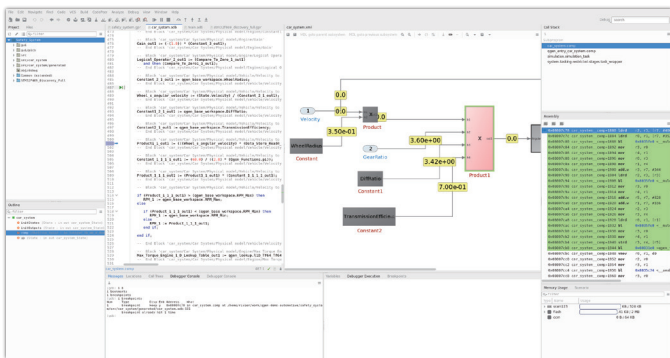


モデルレベルデバッグ手法の紹介

制御エンジニアリングとソフトウェアエンジニアリングの架け橋

AdaCore 言語リサーチディレクタ タッカ タフト

モデルレベルデバッグは、オートコードジェネレータによって生成されたソースコードと制御システムのシミュレーションモデル（Simulink® で作成）を並列に表示できます。エンジニアは、生成されたソースコードの代わりにモデルを使用して、ブレークポイント設定、シグナル値の表示と変更、ステップ実行が可能です。ソースレベルデバッグがアセンブリレベルコードの詳細を知らなくてもソースコードデバッグできる事に似ています。ソースレベルデバッグをベースにしたモデルレベルデバッグは、自動生成コードと手書きコードを組み合わせ、ホスト上でのシミュレーションならびにターゲットプロセッサシステムでも利用可能です。このデバッグは個々のブロックまたはモデル全体でシミュレーション時に記録されたシグナル値とのバックツラックでの比較ができます。モデルレベルデバッグは、生成されたソースコードとモデルを同時に表示するため、制御エンジニアリングとソフトウェア エンジニアリング間の架け橋となります。



QGen モデルレベルデバッガー画面

モデルレベルのデバッグの機能

モデルレベルデバッグは、ホスト（「Software in the loop」/SIL）または組込ターゲットプロセッサ（「Processor in the Loop」/PIL）上でシミュレーションが終了したモデルから生成されたコードの実行をモニタあるいは制御できます。さらに、このデバッグはモデルレベルでのブレークポイント設定やシグナル値の表示が可能でソフトウェアの要素と対応するモデルの要素を結びつけることが可能です。モデルレベルデバッグは、自動生成されたコードセクションまで実行するとモデルとそのシグナル値が表示可能となります。

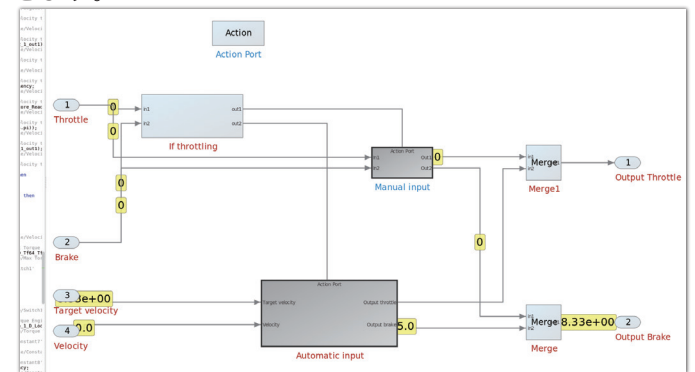
また、デバッグが手書きコードで停止するとソースコードのみを表示、自動生成コードでブレークポイント停止すると、ソースコード行とモデル内の関連ブロックの両方を表示します。つまり、モデルレベルデバッグは、オートコードジェネレータによって作成されたトレサビリティ情報を使用して、シミュレーションのモデル実行とコードの実行を同期させる機能を持っています。このようにして、生成されたソースコードあるいは最終的なアセンブリコードをステップ実行しながら、モデルの動作を解析することができます。モデルレベルデバッグでは、シグナル値を変更したり、プログラムの実行を制御したりして、モデルを特定の状態に移行させることができます。これはモデルへの入

力のみの変更では、実現できない場合があります。モデルの特定部分で問題が発生している場合は、モデルの変更や入力ストリームを新たに作成するためにデバッグの作業を中断することなく、検証作業を継続できます。モデルレベルデバッグの重要な機能は、デバッグ実行中にシグナル値を記録し、シミュレータでモデル内の元のブロックにトレースバックすることができるため、予期しない動作を解決するのに役立ちます。さらにモデル内で特定したポイントをもとに、シミュレータで必要な個所の修正が可能です。

モデルデバッグを利用したクロス開発

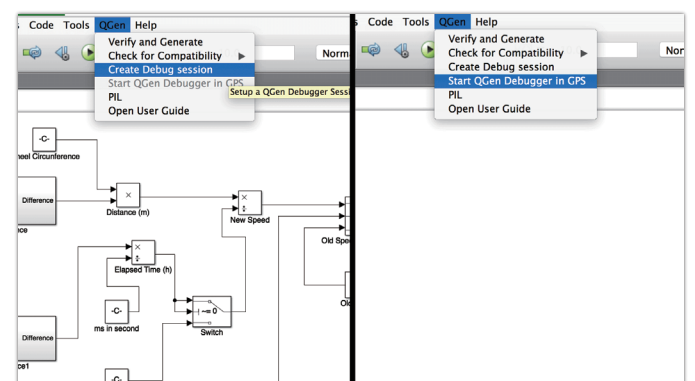
ソースレベルデバッグの多くは、ホスト（デバッグが実行されている PC）とターゲット（エージェントが動作している環境）間をネットワークやその他の接続を利用して、ターゲットハードウェア上のプログラムをデバッグできるクロスデバッグ機能がサポートされています。モデルレベルデバッグでも同様のクロスデバッグ機能に対応しています。オートコードジェネレータとともに、クロスコンパイラツールチェーンとターゲットインタフェース機能を使用して、プラットフォーム上で自動生成コードを直接クロスコンパイルして実行、ホストからモデルレベルでデバッグを行いながらターゲットの実行動作を観察できます。

シミュレーションモデルは、全てのシミュレーションブロック、サブシステムおよびサブモデルを IDE に表示できます。



モデルレベルデバッグの実行例

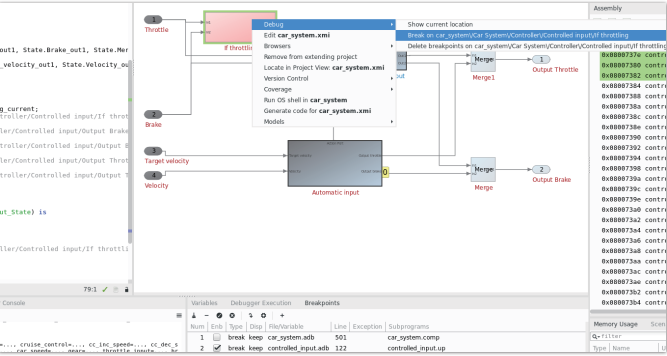
モデルの入力は、標準シミュレータ機能を使用、あるいは、別のモデルを使用して定義できます。



シミュレータを使用してターゲット入力を作成

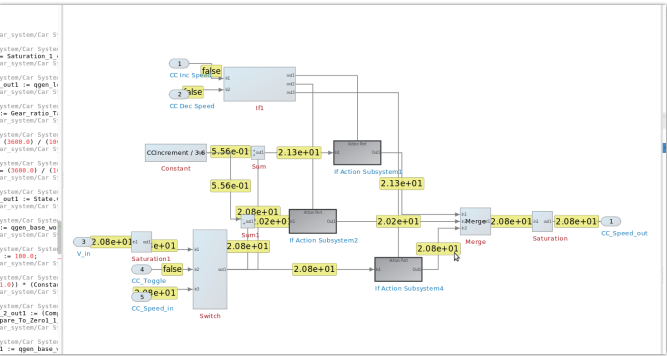
モデルレベルデバッグは、モデルへの全入力を記録し、同じ入力を与えることができるデバッグセッションが作成可能です。この入力を用いてモデルおよび生成されたコードの動作は、後で、同期して観察およびデバッグが可能です。

ソースコードにブレークポイントを設定できるように、モデルレベルデバッグでも、モデルビューからブレークポイントを直接設定することができます。ブレークポイントが設定されたブロックまでコードが到達すると、実行が一時停止されます。



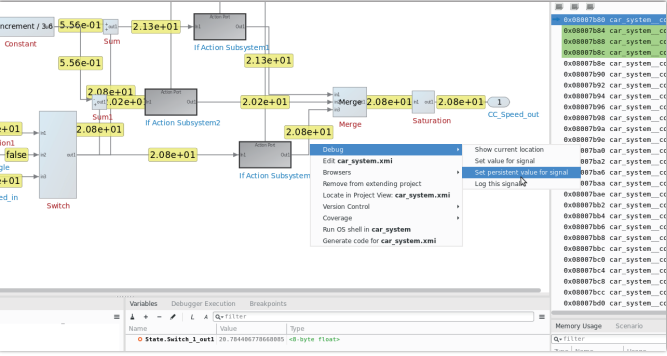
ブロックまたはモデル参照にブレークポイントを設定

モデルレベルデバッグは、コード内の変数とシミュレーションモデルのシグナル間のトレーサビリティにより、シグナル値を更新表示することができます。実行されたブロックをグレースアウトして、更新されたシグナルには新しい値を表示します。



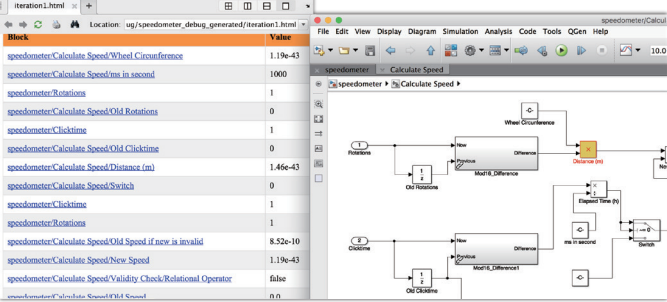
シグナル値を動的に表示する機能

モデルレベルのデバッグは、シミュレーションモデルの特定の振舞を解析する強力な手法を提供しています。それは、1つ以上の反復するシグナル値を変更して、モデルを異なるステートに移行できます。



実行中の信号値の変更

モデルレベルデバッグは、デバッグセッション中に取得したシグナル値の全て、または一部をファイルへ保存する事ができます。



シグナルの記録とシミュレータ内での対応するブロックの強調表示

シミュレータでこのファイルを開くと、各シグナルに対応するシミュレーションブロックへのリンクがあります。そのためコードレベルで発見された問題点に対応するモデルブロックを特定するのに役立ちます。さらに、シミュレータで問題点を調査する事ができます。

生産性の高いモデルベースデバッグ

モデルレベルデバッグは、手書きコードとの組み合わせでも、モデルベースシステムのターゲットでの実行を検証することができます。それは、問題のあるモデルのブロックで停止し、シグナル値を調べ、時間の経過とともに変化する状態を確認できるからです。

手作業で停止することによって、システムの動作（モデル、ソースコード、またはアセンブリコード）を観察することができ、必要に応じて手書きコードで停止することもできます。

したがって、プログラム全体のモニタ、テスト、デバッグが可能のため、モデルブロックとシグナルの制御エンジニアリングの観点と、関数と変数のソフトウェアエンジニアリングの観点との差分を埋めることができます。

モデルレベルデバッグは、システム全体のモデルレベルデバッグ機能と低レベルデバッグ機能とを組み合わせることで、問題点の特定を行います。

AdaCore は、QGen モデルレベル開発スイートの一部として、上記の機能に対応したモデルレベルのデバッグを開発しました。このことは、最終ターゲットでしか問題点を確認できなかったエンジニアが、モデルベースレベルでの作業が可能となりデバッグ効率を飛躍的に改善します。

また、このデバッグは、モデルベースの開発に不慣れなエンジニアに、モデルと自動生成されたコードの間の相関関係を示したり、モデリングの概念をプログラミングの概念に結びつけたりする際に有用です。最終的に、同じシステムを複数の視点から見る事ができるため、システムエンジニアとソフトウェアエンジニアが協力して、提供されるモデルベースシステムの全体的な品質を向上することが可能となります。

Article originally published in EENewsEurope, 1 February 2018

お問い合わせ先

IT Access
アイティアアクセス株式会社

※本文中の会社名、商標、製品名等は各社の商標または登録商標です。

本社 〒222-0033 神奈川県横浜市港北区新横浜 3-17-6
TEL: 045-474-9095 FAX: 045-474-8823
E-mail: info@itaccess.co.jp URL: http://www.itaccess.co.jp