T-VEC Technologies, Inc.

Test Sequences Generation from Control System Models
(制御システムモデルからのテストシーケンス生成)

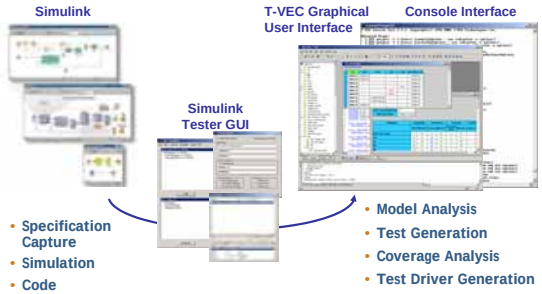
Presented at
JaSST: Japan Symposium on Software Testing

By:
Robert D. Busser
T-VEC Technologies, Inc

Automated Verification and Validation 自動V&V

- Model Analysis モデルの解析
  - Identifies defects in model's logic and input signal domain expectations
    (モデル上の欠陥や、入力値範囲の見込み違いを発見)
- Test Generation From Models モデルからテストを生成
  - Test Vectors - Input values and expected output values for every logical path in the model *(モデル上の全論理パスに対して入力と予測出力値のベクタ)*
  - Test Sequence Vectors - Input values and expected output values for each of instance of a system's execution over multiple cycles
    (テストシーケンスベクタは、マルチサイクルな実行にわたる入力と予測出力値)
- Test Driver Generation テストドライバーを生成
  - Automatic creation of test drivers from test vectors and test sequence vectors *(テストベクタ・シーケンスベクタからテストドライバの自動生成)*
  - Test drivers for application source code, verifies code against model
    (テストドライバでモデルに対してアプリケーションコードを検証)
  - Test drivers for Matlab simulator, helps validate model and verify that the model translator and test vector generator have properly interpreted the model's semantics *(モデルの意図を実証および検証するテストドライバー)*
  - Test driver mechanism based on template description of test drivers can be customized for other platforms *(テストドライバーは如何なる環境へも対応)*

T-VEC Environment with Simulink モデルでの例



- Specification Capture
- Simulation
- Code Generation

モデルツール

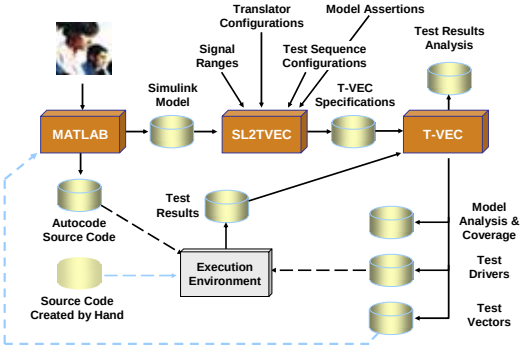
- Model Translation

モデル情報を変換

- Model Analysis
- Test Generation
- Coverage Analysis
- Test Driver Generation
- Test Results Analysis

テスト生成～実行・結果解析

Process Overview



Model Analysis モデルの解析

- Identifies model defects
  - Logic errors - (e.g., unreachable paths)
    ロジックエラー (到達され得ないパスなど)
  - Mathematic exceptions (e.g., divide by zero)
    数式上の考え違い
  - Range specification errors
    入力領域仕様の誤り
  - Type overflows
- Prove model properties through assertions
  - Specify additional relationships for model
    モデル特性を実証する為の、追加記述を指定 (アサーション)

Test Vector Generation テストベクタ生成

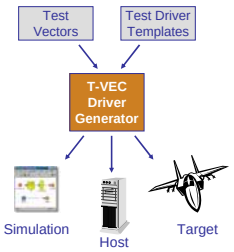
- Unit, integration and system level tests
  (ユニット、統合、システムレベルテスト)
- Test vectors include test inputs, expected test outputs, and traceability to model
  (テストベクタは、入力・予測出力値、モデルへのトレーサビリティ)
- Test output values derived independently from Simulink
  (モデルの依存性なく、予測出力値が生成される)
- Configurable test coverage *(テストカバレッジ、～MCDC レベル)*
  - Default - covers explicit paths (if, switch)
  - Path - covers all paths (implicit and explicit)
  - Condition - covers all conditions (MC/DC)
- Test sequences for testing dynamic aspects of model
  (動的な振舞いの為の、テストシーケンスベクタ)

Test Vector Generation (cont) テストベクタ生成

- All model control paths are identified
  (モデル上の全コントロールパスを確認)
- For each path, constraints are analyzed
  (各パスごとに収束条件が解析される)
- Tests are selected along constraint boundaries
  (テストは収束される領域に沿って選択)
- Test cases stress constraint and input boundaries
  (テストケースは収束条件および入力境界への負荷)
- User can add constraints to model to test specific properties (追加の収束条件は、更なる特性の解析、実証など)

Test Driver Generation テストドライバー生成

- User customizable templates generate multiple outputs
- Included templates:
  - C test drivers for RTW GRT and ERT code generator
  - Simulation scripts for Matlab/Simulink
- Supports test drivers for any language, script, or execution environment
- Templates provided for code coverage tools



テストドライバー生成用テンプレートは、いかなる言語・スクリプト・実行環境に対応。
また、コードカバレッジツールなど別のツールとも柔軟に組合せできる。

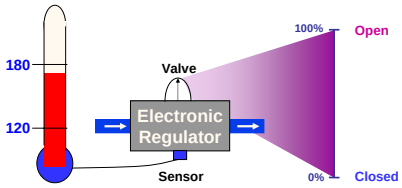
Test Sequence Vectors テストシーケンスベクタ

- A test vector is a single input value / output value set used for testing a single execution cycle
  - State variables are set to initial values specified in model
- Verifying dynamic behavior requires tests that span multiple sample periods
  - State variables initialized to values specified in model
  - State variables updated between each subsystem reference during the vector generation process
- Test Sequence Vectors include test input values and expected output values for multiple cycles of execution
- Provides ability to verify state changes over time and the dynamic response of closed-loop designs to changing input signal values

システムの動的な振舞いの検証には、マルチサンプルなテストベクタが必要。
テストシーケンスベクタは、クローズドループなどの刻々と変化する状態とその応答の検証もできる。

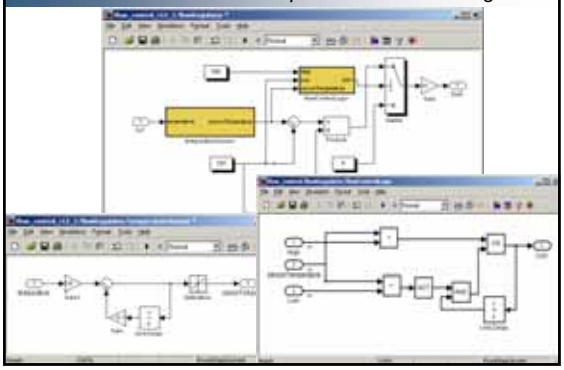
Dynamic Systems Have Time Complexities - Flow Regulator

Temp Range [-100…300]

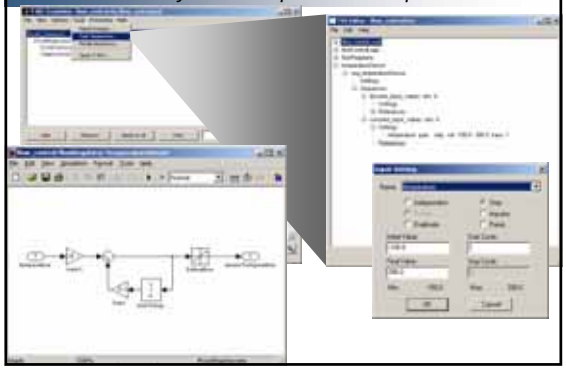


Operational Summary
If **Temp** is less than **120**, then valve is **Closed**
If **Temp** is becomes greater or equal to **180**, then valve is **Open**
and remains **Open** until Temp drops below **120**

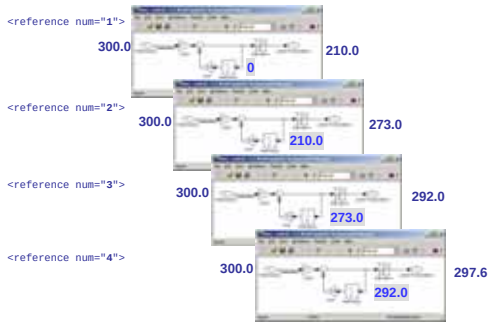
Example Simulink Flow Regulator



TSV Editor – Dynamic Properties of temperatureSensor



Sequences Includes Feedback via Unit Delay

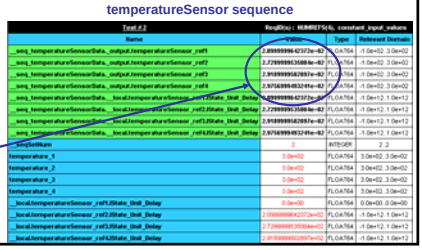


Differences Between Single Cycle and Sequence

- Key information in the single state
- Vector values for temperature selected at high and low bound values
- Unit delay fixed at zero - its initial value
- Temperature sequences from 125, 130 122, to 119
- Signal undergoes first order filter
- Values produced show filtering action over multiple cycles

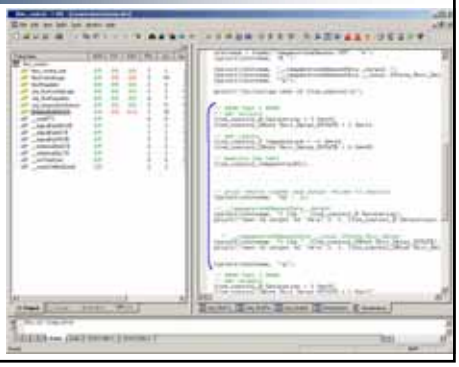
temperatureSensor single cycle

temperatureSensor sequence



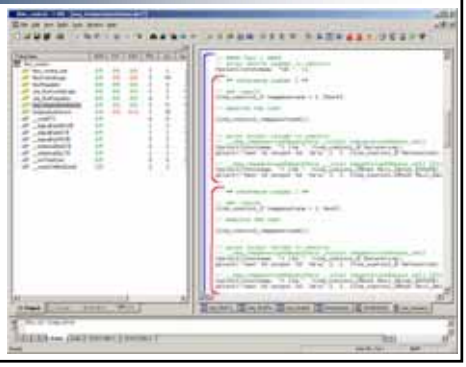
Single Cycle Test Driver

- 1 Test = 1 Cycle of subsystem execution

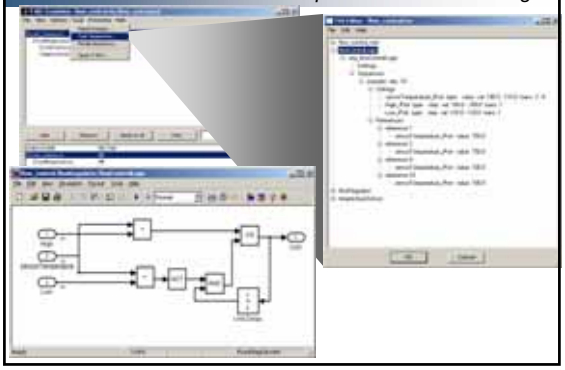


Test Sequence Test Driver

- 1 Test = 4 cycles of subsystem executions

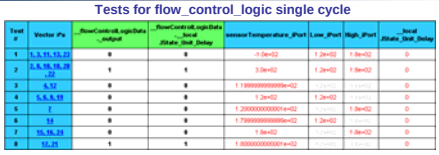


TSV Editor – State Machine Aspects of flowControlLogic

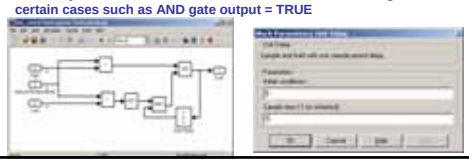


Differences Between Single Cycle and Sequence

Tests for flow_control_logic single cycle

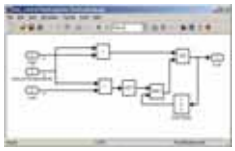


- Vector values for temperature selected at high and low bound values
- Single Cycle tests can cover most of the logic conditions
- UnitDelay blocks fixed at 0 - it's initial value - limits coverage of certain cases such as AND gate output = TRUE

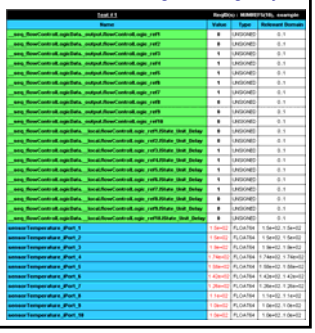


Differences Between Single Cycle and Sequence

- Temperature values based on complex sequence
- Signal goes above the upper limit at cycle 3
- But begins to ramp down falling below lower limit at cycle 8
- However, latching keeps the output value at 1 from the time signal exceeded upper limit until the time it fell below lower limit
  - AND gate value must be 1

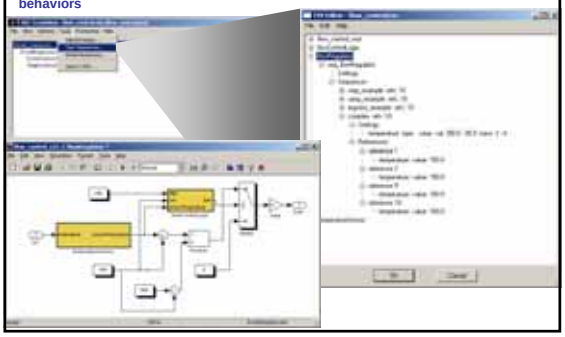


Tests for flow_control_logic involving 10 cycles



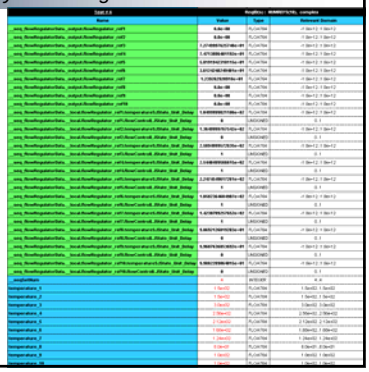
System Integration Behavior Over Time

- Test Sequences can be applied at higher levels to demonstrate system integration behaviors



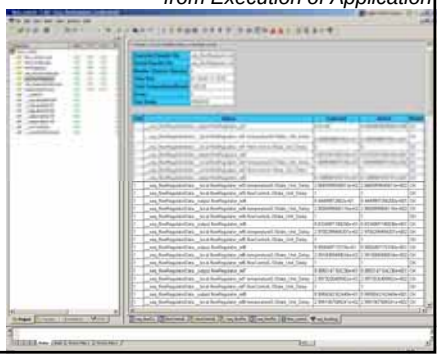
System Integration Behavior Over Time

- The integration of the temperature sensor filtering action with the flow_control_logic latching action can be verified most effectively using the TSV approach



Comparing Expect Results (Model) to Actual Results from Execution of Application

- Comparison of expected output values (T-VEC) and actual values produced by executing the application code over 10 cycles
- All values comparisons passed, indicating application correctly implements the semantics of the flow control model



Summary

- Simulink models can be translated into a standardized form for the purpose of independent analysis outside of Simulink (SL2TVEC)
- Simulink models can be analyzed for defects in logic and computational relationships using an independent application (T-VEC)
- Test vectors and Test sequence vectors can be automatically derived from Simulink models using an independent application (T-VEC)
- Test drivers can be automatically produced from test vectors and test sequence vectors for the purpose of verifying a model's correct implementation (T-VEC)
- Test vectors verify that individual logic conditions and computations are correctly implemented in target application
- Test sequence vectors verify that the dynamic time-sensitive properties of a model are correctly implemented in target application

モデルツールとは独立した環境で、モデル上の欠陥を解析、テストベクタ 及びシーケンスベクタを自動生成。 生成されるテストドライバーはモデルが正しくコード化されたかの検証。 テストベクタはモデル上の各要素の検証、テストシーケンスベクタはモデル上の動的特性の検証に使用される。