Property Specification UI (仮称) 説明資料

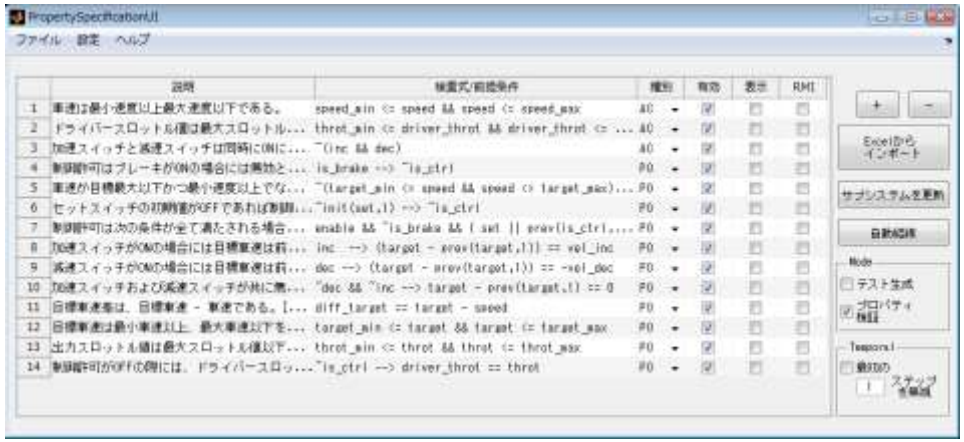
2012/11/20

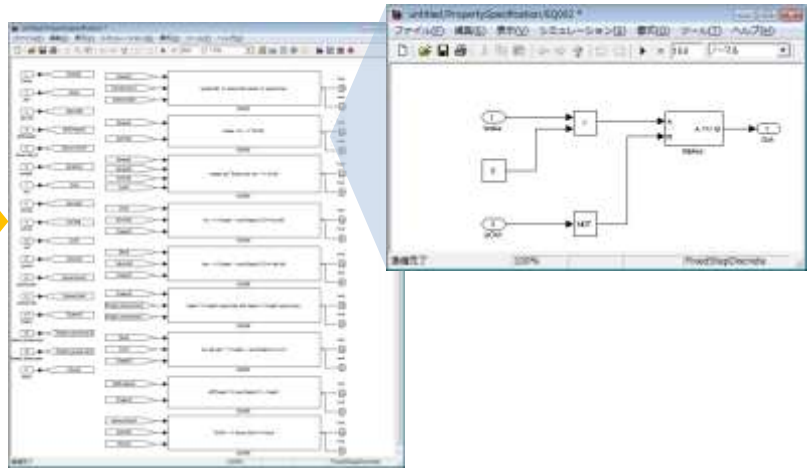
MathWorks

伊藤泰充

Property Specification UIとは

- プロパティ検証・テストケース自動生成において検査式、テスト条件式のモデル化を容易にするユーザインターフェース。
- テキスト記述の検査式・テスト条件式からブロック線図を自動生成することが可能。
  - Simulinkブロックで記述されるためデバッグが容易
- Excelで記述された検証仕様を容易にSimulinkに取り込むことが可能。



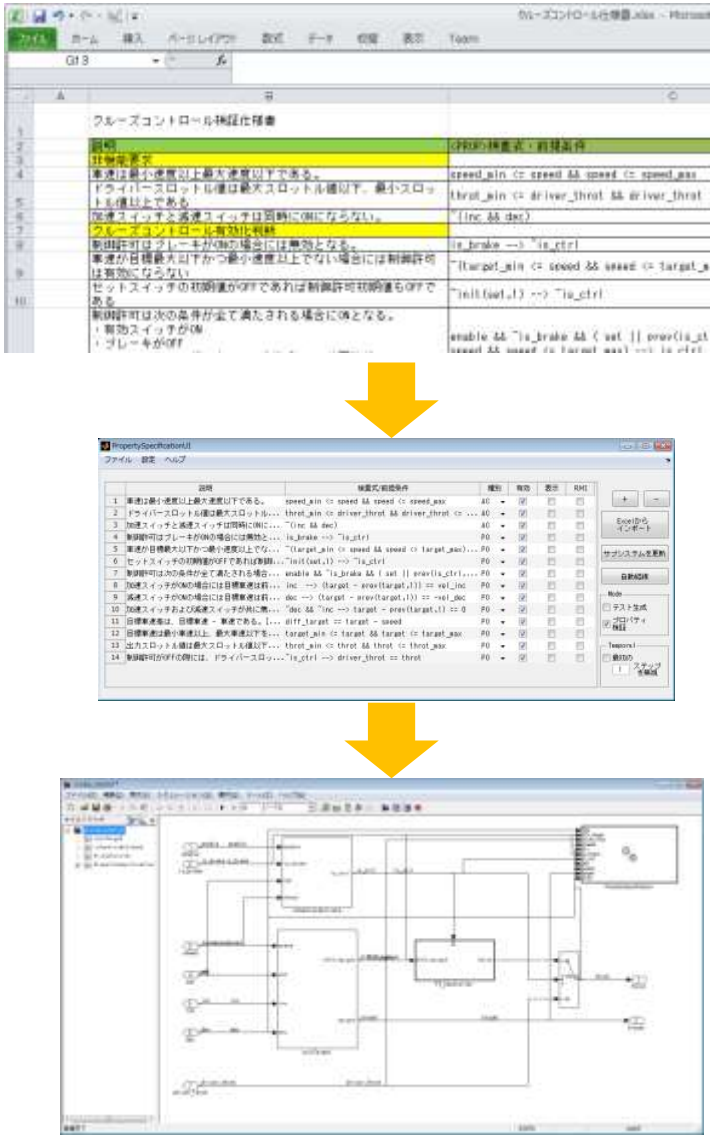


注意事項とお願い

- 本ツールは、プロトタイプ版であり、将来の製品機能として取り込むべく、米国開発チームと共同で開発を進めています。
- 本ツールの利用の結果生じた、いかなる損害についてMathWorksは一切の責任を負いません。
- ご利用状況の把握のため、本ツールを他社・他部署へ配布される場合には、必ずMathWorksの担当営業までご一報下さい。
- フィードバックなどがあれば、是非お知らせください。

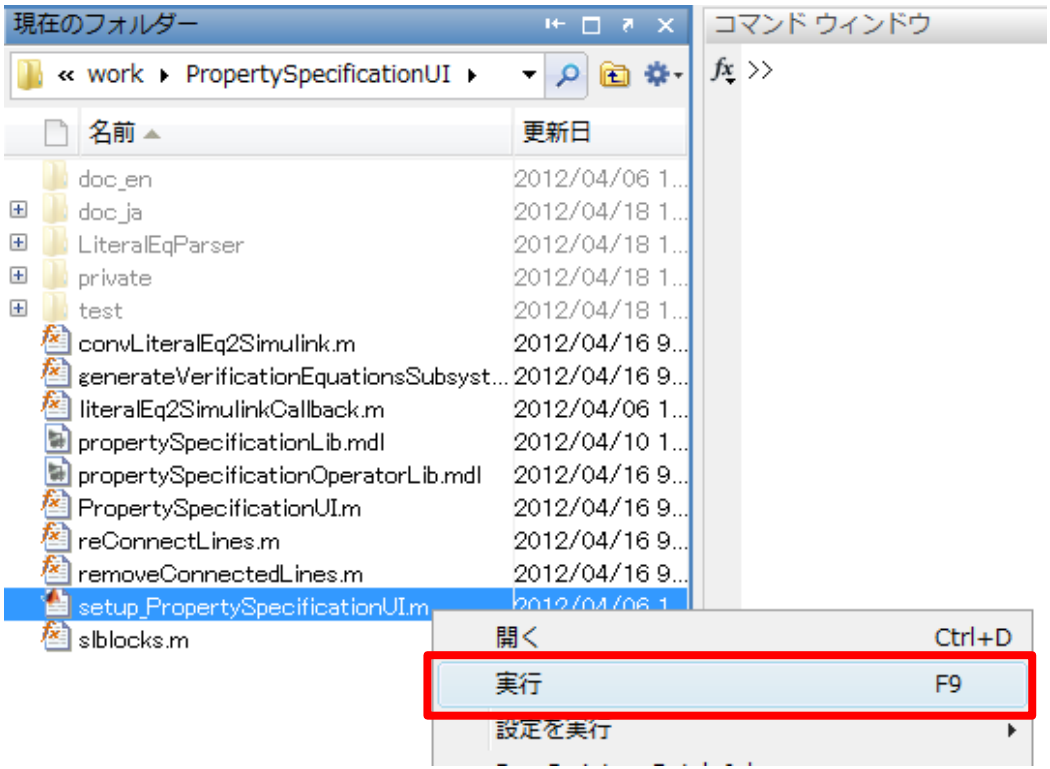
Property Specification UIのワークフロー

- Excel上で検証仕様を記述する
  - 検査項目の説明
  - 検査項目を検査式として記述
- Property Specification UIに取り込む
  - 信号名が検査式内変数と同名であれば自動結線が可能
- プロパティ検証を実行
  - 反例が見つかればデバッグし、要求仕様、検証仕様、モデルをレビュー



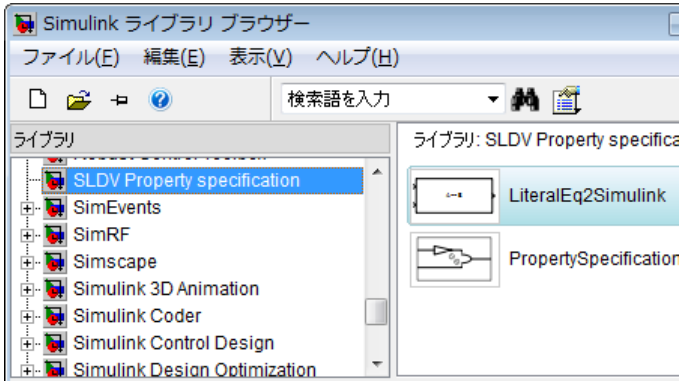
インストール方法

PropertySpecificationUIフォルダ内の
setup_PropertySpecificationUI.m
を実行する



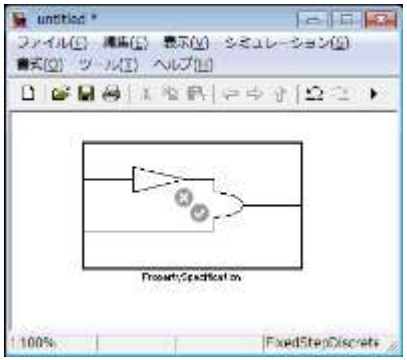
動作の確認(1)

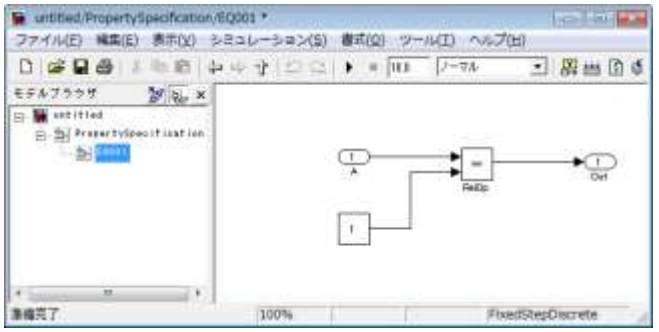
1. Simulinkライブラリブラウザ内に「SLDV Property Specification UI」が存在することを確認
2. 上記ライブラリが無い場合は、「表示」→「ツリー表示を更新」を実行する



動作の確認(2)

1. 新規モデルに「Property Specification」ブロックを配置
2. 「Property Specification」ブロックをWクリックし、「PropertySpecificationUI」の「＋」をクリック
3. 「検査式/前提条件」に「A==1」を入力
4. 「サブシステムを更新」をクリック
5. 「表示」をクリックし、A==1と等価なモデルが作成されていることを確認





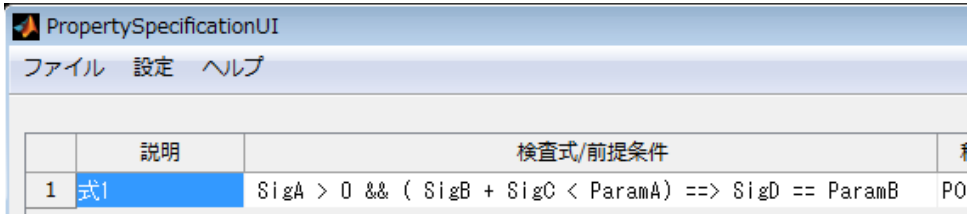
検査式の記述

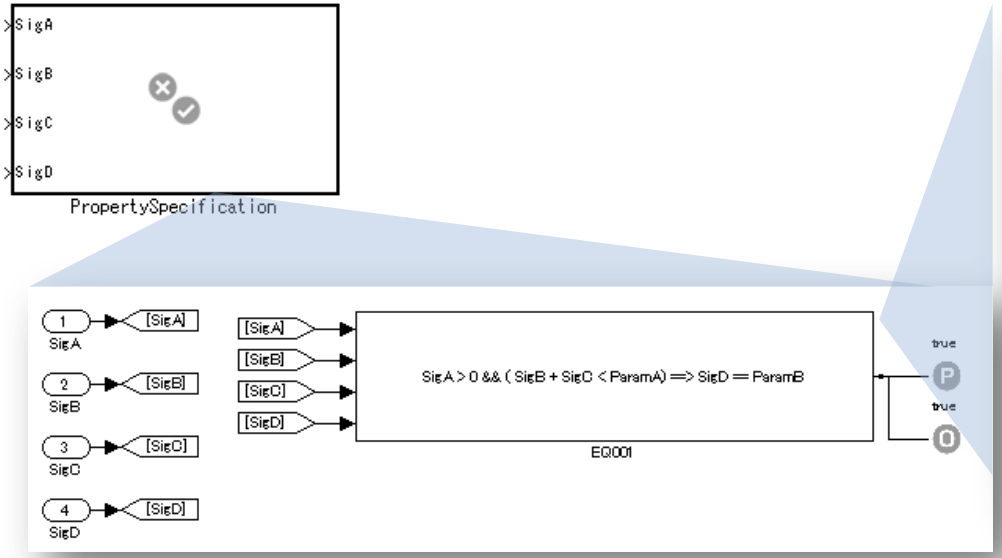
例:

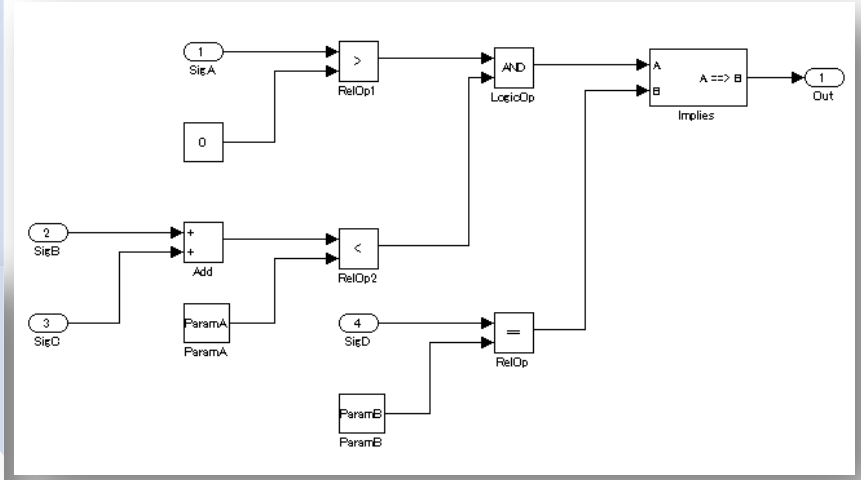
```
SigA > 0 && ( SigB + SigC < ParamA) ==> SigD == ParamB
```

Sig*: 変数 → Inportブロックへ

Param*: 定数（ワークスペース変数）→ Constantブロックへ







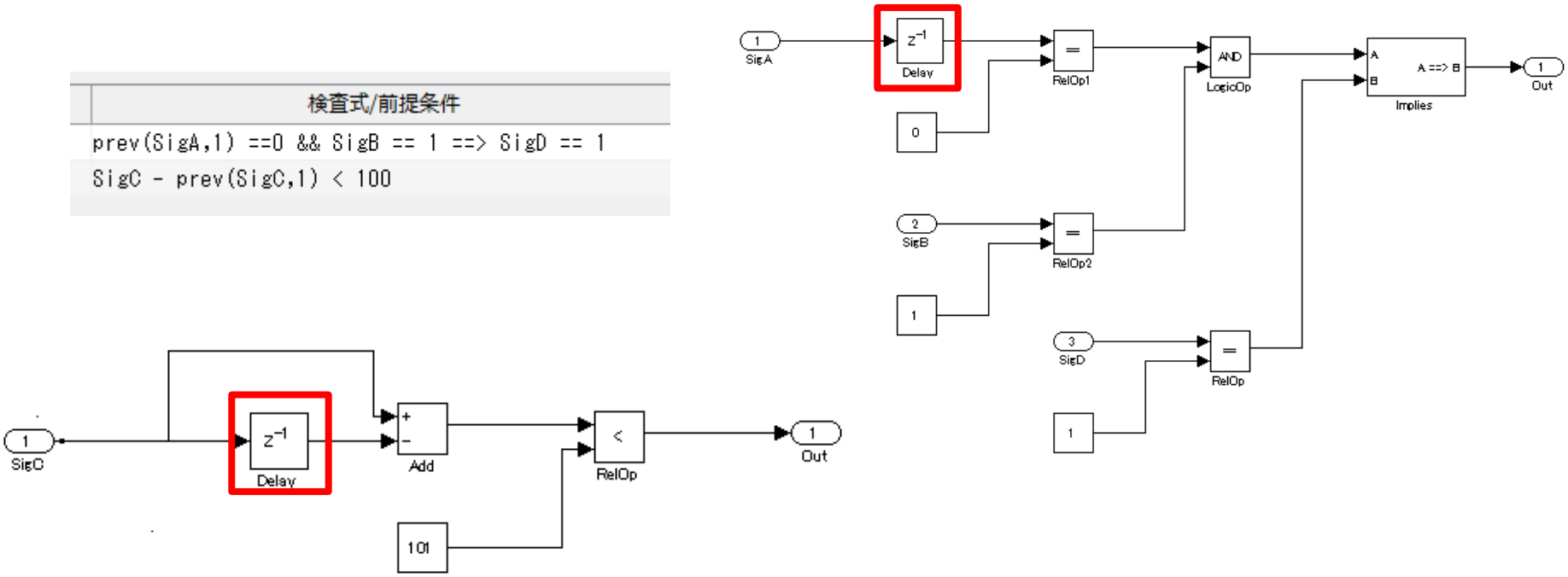
検査式の対応フォーマット

- **対応演算子** 対応演算子とその優先順位は基本的に MATLAB言語の優先順位に従います。
  - カッコ ()
  - 組み込み関数
  - 単項演算 +, -
  - キャスト (single, double, uint8, uint16, uint32, int8, int16, int32, 固定小数点書式 例:fixdt(1,16,2,0))
  - 論理否定 ~
  - 乗算除算 * / %(mod)
  - 加減算 + -
  - 比較演算 ==, <, >, <=, >=, ~=
  - AND &&
  - OR ||
  - ならば(Implies) ==>, WithinImplies ==>>, Implies with Test Objective =@>
- **組み込み関数**
  - **abs**([expr]) : 絶対値
  - **prev**([expr],[Integer]) : [Integer] ステップ前の値
  - **init**([expr],[Integer]) ステップ0から [Integer] ステップ間は [expr] が成立する
  - **withinImplies**([expr],[expr]) Simulink Design Verifierで提供されるWithin Impliesブロック
  - **after**([expr],[INTEGER]) : [expr] が [INTEGER] ステップ以上成立すれば真を出力
  - **extend**([expr],[INTEGER]) Simulink Design Verifierで提供されるExtenderブロック
  - **detect**([expr],[INTEGER]) Simulink Design Verifierで提供されるDetectorブロック

時相論理記述関数: prev

- prev(*expr* , *n*)
  - expr を n ステップ遅らす

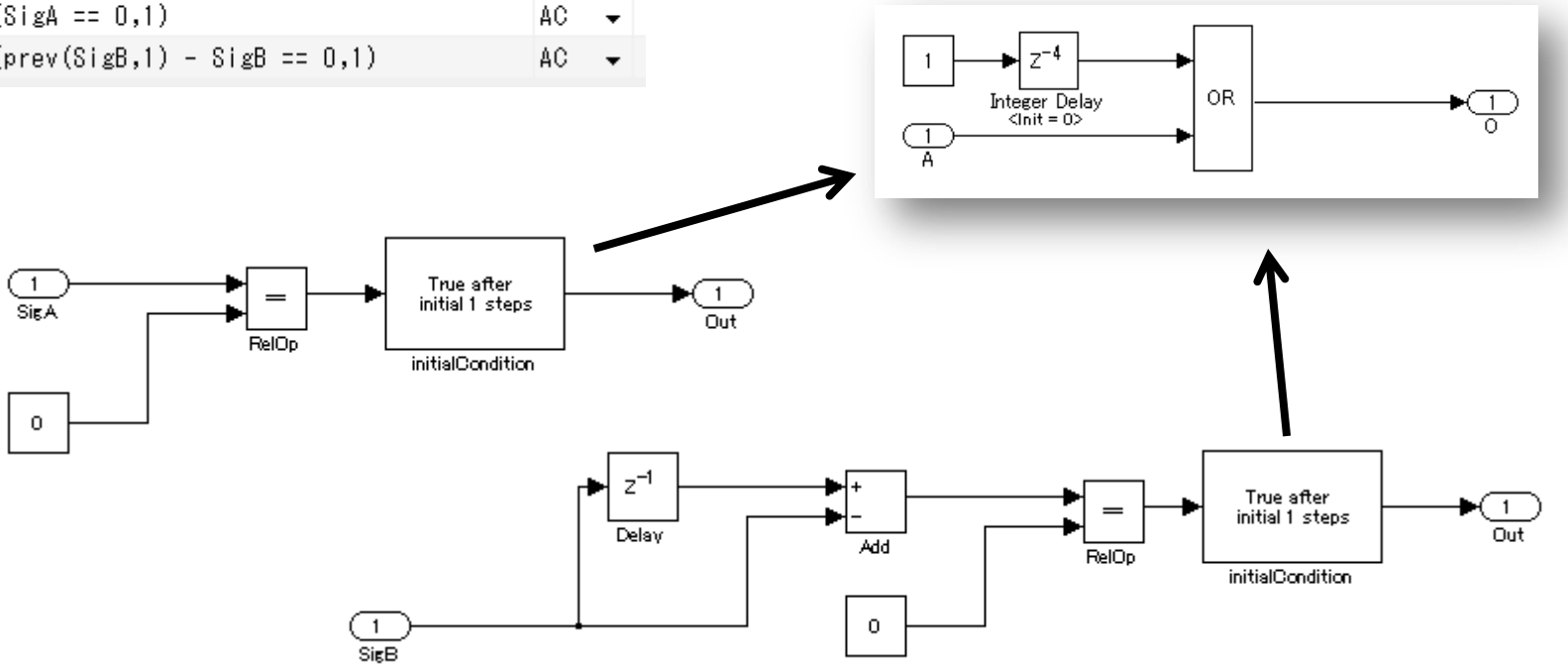
検査式/前提条件
prev(SigA,1) ==0 && SigB == 1 ==> SigD == 1
SigC - prev(SigC,1) < 100



時相論理記述関数: init

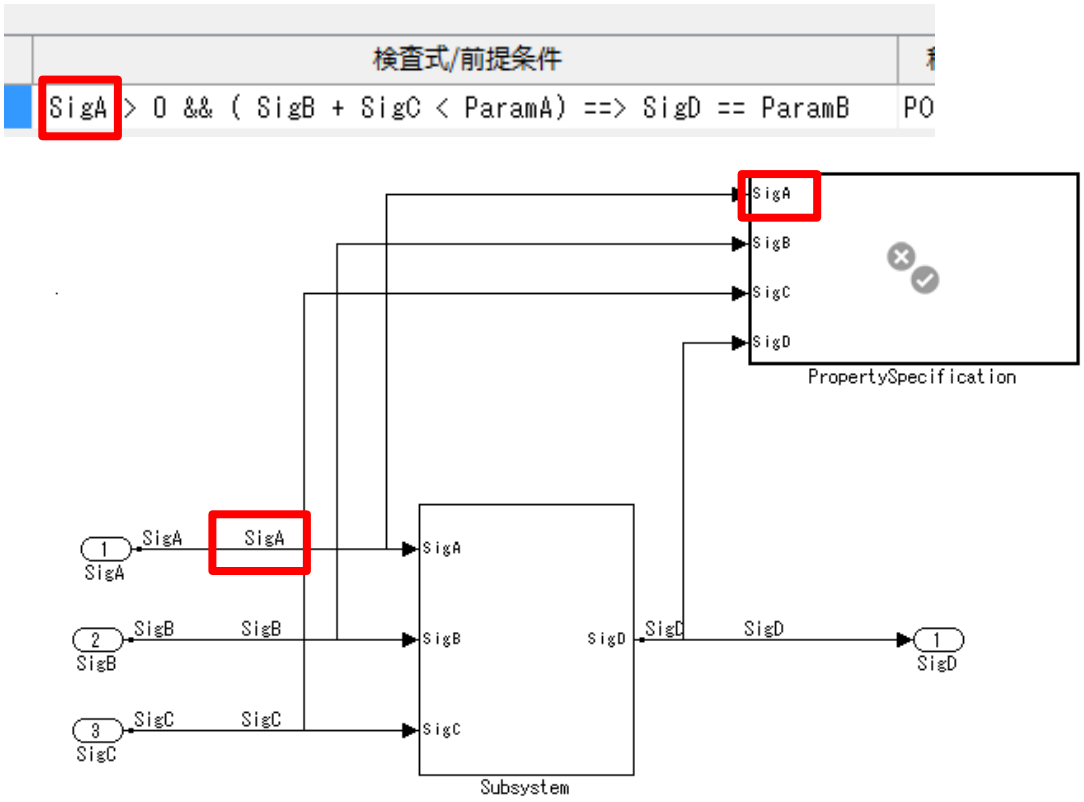
- init(*expr* , *n*)
  - expr がシミュレーション開始時からnステップ成立する

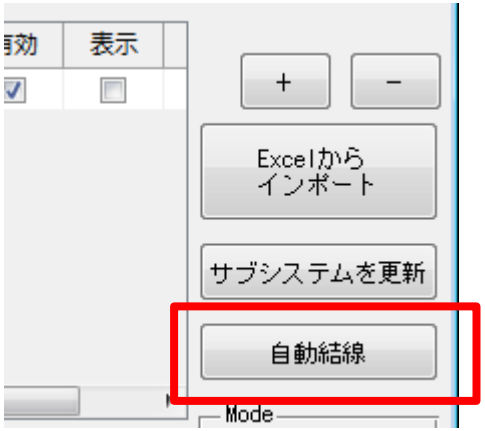
検査式/前提条件	種別
init(SigA == 0,1)	AC
init(prev(SigB,1) - SigB == 0,1)	AC



自動結線機能

プロパティで指定した変数と同名の信号があれば自動的に結線することが可能

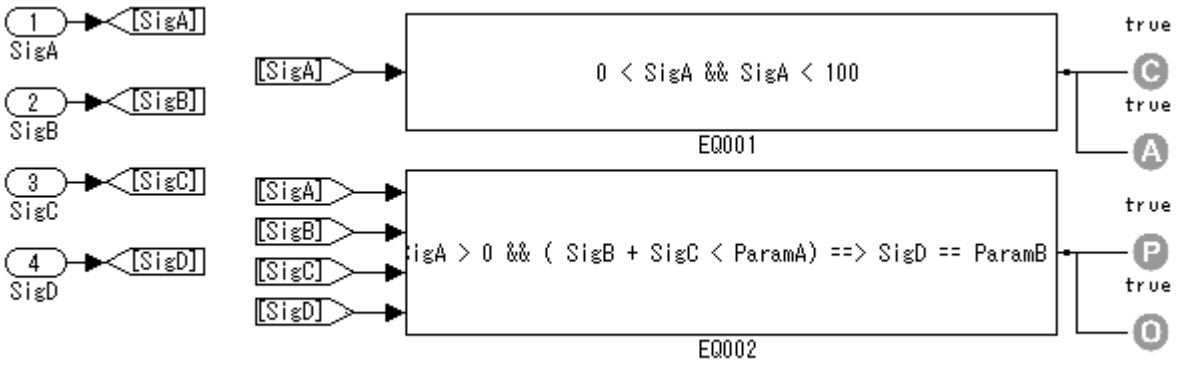




検査式と前提条件

- 種別が「AC」の場合は前提条件(Assumption, Test Condition)に接続される。
- 種別が「PO」の場合はオブジェクティブ(Proof Objective, Test Objective)に接続される。

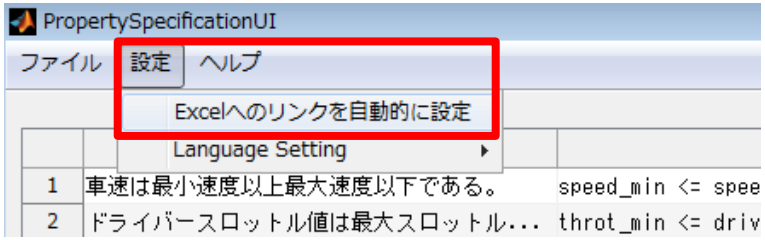
	説明	検査式/前提条件	種別	有効
1	前提1	0 < SigA && SigA < 100	AC	☑
2	検査式1	SigA > 0 && (SigB + SigC < ParamA) ==> SigD == ParamB	PO	☑



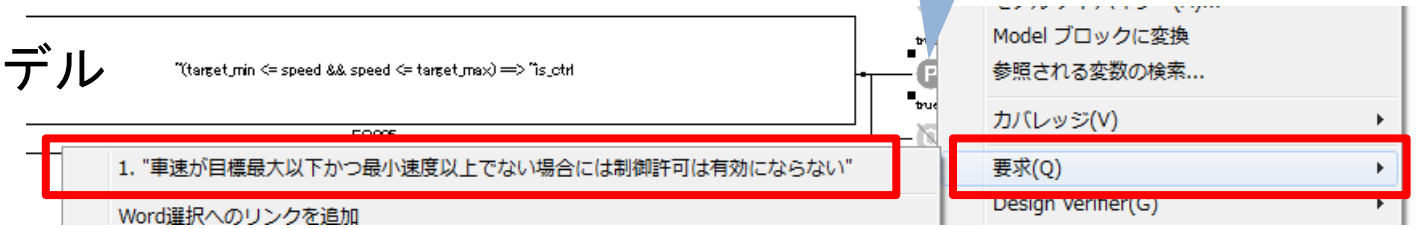
トレーサビリティ設定

Simulink Verification and Validationの要求トレーサビリティ機能を利用して自動的にExcelと命題ブロック間に要求リンクを設定

※開発中の機能です



プロパティモデル



検証仕様書Excel