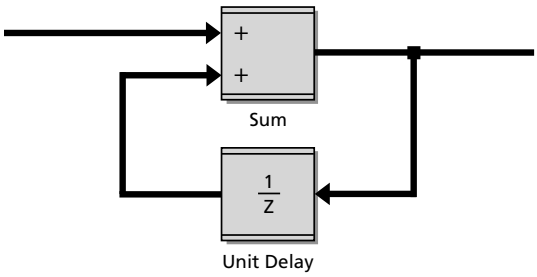
実行時エラーの追跡

- TargetLink コードの自動実行時エラー解析
- コード解析結果からモデルに直接移動
- ツールの統合で高精度の解析を保証

量産コード生成ツール TargetLink の諸機能のうち、ECU 用ソフトウエア開発に役立つのは、ファンクションコードの生成機能だけではありません。TargetLink はユーザーに、ソフトウエアの検証と妥当性確認のための幅広い機能を提供します。その一部は TargetLink 自体に組み込まれています。また、MTest のように拡張ソリューションとして提供されるものもあります。その TargetLink に新たに革新的なツールが加わりました。実行時エラーを防止する TargetLink-PolySpace Integration です。

実行時エラーの原因

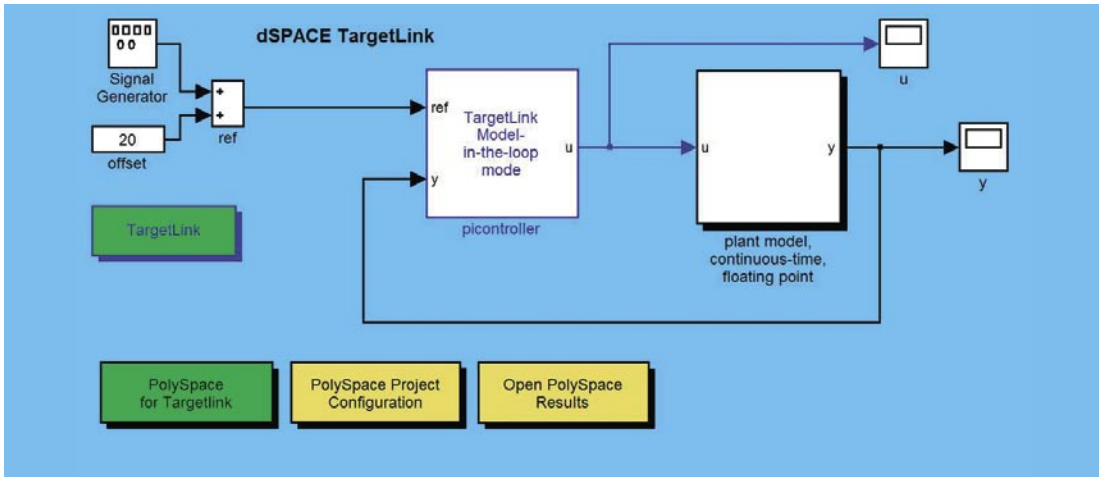
量産コード自動生成ツールは、人間のプログラマーがとうてい及ぶことができないほど正確ですが、ツールで生成したコードに実行時エラーが潜んでいないという保証はありません。理由は、ファンクションの開発中、モデルデザインの過程でエラーが入り込むことにあります。たとえば、モデルレベルでゼロ除算やアウトオブレンジに対する保護がなされていなければ、生成されたコードに実行時エラーが含まれる可能性があります。潜在的にエラーの可能性をはらんだ仕様が 1：1 でコードに変換されるためです。この種の実行時エラーは、モデルレベルに原因があるわけですから、そのレベルで解消してしまうのが一番です。TargetLink-PolySpace Integration を使うと、このプロセスを大幅に簡略化できます。



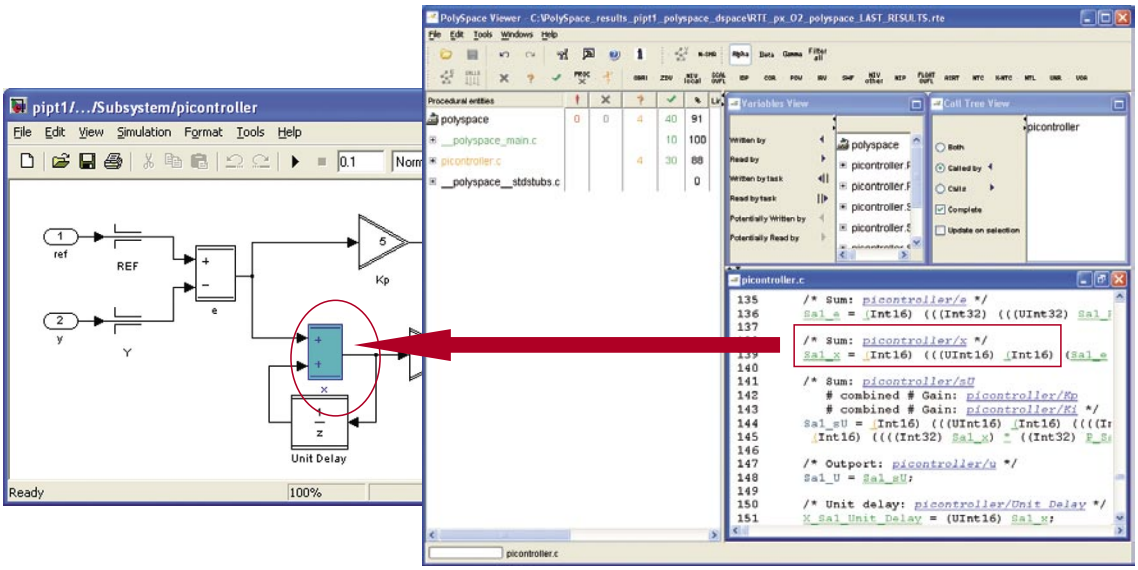
▲ オーバフロー／アンダフローを引き起こす可能性のあるフィードバックを含むモデルフラグメント

TargetLink-PolySpace Integration を使ってできること

この場合の Integration（統合）は、TargetLink を PolySpace Verifier と直接連動させられることを意味しています。これは抽出インタープリテーションと呼ばれる手法を用いて、生成されたコードを静的に解析します。この手法では、解析結果として数学的実証に匹敵する精度で実行時エラーの情報が返されます。個々のコードフラグメントは、実行時エラーがまったく発生しない、実行時エラーが必ず発生する、決して実行されることがない（デッドコード）、または時に実行時エラーが発生する可能性がある、のいずれかに分類されます。開発者が綿密に分析する必要があるのは、



▶ 追加の TargetLink-PolySpace ブロックを使い、モデルレベルからコード解析を設定し、実行します。



▲ PolySpace Viewer では、コード中の実行時エラーを起こす可能性のある箇所が色分けして表示され、そこからモデルの対応する箇所に直接移動できます。

実行時エラーが実際に発生するかどうかを PolySpace Verifier の抽出によっても正確に断定できなかった最後のグループだけです。

TargetLink-PolySpace Integration の利点

両方のツールを使用すると、TargetLink-Polyspace Integration が持つ以下の利点を最大限活用できるようになります。

- 数回クリックするだけでモデルレベルからコード解析が始まります。PolySpace Verifier と、コードが属する TargetLink サブシステムの設定パラメータもモデル内で指定します。
- PolySpace Viewer に、生成されたコードのタイプが色分けして表示されます（緑色＝実行時エラーがまったく発生しない、オレンジ＝時に実行時エラーが発生する可能性がある、など）。ユーザーはコードから、モデルの対応する箇所に直接移動できます。それにより、問題のありそうな箇所をモデルに遡ってトレースし、精査し、必要に応じて訂正する作業が容易になります。
- 適合可能なパラメータや入力値の最大値／最小値など、モデルレベルの追加情報が得られることで、解析精度が大幅に向上します。PolySpace Verifier ではこうした情報を dSPACE データディクショナリから読み出し、解析に使用します。それにより、実行時の動作を精密に判断できないコード行の数を減らすことができます。
- PolySpace Verifier は、compute-through-overflow と呼ばれる手法を使って最適化された TargetLink のコード生成を明示的に認識します。それにより生成コードの解析が一段と容易になります。

PolySpace と dSPACE により実現されたツールの統合は、開発プロセスの時間短縮に寄与するだけでなく、生成された量産コードの検証を容易にします。

用語解説

デッドコード –
決して実行されることのないコードフラグメント

Compute through overflow –
最終結果に問題がなければ、計算の中間段階で発生したオーバフローに目をつぶるような数値演算手法

抽出インタープリテーション –
抽出を利用して処理負荷を少なくする、プログラムのセマンティクスの解析手法

TargetLink-PolySpace Integration（モデルベース設計用 PolySpace）の詳細については、PolySpace (contact@polyspace.com) にお問い合わせください。