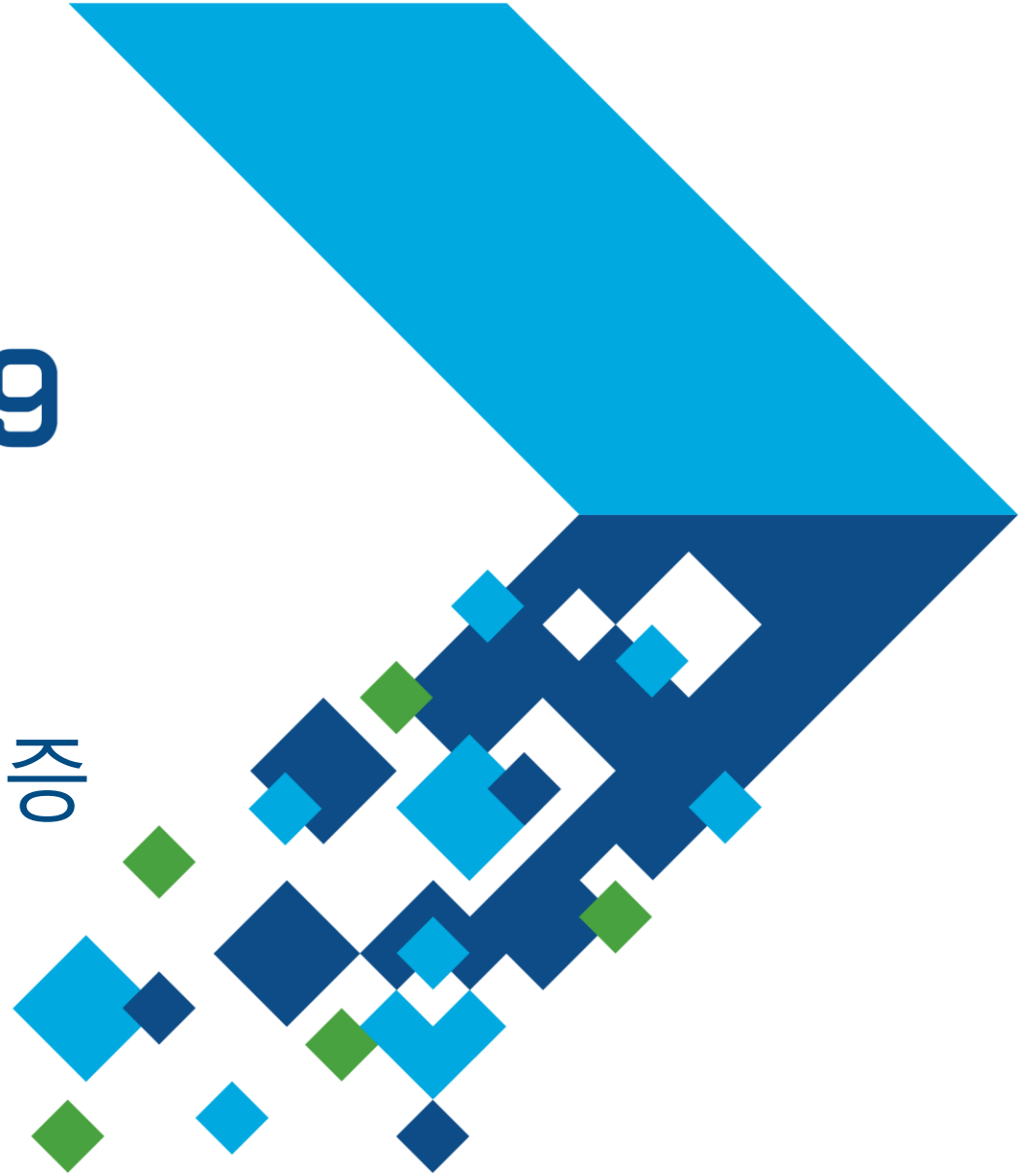
MATLAB EXPO 2019

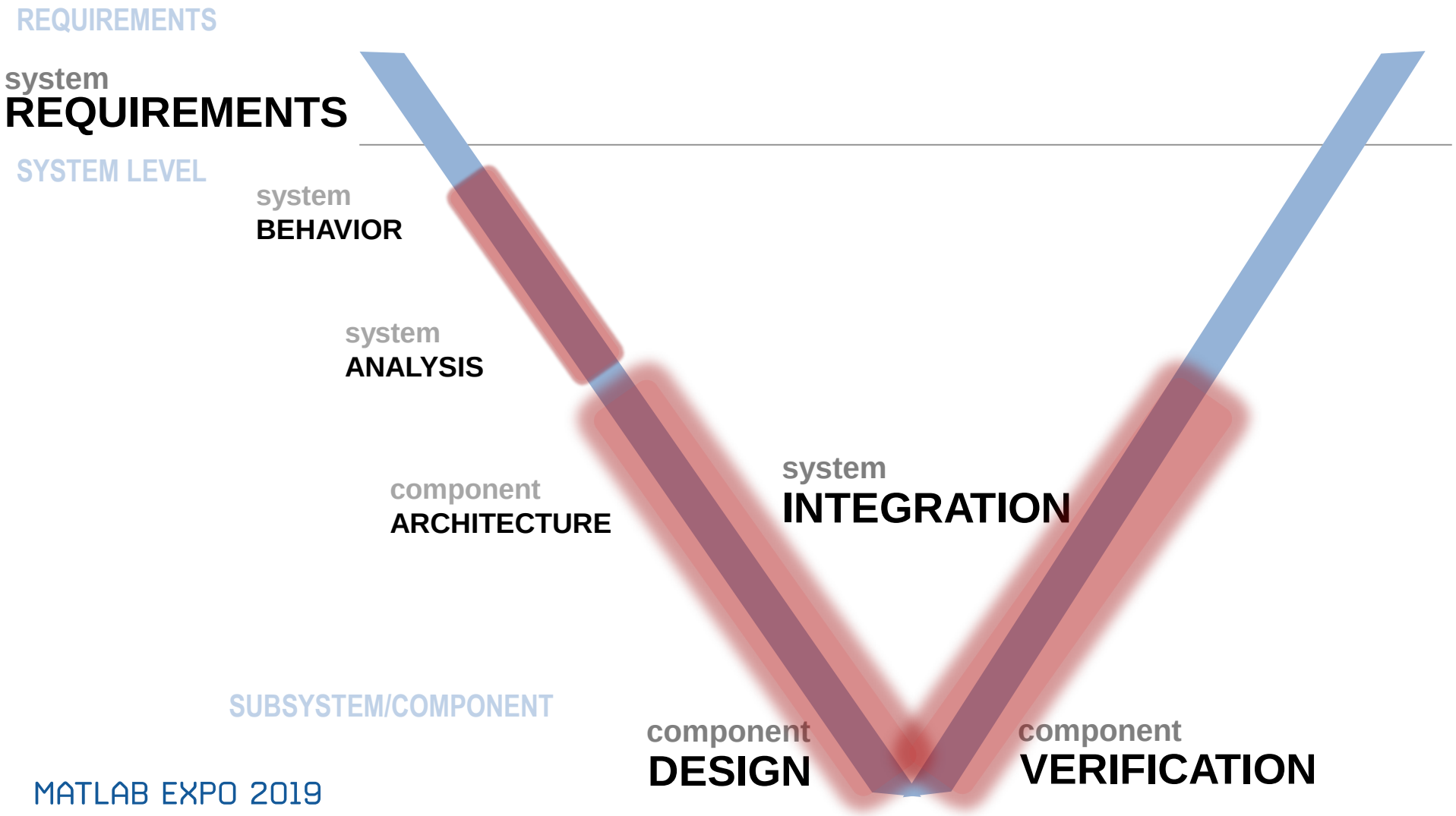
Simulink 기반 Legacy C/C++ Code 통합, 결과 시각화 및 검증 방안

유성재



Model-Based Design

Systematic use of models throughout the development process



Legacy Code? Hand-written code?

Components Under Development

```
template<class InputString, class Ou
bool unhexlify(const InputString& in
  if (input.size() % 2 != 0) {
    return false;
  }
  output.resize(input.size() / 2);
  int j = 0;
  auto unhex = [](char c) -> int {
    return c >= '0' && c <= '9' ? c - '0' :
           c >= 'A' && c <= 'F' ? c - 'A' + 10 :
           c >= 'a' && c <= 'f' ? c - 'a' + 10 :
           -1;
  };

  for (size_t i = 0; i < input.size(); i += 2) {
    int highBits = unhex(input[i]);
    int lowBits = unhex(input[i + 1]);
    if (highBits < 0 || lowBits < 0) {
      return false;
    }
    output[j++] = (highBits << 4) + lowBits;
  }
  return true;
}
```

Hand Code

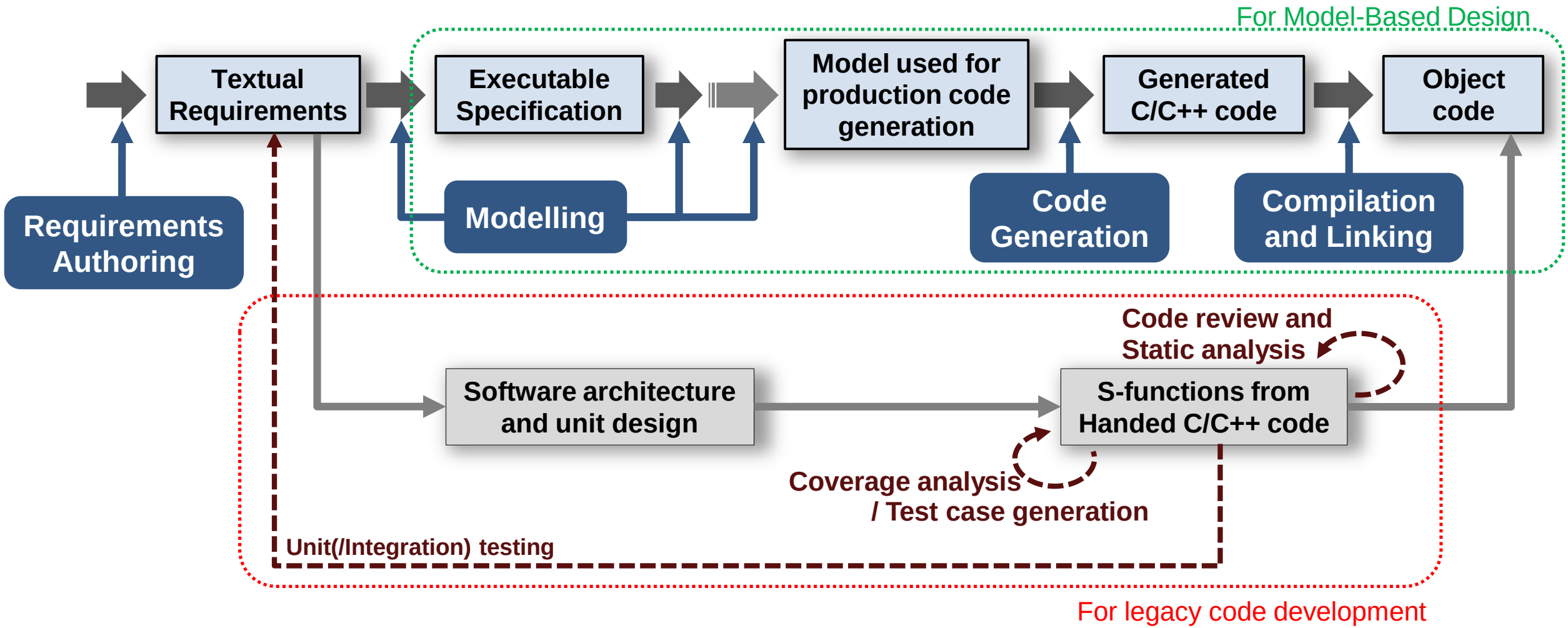
Internal Libraries

Verified Components

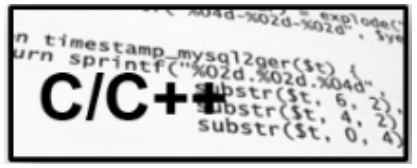
Vendor Libraries

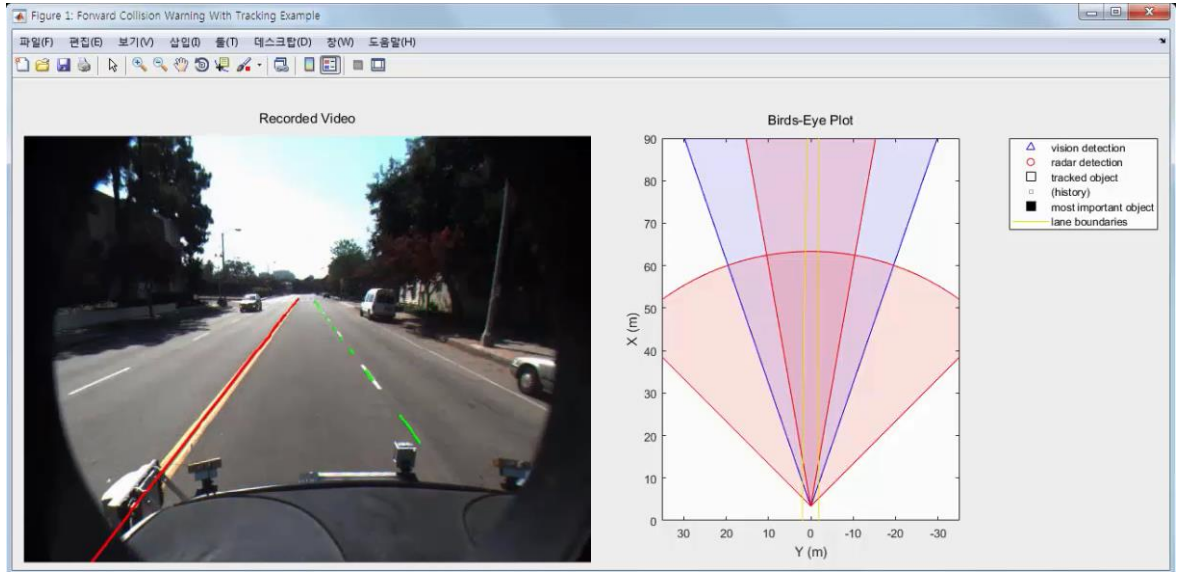
Device Drivers

MBD with Legacy Code!



Challenge?





File Contents/Complexity — **Test 1**

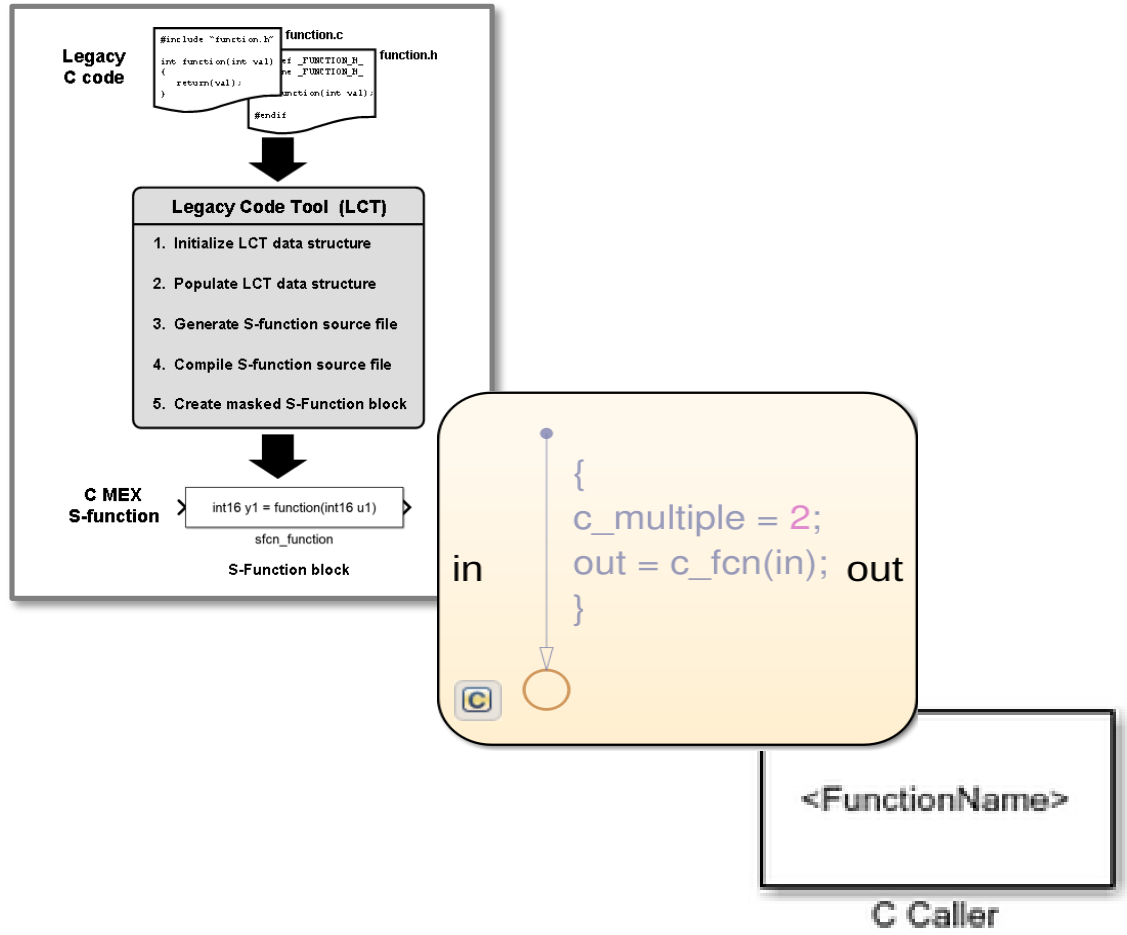
		Decision	Statement	Function	Function call
1 . mExportFunction.c	4	100%	100%	100%	100%
2 . . . function1	2	100%	100%	100%	--
3 . . . mExportFunction_initialize	1	--	100%	100%	100%
4 . . . mExportFunction_terminate	1	--	100%	100%	--

Agenda

- **Legacy code integration using Simulink**
- Visualization using Simulink
- Verification with legacy code

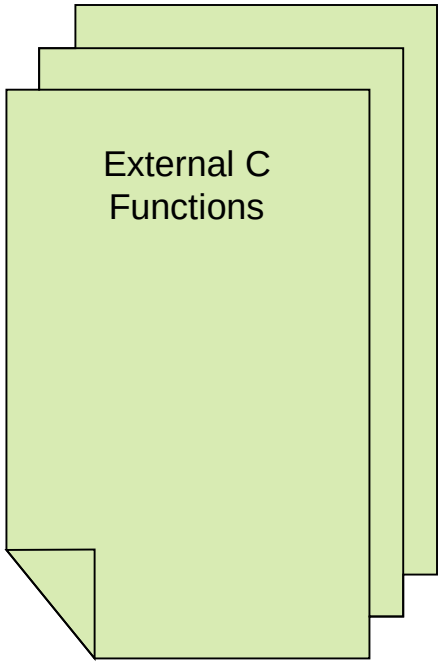
How to Import Legacy Code

- Legacy Code Tool
- Legacy code integration in Stateflow
- C Caller Block

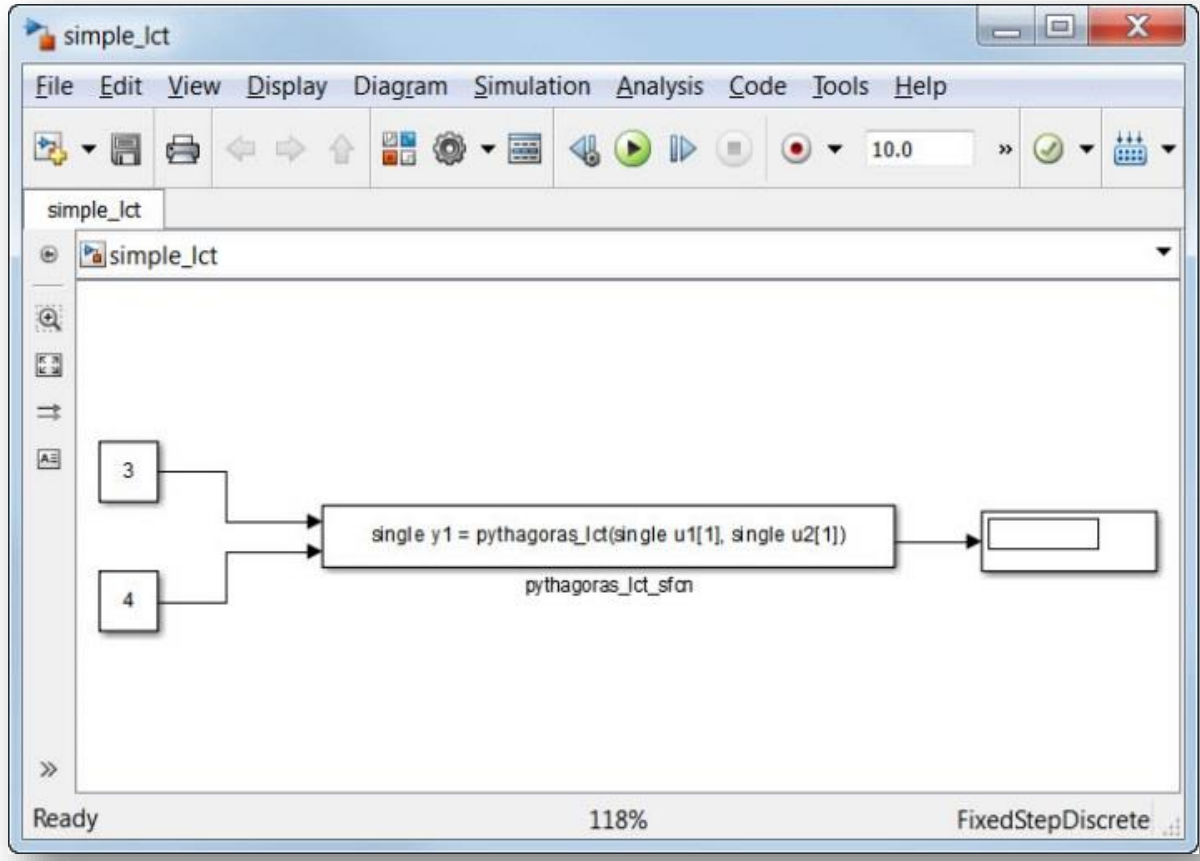


What legacy C code integration in Simulink means?

- Legacy Code Tool enables existing C code to be used in Simulink models

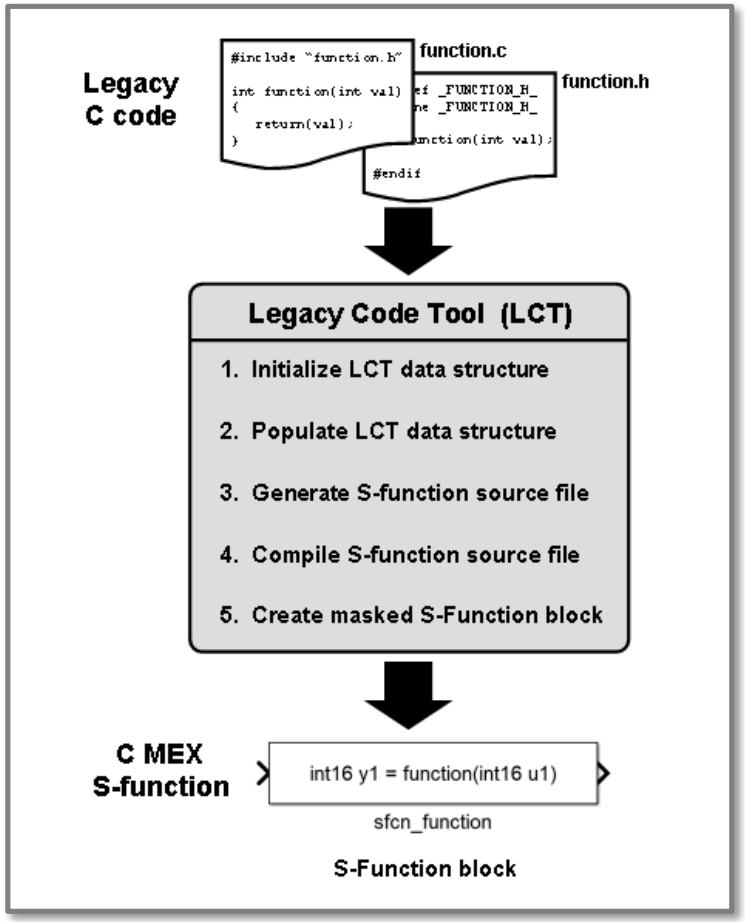


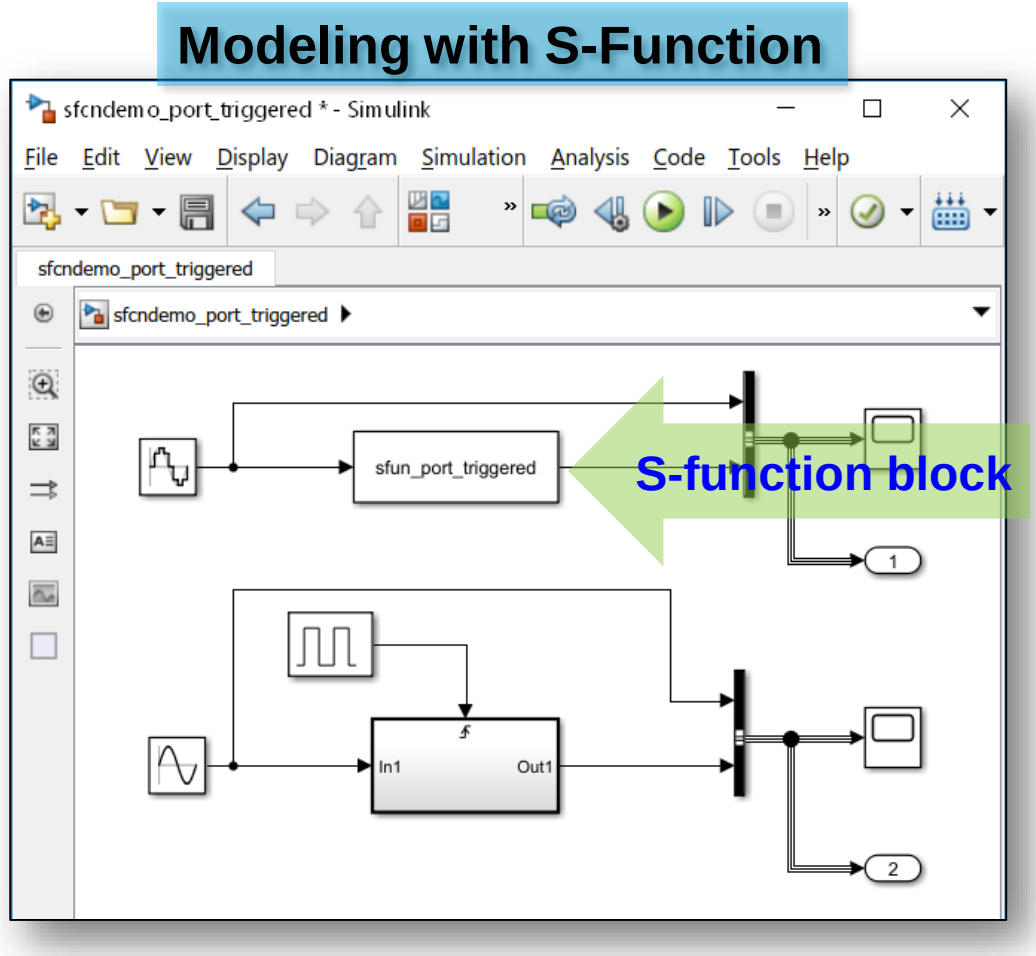
Generating S-function



How to use Legacy Code Tool?

- General procedure for using Legacy Code Tool





Maintenance Problem…

Legacy code

```
/*cruise controller main runnable*/
void CrsCntrlerMain(void)
{
    boolean_t l_u1IncSetLong;
    boolean_t l_u1DecSetLong;

    /* setting increase button is pressed for long ti
    if (u1IncSw == TRUE)
    {
        if (u8IncCnt >= 50)
        {
            l_u1IncSetLong = TRUE;
        }
        else
        {
            u8IncCnt = u8IncCnt + 1;
            l_u1IncSetLong = FALSE;
        }
    }
    else
    {
        u8IncCnt = 0U;
        l_u1IncSetLong = FALSE;
    }

    /* setting decrease button is pressed
    if (u1DecSw == TRUE)
    {
        if (u8DecCnt >= 50)
        {
            l_u1DecSetLong = TRUE;
        }
        else
        {
            u8DecCnt = u8DecCnt + 1;
            l_u1DecSetLong = FALSE;
        }
    }
    else
    {
        u8DecCnt = 0U;
        l_u1DecSetLong = FALSE;
    }
```

Wrapper code

```
#include "CrsCntrl_Wrapper.h"

void CrsCntrl(boolean_t u1, boolean_t u2, boolean_t u3, boolean_t u4, boolean_t u5, boolean_t u6,
              int16_t u7, uint8_t u8, uint8_t u9, int32_t u10, int32_t u11,
              uint8_t *y1, boolean_t *y2, uint8_t *y3, int32_t *y4, int32_t *y5)
{
    u1EnblSw    = u1;
    u1CnclSw    = u2;
    u1SetSw     = u3;
    u1ResumeSw  = u4;
    u1IncSw     = u5;
    u1DecSw     = u6;
    s16BrakeP   = u7;
    u8Key       = u8;
    u8Gear      = u9;
    s32ThrotDrv = u10;
    s32VehSpd   = u11;

    CrsCntrlerMain();

    *y1 = enumReqDrvOut;
    *y2 = u1StatusOut;
    *y3 = enumModeOut;
    *y4 = s32TargetSpOut;
    *y5 = s32ThrotCcOut;
```

Script file

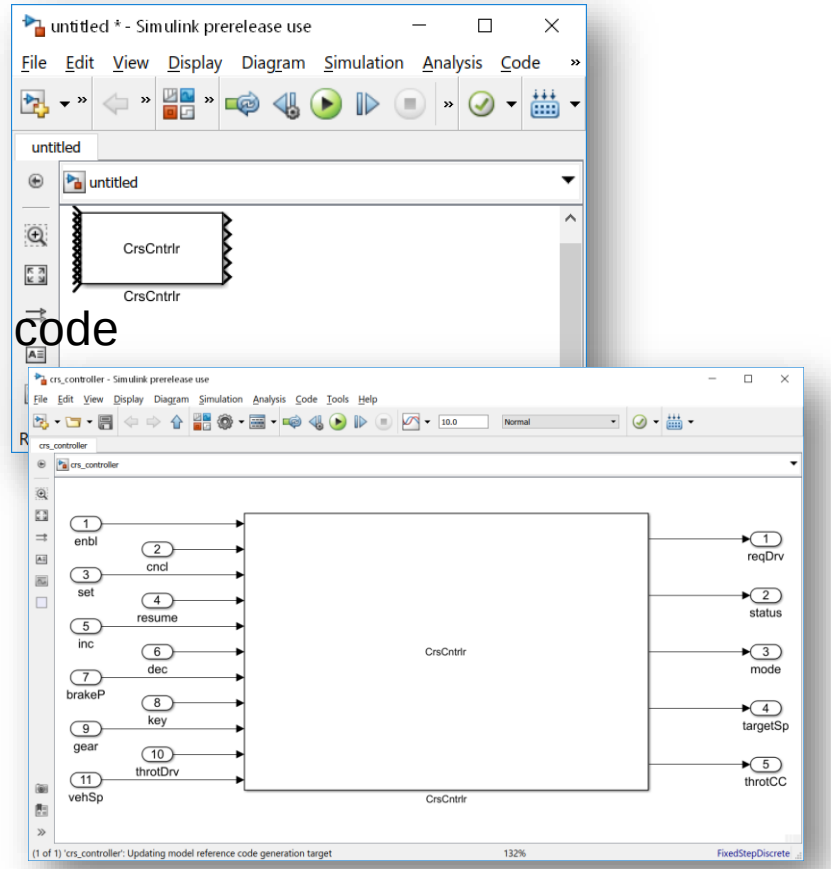
```
def.SourceFiles   = {'CrsCntrl_Wrapper.c', 'CruiseCntrlr.c'};
def.HeaderFiles   = {'CrsCntrl_Wrapper.h', 'CruiseCntrlr.h'};
def.IncPaths = {[defaultDir, '¥files¥legacycode']};
def.SrcPaths = {[defaultDir, '¥files¥legacycode']};

%def.StartFcnSpec  = 'void sbr_initialize(void)';

def.OutputFcnSpec = ['void CrsCntrl(boolean_t u1, boolean_t u2, boolean_t u3, boolean_t u4, boolean_t u5, boolean_t u6,'...
                     'int16_t u7, uint8_t u8, uint8_t u9, int32_t u10, int32_t u11,'...
                     'uint8_t y1[1], boolean_t y2[1], uint8_t y3[1], int32_t y4[1], int32_t y5[1])'];
def.Options.supportCoverageAndDesignVerifier = true; %neccesary for code coverage analysis and test case generation
def.Options.isMacro = true;
% Generate the C-MEX S-function
legacy_code('sfcn_cmex_generate',def);
legacy_code('rtwmakecfg_generate', def);
```

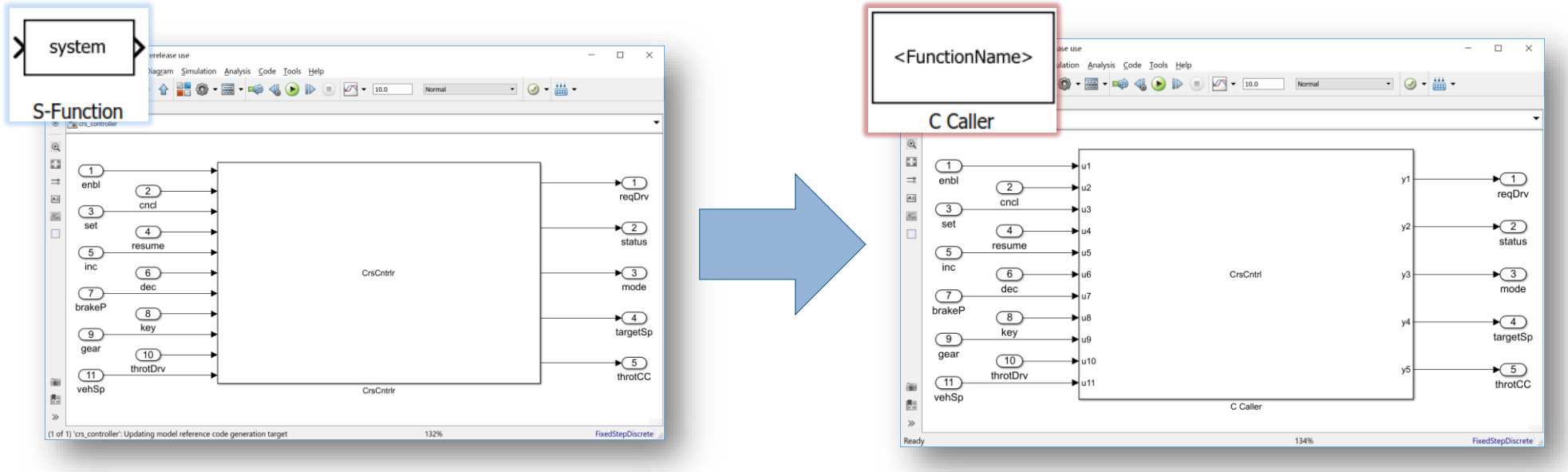
If any changes in code

Modeling



Introducing C Caller Block

C Caller Block makes it easier to call C Functions in Simulink
→ It works for simulation and Code Generation



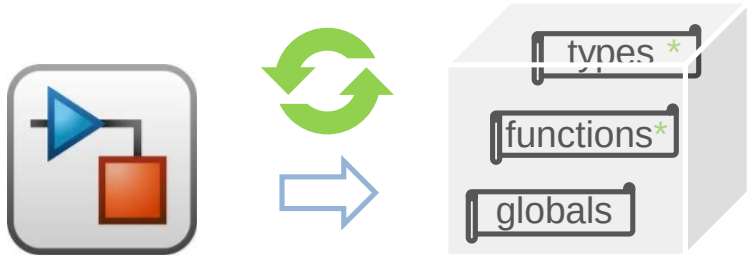
Key Features

- Automate the process

- Tedious
- Error prone
- Hard to maintain

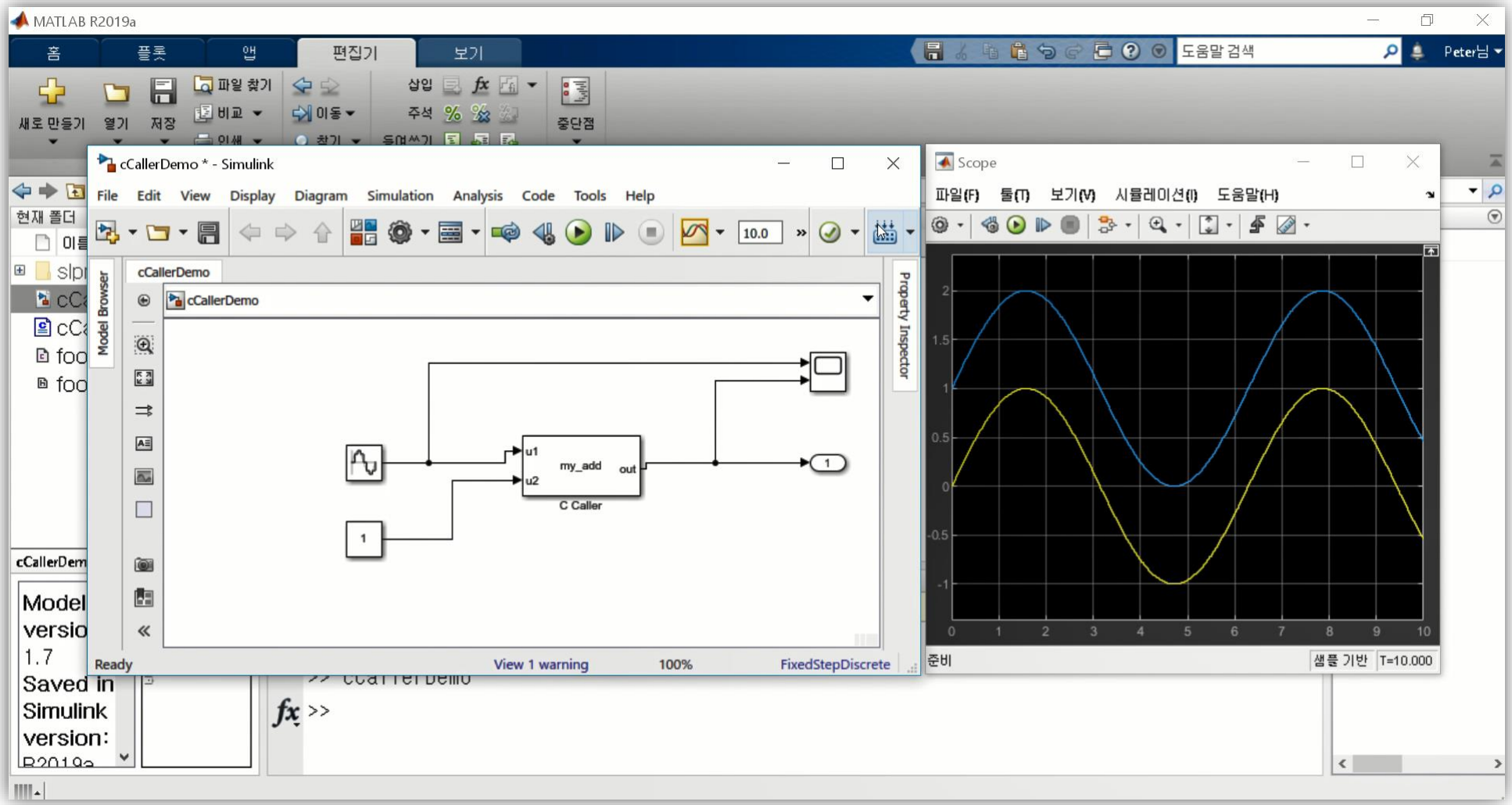
Automate

- Synchronize with custom code changes

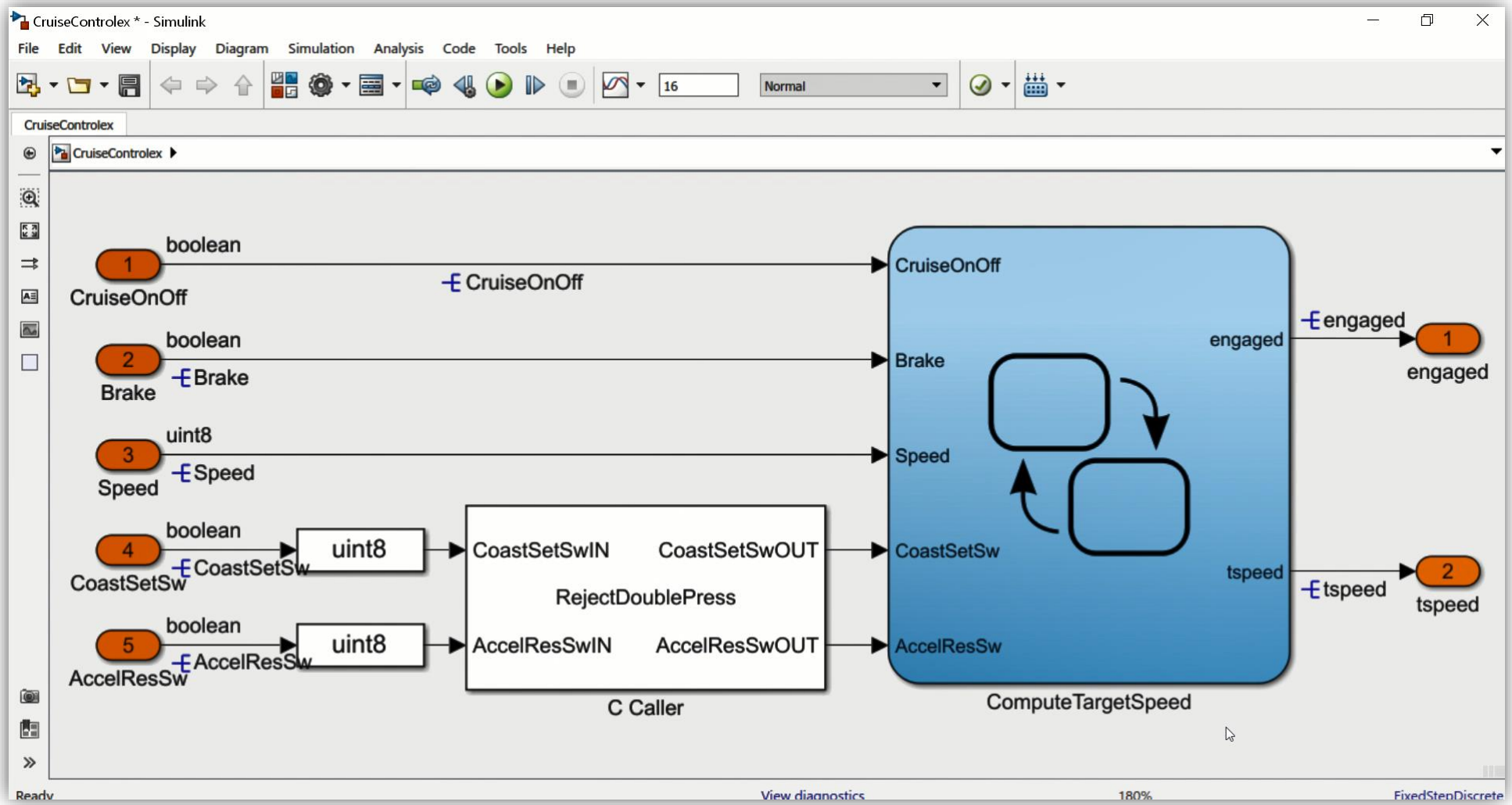


C/C++ Code

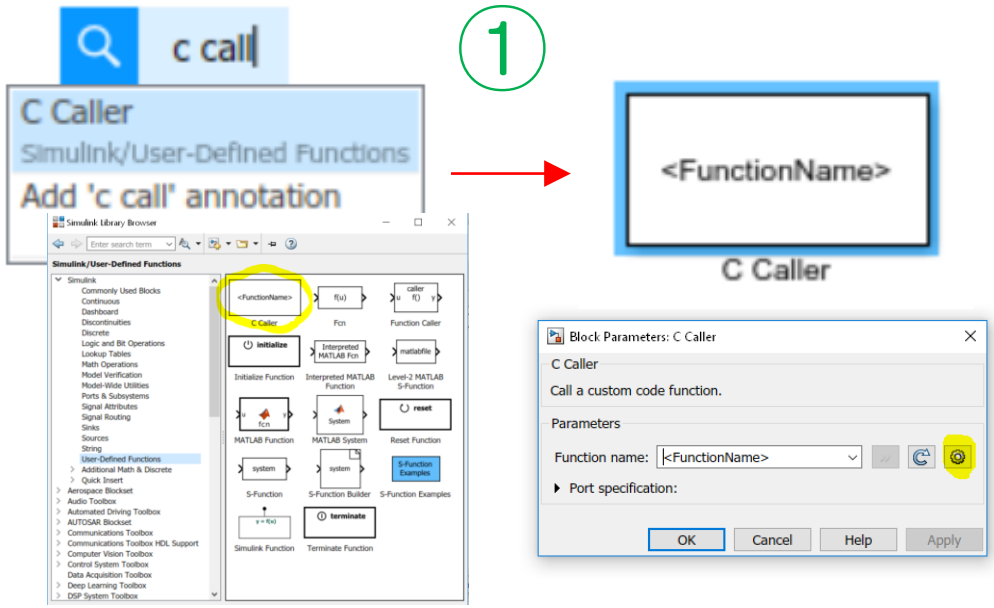
Demo: Simple C Caller

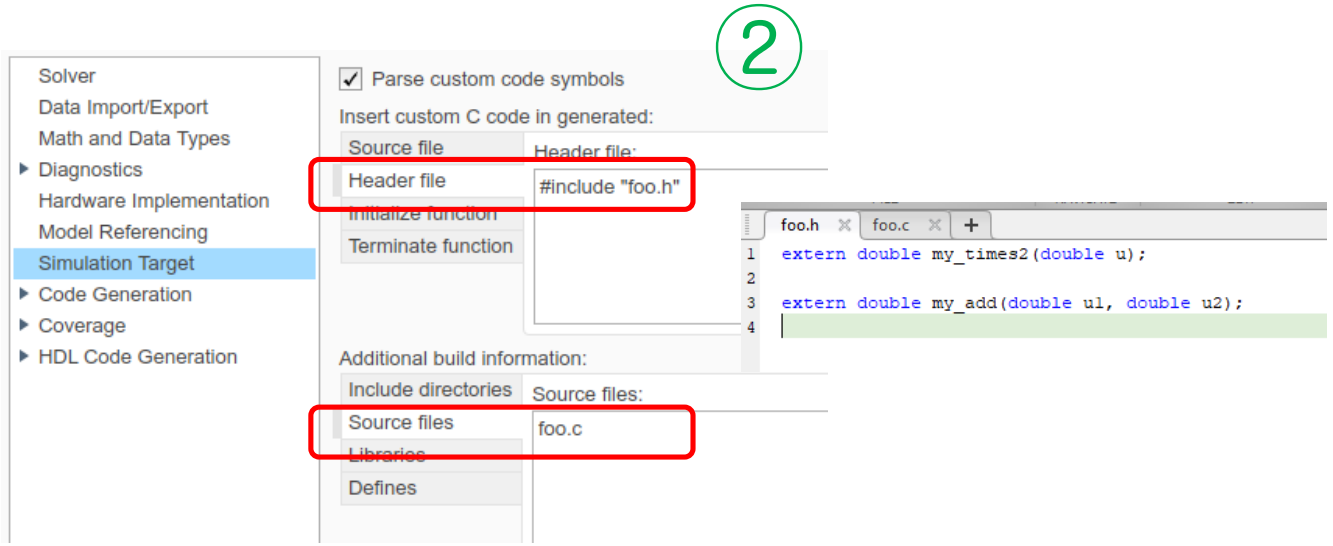


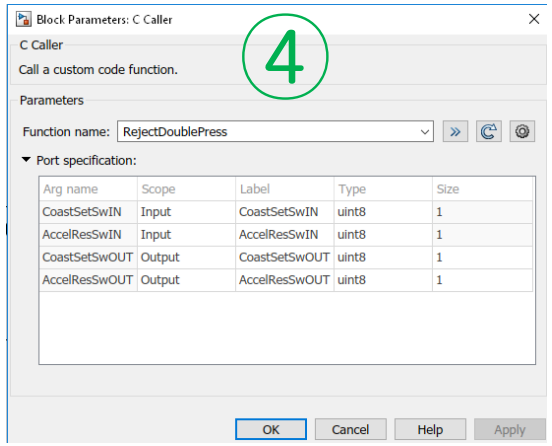
Demo: Integrate C code with Simulink Model

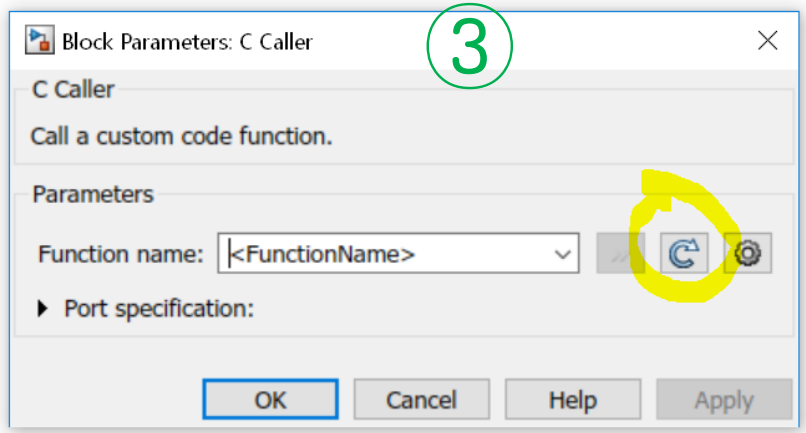


C Caller workflow





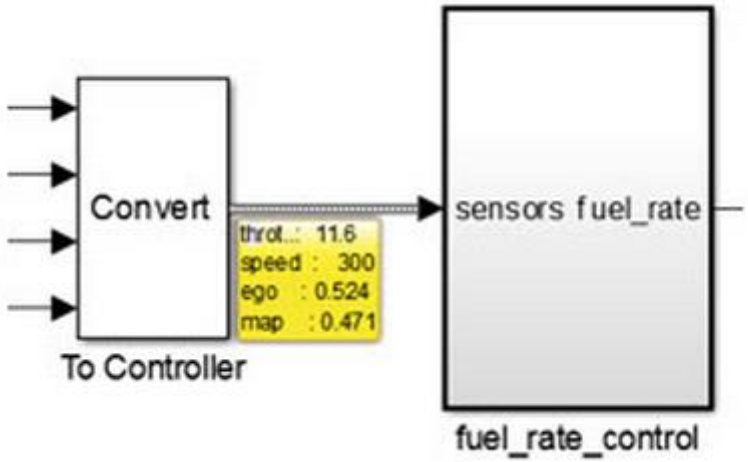


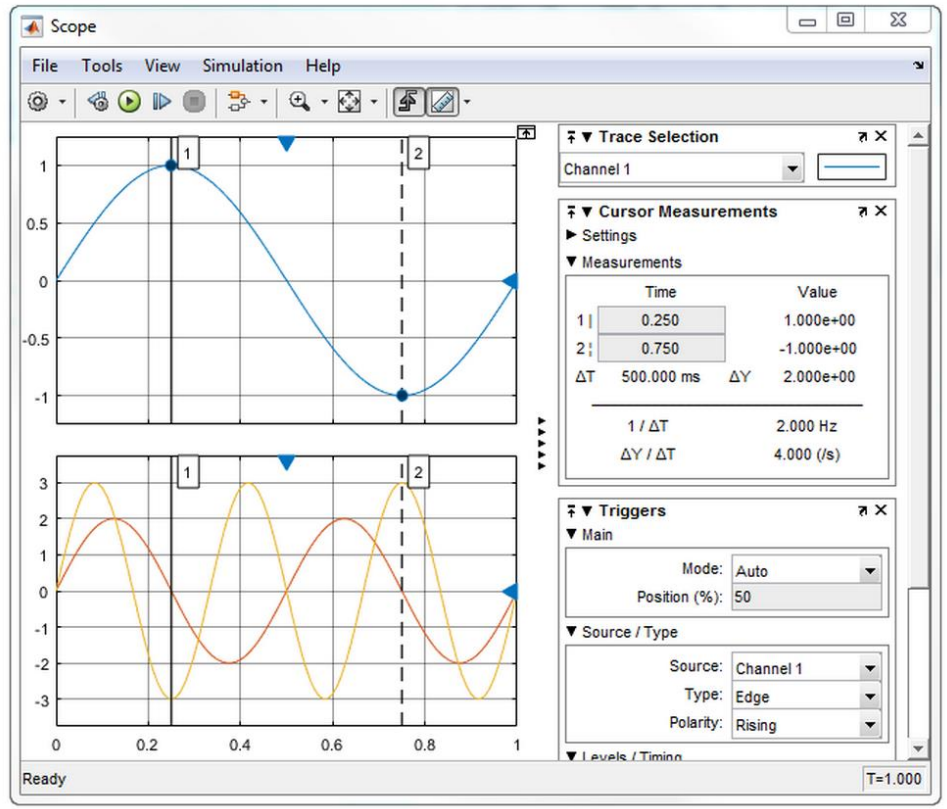


Agenda

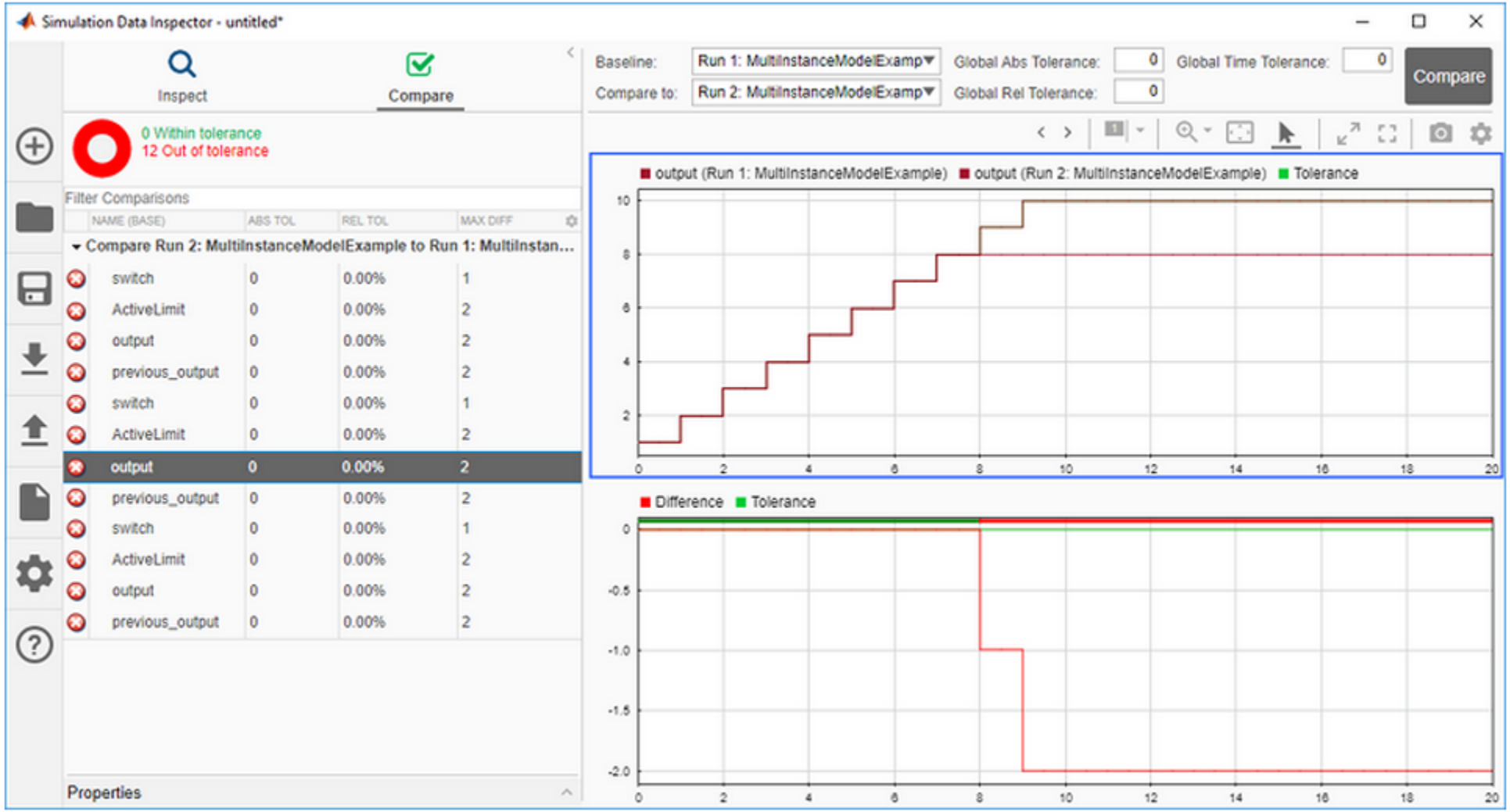
- Legacy code integration using Simulink
- **Visualization using Simulink**
- Verification with legacy code

Visualize Simulation Data



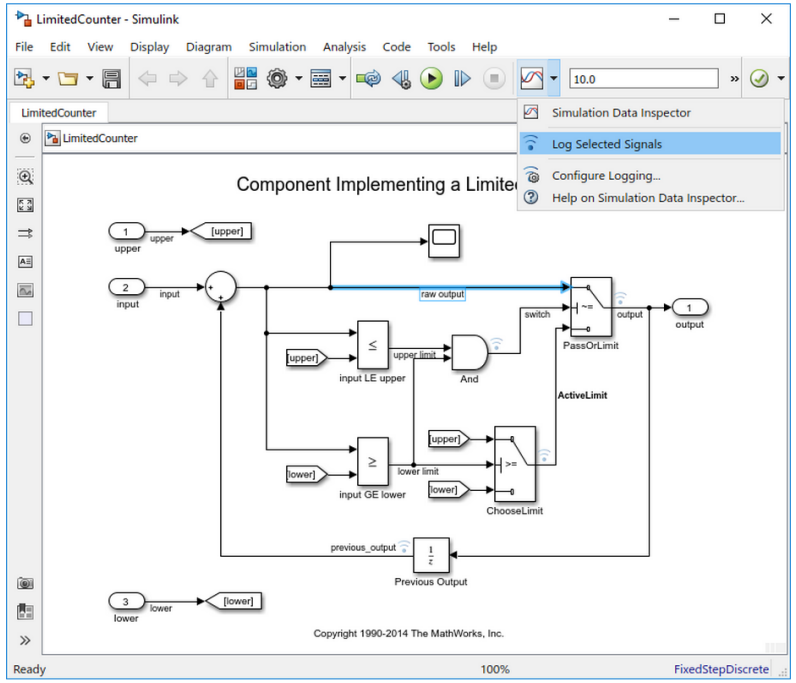


Visualize Simulation Data

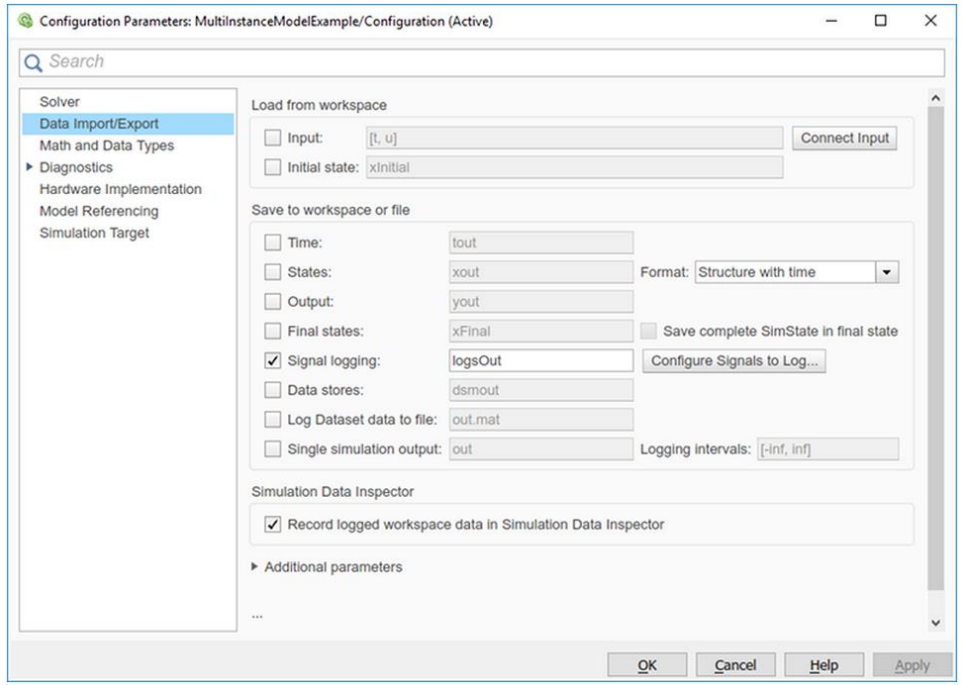


Visualize Simulation Data

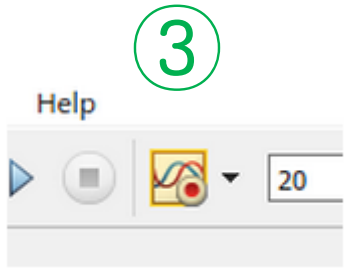
1



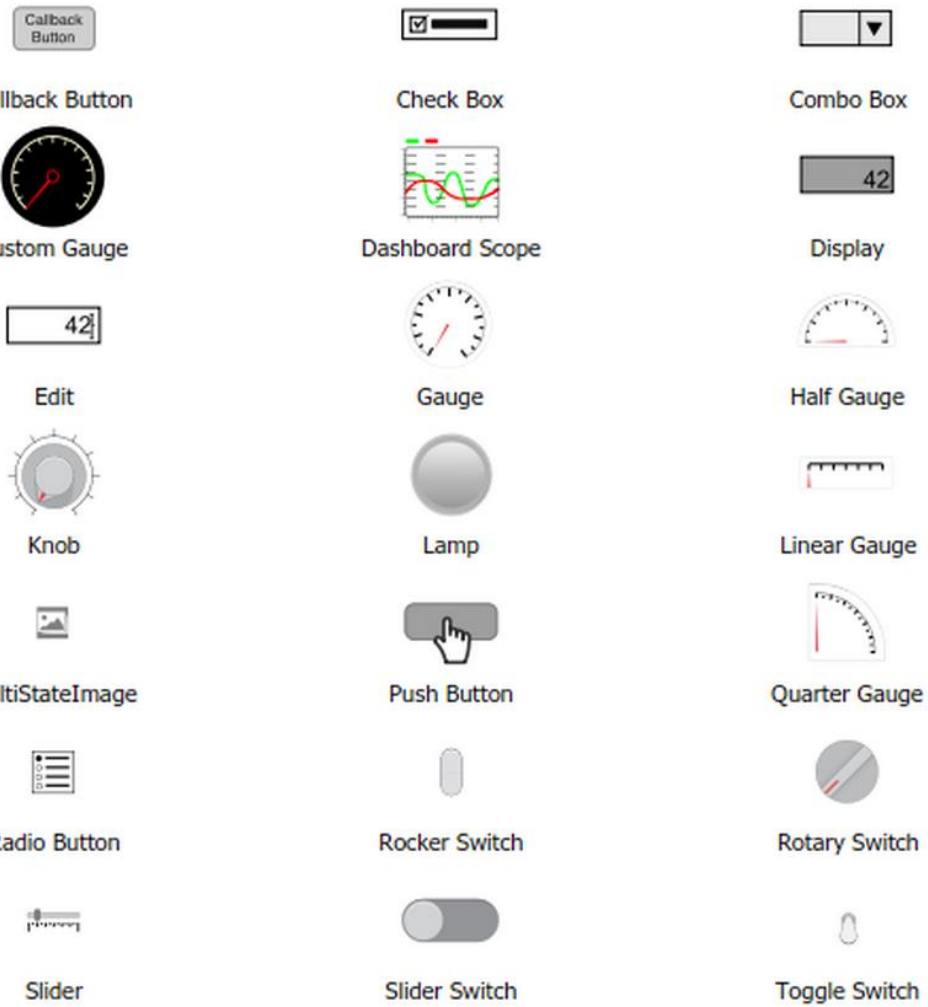
2



3



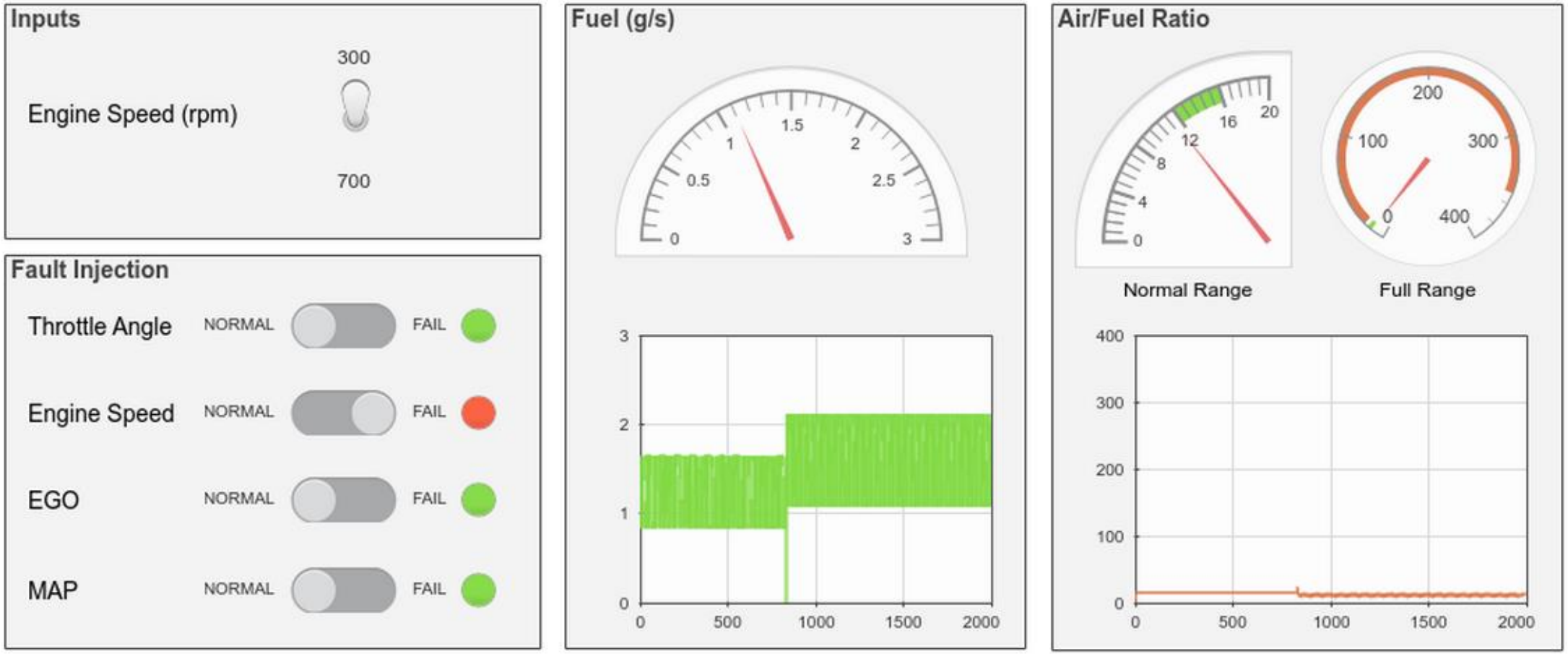
Tune and Visualize Your Model with Dashboard Blocks



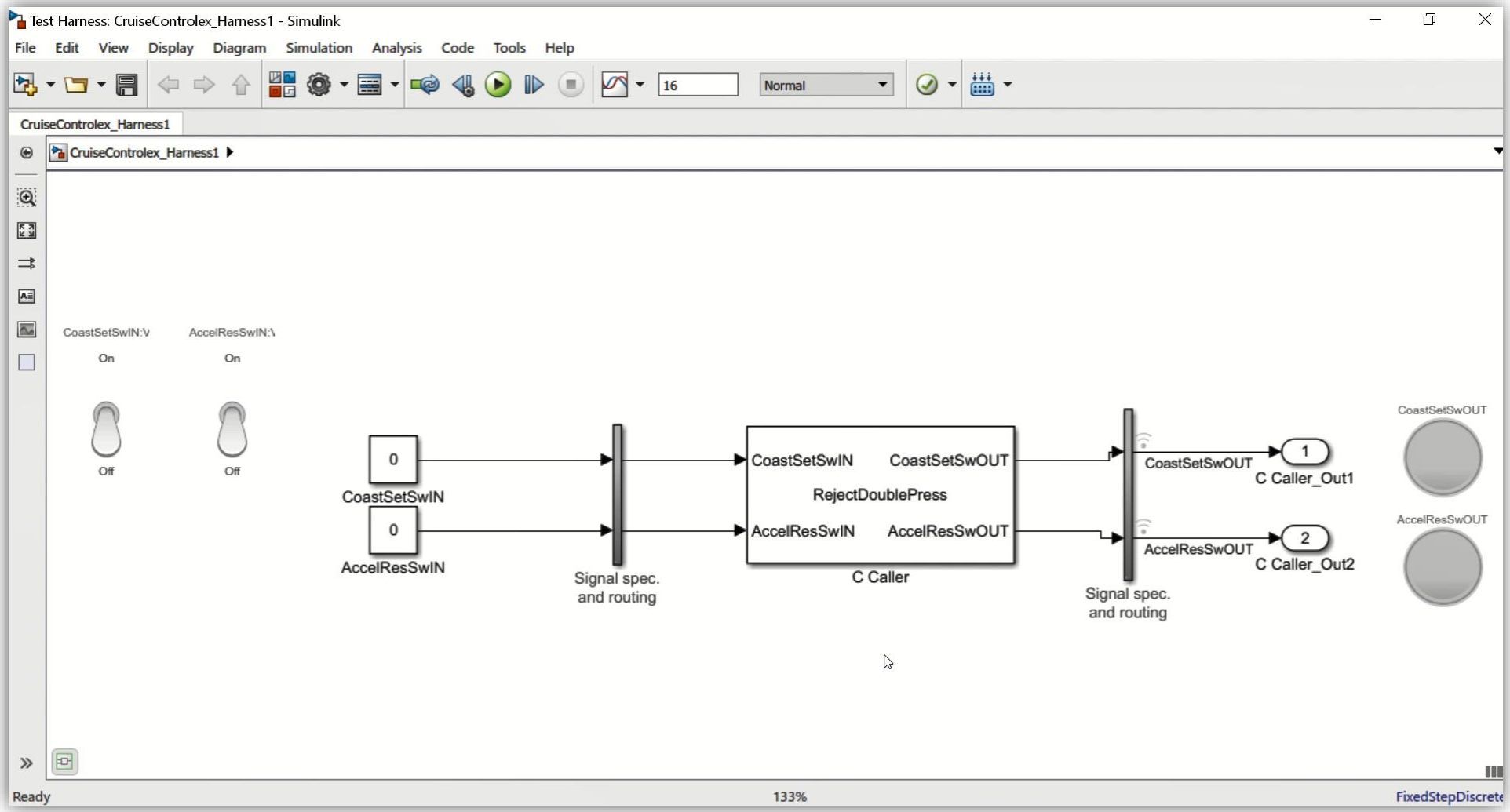
Custom Gage

Flight Instrument Gauges
- Aerospace Blockset -

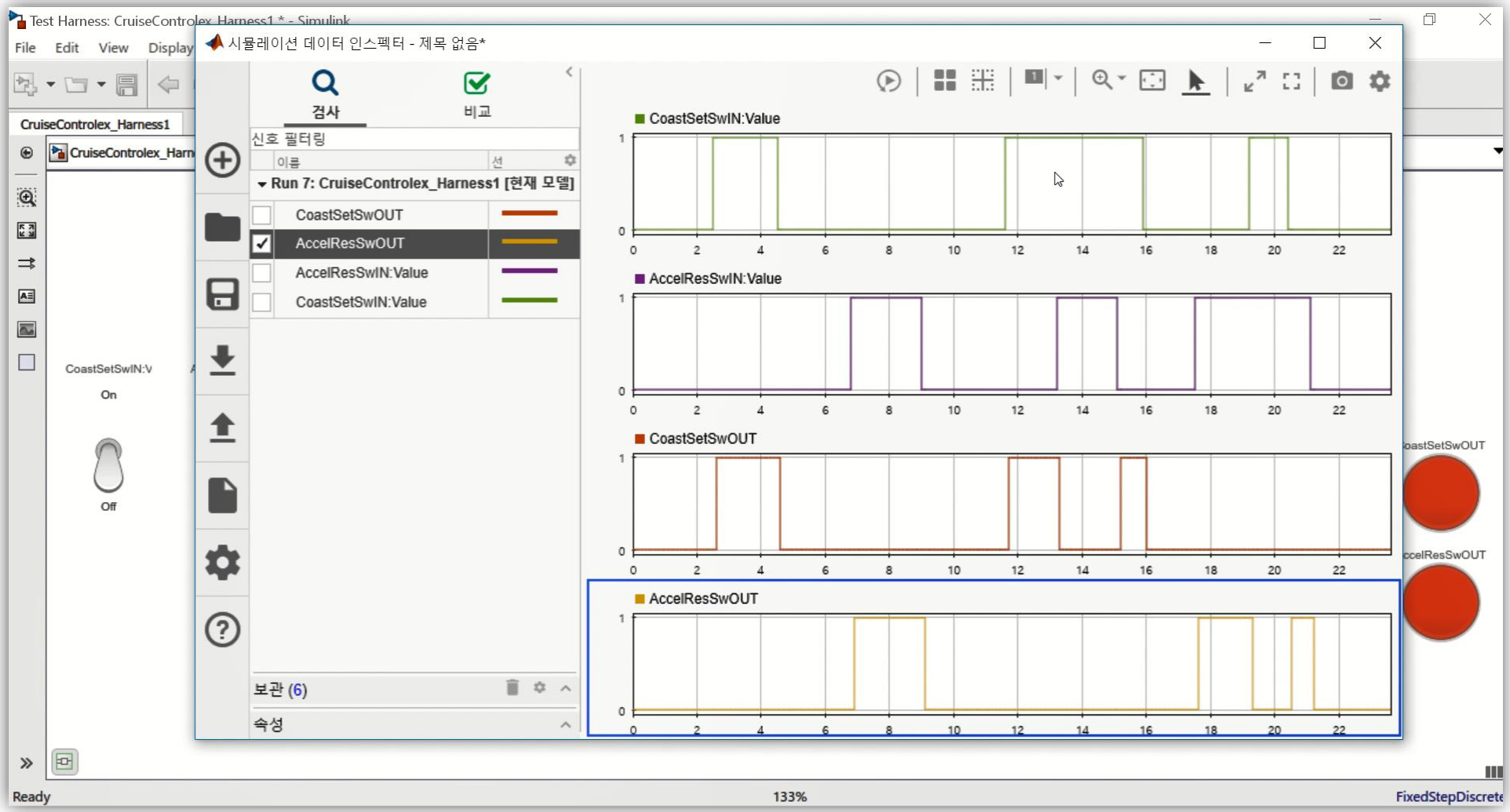
Tune and Visualize Your Model with Dashboard Blocks



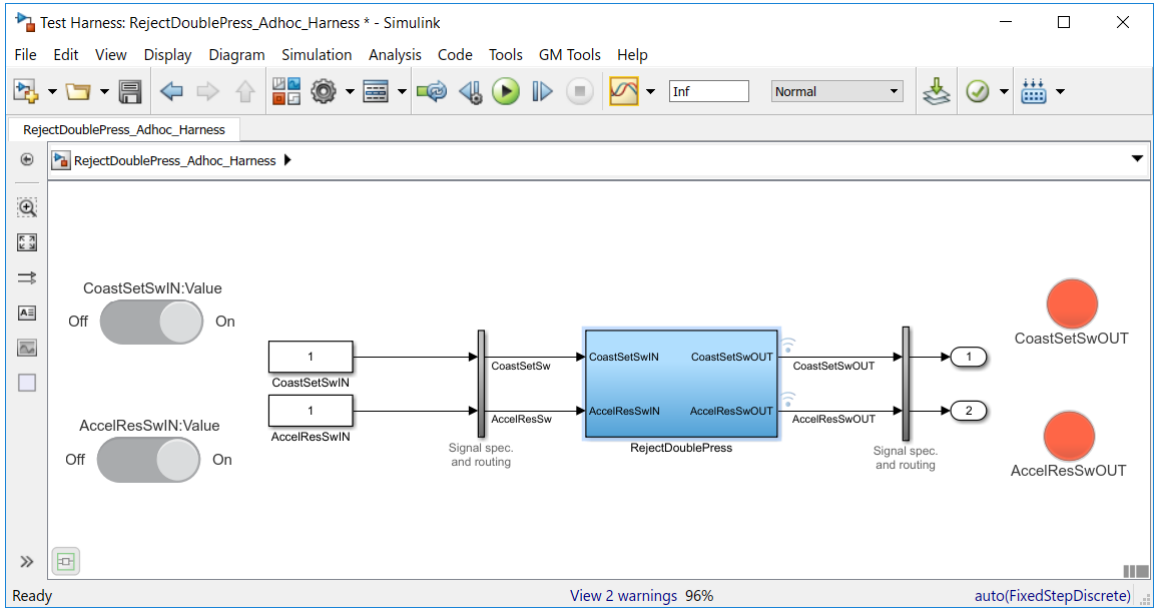
DEMO : Test harness creation with Dashboard

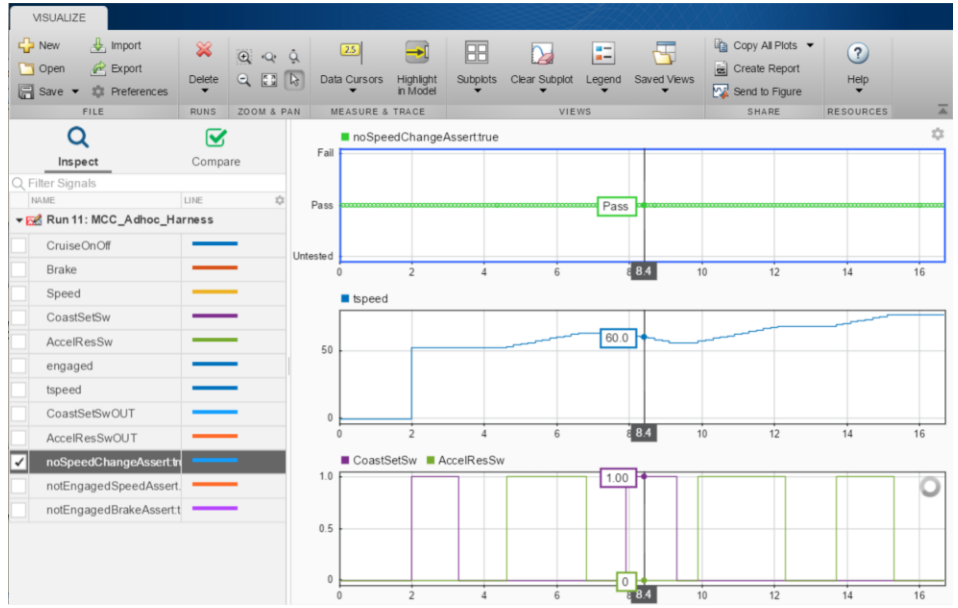


DEMO : Visualize Input and Output data



Visualization Summary

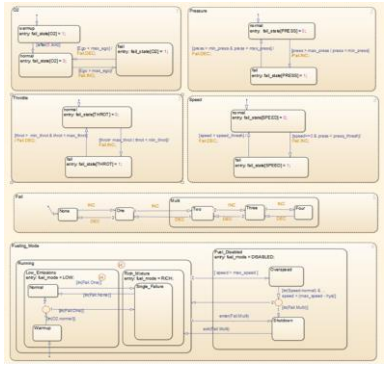




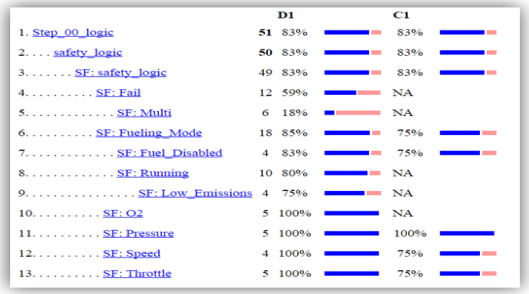
Agenda

- Legacy code integration using Simulink
- Visualization using Simulink
- **Verification with legacy code**

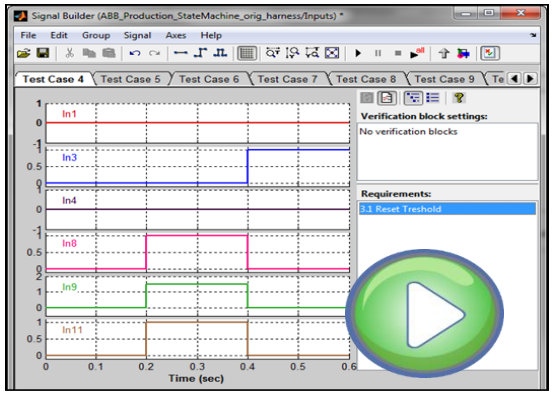
Test Case generation



Design Logic

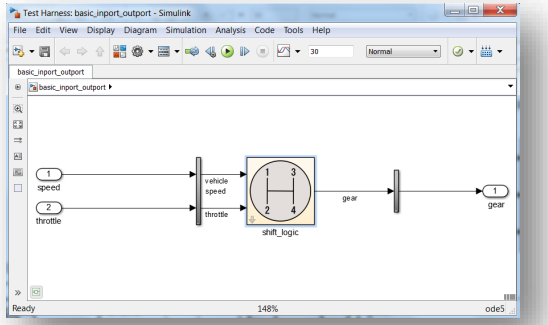


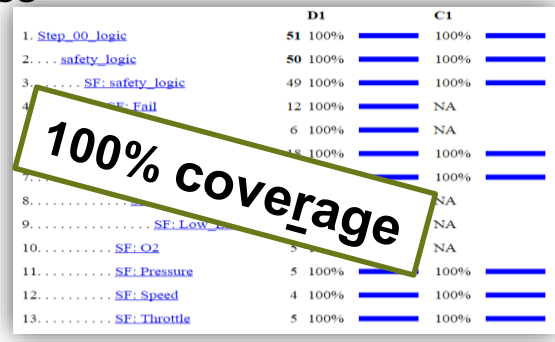
Existing test coverage



Test Cases

Test case generation





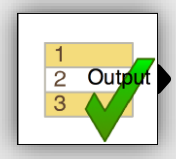
Systematic Functional Testing with Simulink Test

Test Case

Inputs

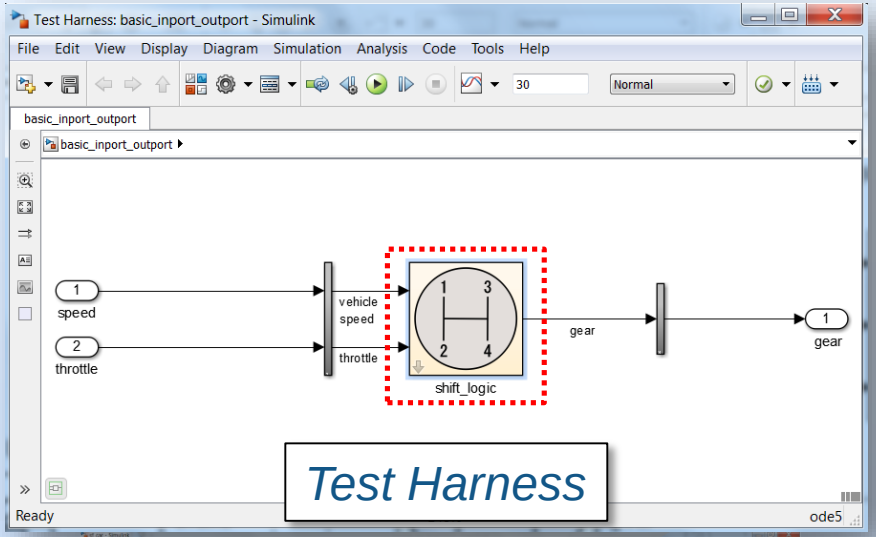
MAT file (input)

Signal Builder

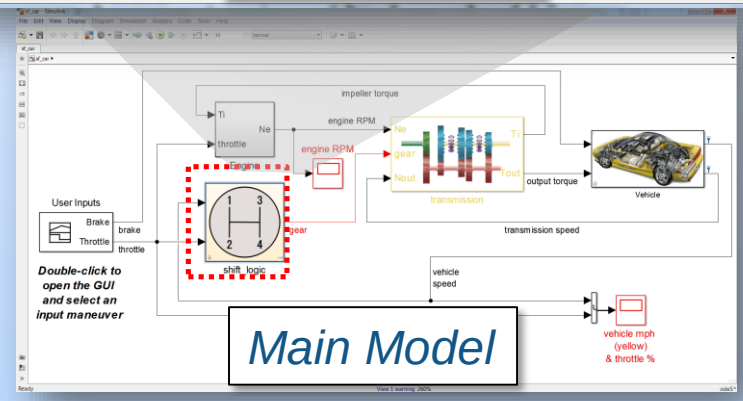


Test Sequence

and more!

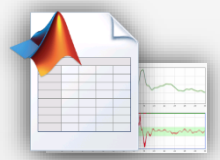


Test Harness

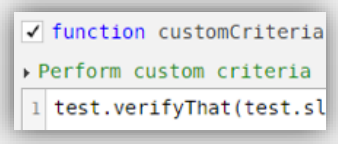


Main Model

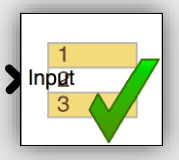
Assessments



MAT file (baseline)



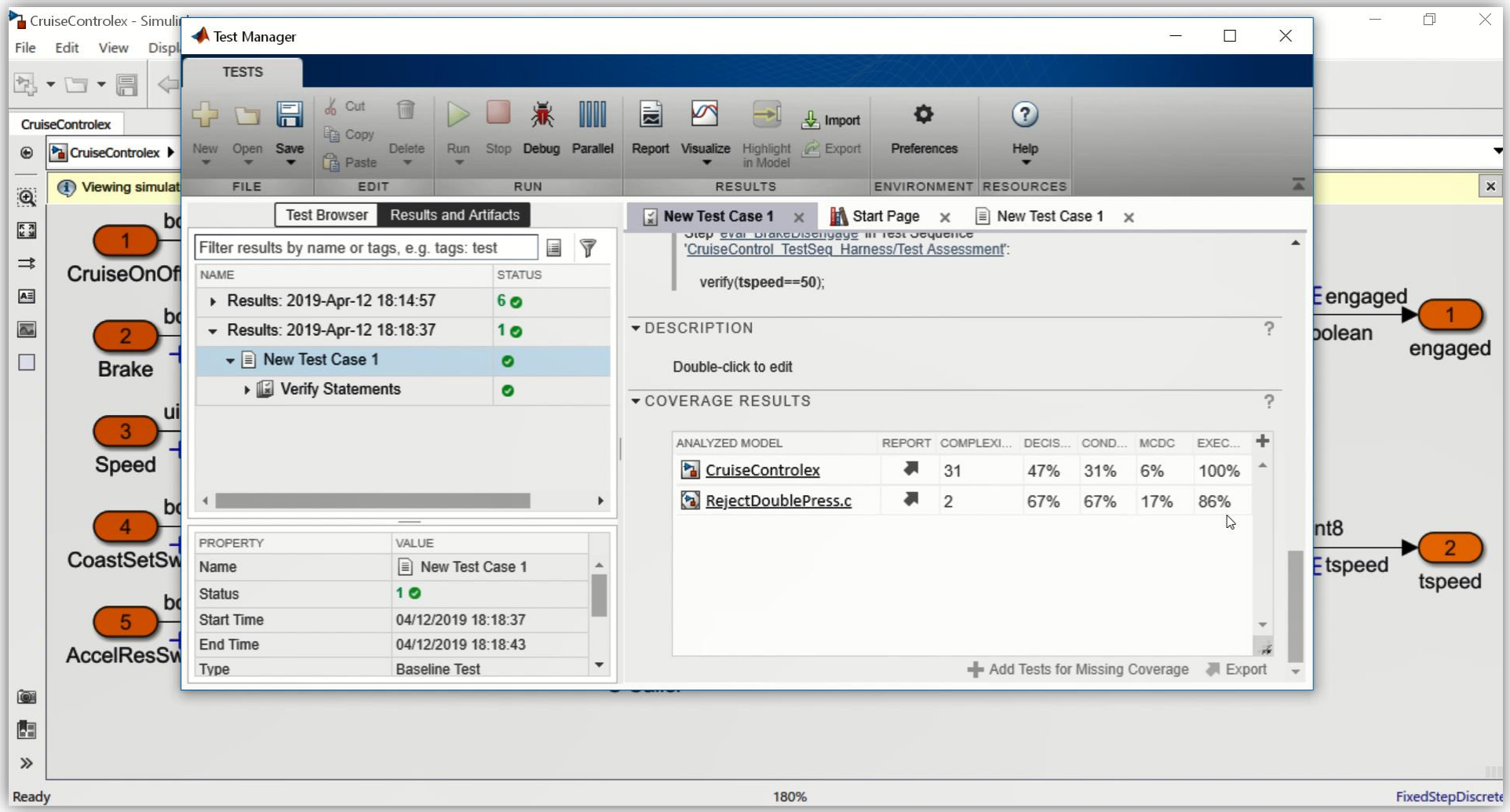
MATLAB Unit Test



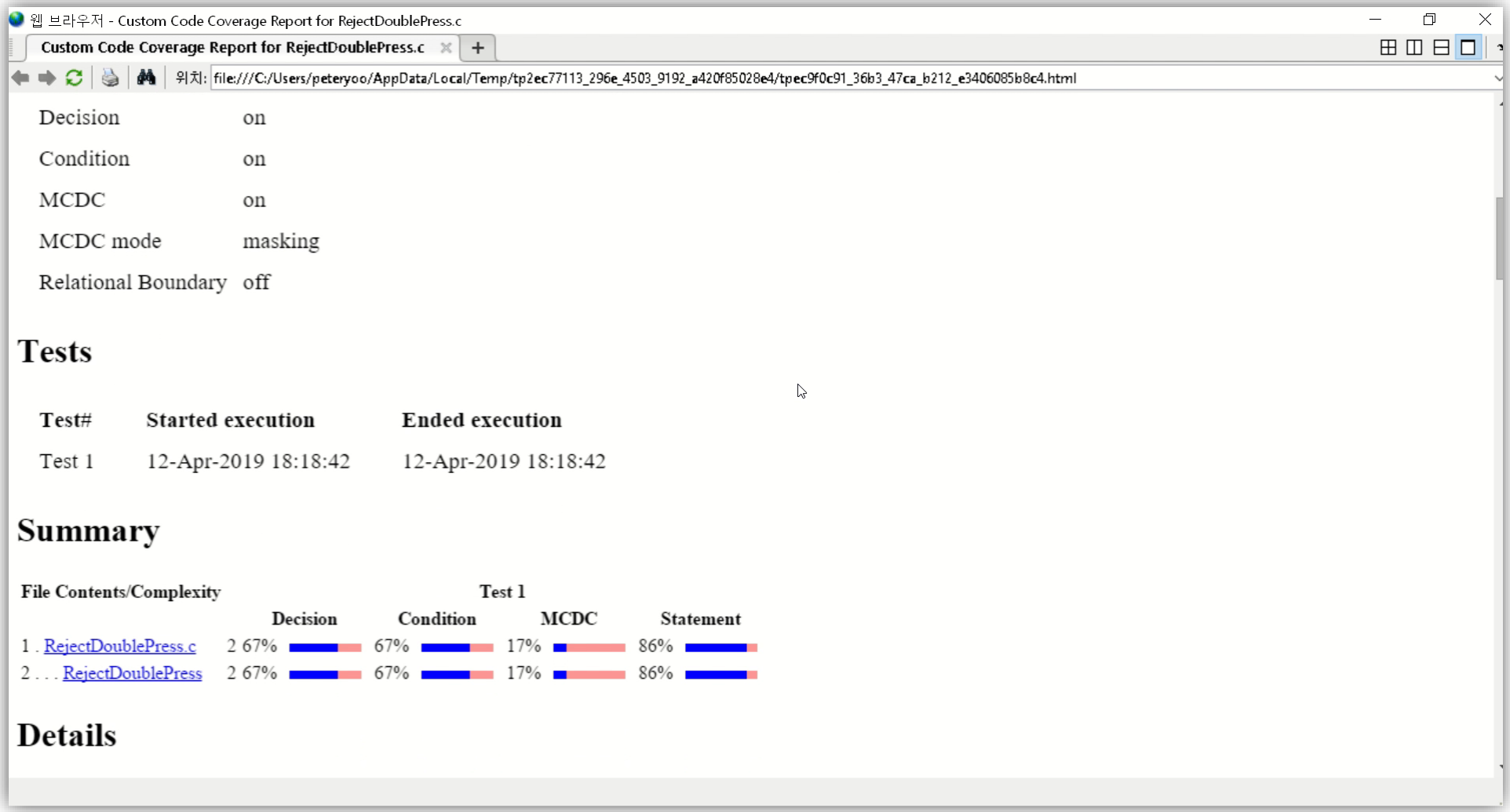
Test Assessment

and more!

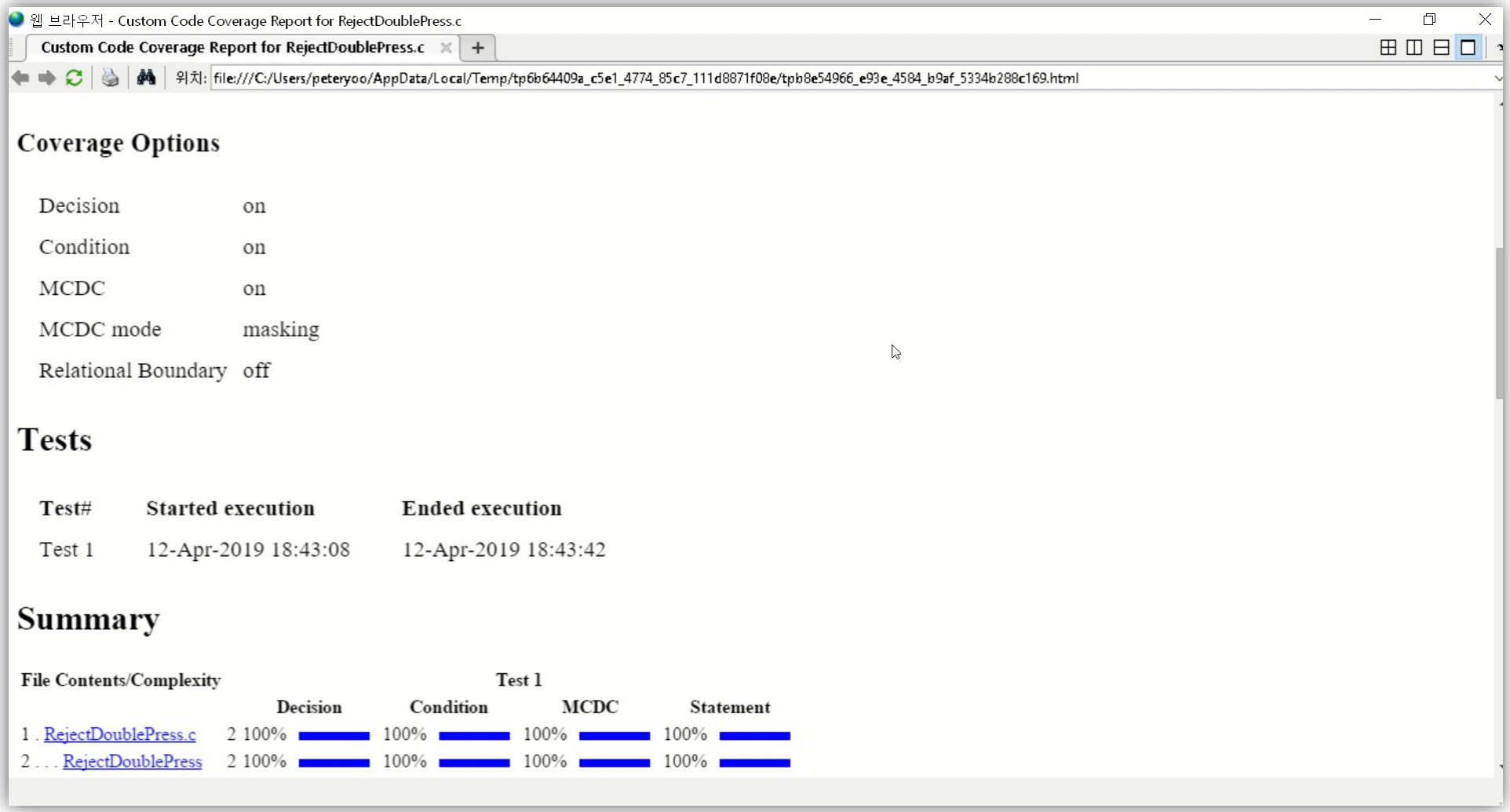
DEMO : Code Coverage with C caller



DEMO : Coverage Report



DEMO : Test Case Generation



Key takeaways

Bring C/C++ code into Simulink

- With full integration into Model-Based Design workflows
- To visualize simulation result
- Enable unified verification environment

MATLAB EXPO 2019

데모 부스와 상담부스로 질문 하시기 바랍니다.

감사합니다