

宇宙機搭載ソフトウェア検証のためのシミュレーション環境の検討

Study on Processor-in-the-loop Simulations for Spacecraft Software Verification

上田 裕子[1]; 高橋 孝[1]; 平野 聡[2]; 邑中 雅樹[2]; 小堀 壮彦[3]

Hiroko, O Ueda[1]; Takashi Takahashi[1]; Satoshi Hirano[2]; Masaki Muranaka[2]; Takehiko Kobori[3]

[1] 宇宙航空機構; [2] 産総研; [3] MSS

[1] JAXA; [2] AIST; [3] MSS

[1. はじめに] 我々は、従来衛星プロジェクトごとに開発されてきた宇宙機シミュレータとは異なり、複数の人工衛星など宇宙機に共通して利用され、また宇宙機の開発ライフサイクルを通じて利用可能な汎用的な宇宙機シミュレーション環境の研究を行っている。そこでは従来と同様にすべてのモジュールがひとつの計算機上で動作するシミュレーション(Full Software Simulation :FSS)が基本的な形態であるが、実際の運用卓や搭載ハードウェアと連携するシミュレーション(Hardware in the Loop Simulation :HILS)、および衛星の姿勢軌道制御系やデータハンドリング系、ミッション系などリアルタイムで動作する搭載ソフトウェア(OBS)および搭載計算機(OBC)を分散オブジェクトとするシミュレーション(Processor in the Loop Simulation :PILS)を同様な枠組みで実現することを目指している。PILS は主に OBS の検証試験をなるべく実際の動作状況に近い環境で実施することが目的であり、ソフトウェアに起因する宇宙機の不具合を未然に予防する重要な手段のひとつである。本研究では PILS を実現するための新たな形態として、リアルタイム OS(RTOS)上で動作する OBS と Linux などの汎用 OS 上で動作する宇宙機シミュレータが連携する分散シミュレーションを検討し、その有効性、実現性を評価している。

[2. 模擬搭載装置とシミュレータの接続実験装置] 宇宙機の搭載計算機には基本的に宇宙用 OBC が用いられ、OBS は特定の OBC 上で動作する。OBS の動作検証には、必要な入力データを供給し出力データを適切に受け取る環境が必要となる。これは汎用計算機上で動作する宇宙機シミュレータと、OBS とシミュレータとのインターフェイス規定するフレームワークを中心とするシミュレーション環境によって実現できる。従来はおおよそ図 1 に示すような 3 通りの形態がとられ、JAXA などの宇宙機関または宇宙機メーカーで実施されてきた。(a)の場合は OBS の動作論理の確認に重点をおくものであり、I/O ドライバの部分などは検証対象とはしない。また OBS を汎用計算機に移植するためにはソースコードが提供され、インターフェイス部分などの改修が許される必要がある。(b)(c)の場合はドライバを含めて OBS の大部分を検証対象に含めることができ、ソースコードは必ずしも必要ないが、(b)では特定の宇宙用 OBC に対応する S/W エミュレータが提供されなければならない。また(c)では OBC または I/O ドライバ毎に専用 H/W インターフェイスを製作する必要がある。これに対して、我々が検討している新たな形態を図 2 に示す。まず図 2(a)は図 1(a)の場合と同様の目的で行う FSS であるが、分散シミュレーションにすることでソースコードの提供が不要になり、また汎用の通信ミドルウェアを介することでインターフェイス改修の負荷を軽減できる。一方、図 2(b)は図 1(c)における専用 H/W インターフェイスをソフトウェアで代用する PILS である。ただしこの場合、OBC には ITRON など汎用 RTOS が搭載されること、および OBS とシミュレータの通信には通信ミドルウェア HORB を利用することを想定している。従来の日本の OBC では必ずしも汎用 RTOS は搭載されず、スケジューラ等の機能を含む専用 OS を開発して用いられている場合が多いが、近年、宇宙機の搭載ソフトウェアに要求される高機能、高信頼性、低コスト、開発期間の短縮を実現するために他業種で実績のある汎用 RTOS を搭載することが検討されている。またシミュレータの実装言語は保守性、再利用性を考慮するとオブジェクト指向言語である Java に移行しつつあると考えられ、Java 対応の汎用高速通信ミドルウェアを利用することでシミュレータ、OBS 双方のインターフェイス作成の負荷が軽減されると考えられる。そこで本研究では図 2(b)の実現性、有効性を評価するため、 μ ITRON が搭載される OBC 上でリアルタイムタスクとして動作し C 言語で作成される OBS と、Linux などの汎用 OS 上で動作し Java および C 言語で作成される宇宙機シミュレータを想定し、その連携には通信ミドルウェア HORB を用いるインターフェイスソフトウェアを整備し、接続実験を行った。

[3. 実験状況] 実験用には OBS として動作周期の異なる簡易な制御系タスクと TT&C タスクを作成した。一方シミュレータ側では、宇宙機ダイナミクスおよび、タスクと通信するセンサ、スラスタ、地上局などのモデルが動作する。これまでに本環境を用いて、通信インターフェイスが正常に機能していることを確認した。また通信インターフェイスは定型的なコードとパラメータで構成できることから、その作成作業を省力化するため自動コード生成ツールを作成し、その機能の評価を行った。



図1. 従来の搭載ソフトウェア検証環境

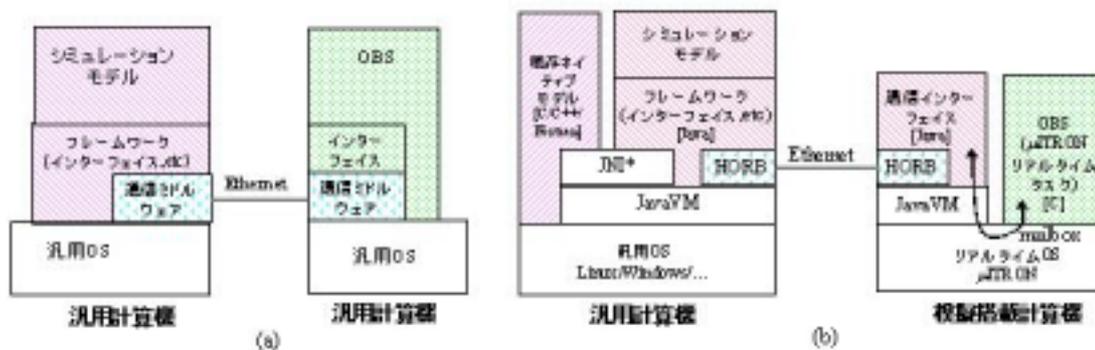


図2. 新たな搭載ソフトウェア検証環境