のECU（電子制御ユニット）、大容量のROM（読み出し専用メモリ）、および大量のソフトウェアコードから構成されていることも珍しくない。これに対応するため、サイバーフィジカルシステム（CPS）が求められている。また、従来の設計方式には、開発プロセスが後半まで進んでから検証フェーズでハードウェアに根本的な問題が発見されても、対応が難しいという課題があるが、モデルベース開発（MBD）のアプローチは、こうした問題への対処法の一つであり、サイバーフィジカルシステムの基盤作りにもなるものである。

1. はじめに

CPS（Cyber Physical System）の特徴は、実世界のプロセスおよびシステムとコンピュータシステムを密接に統合することである¹⁾。CPSは、物理プロセスにおけるダイナミクスとソフトウェアおよびネットワークにおけるダイナミクスを統合し、統合システムに適した分析手法を提供して、モデリングおよび設計の抽象化を可能とする。従来の組込みシステムではデバイス単体での動作が重視されるが、CPSでは相互作用するシステムのネットワークが重視される。同様に、従来の組込みシステムでは演算部品を中心に考えるが、CPSはシステムの演算要素間のインタフェースを主に扱う。CPSは、センサベースの制御システム、組込みシステム、および自律システムをすべて適用範囲として包含するものである。CPSが適用可能なシステムとしては、車載システムやエンターテインメント・家電製品のような機器ベースのシステムから、社会インフラ、エネルギー、交通・輸送、航空宇宙、あるいはヘルスケアなどの統合的システムや、製造システムなどの技術プラットフォームまで、幅広いシステムが考えられる。

エンジン制御、トランスミッション制御、スロットル制御、ブレーキといった車載アプリケーションに実装される

ことの多い機構制御システムは、CAN（Controller Area Network）やFlexRayなどの車載ネットワーク経由で相互通信する専用の組込みコントローラを使用した複数の複雑な物理システムとなるのが一般的である。MBD（Model-based Development）は、こうした複雑なシステムの設計プロセスの効率を高める方法の一つとして取り入れられている²⁾。システム設計段階で、それぞれの物理システムの挙動モデル（「プラントモデル」とも呼ばれる。）と制御モデルとを統合し、抽象化されたシステム実装を生成する。制御モデルの実装では、MATLAB^{※1}/Simulink^{※1}などの主要なツールを使用したアルゴリズムレベルでの実装や、仮想CPU（Central Processing Unit）モデリング手法を用いた、より低い抽象化レベルでの実装が可能である。

仮想CPUモデリング²⁾では、マイクロコントローラ（マイコン）のハードウェア自体のソフトウェアモデルも開発する。このマイコンモデルを物理システムの動作モデルと統合することにより、実際のシステム挙動を評価・検証できる。この手法を用いれば、プラントモデルと制御ソフトウェアアプリケーションを並行して開発できるだけでなく、RTOS（Real-time Operating System）およびデバイスドライバも含めて、それぞれの検証を並行して行うこともできる。

こうしたシステムが、機構システムの各要素と対応する複数のプラントモデルで構成される場合もある。このような要素を別々のドメインに実装し、何らかのインタフェースを介してコントローラ（演算システム）に接続する必要がある。これらの要素が集まって、一つの機構系CPSが形成されるのである。

ここでは、日立アメリカ社（Hitachi America, Ltd.）オー

※1 MATLAB/Simulinkは、米国The MathWorks, Inc.の登録商標である。

トモティブプロダクツリサーチラボラトリ (APL : Automotive Products Research Laboratory) で採用されている基本的なCPS実装手法と、マルチドメイン・協調解析基盤³⁾として実装されたガソリン燃料ポンプのCPS研究事例について述べる。

2. 車載アプリケーションにおけるCPSの必要性

2.1 車載組込みシステムの課題

最近の自動車の各所に使われている車載組込みシステム (パワートレイン, トランスミッション, 走行制御, インフォテインメントを含む) は, 100個以上のECU (Electronic Control Unit : 電子制御ユニット) で構成され, 非常に複雑な機器となる場合がある。こうしたECUは, マイコンベースの組込みシステムである。通常, 物理システムのサブコンポーネントは, それぞれに専用ECUを持ち, これらをネットワークで接続させるために, CANやFlexRayといった車載ネットワーク用のプロトコルが使われる。

最新の車載アプリケーションには, 専用のアプリケーションプログラムを格納するために, 4 Mバイトよりも大きいROM (Read-only Memory) を必要とするものが多い (プログラムメモリ)。また, ソフトウェアコードの量も指数関数的に増加しており, 今後20年間で現状の1,000倍を超えると予測されている。対照的に, マイコンの処理能力の増加は10年間で20倍程度である。コードサイズの増大と計算処理能力の向上との差に対応し, サイクル性能と信頼性を高めるためには, 近い将来にマルチCPUまたはマルチコアアーキテクチャが必要になると予想されている。

2.2 従来の設計手法の非効率性

従来, 車載組込みコントローラの開発は, Vサイクルに準じて行われることが多かった (図1参照)。

Vサイクルは設計と検証の二つのフェーズに大きく分けられる。縦軸は抽象化の度合いを示し, 横軸は時間を表す。設計フェーズは, 自動車メーカーまたはOEM (Original Equipment Manufacturer) において, 抽象化レベルの高い概念設計から始まる。通常, このプロセスは, より低レベルへの一連の抽象化を経て, ティア1 (一次外注先) サプライヤーにおけるシステム設計, そしてティア2 (二次外注先) サプライヤーによって供給される実際のコンポーネントおよびハードウェアに至る。以上が実装フェーズであり, この後の検証フェーズは, 設計の実装時と同じ抽象化レベルにおけるユニット検証で始まる。次に, より高い抽象化レベルでのシステム検証を経て, 最終的な成果物としての製品が完成する。

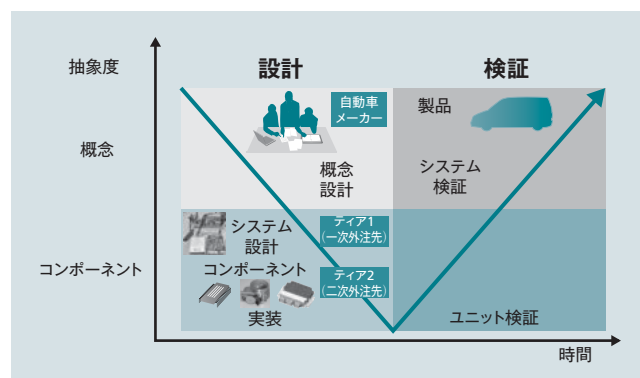


図1 | 従来のVサイクル開発プロセス

車載ECU (Electronic Control Unit) の標準的な開発サイクルと各要素の役割を示す。

従来のVサイクルにおける課題は, 検証フェーズを過ぎてからハードウェアの根本的な欠陥が発見された場合に, 欠陥の原因を特定するために設計フェーズに戻ることであり, 時間とコストの両面で効果的とは言えない点である。ハードウェア上の内在的な欠陥は, 設計フェーズの初期段階, すなわちハードウェア開発への投資前に特定することが重要である。また, 将来の高度なシステムアーキテクチャ (ハードウェアおよびソフトウェア) について研究, 開発, および評価を実施することも, ビジネス機会の拡大と革新のためには重要である。

こうした課題の解決法の一つがMBDである。MBDのアプローチはMILS (Model in the Loop Simulation) とも呼ばれ, CPS開発の基盤となる。MBDとCPSを取り入れた場合のVサイクルを図2に示す。

図2は, MBDのアプローチを採用することによって, 設計フェーズに活動が集中することを示している。こうした活動には, 概念設計の段階で行われるアルゴリズム設計, 挙動解析, ラピッドプロトタイピングなどがある。まずは, MATLAB/SimulinkやSynopsys^{*2} Saber^{*2} などの信

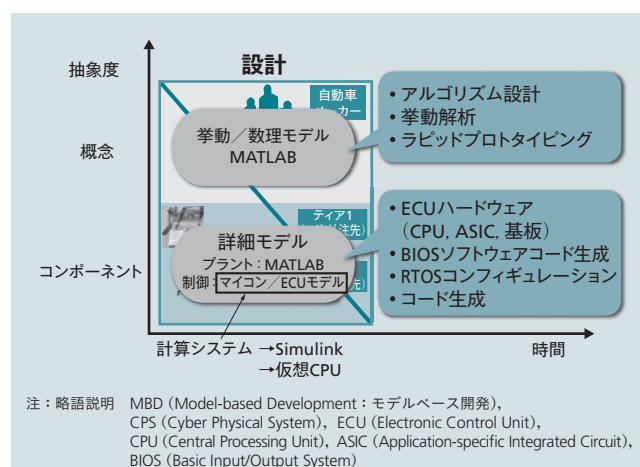


図2 | MBDおよびCPS手法を取り入れたVサイクル

Vサイクル開発におけるMBDの役割とCPSアプローチおよびそれに関連する活動を示す。

号・物理モデルライブラリ解析ツールを用いて挙動モデルまたは数理モデルを開発する。

続いて、より低い抽象化レベルでコンポーネントの解析を行う。主な開発内容は、ECUハードウェアの仕様策定、BIOS (Basic Input/Output System：基本入出力システム) ソフトウェアのコード生成、RTOS コンフィギュレーション、自動コード生成である。MBDでは、このフェーズにおいてプラントモデル (物理システム) と制御/ECU (計算システム) 両方の詳細モデルを開発する。

計算システムはSimulinkあるいは仮想CPUを用いて構築できる。すなわち、CPSでは2種類のコンフィギュレーションが可能である。通常、MBDを用いるときは、Simulinkベースの計算システムを開発した次の段階で自動コード生成を実行し、生成されたソフトウェアを実ハードウェアECUに実装する。これはPILS (Processor in the Loop Simulation) と呼ばれる。ただし、ここで述べる作業には、CPUの挙動解析もモデルに含める新たな手法が採用されている。こうした手法は「仮想PILS」とも呼べるが、用語の一貫性の観点から、ここでは「仮想CPUモデリング」と呼ぶ。次に、CPSに関して、Simulinkの手法を用いた場合、および仮想CPUモデルの手法を用いた場合について、それぞれ詳しく説明する。

3. 車載アプリケーションのCPS

3.1 SimulinkベースのCPS

Simulinkを用いたCPSの概略を図3に示す。

このシステムはプラントモデル (物理システム) および制御/ソフトECUモデル (計算システム) という二つの主要部分で構成されている。プラントモデルは、ガソリン車のエンジンや、HEV (Hybrid Electric Vehicle) またはEV (Electric Vehicle) の電気モータなど、車両の物理システムをモデル化したものである。制御モデルは、プラントモデルに期待どおりの機能を達成させるための制御ロジックを実装する。これらのモデルにより、制御パラメータも視覚化できる。

Simulinkは解析に使用され、プラントモデルと制御モデルとの通信は、物理量や、工学単位 (圧力ならばパスカル、電流ならばアンペア) で表現される各パラメータとしてやり取りされる。また、このシステムではフィードバック制御が用いられており、制御モデル上で実行される制御アルゴリズムに基づいて、ユーザー/運転者からの入力とアクチュエータからの入力が共にプラントモデルに適用される。ここで重要な点は、プラントモデルと制御モデルとの

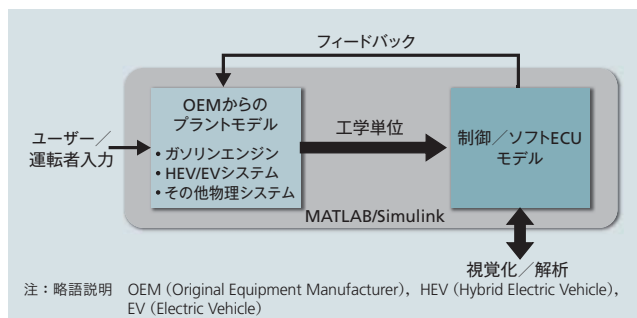


図3 | SimulinkベースのCPS

CPSシミュレーションモデルを示す。

連関をつかさどるために、両モデルを解析システム内で直接接続していることである。

別々のシミュレータで実行されている複数の異なったモデルを使用する場合は、シミュレータ間を接続するための協調解析環境を実装する必要がある。協調解析環境は、各シミュレータがそれをサポートしていれば、モデル間の直接接続あるいはコシミュレーションバス⁴⁾によって実装できる。この環境は、使用するシミュレータでサポートされているか、もしくはC/C++などの高級プログラミング言語で実装可能となっている必要があるが、コシミュレーションバスを使用して実装する場合、モデルの物理的な場所は同一のPC (Personal Computer) 上でなくてもよい。

コシミュレーションバスの使用には、ほかに、(1) 各モデルがそれぞれのネイティブドメインにあるままで協調するマルチドメイン/マルチフィジックス・システム解析が可能になる、(2) マルチドメインの相互作用から、堅牢(ろう)な設計手法が生まれる、(3) コシミュレーションバスは、CPSの計算システムと物理システム間のインタフェースを実装するための枠組み (フレームワーク) にもなる、といった利点がある。

3.2 仮想CPUベースのCPS

仮想CPUベースの概略を図4に示す。

仮想CPUをベースとしたCPSのシステムアーキテクチャは、SimulinkベースのCPSアーキテクチャを拡張したものである。この場合、制御モデルは仮想CPUシミュレータを使用して実装される。仮想CPUシミュレータは、ユーザーが規定する特定のマイコンのマイコンモデル、または仮想CPUモデルをエミュレートできる。マイコンモデルは、フリースケール社 (Freescale Semiconductor, Inc.), ルネサスエレクトロニクス株式会社, インフィニオン社 (Infineon Technologies AG) をはじめとする主要な車載半導体製造会社のモデルが利用可能である。

これら各社のモデルでは、マイコンをエミュレートすることにより、実際のマイコンと同じオブジェクトコードを

※2 Synopsysは、Synopsys, Inc.の登録商標であり、Saberは、Sabermark Limited PartnershipがSynopsys, Inc.に使用許諾している登録商標である。

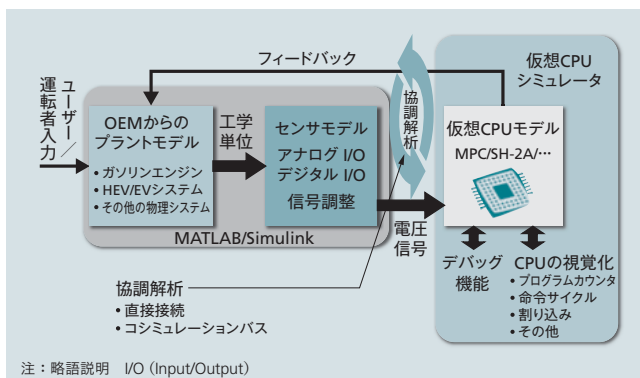


図4 | 仮想CPUベースのCPS

従来のSimulinkベースの制御モデルを仮想CPUモデルに置き換えたときの構成を示す。

実行できる。このシミュレータを使用すれば、プログラムカウンタ、命令サイクル、割り込み、ソフトウェアステートなど、マイコン内部の情報にもアクセスできる。

実際のマイコンと同様に、これらのモデルはアナログまたはデジタル電圧信号を処理するので、工学単位に基づく直接処理はできない。したがって、プラントモデルと仮想CPUモデル間で必要な信号調整を実行するために、インタフェースとして機能する別のセンサモデル（図4のセンサモデル）を実装する必要がある。言い換えれば、このブロックはアナログおよびデジタルの入出力（I/O：Input/Output）処理と、工学単位を仮想CPUモデルで処理できるようにするための電圧信号変換を行う。残りのシステムアーキテクチャは、フィードバックループ、およびユーザー／運転者入力も含めて同じである。

もう一つの留意点は、協調解析の使用である。プラントモデルとセンサモデルはMATLAB/Simulinkを用いて実装されるのが一般的だが、仮想CPUは前述のように専用の仮想CPUシミュレータを用いて実装される。そのため、これらシミュレータ間での協調解析が必要であり、これは直接接続によって、もしくはコシミュレーションバスを使用して実現できる。完全にSimulink内で実行されるシステムには直接接続の方法を採ることができるが、直接接続がサポートされていないシミュレータの場合はコシミュレーションバスを用いる必要がある。これについては、以下の研究事例で述べる。

4. 車載CPSの研究事例：ガソリン燃料ポンプのシミュレーション

日立グループが開発したガソリン燃料ポンプ用解析モデルを使用して、Simulinkベースと仮想CPUベースの手法でそれぞれ得られた結果を比較し、CPSの実装を検討した。

4.1 Simulinkベースのガソリン燃料ポンプCPS

コシミュレーションバスを用いてプラントモデル（物理

システム）を実装した（図5参照）。

このガソリン燃料ポンプの物理システムは、電気機械解析ツールを用いて計算されるドライバ回路と、流体力学解析ツールで計算される、ポンプモデルおよび吸気弁モデルの三つの部分で構成されている。コシミュレーションバスを使用し、ドライバ回路は1台目のPCで実行される電気機械解析ツールに実装され、実ポンプのポンプモデルおよび吸気弁モデルは、2台目のPCで実行される流体力学解析ツールに実装された。

計算システム（制御モデル）はSimulinkを使用して実装された（図6参照）。すべて統合されたSimulinkベースのガソリン燃料ポンプCPSを図7に示す。

一定のバッテリー電圧入力レベルにおける目標圧力と圧力フィードバックとの差に基づいてコントローラ出力を生成するために、比例積分（PI：Proportional Integral）制御が使用されている。最終目標はこの制御アルゴリズムをハードウェア（この場合は仮想）に実装することであり、コントローラ出力に応じてソレノイドにトリガをかけると消費エネルギーが増えることから、PWM（Pulse Width Modulation）デューティサイクル制御が追加された。吸気弁を開くにはソレノイドに完全にトリガをかけなくてはならないが、開いたままで保持するための力はより小さいため、ソレノイドに流す電流量は低減することができた。このソレノイド電流の低減率は、PWMデューティ比に基づいて決定された。数回のテストと経験を踏まえて、さまざま

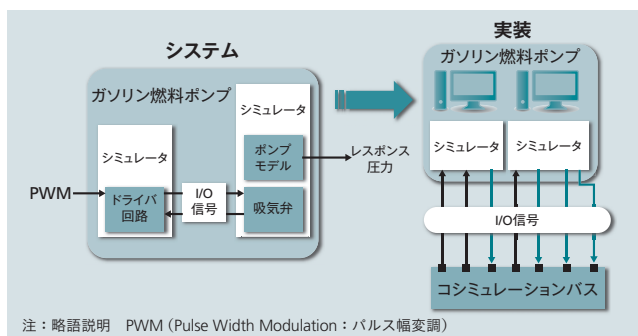


図5 | ガソリン燃料ポンプの物理システム

ガソリン燃料ポンプシステムとその解析モデル、およびコシミュレーションバスを用いるマルチPC（Personal Computer）実装を示す。

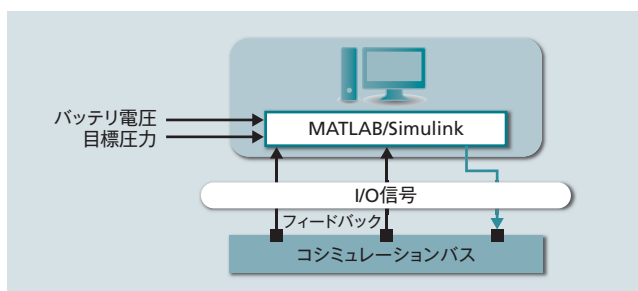


図6 | ガソリン燃料ポンプの計算システム

ガソリン燃料ポンプ制御システムの解析モデルをコシミュレーションバスに接続する。

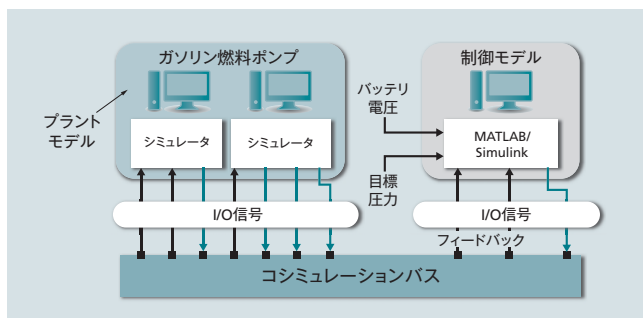


図7 | Simulinkベースのガソリン燃料ポンプCPS
このSimulinkベースのCPSではコシミュレーションバスを使用している。

また開始アングルのデューティ比と、エンジン回転速度およびバッテリー電圧との関係に関するデータが収集された。これらのデータから生成される参照テーブルを使用してデューティサイクル値¹⁾を取得できた。また、PWMベースの制御により、仮想CPUベースの手法でこのCPSを実装することも可能になった。

4.2 仮想CPUベースのガソリン燃料ポンプCPS

仮想CPUベースのCPSにおいて何よりも必要なものは、Simulinkに基づくCPSで使用されているソフトウェア制御アルゴリズムのオブジェクトコードソフトウェアである。これを得るために、自動コード生成を用いてSimulink制御モデルからソフトウェアのアプリケーションレイヤを生成した。今回の手法では、仮想CPUとしてルネサスエレクトロニクス製のSH-2A^{※3}マイコンを指定した。SH-2A用のデバイスドライバソフトウェアは手動でプログラミングし、車載電子機器向けRTOSおよびインタフェースのコンフィギュレーションも、このアプリケーションで使用できるようにした(図8参照)。

仮想CPU解析には、仮想CPUシミュレータとMATLAB/Simulinkの二つのシミュレータを併せて使用した。したがって、この仮想CPU解析ではこれら二つのシミュレータ間で1対1の直接接続による協調解析が行われた。仮想CPUがコシミュレーションバスを直接サポートしていなかったため、システム全体では仮想CPUのMATLAB/Simulink部分をコシミュレーションバスへのインタフェースとして使用した。仮想CPUベースのガソリン燃料ポンプCPSを用いたシステムを図9に示す。

このシステムではPI制御アルゴリズムの仮想チューニングが可能のため、今後のポンプコントローラ開発の加速に貢献できるとみられる。

※3 SH-2Aは、ルネサスエレクトロニクス株式会社の高性能32ビットマイコン「SuperHファミリ」のコアの一つであり、SuperHは同社の商標である。

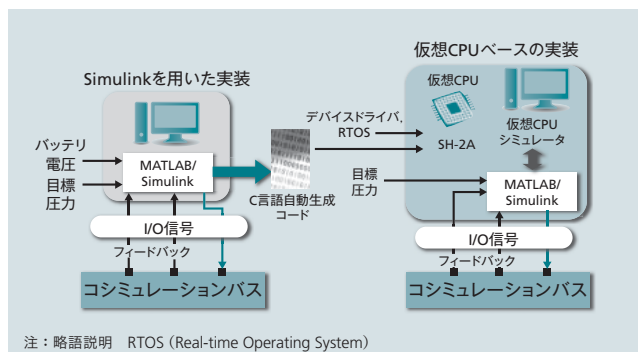


図8 | 「SH-2A」CPUの自動コード生成

Simulinkベースの制御モデルは、C言語によるプログラムコードに変換され、デバイスドライバおよびRTOSとともにコンパイルされたうえで、ルネサスエレクトロニクス株式会社製の「SH-2A」CPU上で実行される。

4.3 SimulinkベースのCPSと

仮想CPUベースのCPSとの相関関係

Simulinkベースと仮想CPUベースの両手法を採用してCPSを実装した後の段階として、同一の解析条件とコンフィギュレーションでこれら二つの手法を用いた場合のガソリン燃料ポンプ解析の挙動の関係を観察した(図10参照)。

この結果、ガソリン燃料ポンプ制御システムに対し、同一解析条件(目標圧力は同じでエンジン回転速度を変える条件)でSimulinkと仮想CPUベースの二つの手法を適用した場合、双方のポンプ挙動の解析結果は極めてよく一致することがわかった。

5. CPSを用いたさらなる成功事例

日立アメリカ社APLにおいてガソリン燃料ポンプのCPSの実装に成功した後、コシミュレーションバス手法が実現する柔軟性を多拠点での協調解析で実証することとした。多拠点間のCPS実装を行う場合、データ転送時間が長くなり、解析に時間がかかるというトレードオフが存在する。今回はEthernet^{※4}ベースのTCP/IP (Transmission Control Protocol/Internet Protocol) ネットワーク上の遅延による速度低下が問題となった。例えばサンプリング

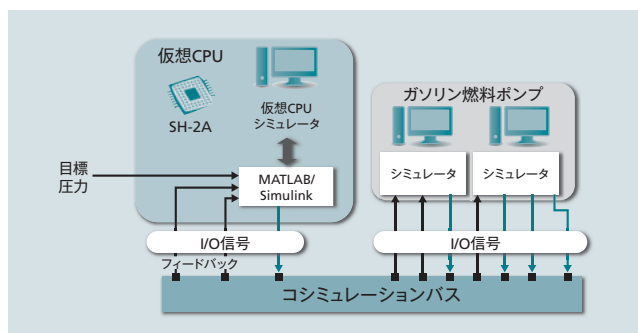


図9 | 仮想CPUベースのガソリン燃料ポンプCPS

仮想SH-2A制御モデルはコシミュレーションバス経由でガソリン燃料ポンプのプラントモデルと統合する。

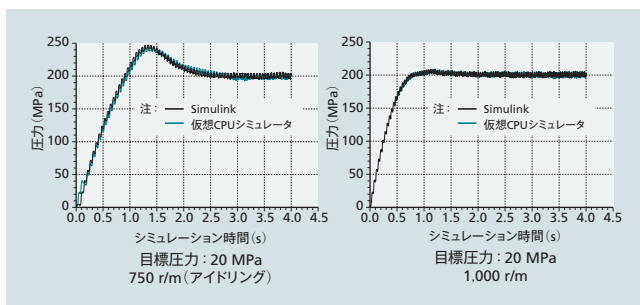


図10 | Simulinkベースと仮想CPUベースのガソリン燃料ポンプCPSの相関関係

この比較は、2種類の目標圧力設定におけるガソリン燃料ポンプのPI (Proportional Integral) 制御プロファイルを示している。

周波数 100 μ s の場合、日米2拠点間の協調解析の速度は、同様な協調解析を米国内で行う場合と比較して3倍の時間がかかった (図11 参照)。

同図に示すように、ガソリン燃料ポンプの物理モデルは日米それぞれ1台ずつのPCに、計算システムモデルは米国の1台のPCに実装された。本実証試験では、データ転送時間の問題を除けば日本と米国の2拠点にまたがる協調解析が可能であることが実証でき、海外拠点間での柔軟性のあるCPS実装が可能であることが証明できたとと言える。

※4 Ethernetおよびイーサネットは、富士ゼロックス株式会社の登録商標である。

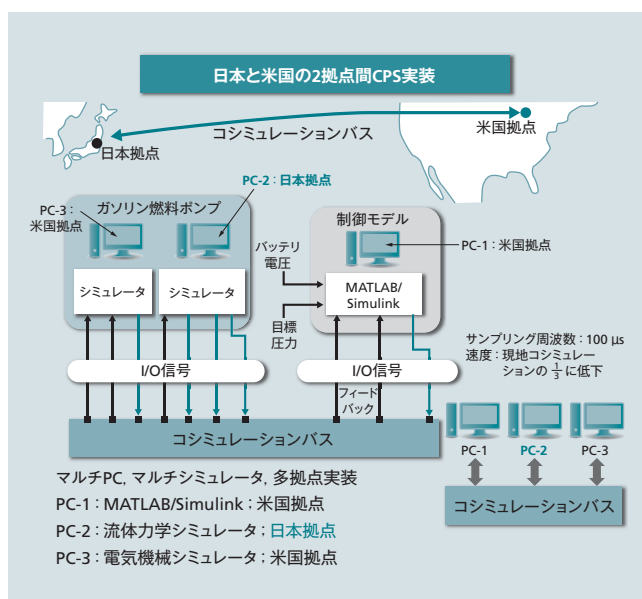


図11 | 海外拠点間CPS実装

このガソリン燃料ポンプ制御システムの分散実装実験では日立グループの日本・米国2拠点を1本のコシミュレーションバスでつないだ。

6. おわりに

ここでは、日立アメリカ社APLで採用されている基本的なCPS実装手法と、マルチドメイン・コシミュレーション基盤として実装されたガソリン燃料ポンプのCPS研究事例について述べた。

PI制御アルゴリズムの仮想チューニングを実現し、開発プロセス全体を加速させることを目的に、ガソリン燃料ポンプ用の車載CPSを実装した。システムの挙動解析はまずSimulinkを用いて実施し、その後に仮想CPUベースの手法を用いて実施した。この二つの手法は良好な相関を示した。多国間CPS実装のテストと検証も実施した結果、これらシステムの多拠点実装が堅牢なマルチフィジックス設計手法として実施可能であることが証明された。

参考文献など

- 1) S. Phatak, et al.: "Cyber Physical System: A Virtual CPU Based Mechatronic Simulation," IFAC (2010.9)
- 2) G. Saikal, et al.: "Virtual Embedded Mechatronics System," SAE Technical Paper # 2006-01-0861
- 3) J. Sinnamon, et al.: "Co-Simulation Analysis of Transient Response and Control for Engines with Variable Valvetrains," SAE Technical Paper, 2007-01-1283
- 4) ChiasTek, 2009, <http://www.chiastek.com>

執筆者紹介



Sujit S. Phatak

2007年Hitachi America, Ltd.入社, Research & Development Division, Automotive Products Research Laboratory 所属
 現在, 組込みシステムと仮想プロトタイプリングシステムのモデルベース開発に従事
 IEEE会員, SAE会員



DJ McCune

2003年Hitachi America, Ltd.入社, Research & Development Division, Automotive Products Research Laboratory 所属
 現在, サイバーフィジカルシステムのモデリングとシミュレーションの研究に従事
 SAE会員



George Saikal

1990年Hitachi America, Ltd.入社, Research & Development Division, Automotive Products Research Laboratory 所属
 現在, 研究開発部門のバイスプレジデントとして自動車分野のさまざまな研究開発に従事
 Ph.D.
 IEEE会員, SAE会員



勝 康夫

1999年日立製作所入社, 中央研究所 情報システム研究センタ 所属
 現在, 組込み制御システム向けマイクロコントローラモデルを使用した仮想プロトタイプリングシステムの研究に従事
 SAE会員, IEICE会員