

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。

PropertySpecificationUI

ファイル 設定 ヘルプ

説明	検査式/前提条件	種別	有効	表示	RMI
1 車速は最小速度以上最大速度以下である。	<code>speed_min <= speed && speed <= speed_max</code>	AO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 ドライバースロットル(値は最大スロットル)...	<code>throt_min <= driver_throt && driver_throt <= ... 40</code>	AO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 加速スイッチと減速スイッチは同時にONに...	<code>!(inc && dec)</code>	AO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 制動可能なブレーキがONの場合には制動と...	<code>is_brake ==> !is_str</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5 車速が目標値以下かつ最小速度以上でな...	<code>!(target_min <= speed && speed < target_max)...</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6 セットスイッチの初期値がOFFであれば制動...	<code>!init(iset,1) ==> !is_str</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7 制動可能な条件が全て満たされる場合...	<code>enable && !is_brake && !iset prev(is_str)...</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8 加速スイッチがONの場合には目標車速は前...	<code>inc ==> (target - prev(target,1)) == vel_inc</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9 減速スイッチがONの場合には目標車速は前...	<code>dec ==> (target - prev(target,1)) == -vel_dec</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10 加速スイッチおよび減速スイッチが共に無...	<code>dec && inc ==> target - prev(target,1) == 0</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11 目標車速は、目標車速 - 車速である。[...	<code>diff_target == target - speed</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12 目標車速は最小車速以上、最大車速以下を...	<code>target_min <= target && target <= target_max</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13 出力スロットル値は最大スロットル値以下...	<code>throt_min <= throt && throt <= throt_max</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14 制動可能なOFF状態には、ドライバースロ...	<code>!is_str ==> driver_throt == throt</code>	P0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Excelからインポート

サブシステムを参照

自動生成

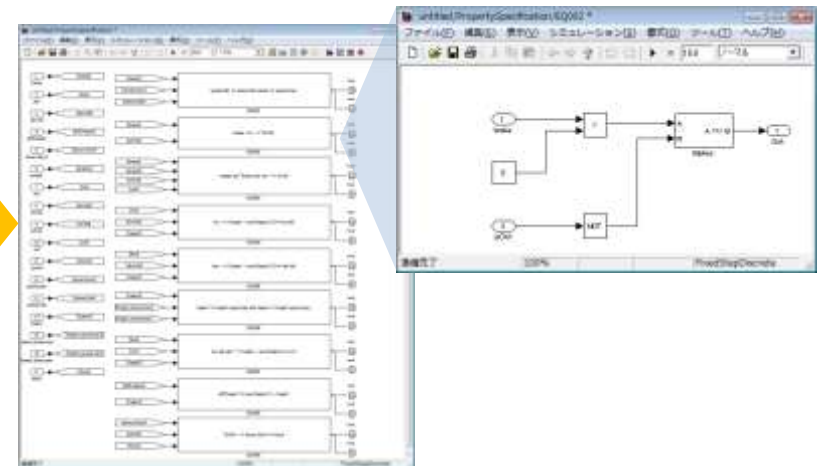
Mode

テスト生成

プロパティ検証

Temporals

ステップ監視



注意事項とお願い

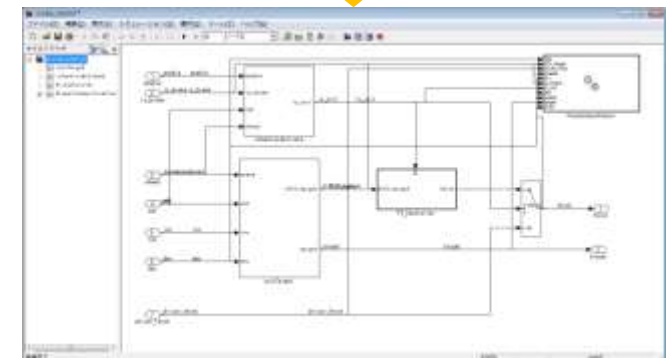
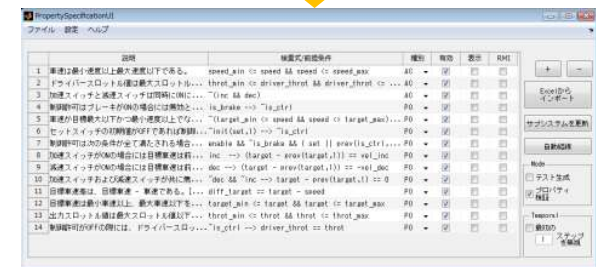
- 本ツールは、プロトタイプ版であり、将来の製品機能として取り込むべく、米国開発チームと共同で開発を進めています。
- 本ツールの利用の結果生じた、いかなる損害についてMathWorksは一切の責任を負いません。
- ご利用状況の把握のため、本ツールを他社・他部署へ配布される場合には、必ずMathWorksの担当営業までご一報下さい。
- フィードバックなどがあれば、是非お知らせください。

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

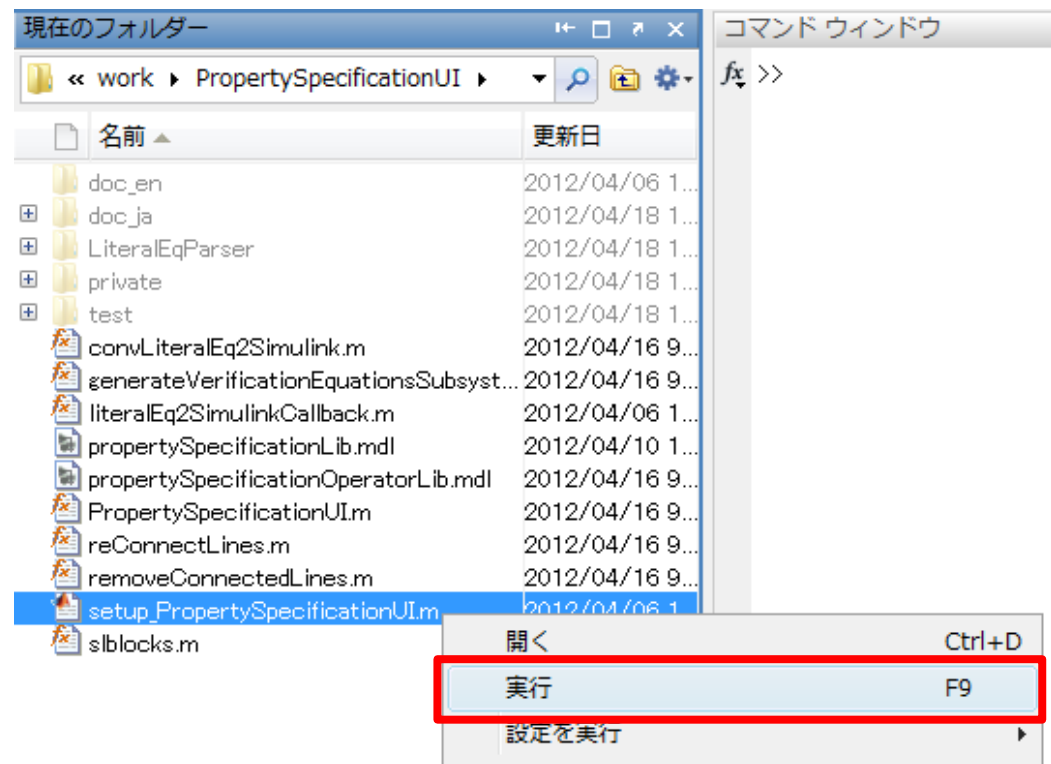


説明	CPDの検査式、前提条件
車速は最小速度以上最大速度以下である。	<code>speed_min <= speed && speed <= speed_max</code>
ドライブスロットルは最大スロットル以下、最小スロットル以上である。	<code>throt_min <= driver_throt && driver_throt <= throt_max</code>
加速スイッチと減速スイッチは同時にONにならない。	<code>!(inc && dec)</code>
ブレーキスイッチがONの場合には加速スイッチがOFFである。	<code>is_brake ==> !is_ctl</code>
車速が目標値以下かつ最小速度以上でない場合には制動許可は有効にならない。	<code>!(target_min <= speed && speed <= target_max)</code>
セットスイッチの初期値がOFFであれば制動許可初期値もOFFである。	<code>!init(target) ==> !is_ctl</code>
制動許可は次の条件が全て満たされる場合にONとなる。 ・ 加速スイッチがOFF ・ ブレーキがOFF	<code>enable && !is_brake && !set prev(is_ctl) && enable && used <= target_max ==> is_ctl</code>



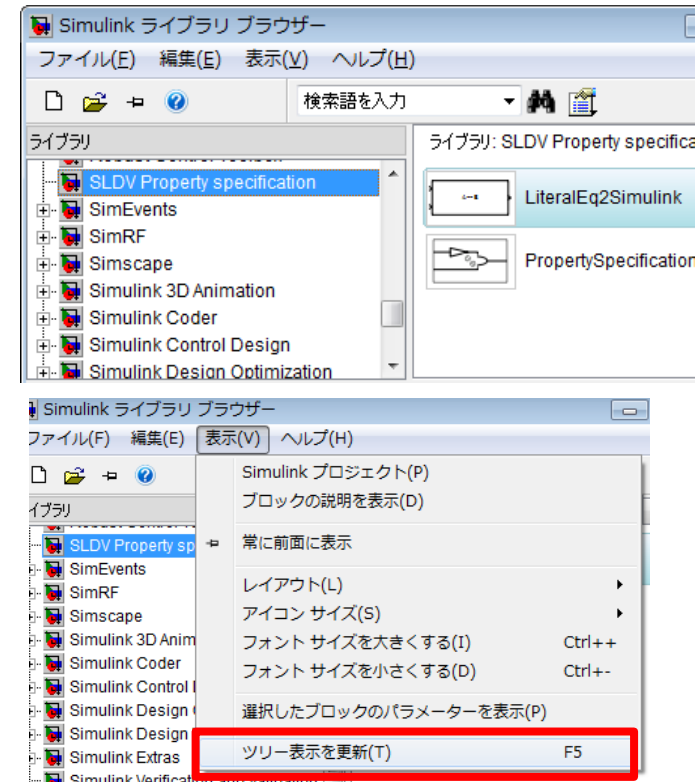
インストール方法

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



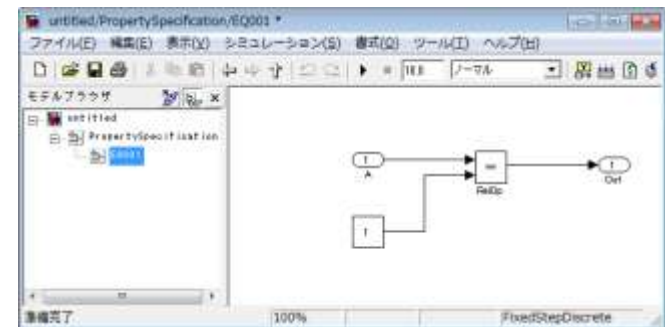
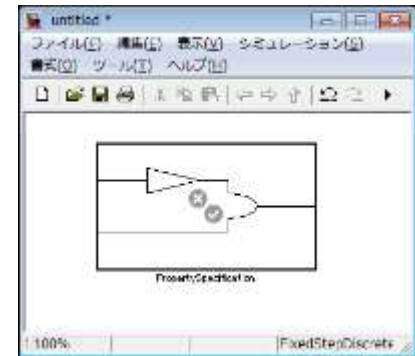
動作の確認(1)

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

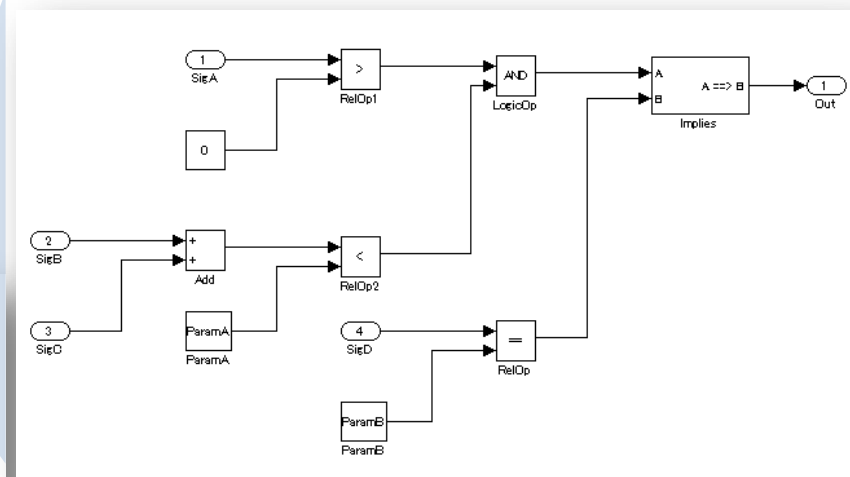
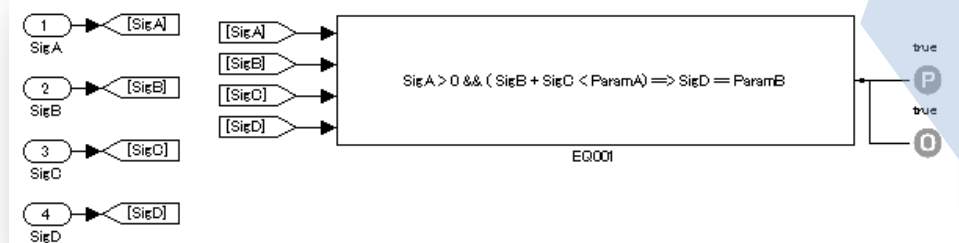
例:

$\text{SigA} > 0 \ \&\& \ (\text{SigB} + \text{SigC} < \text{ParamA}) \implies \text{SigD} == \text{ParamB}$

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

PropertySpecificationUI			
ファイル 設定 ヘルプ			
	説明	検査式/前提条件	
1	式1	$\text{SigA} > 0 \ \&\& \ (\text{SigB} + \text{SigC} < \text{ParamA}) \implies \text{SigD} == \text{ParamB}$	P0



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

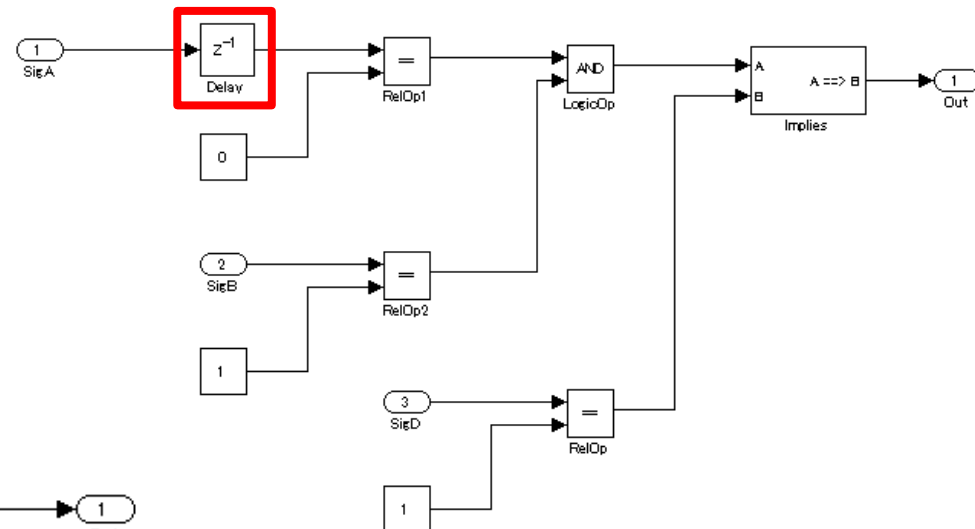
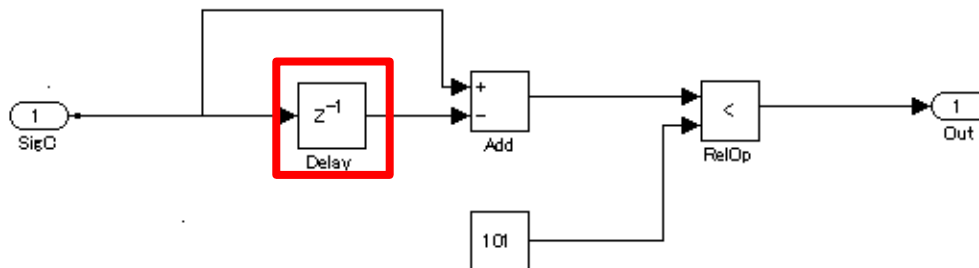
時相論理記述関数: prev

- $\text{prev}(\text{expr}, n)$
 - expr を n ステップ遅らす

検査式/前提条件

$\text{prev}(\text{SigA}, 1) == 0 \ \&\& \ \text{SigB} == 1 \implies \text{SigD} == 1$

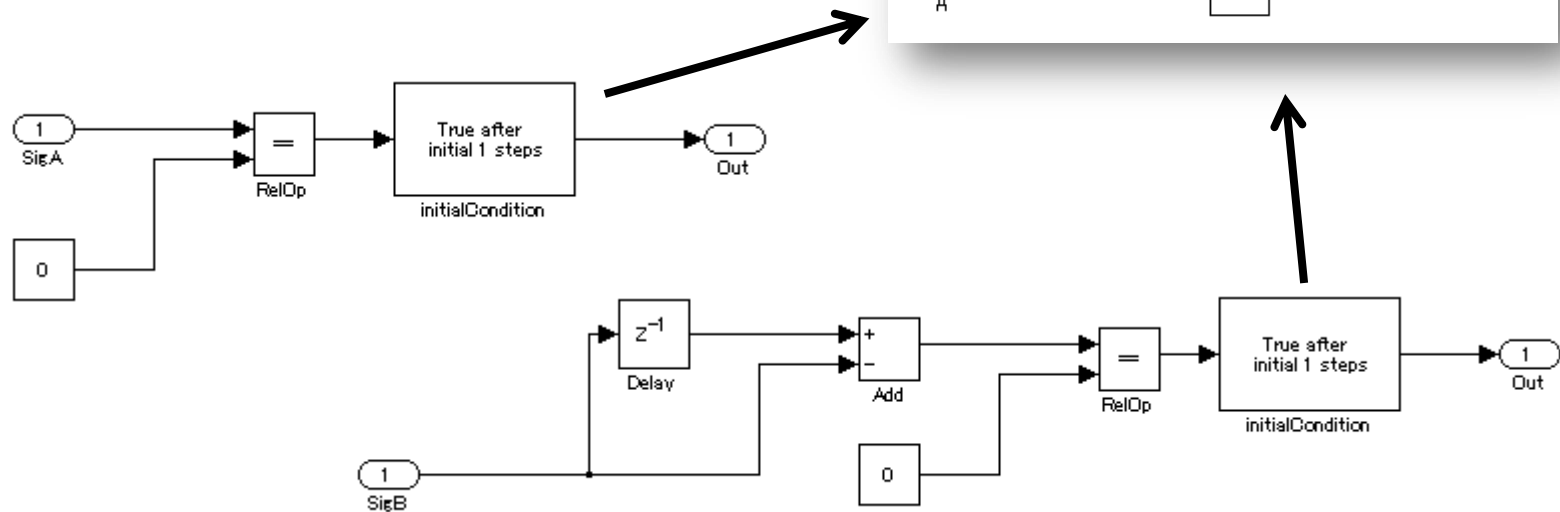
$\text{SigC} - \text{prev}(\text{SigC}, 1) < 100$



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

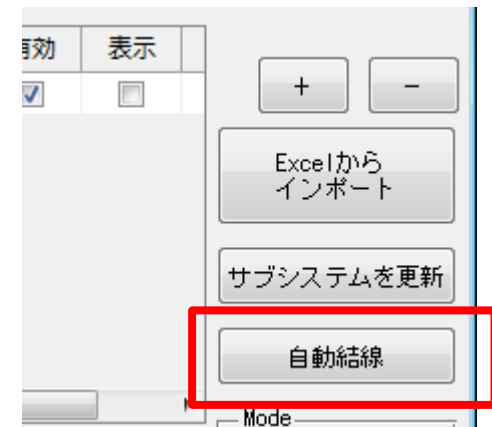
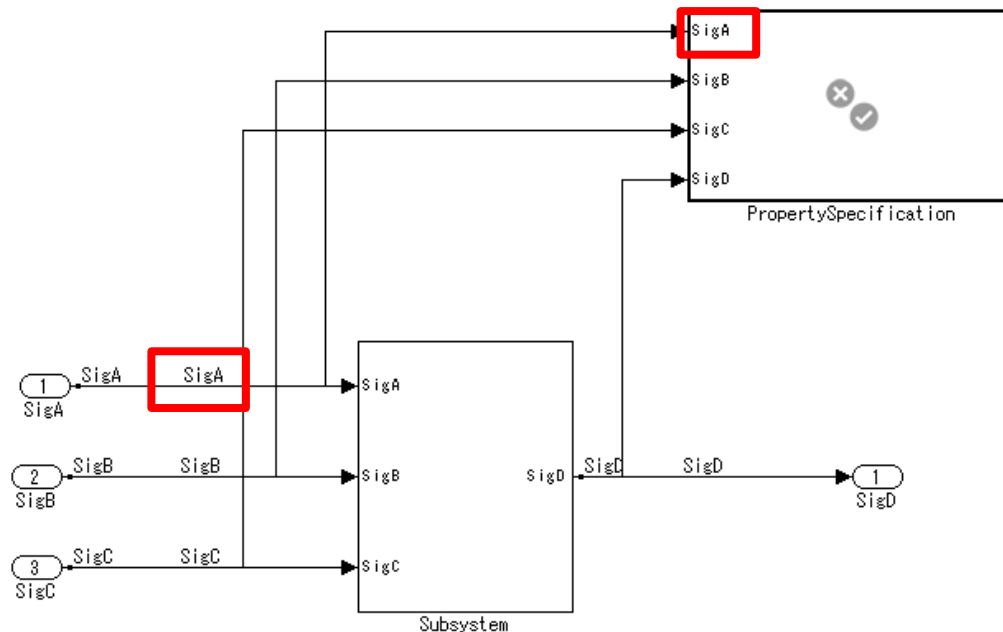
検査式/前提条件	種別
<code>init(SigA == 0, 1)</code>	AC ▼
<code>init(prev(SigB, 1) - SigB == 0, 1)</code>	AC ▼



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

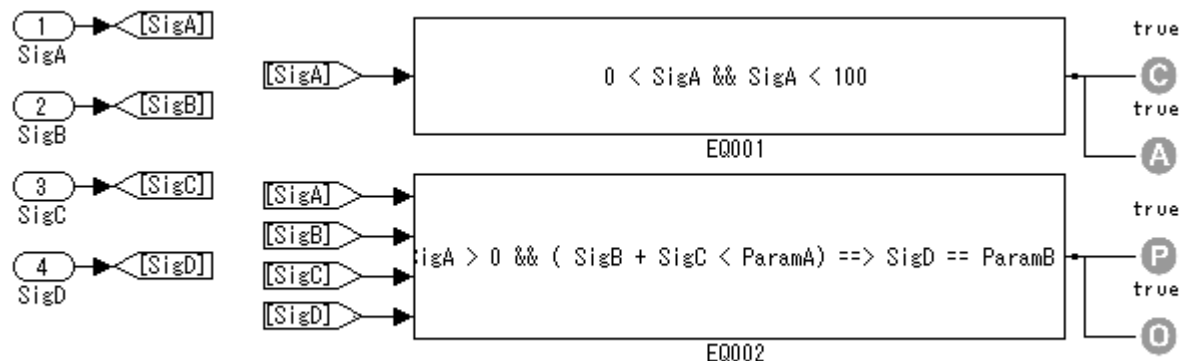
検査式/前提条件	
<code>SigA > 0 && (SigB + SigC < ParamA) ==> SigD == ParamB</code>	P0



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。

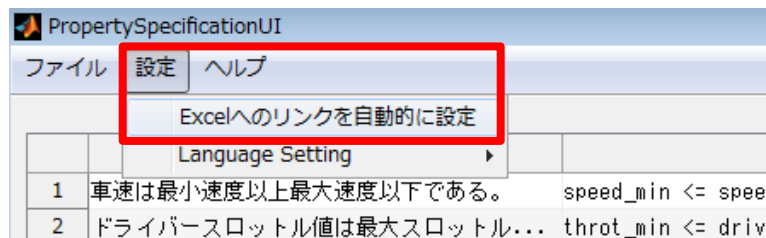
	説明	検査式/前提条件	種別	有効
1	前提1	$0 < \text{SigA} \ \&\& \ \text{SigA} < 100$	AC ▼	<input checked="" type="checkbox"/>
2	検査式1	$\text{SigA} > 0 \ \&\& \ (\text{SigB} + \text{SigC} < \text{ParamA}) \Rightarrow \text{SigD} == \text{ParamB}$	PO ▼	<input checked="" type="checkbox"/>



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



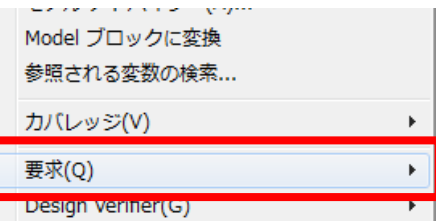
ブロックを右
クリック

プロパティモデル

$\sim(target_min \leq speed \ \&\& \ speed \leq target_max) \Rightarrow \sim is_ctrl$

1. "車速が目標最大以下かつ最小速度以上でない場合には制御許可は有効にならない"

Word選択へのリンクを追加



検証仕様書Excel

加減速スイッチと減速スイッチは同時にONにはならない。	$(inc_brake \ \&\& \ dec_brake) \Rightarrow 0$	AC
クルーズコントロール有効化判断		
制御許可はブレーキがONの場合には無効となる。	$is_brake \Rightarrow \sim is_ctrl$	PO
車速が目標最大以下かつ最小速度以上でない場合には制御許可は有効にならない	$\sim(target_min \leq speed \ \&\& \ speed \leq target_max) \Rightarrow \sim is_ctrl$	PO
ヒットスイッチの初期値がOFFであれば制御許可初期値もOFFである	$init(set,1) \Rightarrow \sim is_ctrl$	PO