

MATLAB/Simulink Training for Smart Model Car

류성연 차장

Senior Application Engineer

MathWorks Korea

Agenda

- 모델 기반 설계 개요
- 시뮬링크를 이용한 개발 환경 구축
- 자율 주행 개발 방안
- 모델링 및 코드 생성 실습

Headquarters

Natick, MA USA

North America

United States



Europe

France
Germany
Ireland
Italy
Netherlands
Spain
Sweden
Switzerland
UK

Asia-Pacific

Australia
China
India
Japan
Korea



3 million+
users

in more than 180
countries


4000+
staff

in 31 offices around
the world


\$900+
million

in 2017 revenues with
60% from outside the US

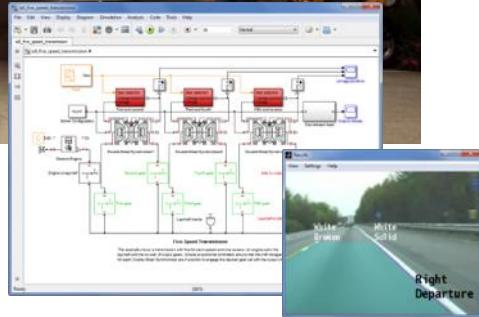
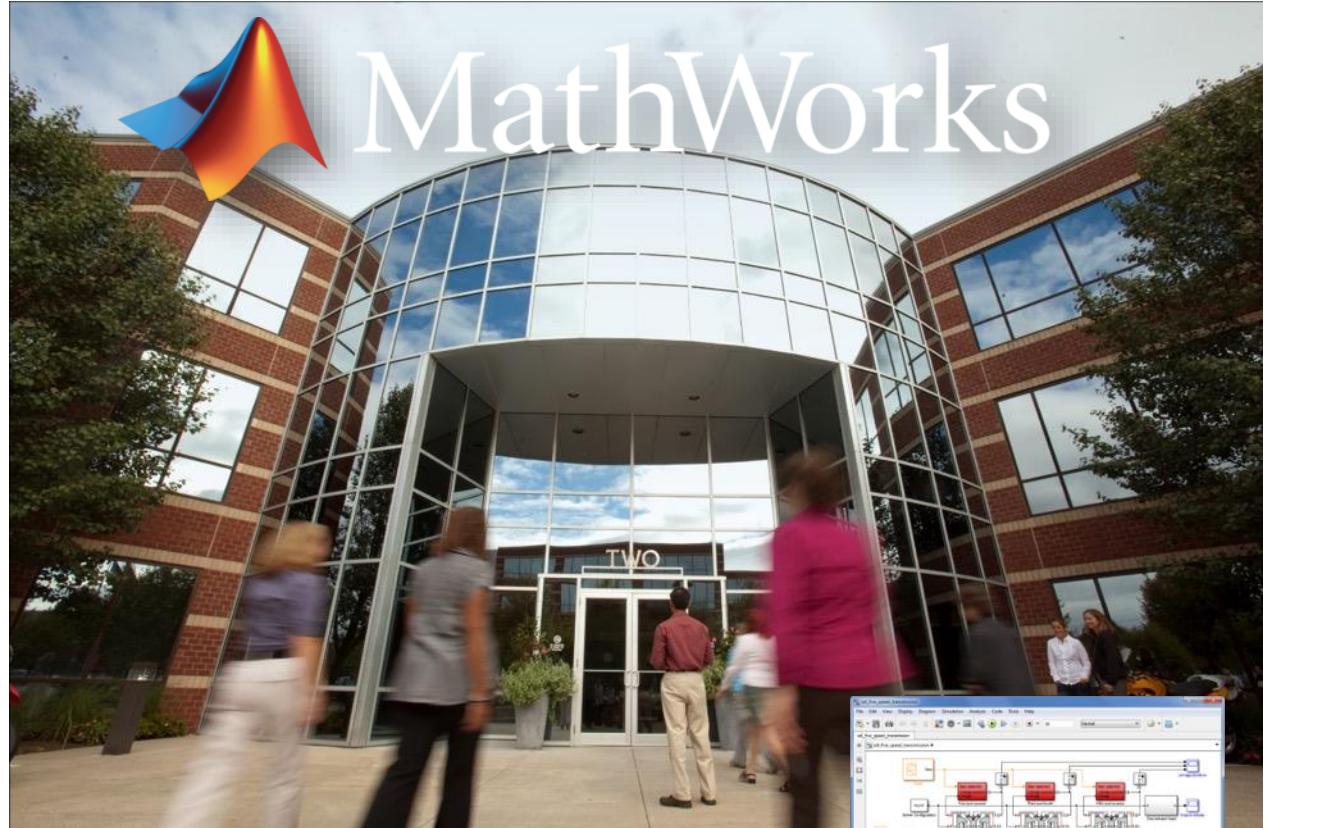


**Privately
held**

and profitable every year

MathWorks

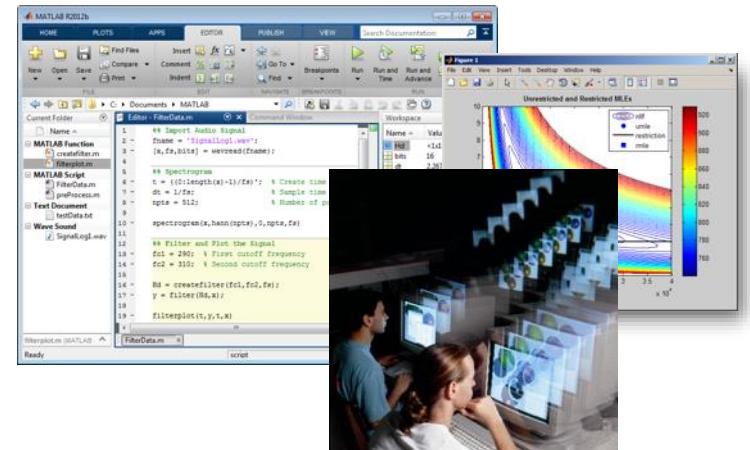
Accelerating the Pace of Engineering and Science



MATLAB®

The leading environment for technical computing

The industry-standard, high-level programming language for algorithm development, numeric computation, data analysis and visualization



SIMULINK®

The leading environment for modeling, simulating, and implementing dynamic and embedded systems

Applications in controls, signal processing, physical modeling, robotics, and other system engineering areas

Key Industries

- Aerospace and defense
- Automotive
- Biotech and pharmaceutical
- Communications
- Education
- Electronics and semiconductors
- Energy production
- Financial services
- Industrial automation and machinery
- Medical devices
- Software
- Internet



MathWorks Product Overview

<p>Event-Based Modeling</p> <ul style="list-style-type: none"> • Stateflow • SimEvents 	<p>Physical Modeling</p> <ul style="list-style-type: none"> • Simscape • Simscape Multibody • Simscape Driveline • Simscape Fluids • Simscape Electrical
<p>Real-Time Simulation and Testing</p> <ul style="list-style-type: none"> • Simulink Real-Time 	<p>Verification, Validation and Test</p> <ul style="list-style-type: none"> • Simulink Requirements • Simulink Check • Simulink Coverage • Simulink Design Verifier • Simulink Test • Polyspace
	<p>Simulation Graphics and Reporting</p> <ul style="list-style-type: none"> • Simulink 3D animation • Simulink Report Generator

<p>Parallel Computing</p> <ul style="list-style-type: none"> • Parallel Computing Toolbox 	<p>SIMULINK®</p> <p>Simulation and Model-Based Design</p> <p>MATLAB®</p> <p>The Language of Technical Computing</p> <p>Code Generation</p> <ul style="list-style-type: none"> • MATLAB Coder • Simulink Coder • Embedded Coder • Fixed-Point Designer • GPU Coder • HDL Coder • PLC Coder
<p>Math, Statistics, and Optimization</p> <ul style="list-style-type: none"> • Statistics and Machine Learning Toolbox • Curve Fitting Toolbox • Deep Learning Toolbox 	<p>Application Deployment</p> <ul style="list-style-type: none"> • MATLAB Compiler <p>Database Access and Reporting</p> <ul style="list-style-type: none"> • Database Toolbox • MATLAB Report Generator

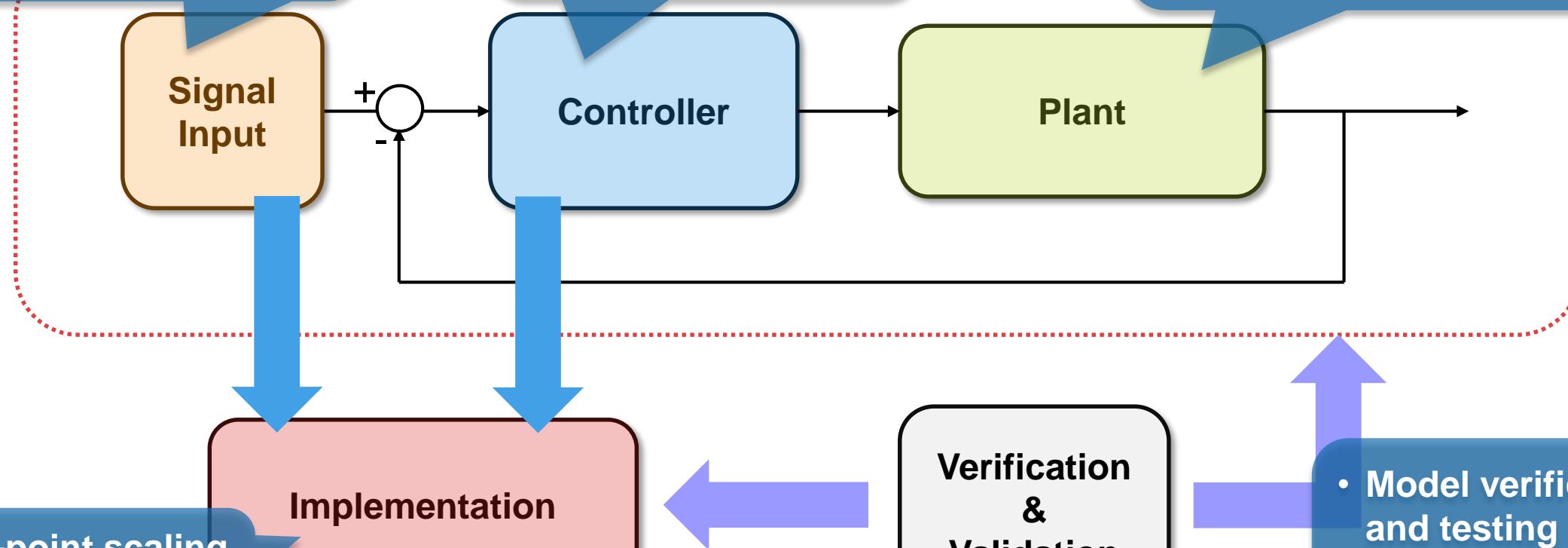
<p>Applications</p>
<p>Control Systems</p> <ul style="list-style-type: none"> • Control System Toolbox • Robotics System Toolbox • Powertrain Blockset • Vehicle Dynamics Blockset
<p>Signal Processing and Communications</p> <ul style="list-style-type: none"> • Computer Vision System Toolbox
<p>Image Processing and Computer Vision</p> <ul style="list-style-type: none"> • Image Processing Toolbox • Automated Driving System Toolbox
<p>Test and Measurement</p> <ul style="list-style-type: none"> • Instrumentation Control Toolbox • Vehicle Network Toolbox
<p>Computational Finance</p>
<p>Computational Biology</p>

MATLAB/Simulink for Control System Design

- Data acquisition
- Data analytics
- Image processing
- Computer vision

- Control analysis
- Control design
- Design Optimization
- Control logic

- Environmental modeling
- System identification
- Physical modeling
- Parameter estimation

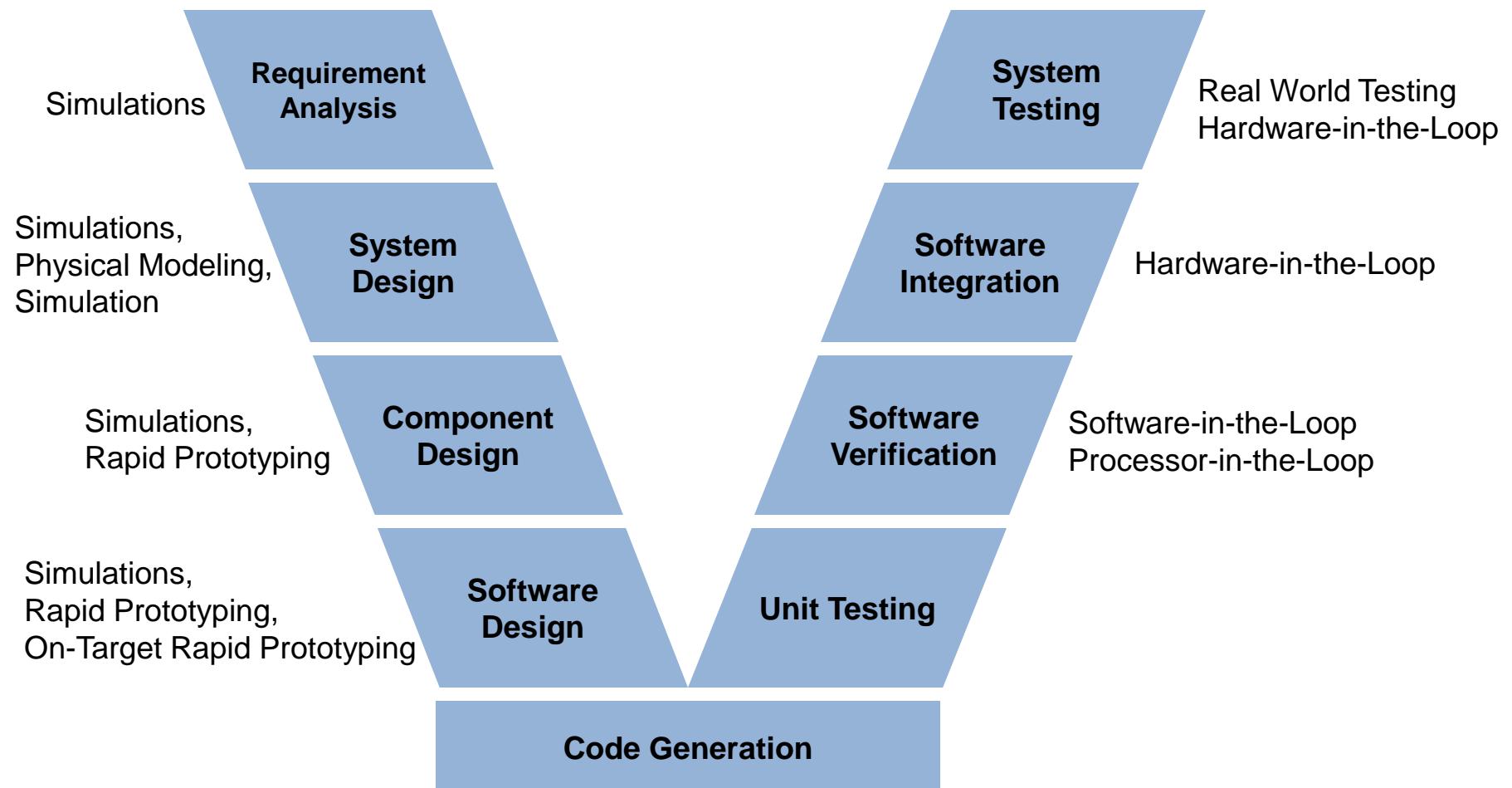


- Fixed-point scaling
- Automatic code generation

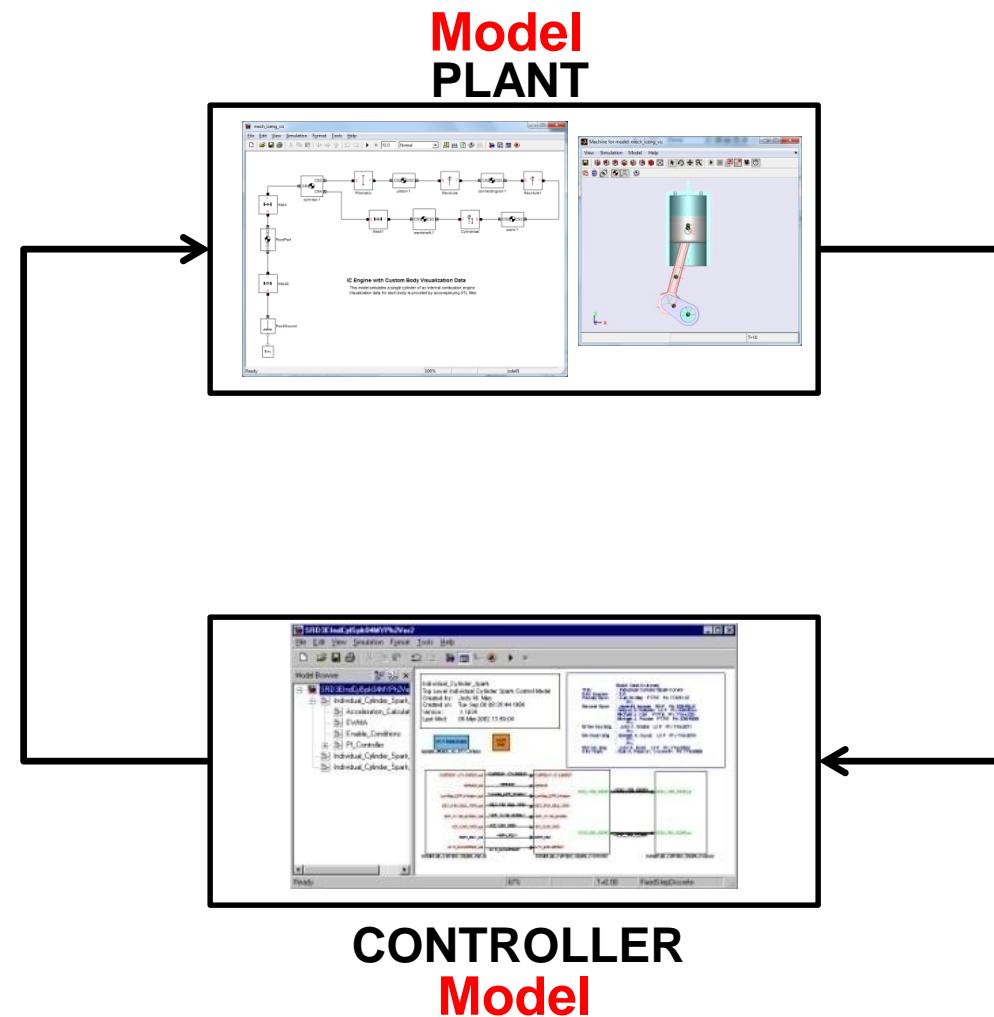
- Model verification and testing
- Real-time testing
- Code verification

Model-Based Design Overview

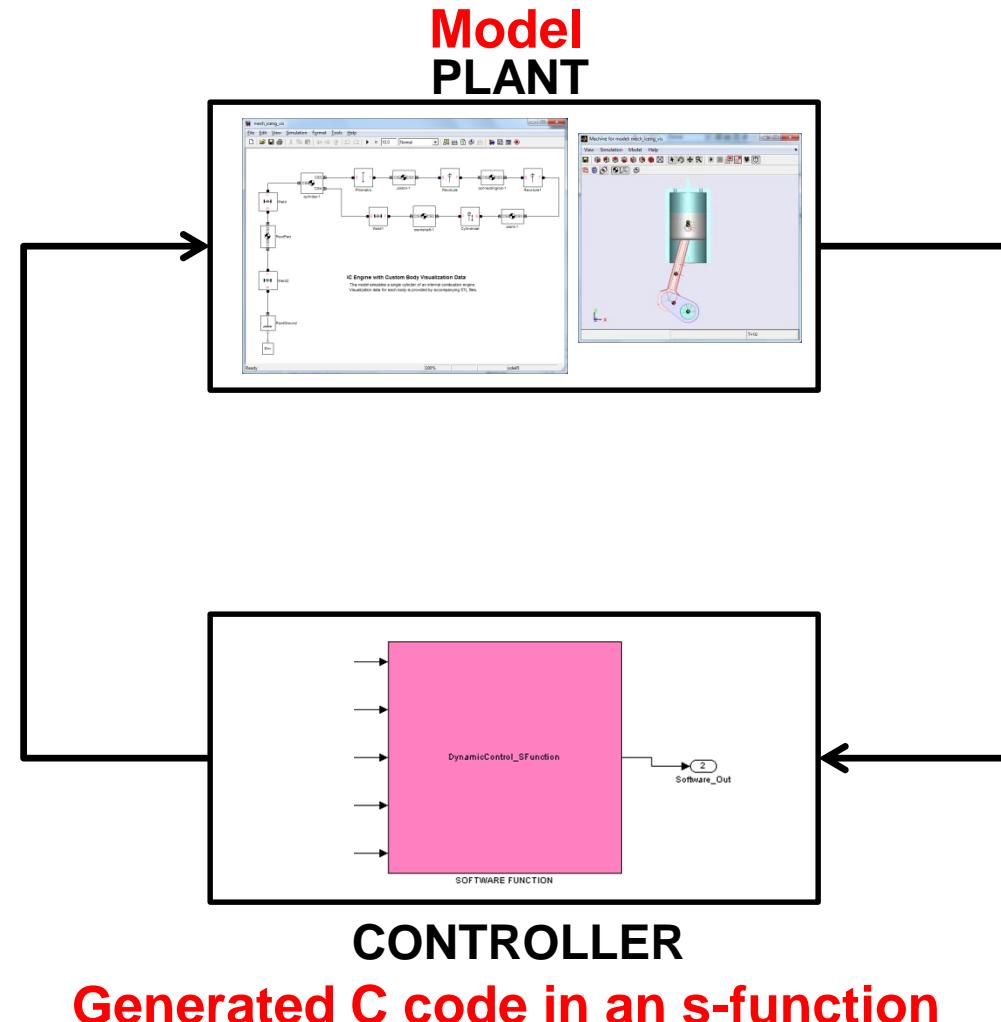
Model Based Design Process



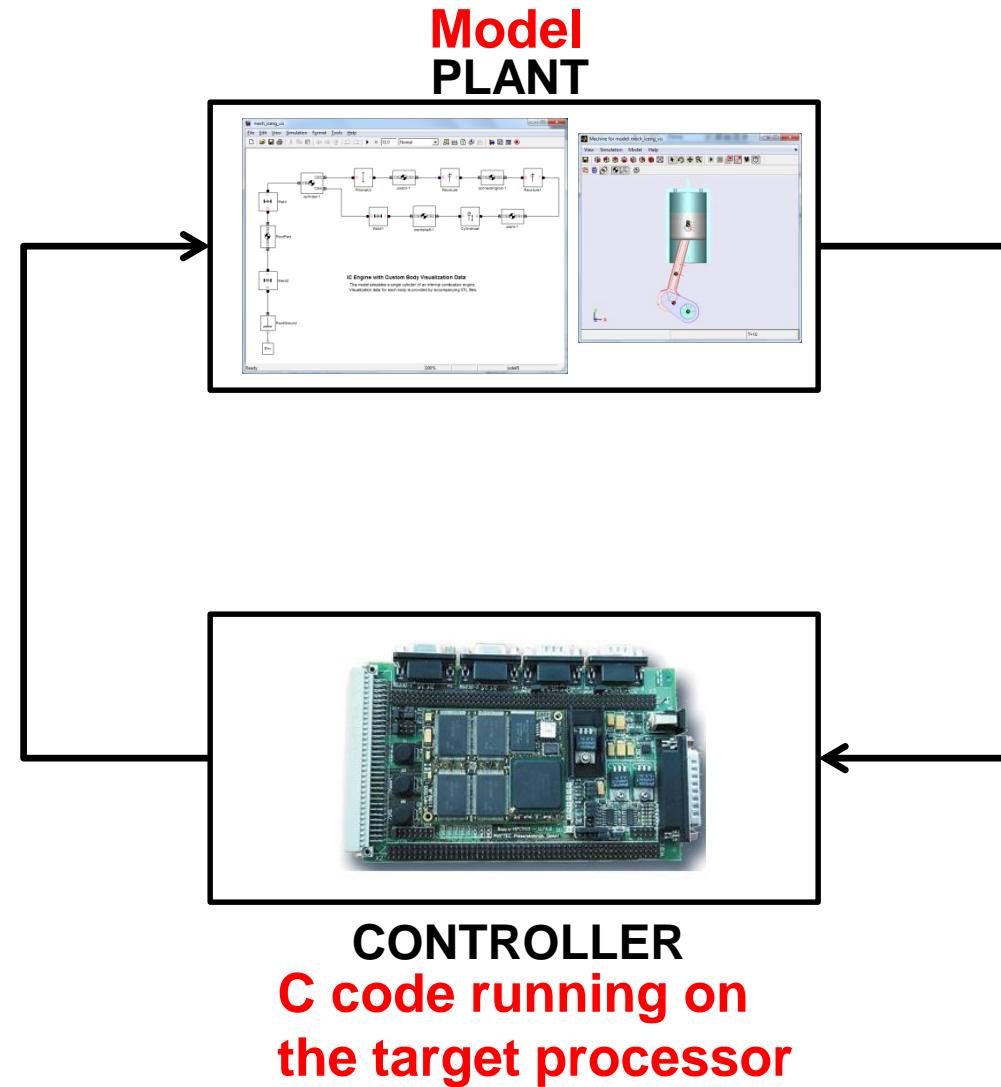
Model Simulation



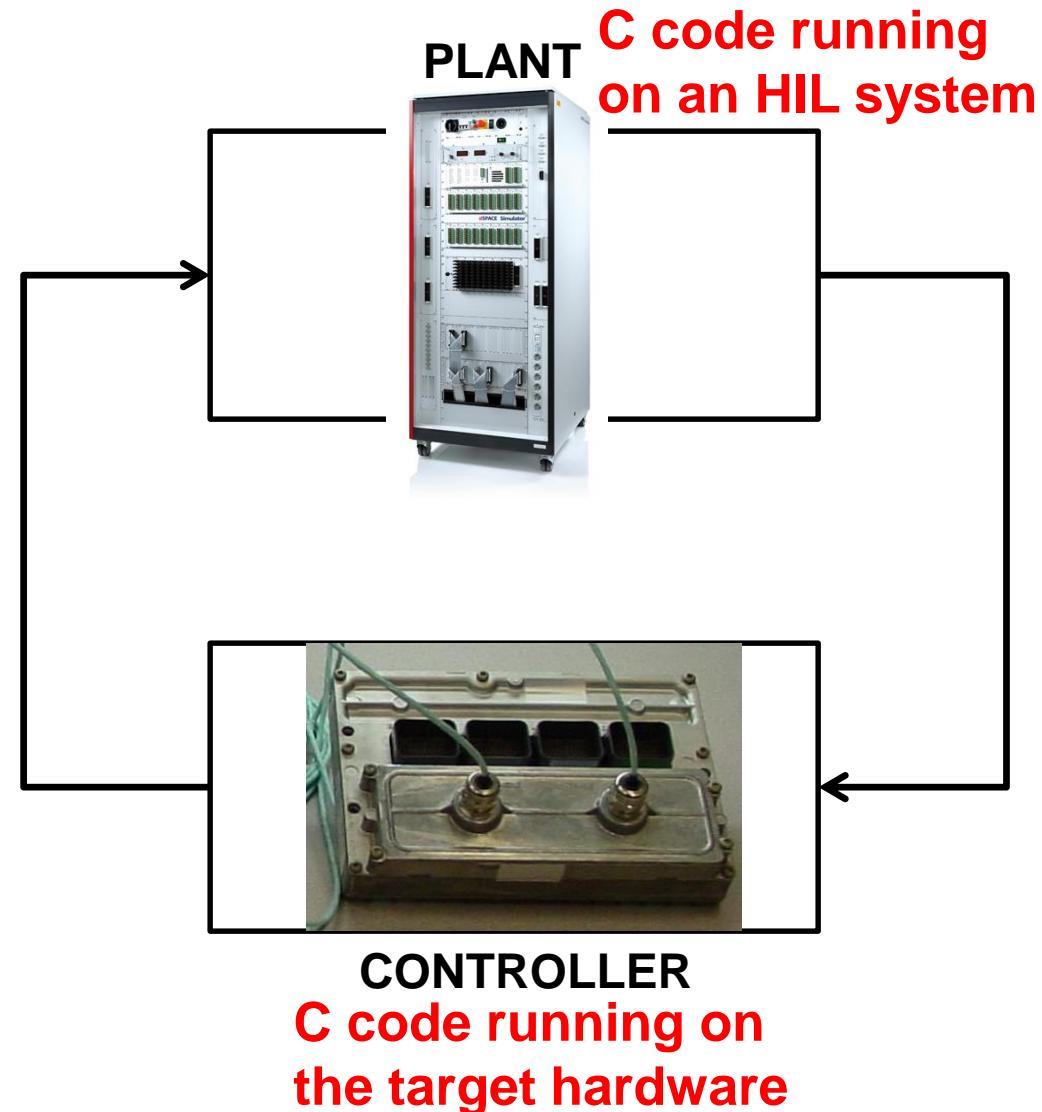
Software-in-the-Loop (SIL)



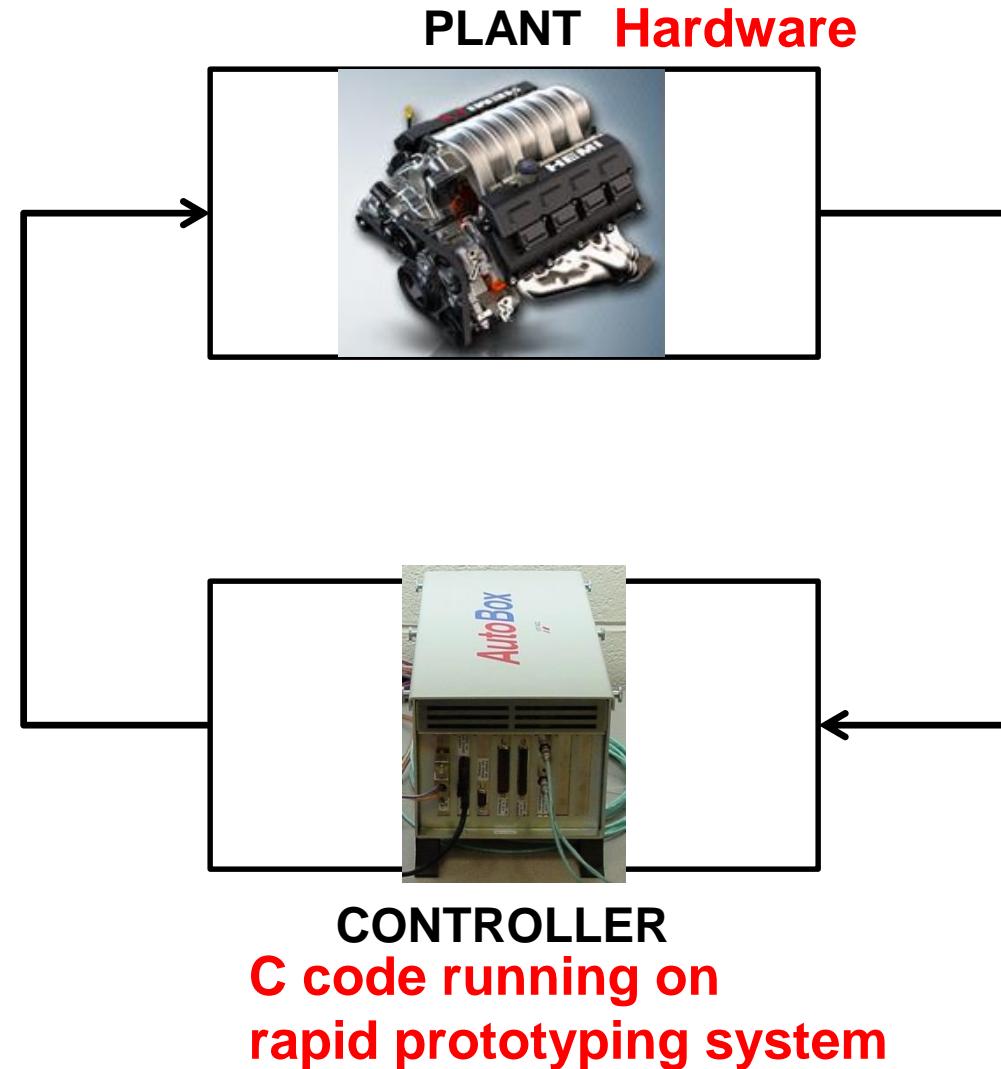
Processor-in-the-Loop (PIL)



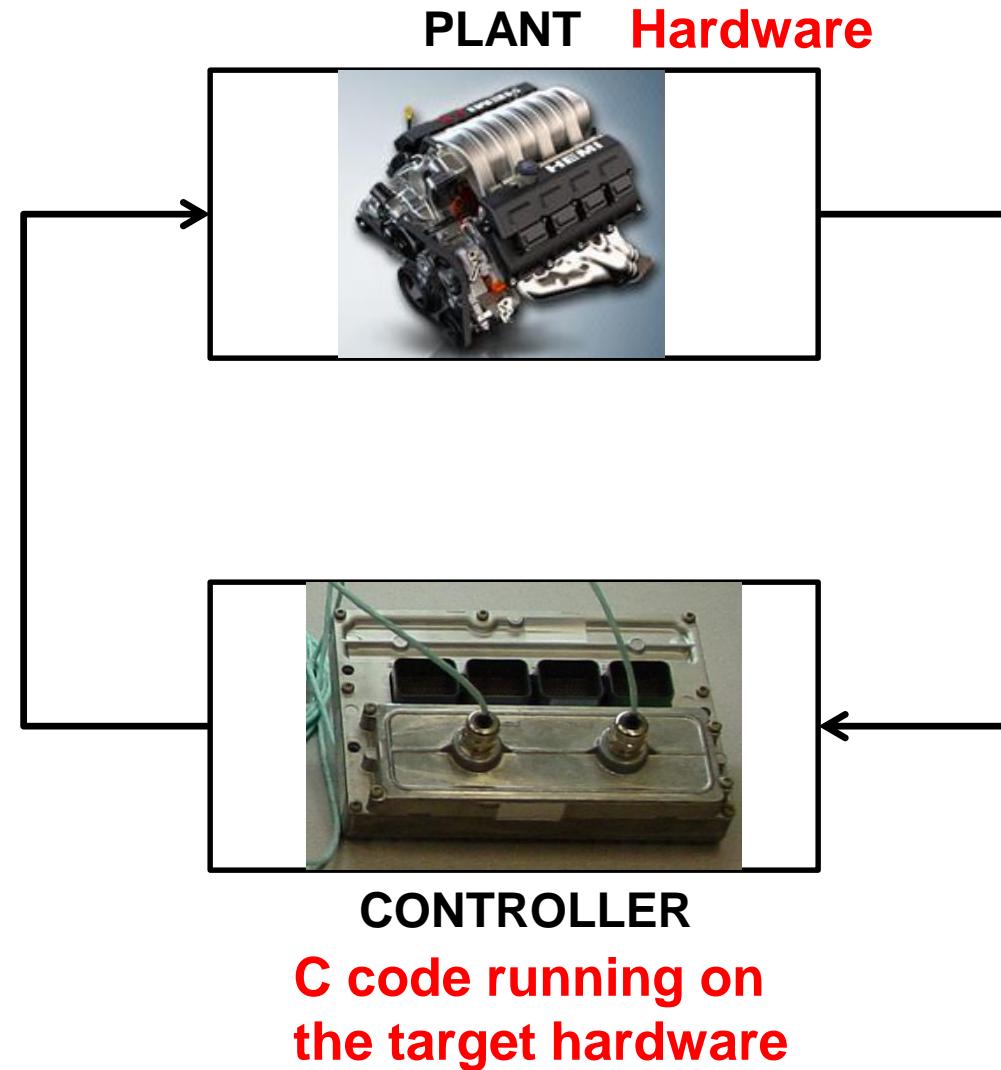
Hardware-in-the-Loop (HIL)



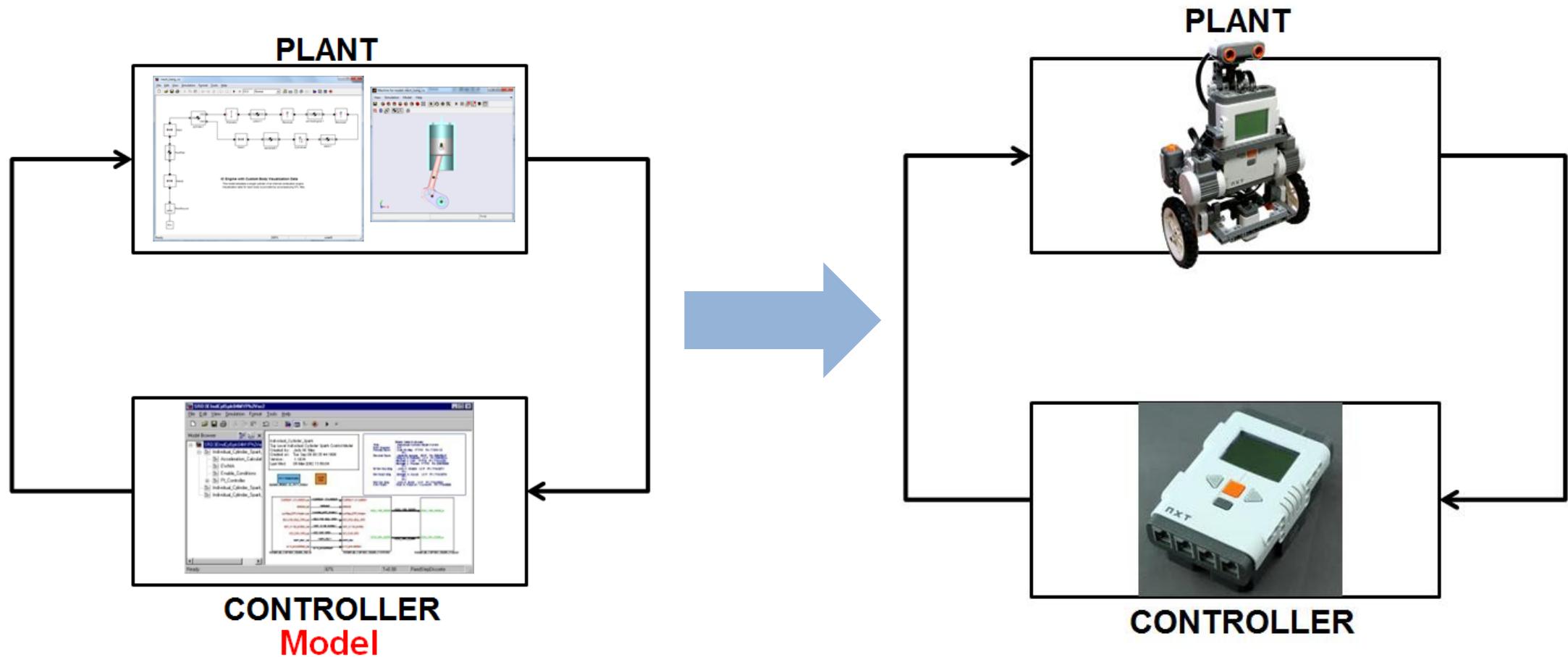
Rapid Prototyping (RP/RCP/RPC)



Deployment



Customizable Process



Lockheed Martin F-35B

Joint Strike Fighter



Scania

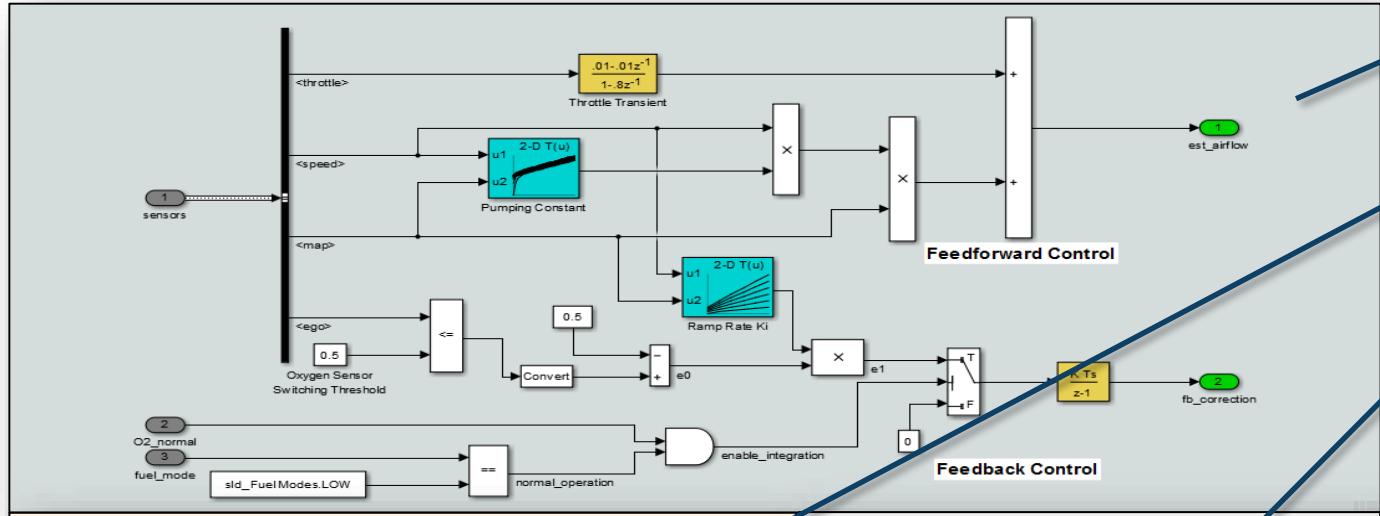
Automatic Emergency Braking

50 km/h - sudden brake



Software Modeling Tools

Simulink

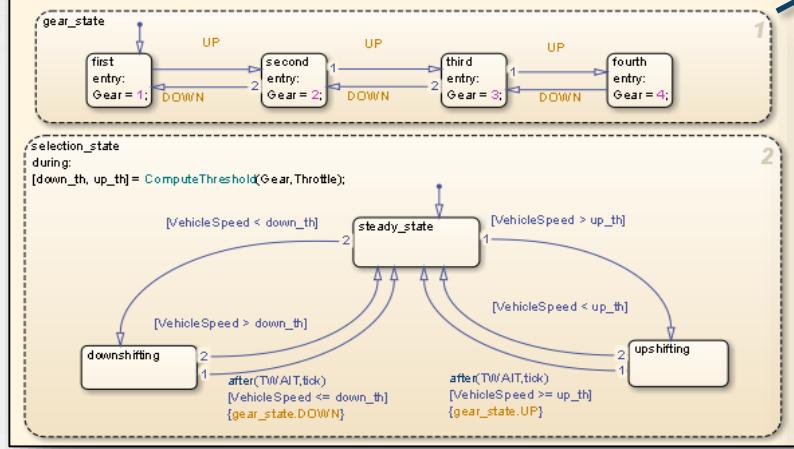


Block oriented for system design

State machines for event-based logic

Textual for concise numerics

Component based modeling



Stateflow

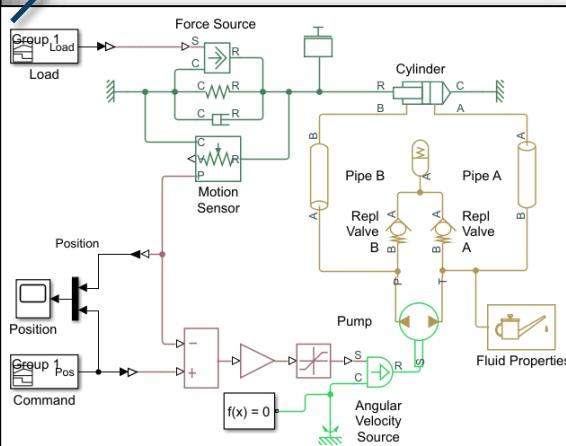
```
% Predicted state and covariance
x_prd = A * x_est;
p_prd = A * p_est * A' + Q;

% Estimation
S = H * p_prd' * H' + R;
B = H * p_prd';
klm_gain = (S \ B)';

% Estimated state and covariance
x_est = x_prd + klm_gain * (z - H * x_prd);
p_est = p_prd - klm_gain * H * p_prd;

% Compute the estimated measurements
y = H * x_est;
```

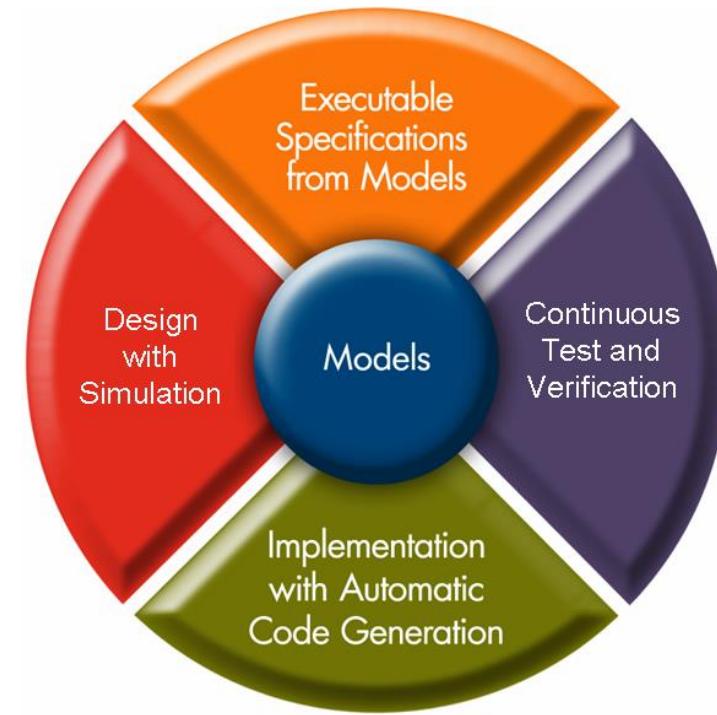
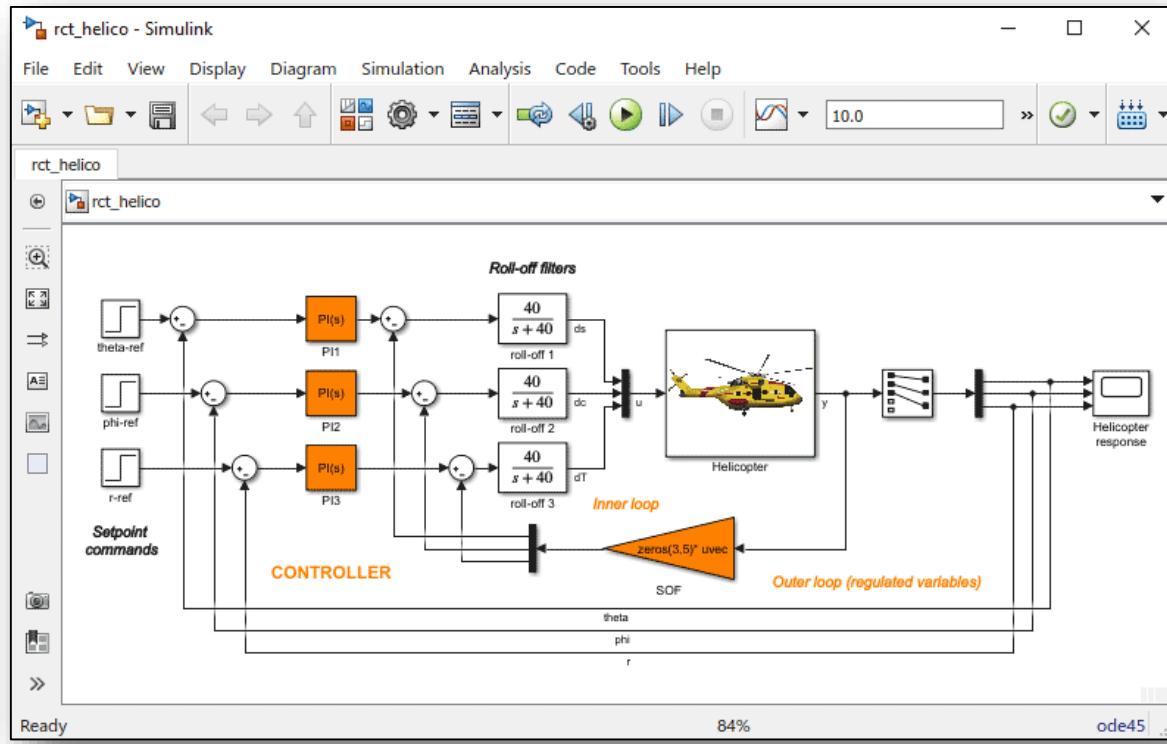
MATLAB



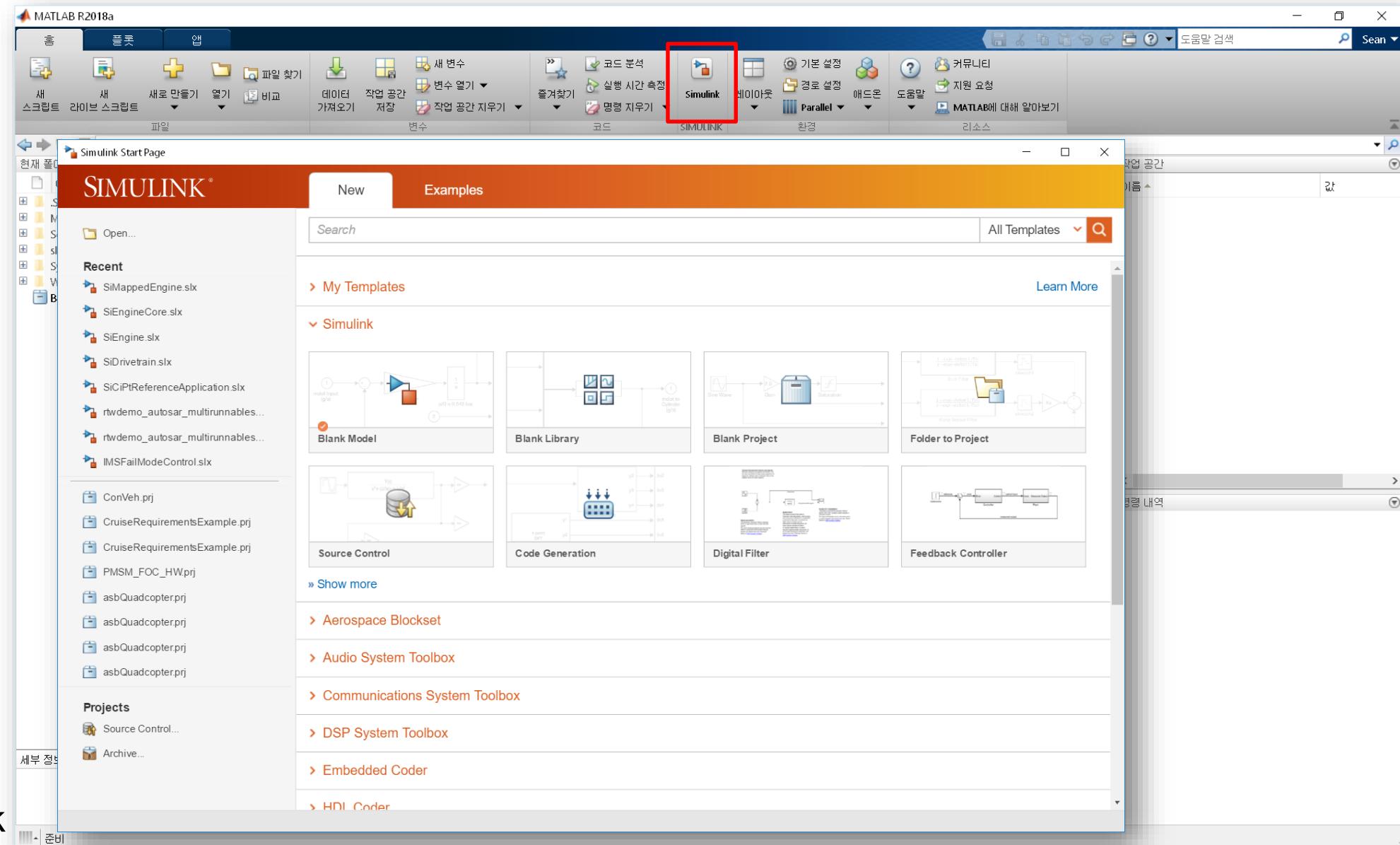
Simscape

What is Simulink?

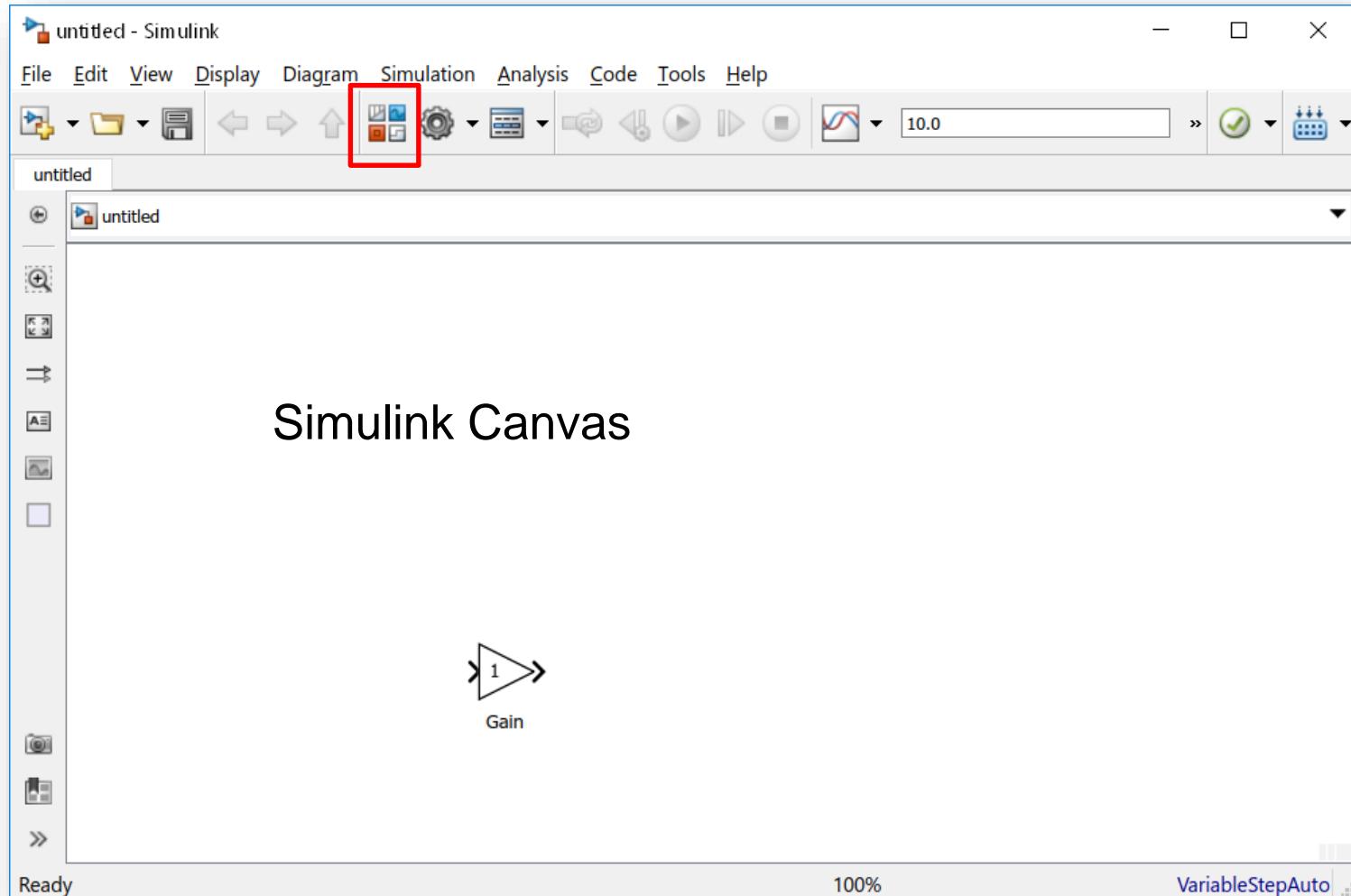
- Simulink is **Graphical Modeling**
- Simulink is **Time and Event Domain Simulation**
- Simulink is **Model-Based Design**



The Simulink® Desktop

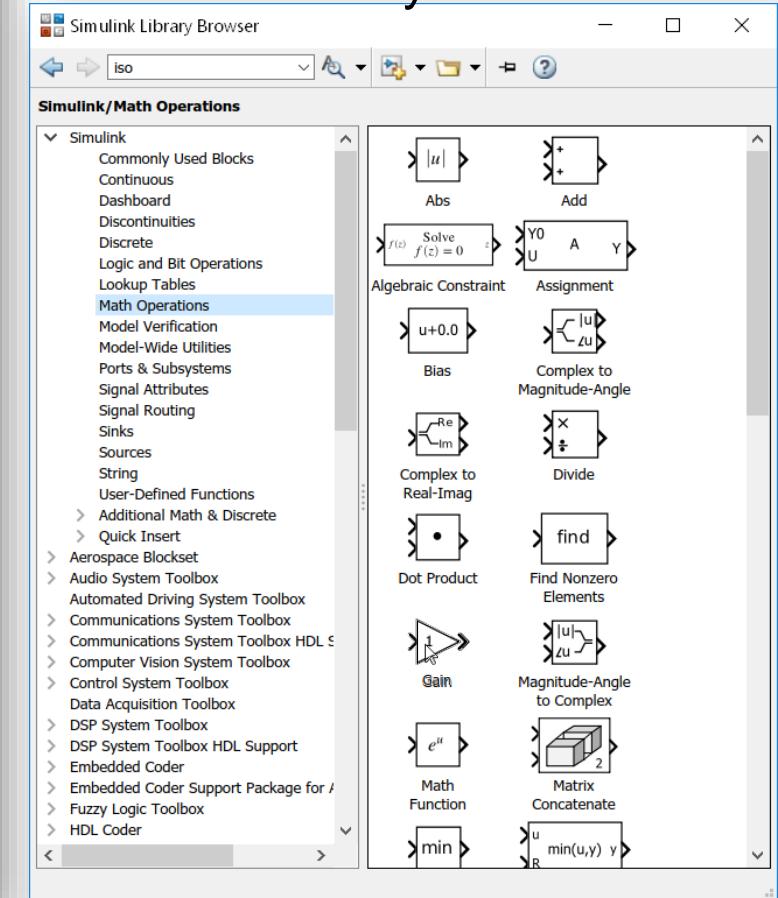


The Simulink® Desktop



Simulink Canvas

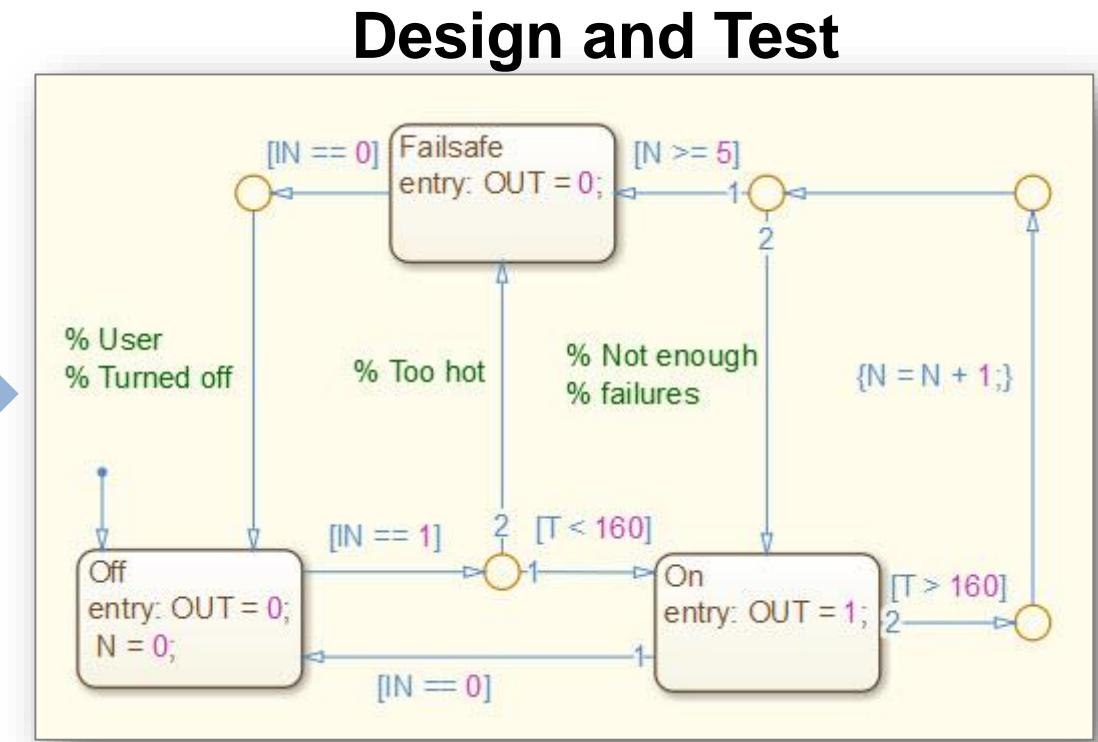
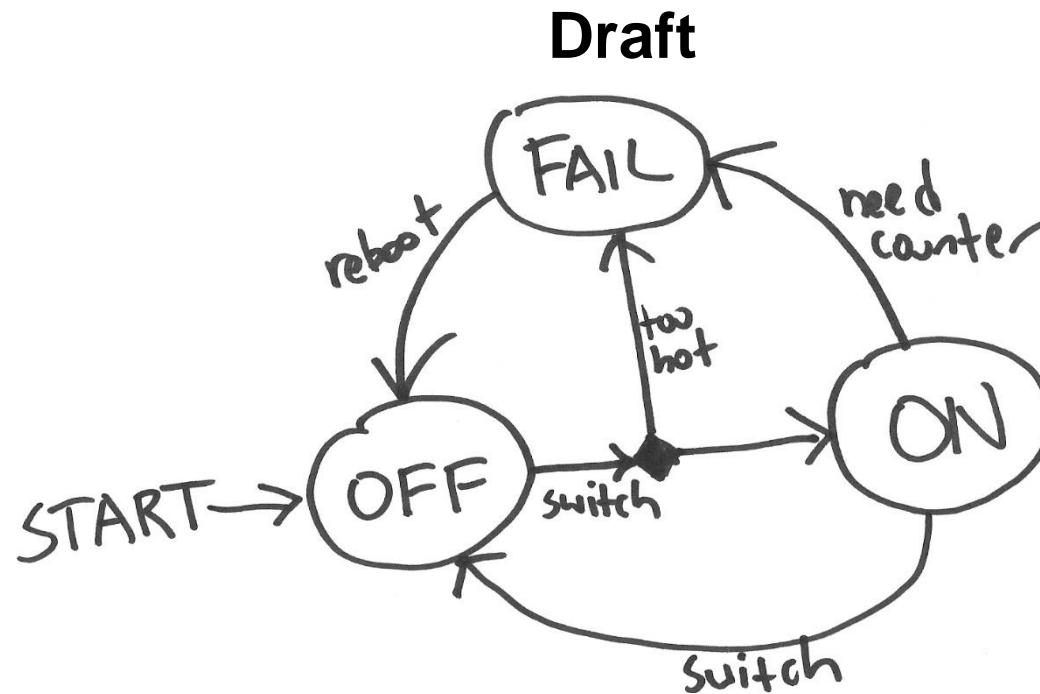
Simulink Library Browser



>> slLibraryBrowser

What is Stateflow?

- Stateflow is a state machine design environment

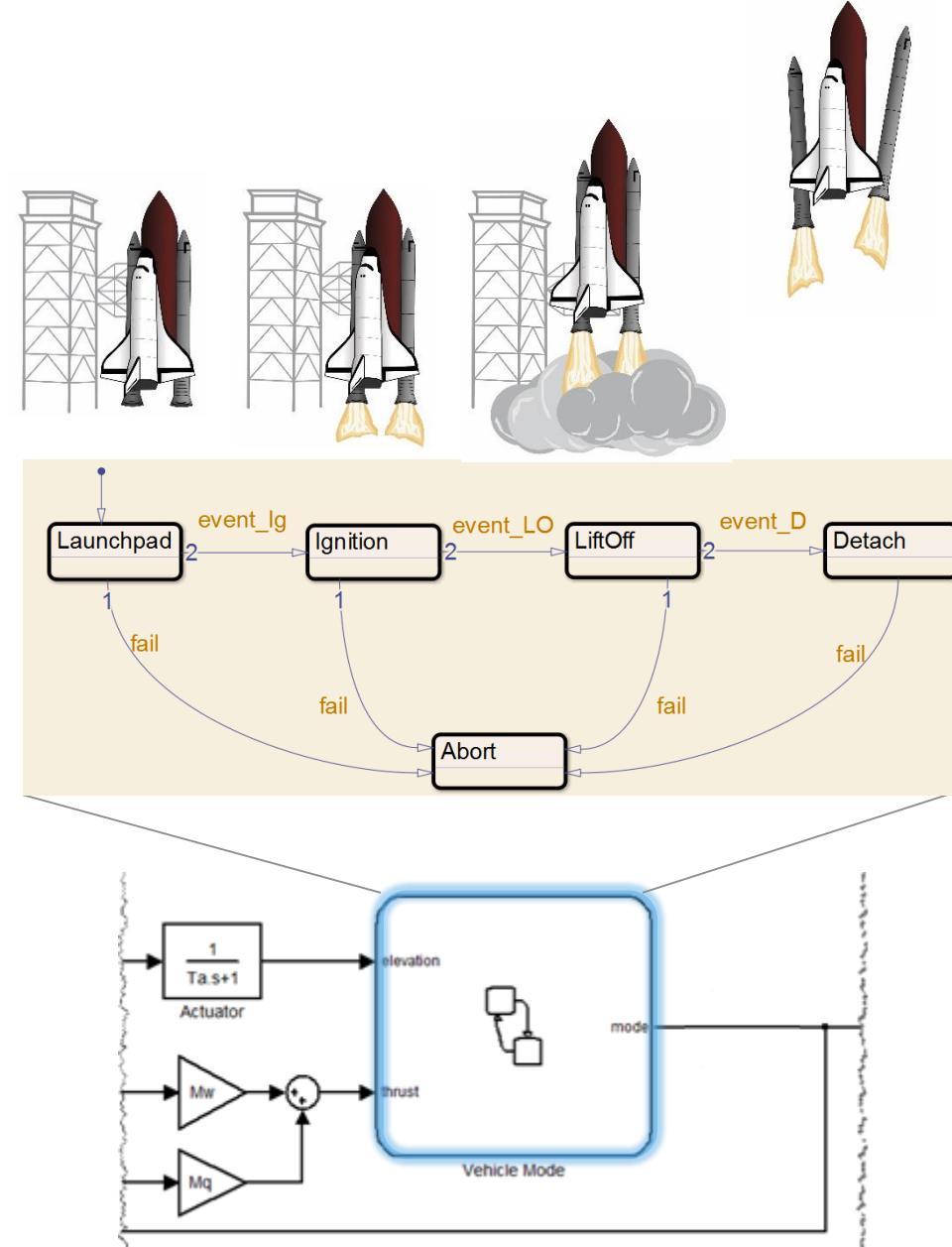


Why Stateflow?

Extend Simulink with state charts and flow graphs

Design supervisory control, scheduling, and mode logic

Model state discontinuities and instantaneous events



How Does Stateflow Work with Simulink?

Simulink models
continuous changes in
dynamic systems.

Stateflow models
instantaneous changes in
dynamic systems.

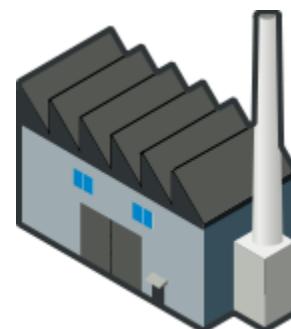
Real-world systems have to respond to both continuous and instantaneous changes.



suspension dynamics
gear changes



propulsion system
liftoff stages



manufacturing robot
operation modes

*Use both Simulink and Stateflow so that you
can use the right tool for the right job.*

Model-Based Design with MATLAB

- Limitations for code generation

disparity

Disparity map between stereo images

[collapse all in page](#)

Syntax

```
disparityMap = disparity(I1,I2)
d = disparity(I1,I2,Name,Value)
```

Description

di ... for a pair of stereo images, I1 and I2.

... function of parameters and corresponding values. One or more

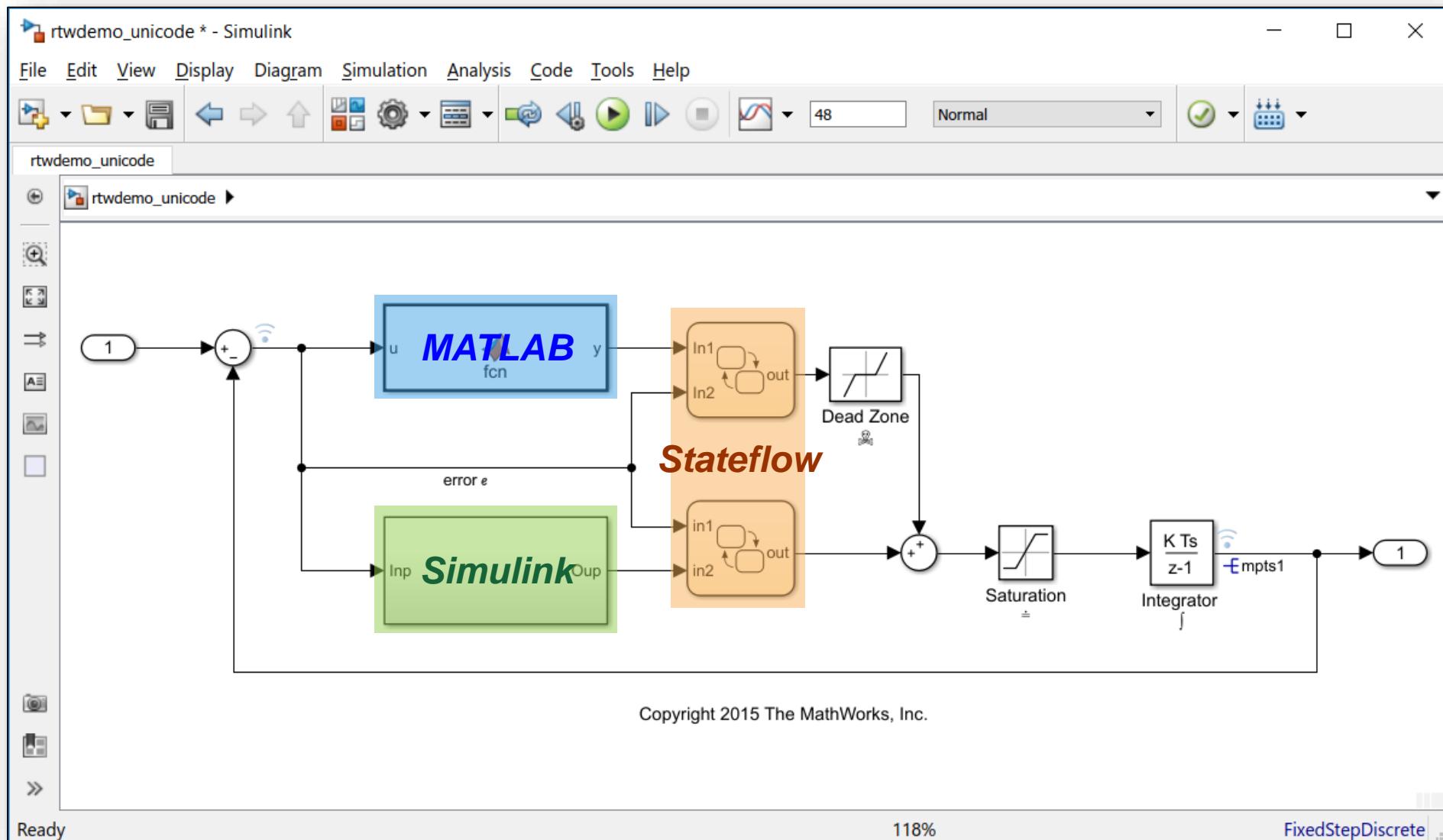
References

- [1] Konolige, K., *Small Vision Systems: Hardware and Implementation*, Proceedings of the 8th International Sy...,
- [2] Bradski, G. and A. Kaehler, *Learning OpenCV : Computer Vision with the OpenCV Library*, O'Reilly, Sebastopol, CA, 2008.
- [3] Hirschmuller, H., *Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information*, International Conference on Computer Vision and Pattern Recognition, 2005.

Extended Capabilities

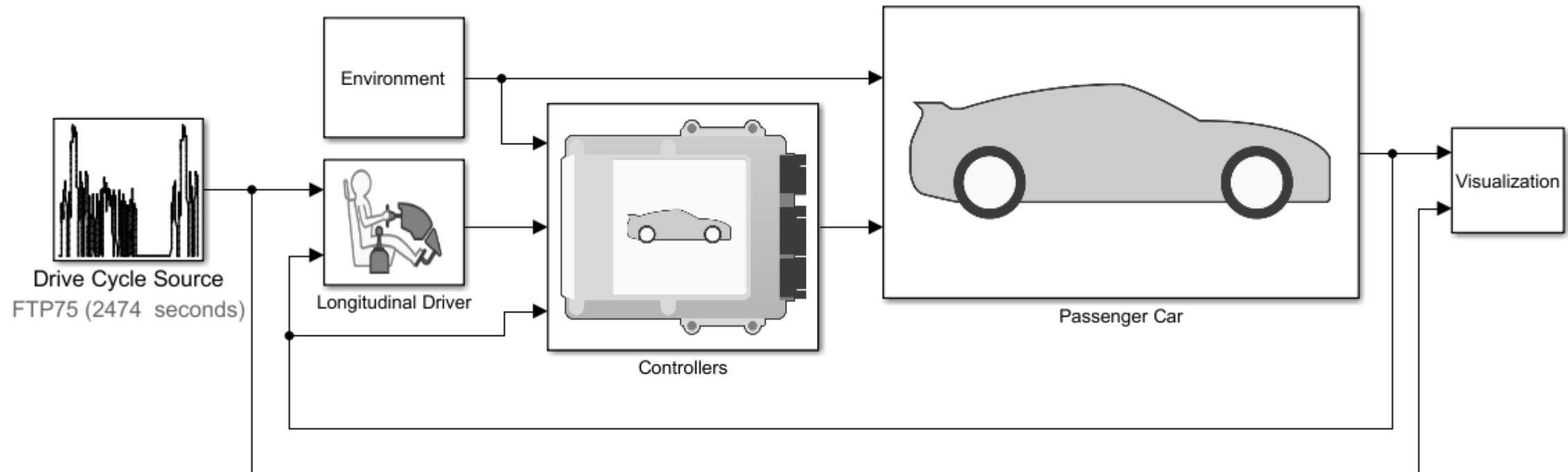
> **C/C++ Code Generation**
Generate C and C++ code using MATLAB® Coder™.

Software Modeling Example



Model-Based Design for Vehicle Applications

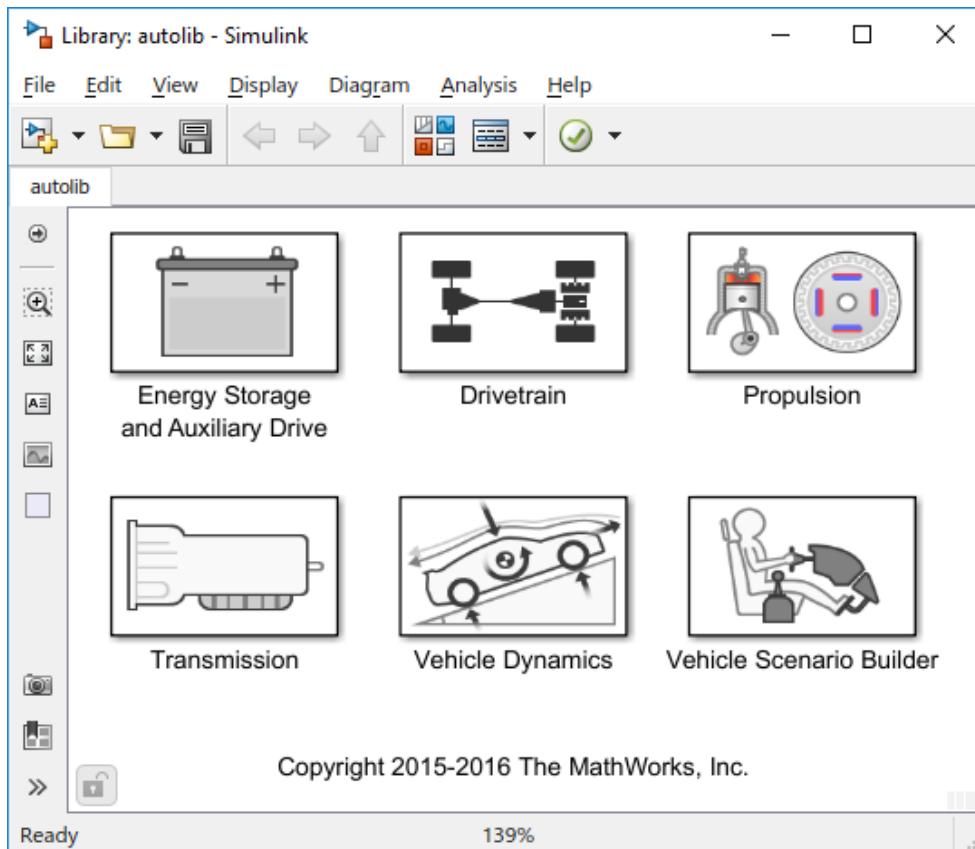
MathWorks Solution for Full Vehicle Simulation



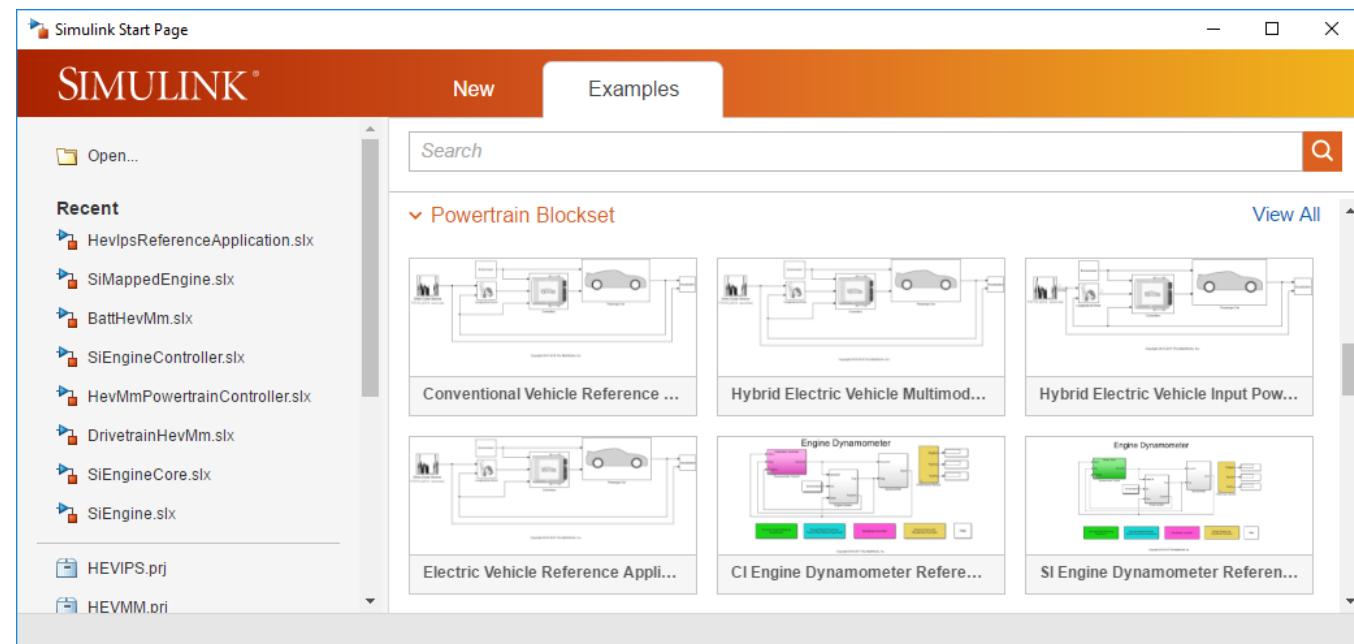
Lower the barrier to entry for Model-Based Design

Powertrain Blockset Features

Library of blocks



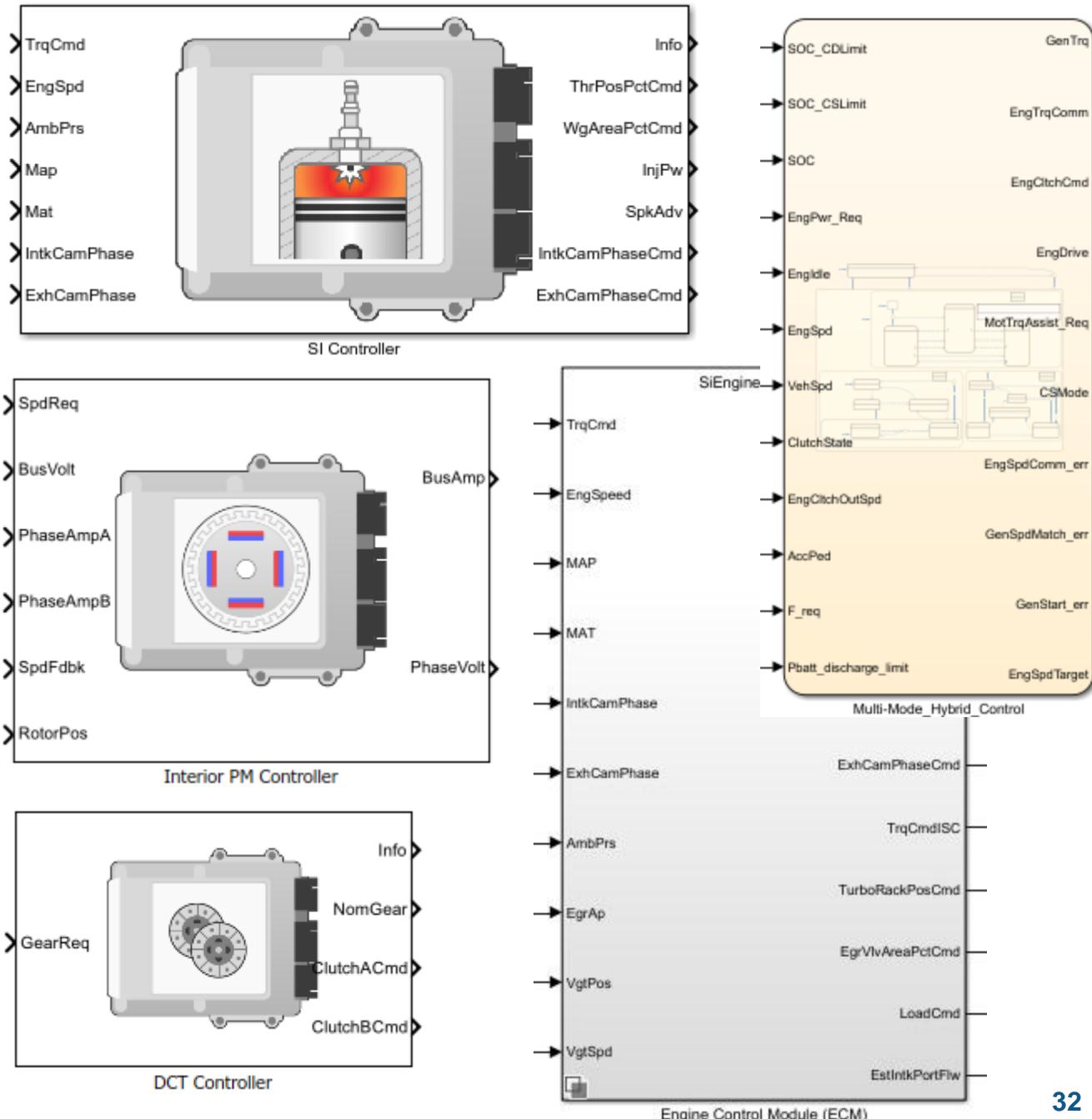
Pre-built reference applications



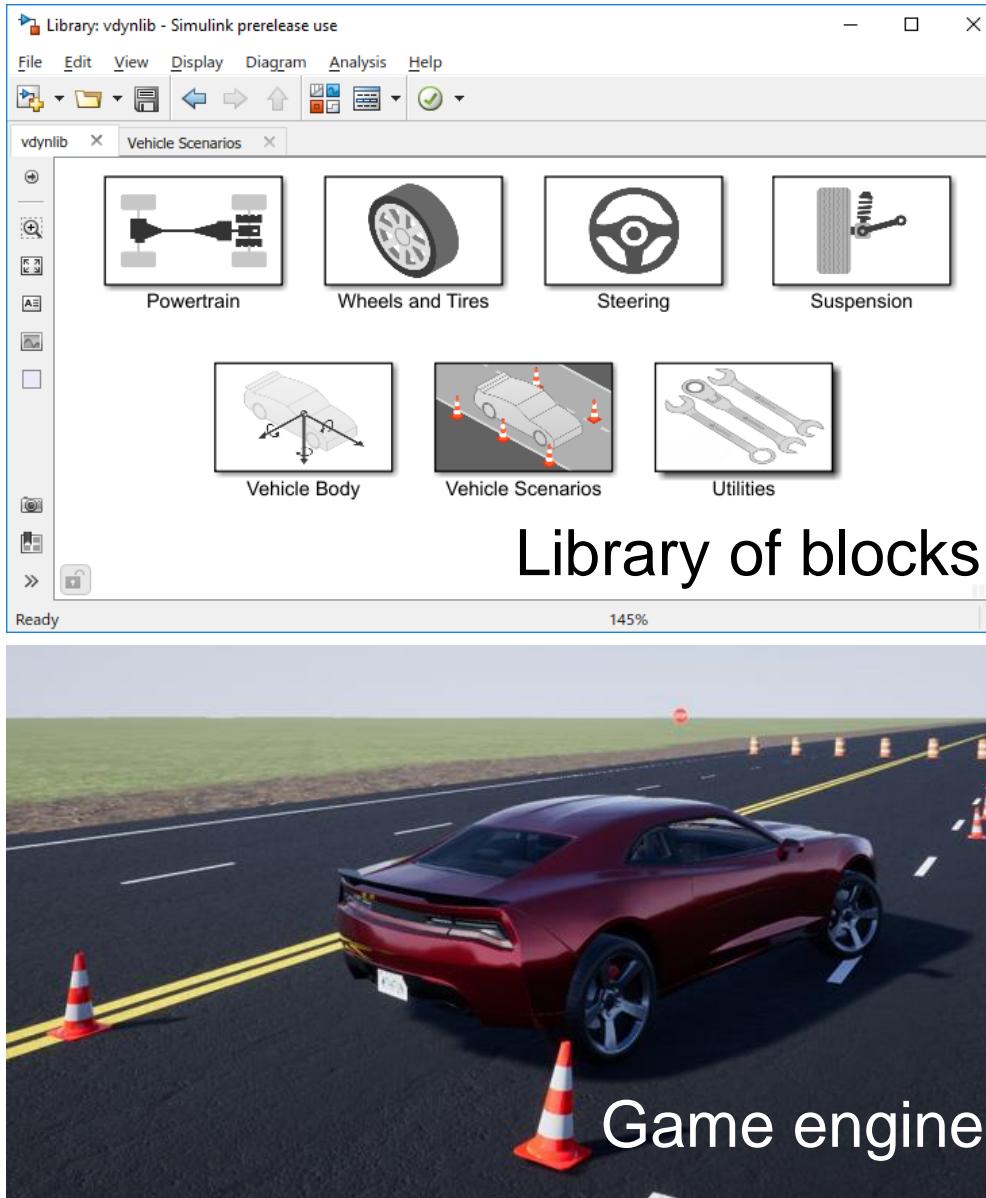
- Conventional (Spark Ignition / Compression Ignition) vehicle
- Electric vehicle (EV)
- Multi-mode / Power Split / P2 hybrid electric vehicle (HEV)
- Virtual engine dynamometer system

Library: Controllers

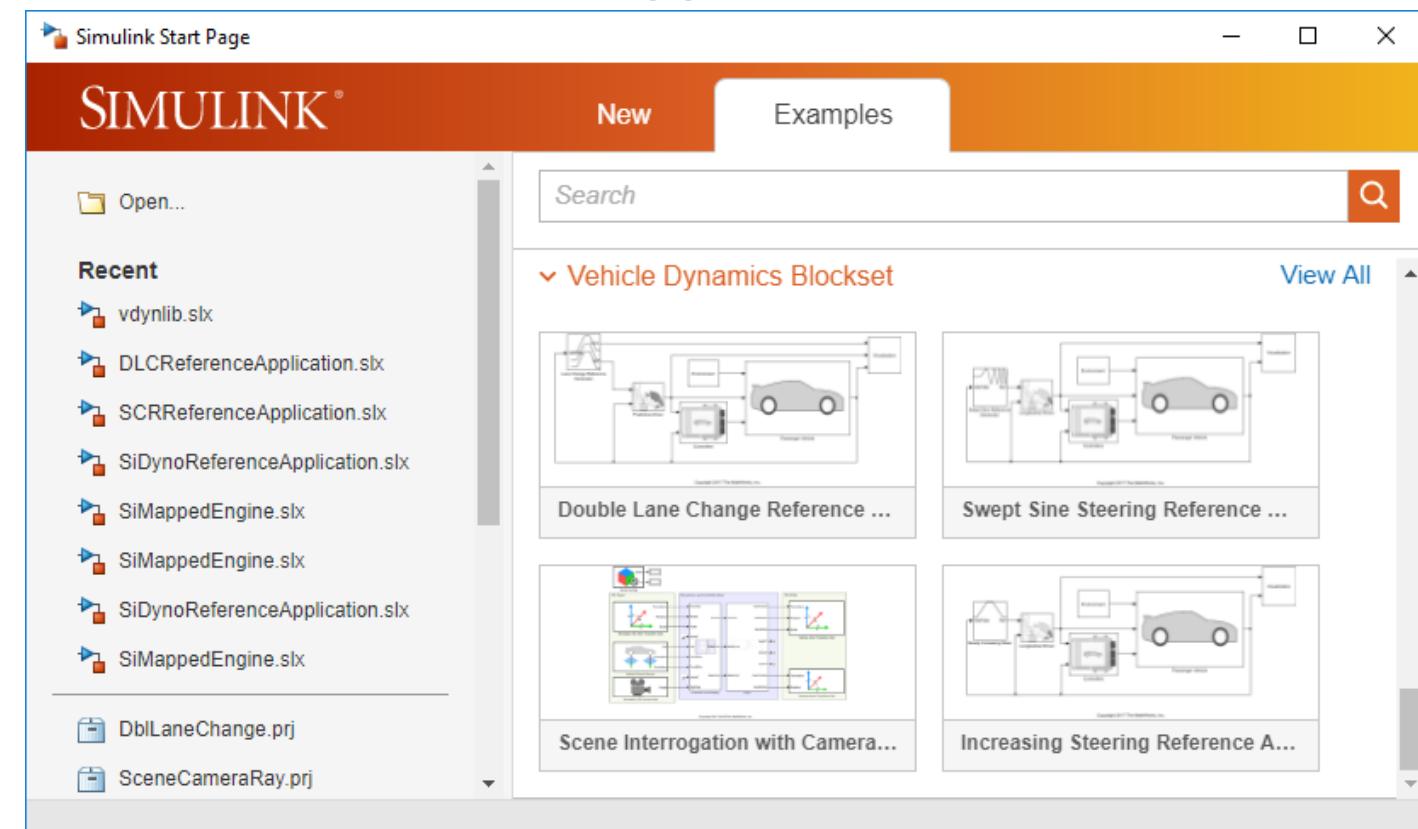
- Libraries include basic component and subsystem controllers
 - Powertrain Control Module (PCM)
 - Hybrid Control Module (HCM)
 - Engine Control Module (ECM)
 - Transmission Control Module (TCM)
 - Component controllers
 - State estimators
- Like plant models, the controllers are open and reconfigurable
- Provides realistic starting point for your own controller development



Vehicle Dynamics Blockset Features



Pre-built reference applications



Vehicle Dynamics Blockset

R2018a

- Model and simulate vehicle dynamics in a virtual 3D environment
- Use Vehicle Dynamics Blockset for:
 - Ride & handling: characterize vehicle performance under standard driving maneuvers
 - Chassis controls: design and test chassis control systems
 - ADAS / AD: create virtual 3D test ground for ADAS and automated driving features



Ride & handling

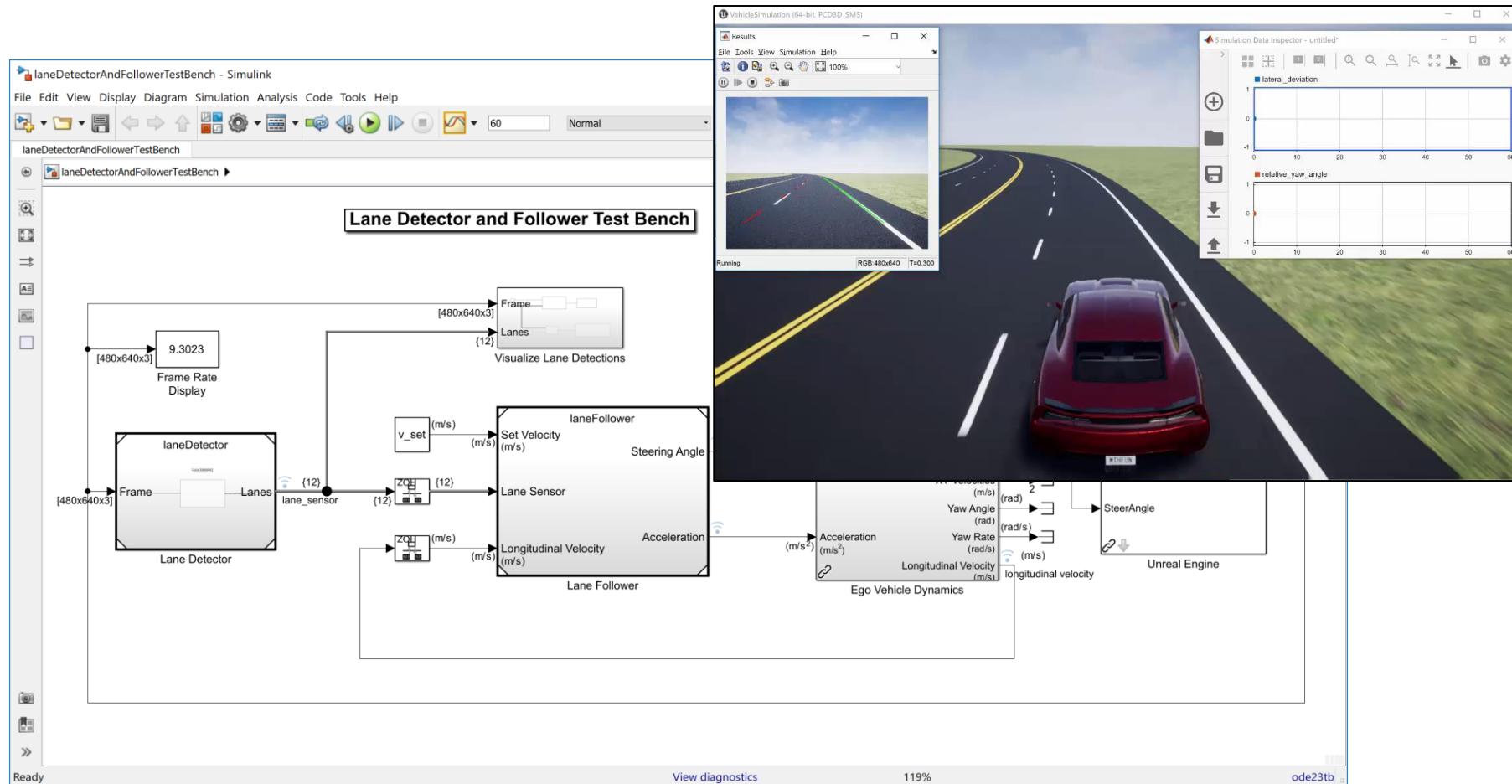


Chassis controls

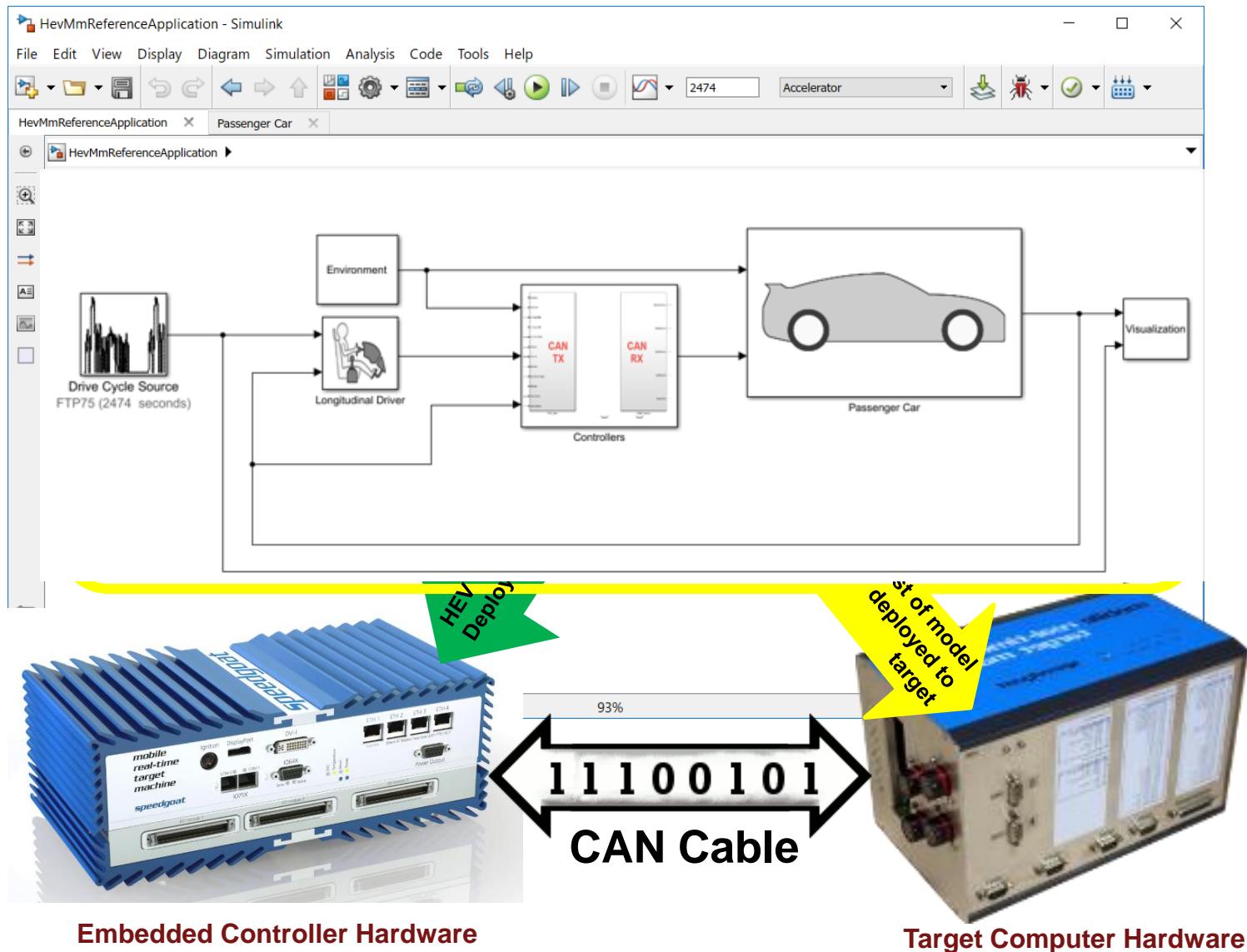


ADAS / AD

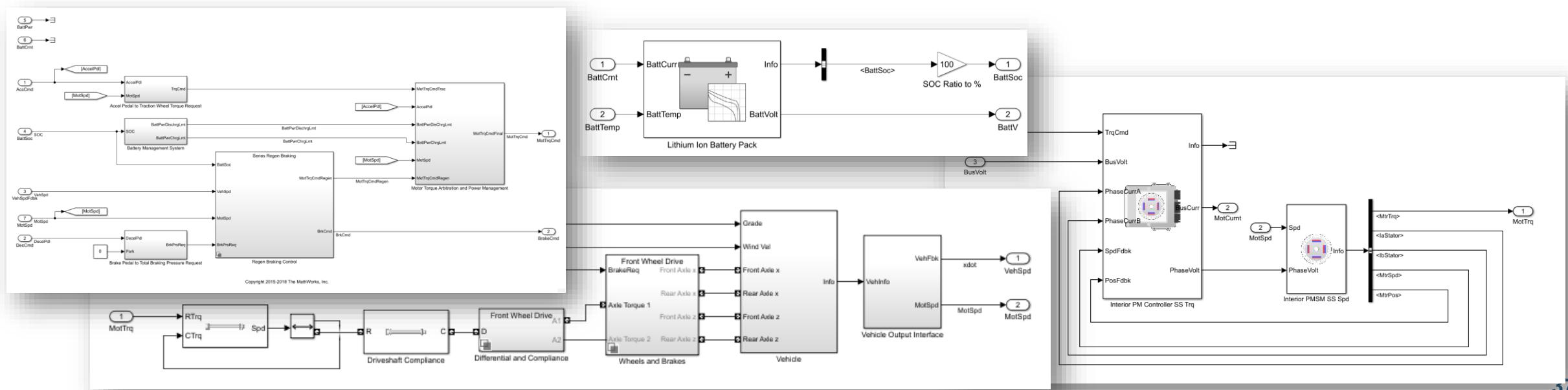
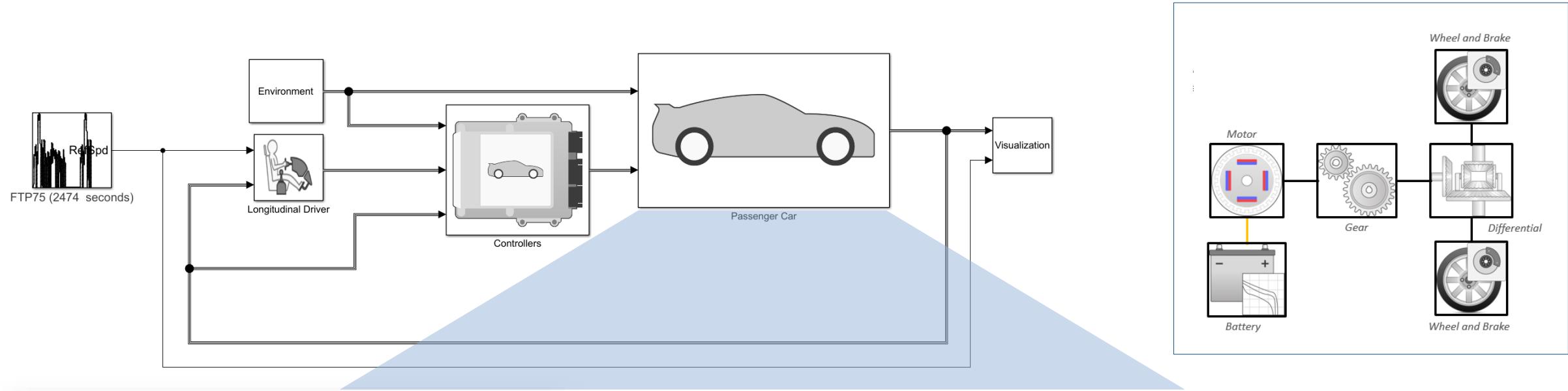
Lane detector and follower system test bench simulates vehicle dynamics with Unreal Engine to synthesize camera images



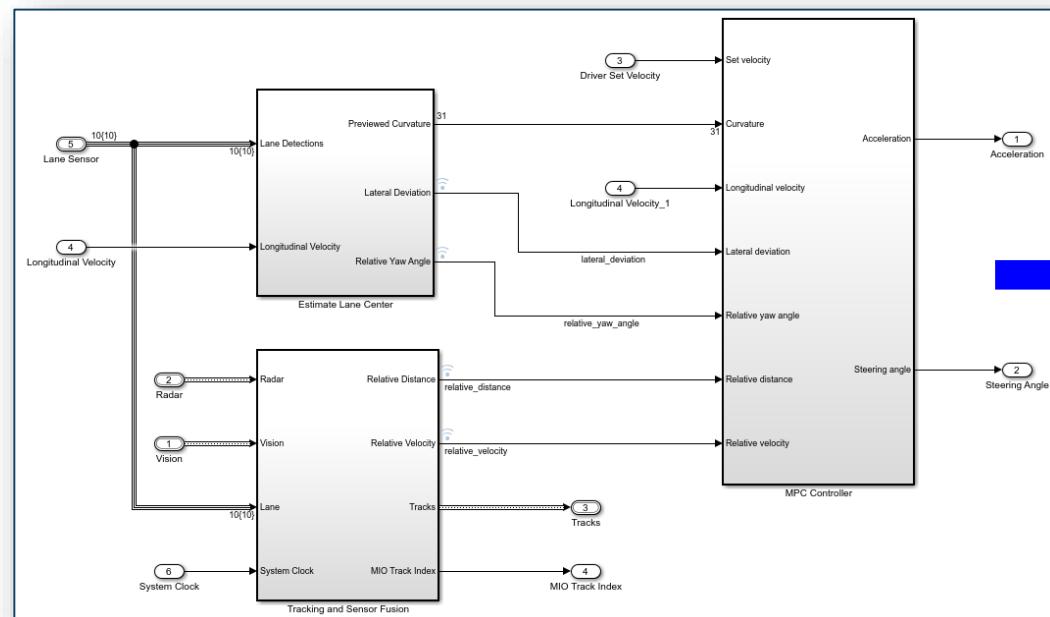
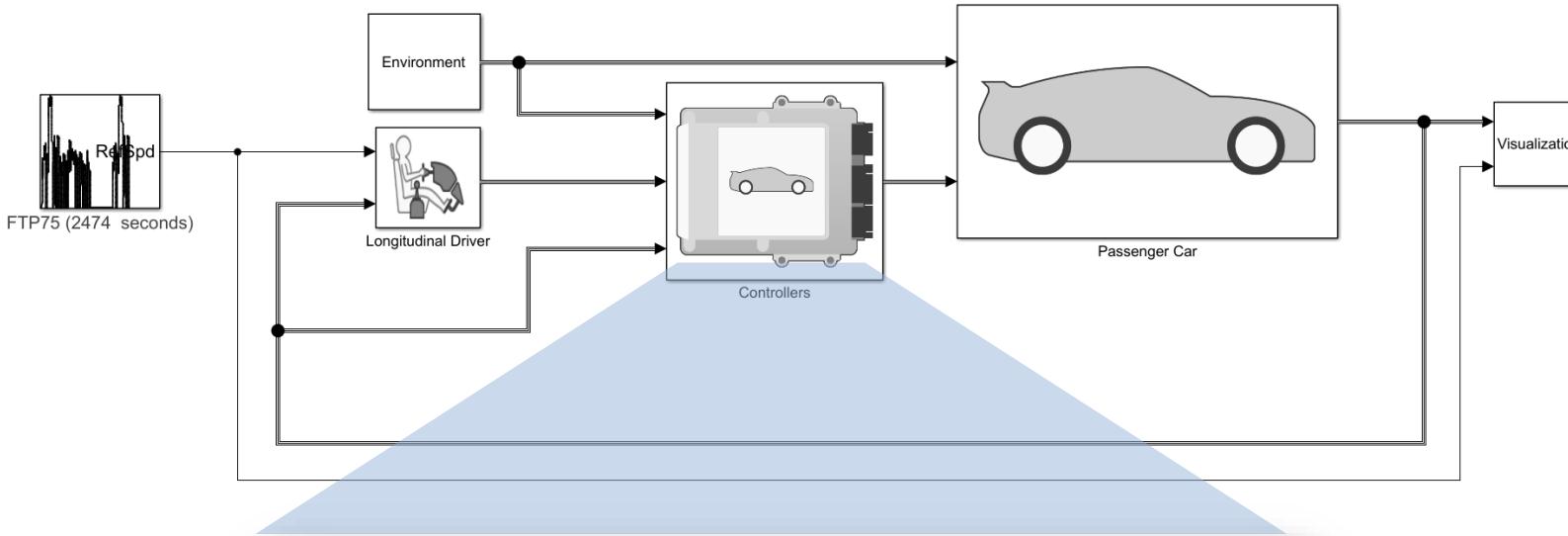
HIL Testing



EV Modeling Example



Controller Design



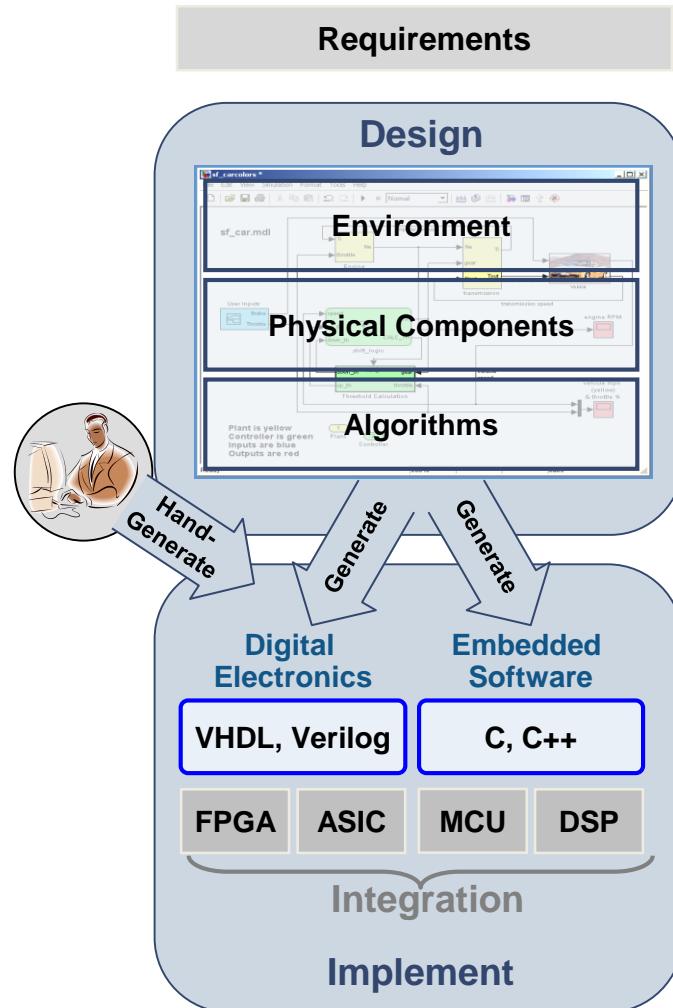
Automatic code generation

```

Code Generation Report
Find: Match Case
Contents
Summary Subsystem Report Code Interface Report Traceability Report Static Code Metrics Report
Generated Code
Main file crt_main.c
Model files crs_controller.c (6)
Utility files (1)
Generated Code
Block requirements for <Root>/DriverSwRequest:
41 reqMode rtb_Switch4;
42 reqMode rtb_reqQr;
43 boolean_T rtb_relop3_h;
44 opMode rtb_Switch_j;
45 real_T rtb_UnitDelay;
46 real_T rtb_Switch;
47 boolean_T rtb_out;
48 boolean_I rtb_out_f;
49 Type: Boolean_T
50
51 /* Outputs for Atomic SubSystem: <Root>/DriverSwRequest */
52 * Block requirements for <Root>/DriverSwRequest:
53 * 1. #1: Driver_Switch_Request_Handling
54 */
55 /* MATLAB Function: 'c599/counter' incorporates:
56 * Constant: 'c599/constant'
57 * Import: 'cRoot/lnk'
58 */
59 /* MATLAB Function: 'DriverSwRequest/increment/counter/counter': 'c5100/1.12'
60 /* if enabled */
61 /* <c5100/1.12> if isempty(cnt) */
62 if (crs_controller_U.lnc) {
63 /* output true if countValue is reached */
64 /* <c5100/1.14> if cnt >= countValue */
65 if (crs_controller_DW.cnt >= 50) {
66 /* <c5100/1.15> out = true; */
67 rtb_out = true;
68 }

```

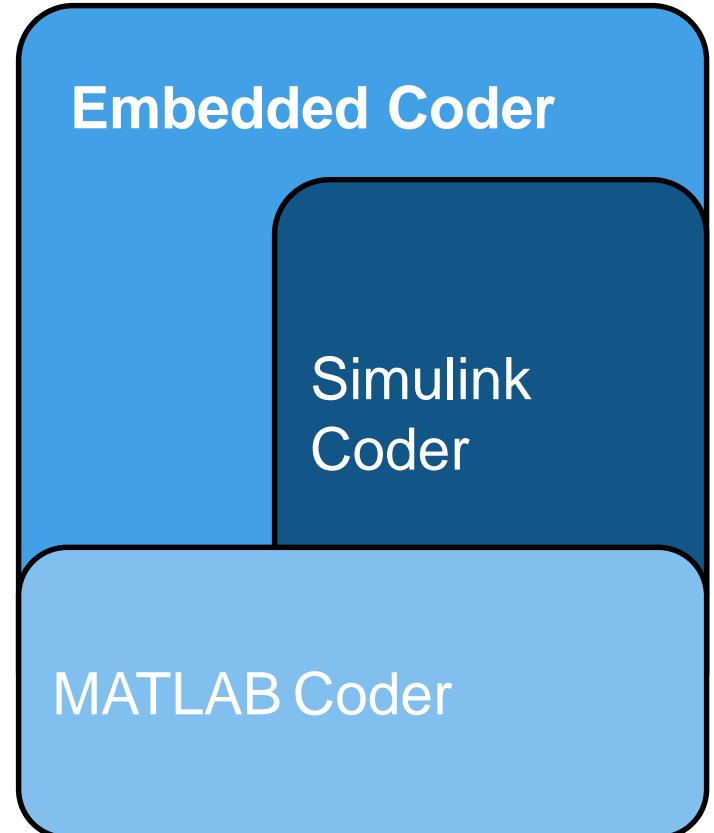
Code Generation Overview



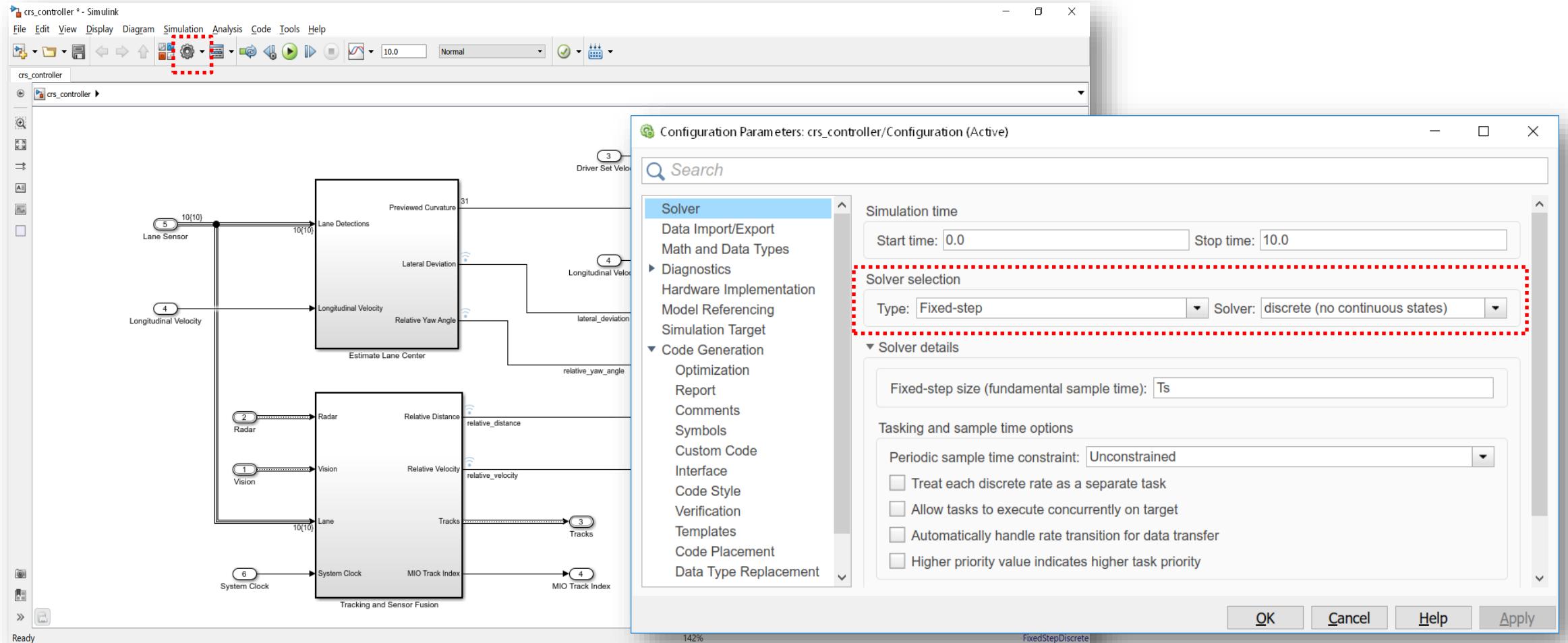
- ISO ANSI-C production code generation with Embedded Coder
- Use of Legacy Code Tool to make a call to existing C code in generated code from Simulink models
- HTML Report Generation for Review and Traceability

Code Generation Products

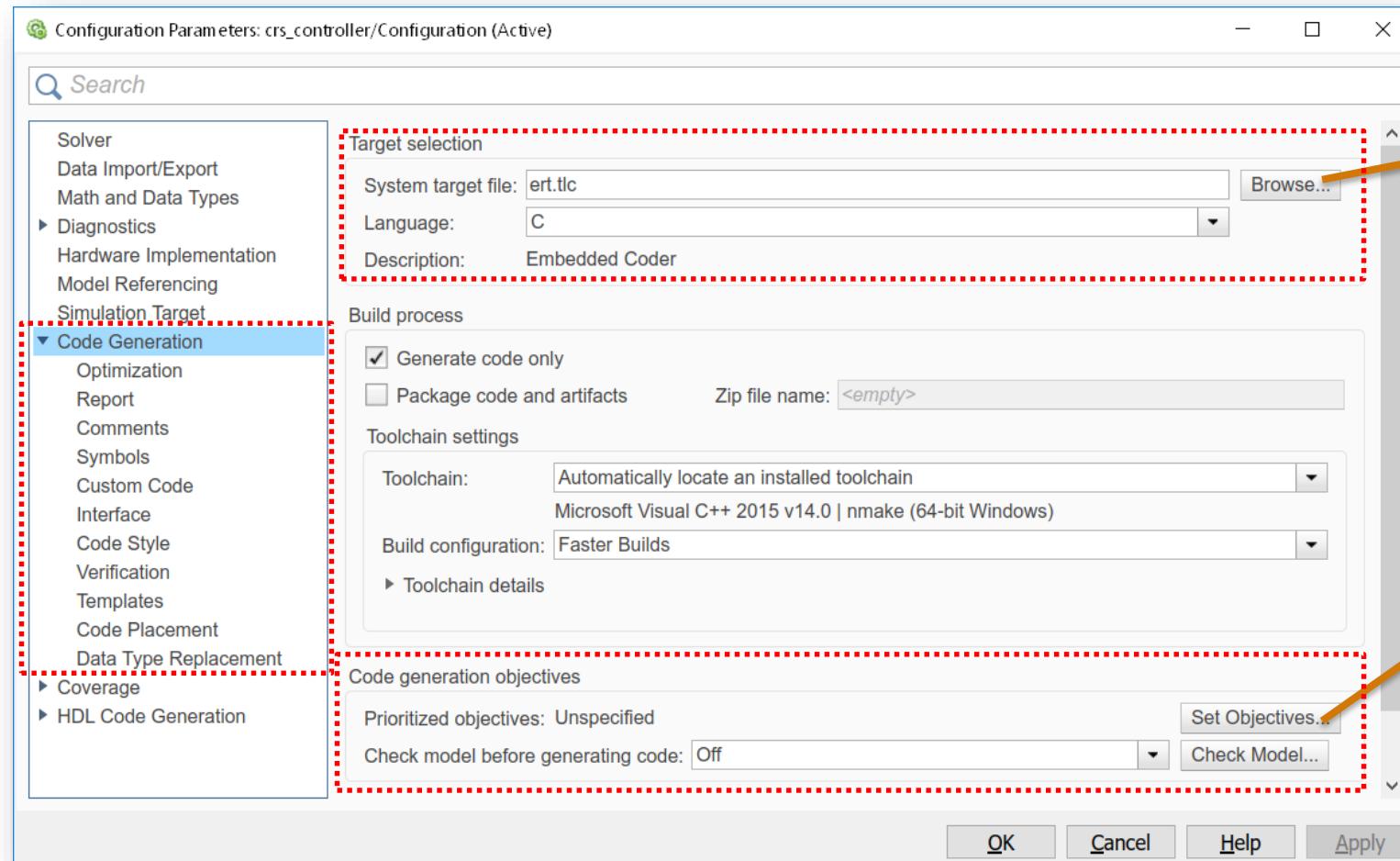
- **MATLAB Coder**
 - Generate C and C++ code from MATLAB code
- **Simulink Coder**
 - Generate C and C++ code from Simulink and Stateflow models
- **Embedded Coder**
 - Generate C and C++ code optimized for embedded systems



Prepare a Model for Embedded Code Generation



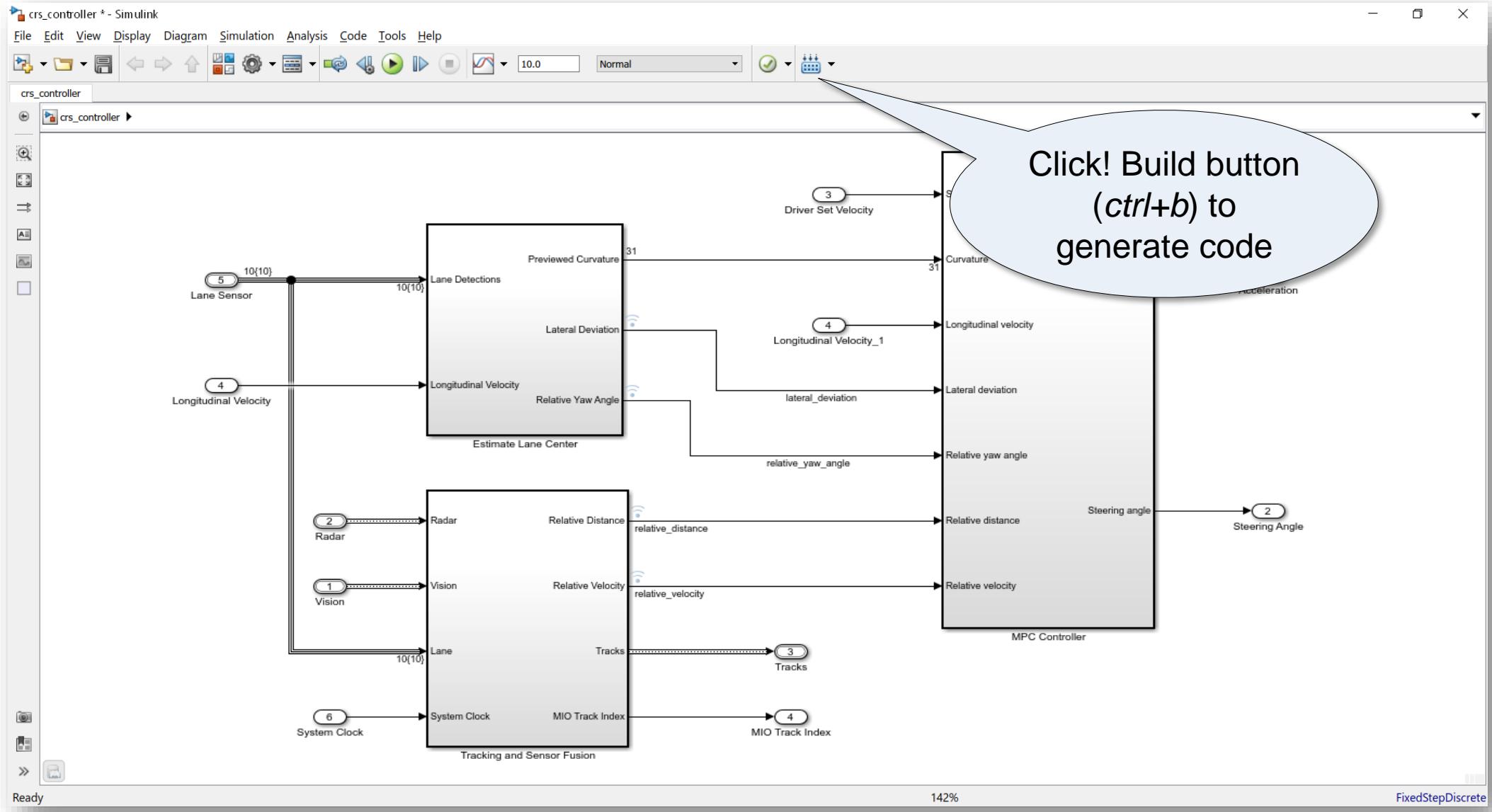
Prepare a Model for Embedded Code Generation



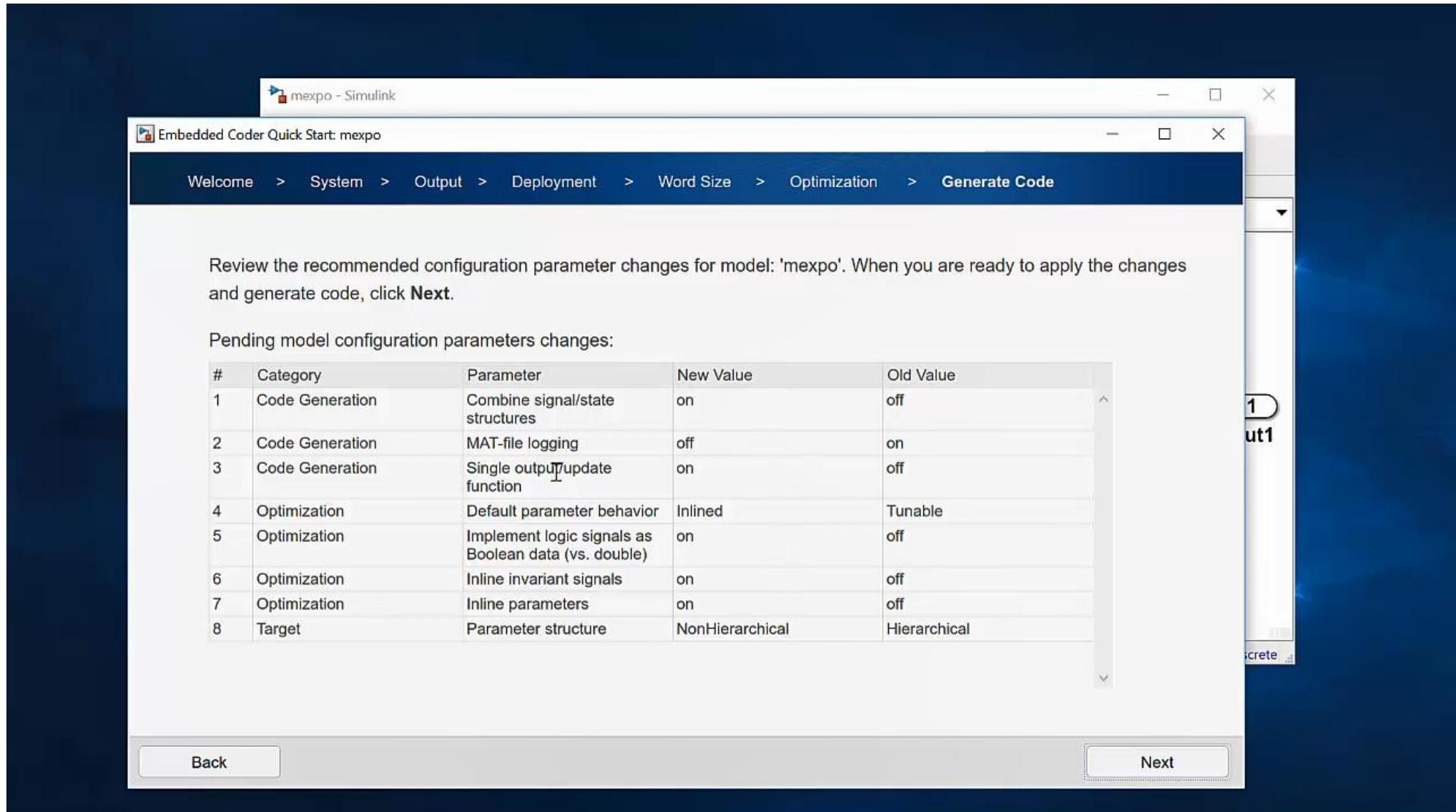
The image shows two overlapping dialog boxes:

- System Target File Browser:** Shows a list of system target files with their descriptions. The entry 'ert.tlc' is highlighted with a red dashed box and has an orange arrow pointing from the 'Target selection' field in the main dialog. The full path 'C:\MATLAB\R2017b\rtw\c\ert\ert.tlc' is shown at the bottom.
- Set Objectives - Code Generation Advisor:** Shows a list of available objectives and selected prioritized objectives. The selected objectives ('Execution efficiency, RAM efficiency, MISRA C:2012 guidelines, Safety precaution') are highlighted with a red dashed box and have an orange arrow pointing from the 'Prioritized objectives' field in the main dialog.

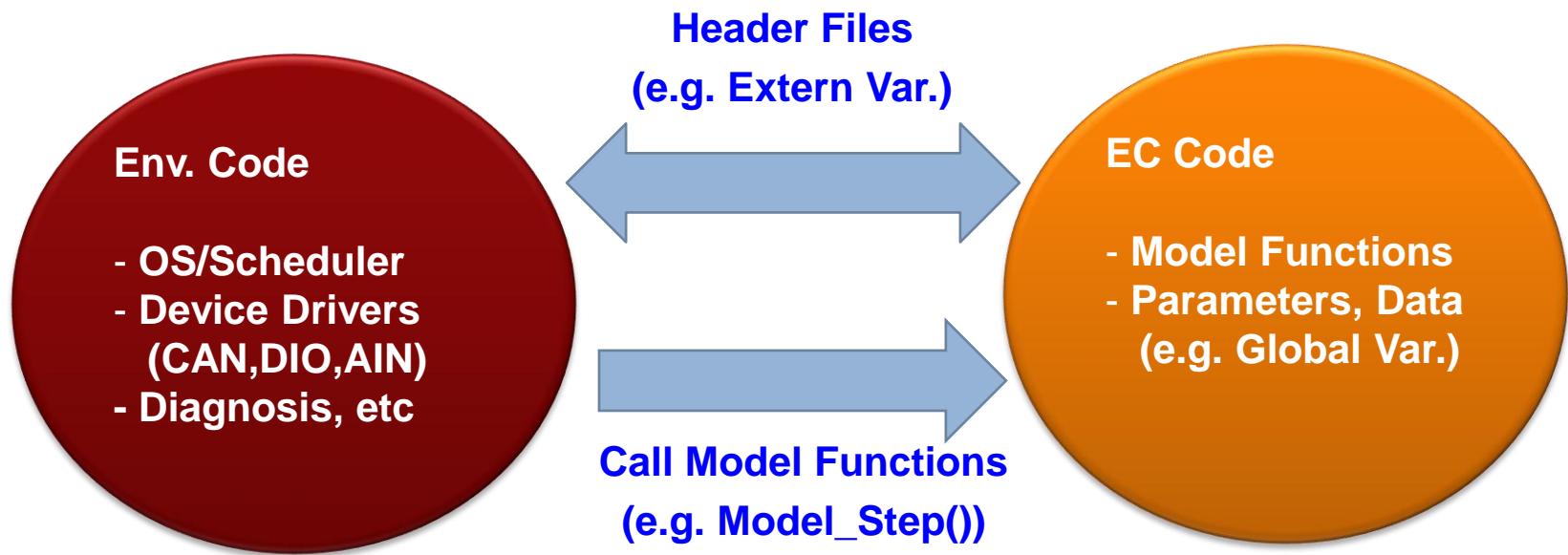
Generating Code



Demo: Embedded Coder Quick Start



Interfacing Generated Code



- Share entry points, data and parameters with header files
- Call generated model functions from OS/Scheduler

Implementation

Example : Integrating Generated Controller Code with an Embedded Software Project

Embedded Software Project

```
Main()
{
    Commands_Init
    PWM_Init
    ADC_Init
    Encoder_Init

    Controller_Init

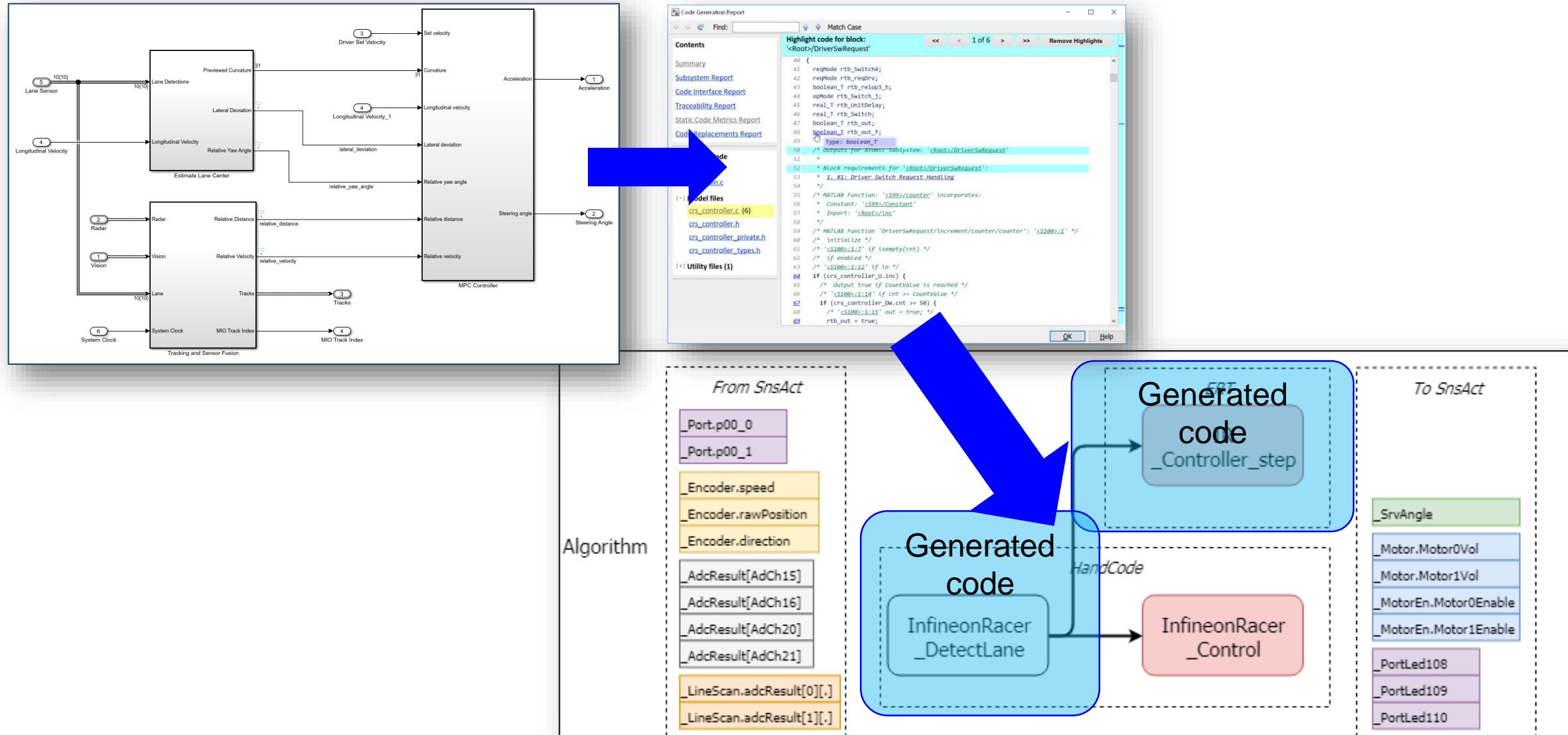
    while(1) { }
}
```

```
ADC_InterruptServiceRountine_25kHz()
{
    Commands_Read
    ADC_Read
    Encoder_Read

    Controller_Step

    PWM_Write
}
```

Software Integration Example for SMCC



Self-study Resources

MathWorks® 제품 솔루션 아카데미아 지원 커뮤니티 이벤트 안내

Academia

Student Home | MATLAB Student | Examples | Student Competitions | Books | Hardware Support

MathWorks.comLH 검색

MATLAB and Simulink Racing Lounge

The MATLAB and Simulink Racing Lounge features a video series covering topics relevant to automotive student teams. Everything from the basics of MATLAB and Simulink to in-depth details on improving your racecar development is explored. Automotive student teams also share their keys to success with examples.

Here's how to stay connected and find out about new videos, training, tips and tricks, File Exchange entries, or get technical advice:

- ☛ Join the MATLAB and Simulink Racing Lounge Facebook group
- ☛ Learn from Physical Modeling Training designed for automotive racing teams (12 Videos)
- ✉ Email engineers dedicated to helping student teams at racinglounge@mathworks.com
- ☛ Find examples by the MathWorks Student Competitions Team on MATLAB Central File Exchange

Most Viewed

MATLAB and Simulink Racing Lounge: Developing Algorithms for ADAS Systems with MATLAB and Simulink

MATLAB and Simulink Racing Lounge: Battery Modeling with Simulink

MATLAB and Simulink Racing Lounge: Lap Time Simulation; Essential Part of Concept Development

MATLAB and Simulink Racing Lounge: Vehicle Modeling, Part 1: Simulink

MATLAB and Simulink Racing Lounge: Tire Modeling; Extracting Results from a Large Data Set

Most Recent

BAJA SAE: How Can MATLAB and Simulink Be Used to Design an ATV?

MATLAB and Simulink Racing Lounge: Vehicle Modeling, Part 4: Simscape Multibody

MATLAB and Simulink Racing Lounge: Vehicle Modeling, Part 3: Simscape

MATLAB and Simulink Racing Lounge: Remote Control Racecar, Part 2: Integrating CAN Data into Your Simulation

MATLAB and Simulink Racing Lounge: Remote Control Racecar, Part 1: Programming ECUs Using Simulink Hardware Support

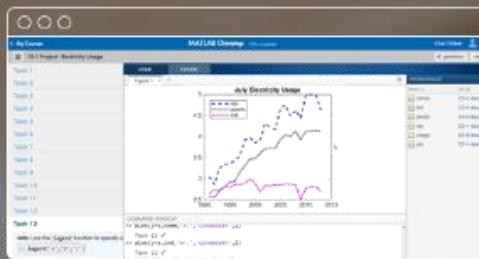
Self-Study Resources for Embedded Code Generation

- Embedded Coder Quick Start Video
https://kr.mathworks.com/videos/coder-summit-2018-how-to-generate-production-code-in-5-minutes--1522057622892.html?s_tid=srchttitle
- Simulink와 Embedded Coder를 이용한 최적 코드 생성
<http://www.matlabexpo.com/kr/2017/proceedings/better-than-hand-generating-highly-optimized-code-using-simulink-embedded-coder.pdf>
- C code generation from Simulink model(webinar)
https://kr.mathworks.com/videos/software-design-and-c-code-generation-using-simulink-116860.html?elqsid=1524127259550&potential_use=Commercial
- Other Embedded Coder Videos
https://kr.mathworks.com/products/embedded-coder/videos.html?s_tid=srchttitle

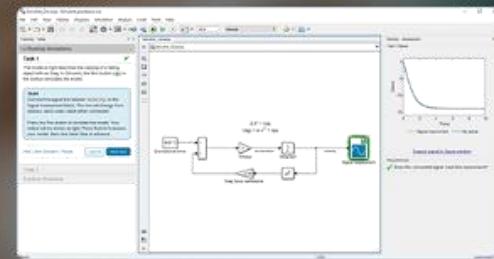
MATLAB 및 Simulink 튜토리얼 학습하기

온라인 자기 주도형 교육 과정에서 **MATLAB** 을 직접 사용해보실 수 있습니다.

- 웹 브라우저에서 프로그래밍 연습
- 짧은 비디오 데모
- 즉각적인 피드백

[모든 과정 둘러보기](#)


MATLAB Onramp
무료 - **MATLAB**의 필수 기능을 학습합니다.

[시작](#)


Simulink Onramp
무료 - **Simulink**의 필수 기능을 빠르게 학습합니다.

[제품정보](#)

MATLAB 기본 배우기

MATLAB® 은 전 세계 수백만 명의 엔지니어와 과학자가 이용하는 고급 언어(high-level language)이자 대화형 환경입니다. 신호 및 이미지 처리, 통신, 제어 시스템 및 계산 금융을 포함한 여러 분야에서 아이디어를 시각화할 수 있도록 돕습니다.

<https://kr.mathworks.com/support/learn-with-matlab-tutorials.html>

Simulink 기본 배우기

Simulink® 는 블록 다이어그램 환경으로 구성된 멀티 도메인 시뮬레이션 및 모델 기반 설계를 위한 도구입니다. 시뮬레이션, 자동 코드 생성과 임베디드 시스템의 지속적인 테스트 및 검증을 지원합니다. 그래픽 편집기, 사용자 정의 가능한 블록 라이브러리, 모델링을 위한 솔버를 활용하는 동적 시스템을 시뮬레이션합니다.

Prerequisites for Development Environment

AurixRacer 다운로드

realsosy / AurixRacer

Watch 1 | Star 2 | Fork 3

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Smart Model Car Contest contents

smcc aurix infineon aurixracer

53 commits 2 branches 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

wootalk Update index.md Latest commit 1d4ef3a 8 days ago

docs Update index.md 8 days ago

src no message 13 days ago

.gitignore no message a month ago

README.md Update README.md 8 days ago

mkdocs.yml Move simulink part from AurixTutorial a month ago

README.md

AurixRacer

Purpose & Scope

- 지능형 모형차 대회에 사용할 수 있는 Software Platform 을 제공
- 아울러 AURIX를 활용한 학부생 실습용 프로젝트 예제

<https://github.com/realsosy/AurixRacer>

개발환경 설치

AurixRacer Documentation

- Home
- AurixRacer Guide
- Getting Started**
 - Getting Started
 - Objectives
 - References
 - Example Description
 - 개발환경 설치**
 - Project Build
 - Shell 을 이용한 동작 확인
 - 추가적인 설명
- User Guide
- Connection Guide
- Simulink as a Programming Language

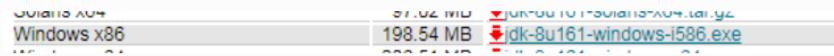
DigitalOcean

New: DigitalOcean Marketplace Self-

개발환경 설치

HighTec Installation

- Dependency: [Java SE Development Kit \(32bit\)](#)
 - Windows 32bit 64bit 운영체제 모두 32bit 버전으로 다운받아야 함 (최신 버전으로 받으면 됨)



- [Tricore tool chain 홈페이지](#)에서 개인 정보 입력 후 Generate License File & Download 버튼 클릭
 - 이때 입력한 개인정보는 HighTec 다운로드 완료 후 License 인증 시 필요
 - 다운로드 완료한 "license.lic" 파일은 "C:\HighTec\licenses"로 복사

HighTec Free TriCore™ Entry Tool Chain
The easiest way to start with a C/C++ compiler for TriCore or AURIX, as well as a TriBoard or AURIX Starter Kit.
Version 4.9.1.0-Infineon-1.1 (includes support for AURIX 2G TC39xx, AURIX TC29x, TC27x, TC26x, TC23x, TC22x and TC21xx)

Instructions:

- Please fill out the form to the right
- You will be sent an e-mail, which contains your license file and the download link
- Download the installation package
- Unzip the installation package and start "Setup.exe" to install the toolchain
- Copy the license file to the folder "C:\HighTec\licenses"
- Read the "Getting Started" manual for additional information

Getting Started

Need a multi-core RTOS (IEC61508 SIL2), MCAL or SafeTlib?

Contact sales
 Contact support

Need technical assistance or training?

Contact support
 Contact sales

Name * John Doe

E-mail address * E-mail address

Company * Company

Department * Department

Job title * Job title

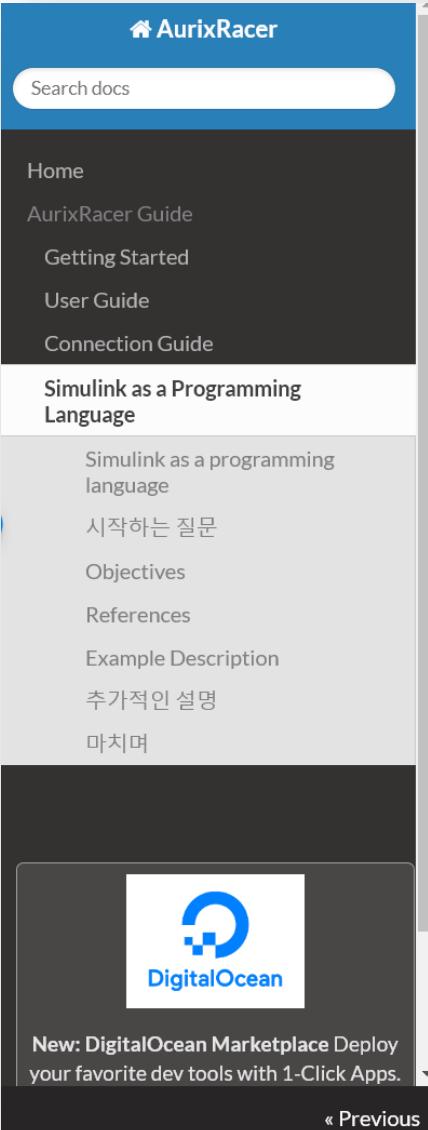
Phone number * Phone number

MAC-address * 11:22:33:44:55:66

Enter below text *

- [free_tricore_entry_tool_chain.zip](#) 다운로드 후 압축 해제
- 압축 해제된 파일 중 setup.exe를 실행 후 설치 진행

시뮬링크 개발 환경 및 자동 생성 코드 사용 방법 가이드



The screenshot shows a documentation page for 'AurixRacer'. The top navigation bar includes links for Home, AurixRacer Guide, Getting Started, User Guide, Connection Guide, and Simulink as a Programming Language. The current page is 'Simulink as a Programming Language', which has sub-links for 'Simulink as a programming language', '시작하는 질문', 'Objectives', 'References', 'Example Description', '추가적인 설명', and '마치며'. A sidebar on the left features a 'DigitalOcean' logo and a call-to-action: 'New: DigitalOcean Marketplace Deploy your favorite dev tools with 1-Click Apps.' Below the sidebar is a 'Previous' button.

Docs » AurixRacer Guide » Simulink as a Programming Language

Simulink as a programming language

시작하는 질문

- 제어 공학에서 시뮬레이션을 하고 분석하는 것을 열심히 배웠는데.... 이 때 만든 알고리즘을 인피니언레이서에도 활용하려면 어떻게 해야 할까?

학교에서 마이크로컨트롤러 수업을 들으면 하드웨어적인 입출력에 대한 처리에 대해서 주로 배우게 됩니다. 그리고 제어공학에서는 전달함수, 상태방정식 등 이론적인 내용을 바탕으로 제어알고리즘을 해석적으로, 시뮬레이션을 활용해서 설계하는 방법을 배우게 됩니다. 실제 제어시스템을 만들려면, 아니 지금 여러분이 만들고 있는 인피니언레이서에도 이 지식이 모두 결합되어야 합니다. 이때 위와 같은 질문을 하게 됩니다.

많은 엔지니어들이 노력해서 이 두 가지, 즉 제어공학과 마이크로컨트롤러 프로그래밍, 분야의 작업을 효율적으로 연결하는 방법을 만들어 냈습니다. 모델을 사용하여 제어기의 구조와 이득값 등을 시뮬레이션 단계에서 설계하고, 자동코드 생성도구를 활용하여 마이크로컨트롤러에서 실행할 수 있는 C 소스코드를 만들어냅니다. 전체 소프트웨어에서 이 부분은 주로 알고리즘 영역에 해당합니다. 이 알고리즘을 입출력을 담당하는 다른 부분과 연결하여 전체 소프트웨어를 만들어 낼 수 있습니다. 이와 같은 개발 방법을 모델기반의 개발 방법(MBDP:Model-Based Development Process)라고 부르고 있습니다. 복잡한 제어 알고리즘을 개발하는 경우에는 매우 효율적인 개발 방법입니다. 이 방법으로 알고리즘을 개발하면, 시뮬링크가 프로그래밍 언어의 역할을 하고 마이크로컨트롤러에서 실행되는 C 소스코드는 자동코드 생성도구가 만들어 내게 됩니다.

Objectives

- 모델기반 개발방법론을 이해하고
- 필요한 부분에 적용하여 활용할 수 있도록 한다.

New Development Platform (Optional)

Third-Party 제품 및 서비스

Search Third-Party Products

써드파티 제품 ▾

개요 | [Become a Partner](#) | [Search Products](#) | [Search Services](#)

Hitex AURIX Target for Simulink

Simulink Blockset for targeting control algorithms onto Infineon AURIX MCUs

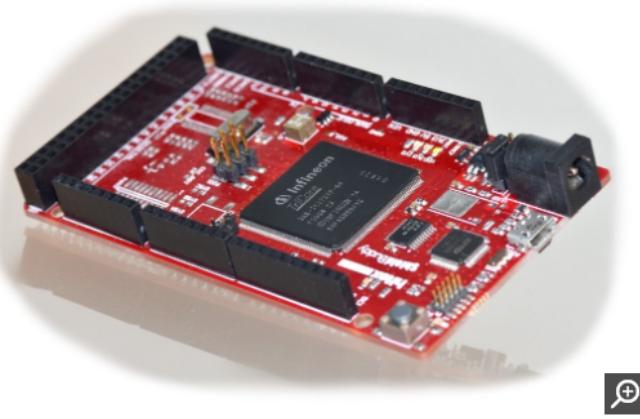
Highlights

- Enables Simulink® users to easily target Infineon AURIX™ MCUs with their models
- Supports MathWorks® code generation via Embedded Coder®
- Allows users to quickly move existing algorithms into AURIX™ applications
- Full support for the Hitex ShieldBuddy TC275
- Multicore operation supports full power of the AURIX TC275 MCU

Description

Infineon's AURIX™ family of microcontrollers serves the precise needs of the automotive industry in terms of performance and safety. The innovative multicore architecture, based on up to three independent 32-bit TriCore™ CPUs, has been designed to meet the highest safety standards, while increasing performance significantly. Using the AURIX platform, automotive developers will be able to control powertrain (including hybrid and electrical vehicles) and safety applications (such as steering, braking, airbag, and advanced driver assistance systems) with a single MCU platform.

Developing on AURIX can reduce MCU safety development time by 30%, while maintaining power consumption of a single-core microcontroller and leveraging a performance surplus of 50-100% for more functionality now and in the future. AURIX also protects IP



Required Products

- [MATLAB](#)
- [Simulink](#)
- [Embedded Coder](#)
- [MATLAB Coder](#)
- [Simulink Coder](#)

Recommended Products

- [Fixed-Point Designer](#)
- [Stateflow](#)

Platforms

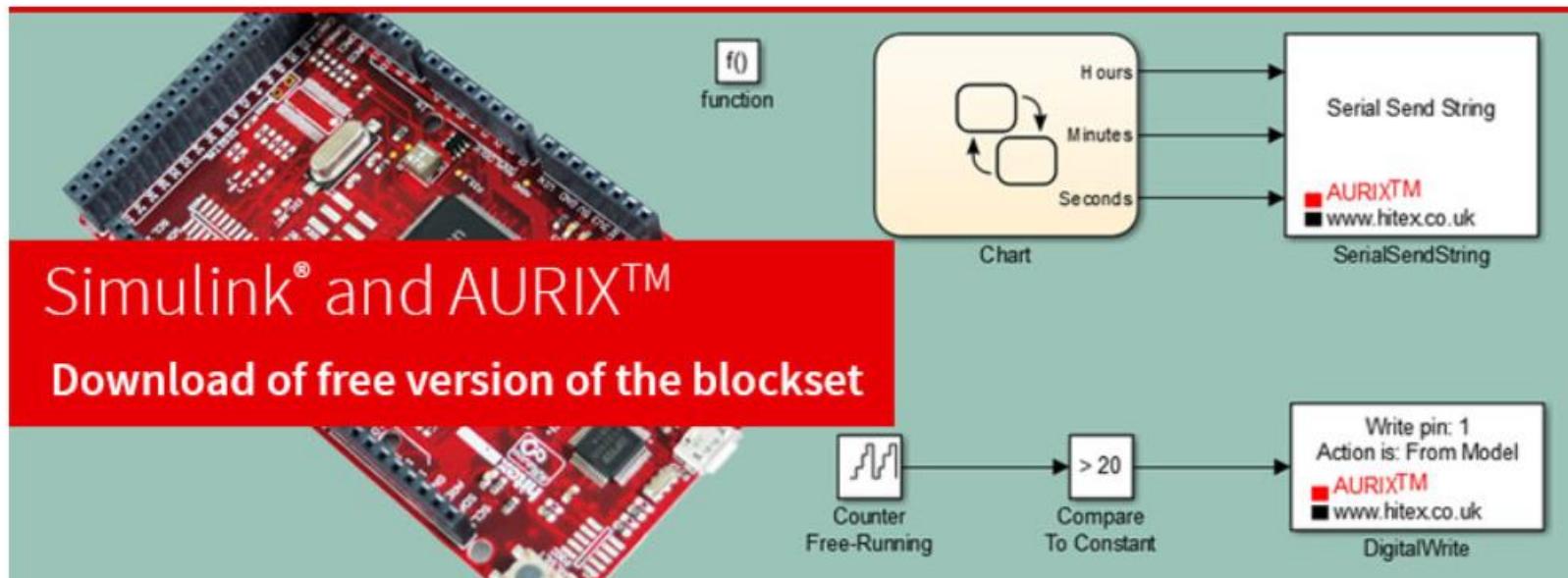
- Windows

Support

- Consulting
- E-mail
- System integration
- Telephone
- Training

Product Type

- Embedded Hardware - MCU, DSP, FPGA
- Embedded Software - Tools, IDE,



Simulink® and AURIX™: Download free version of blockset!

Simulink is well established as a modelling tool particularly for Automotive and Aerospace applications; Hitex (UK) Ltd. has now developed a free blockset that supports the ShieldBuddy TC275. This blockset will allow engineers to easily reuse and port existing control algorithm models to AURIX™ without the need for major source code rework. Multicore operation is also supported in the paid version to leverage the full power of the AURIX TC275. Learn more and download the Free blockset.

You will get an email with the download link after submission.

Email *

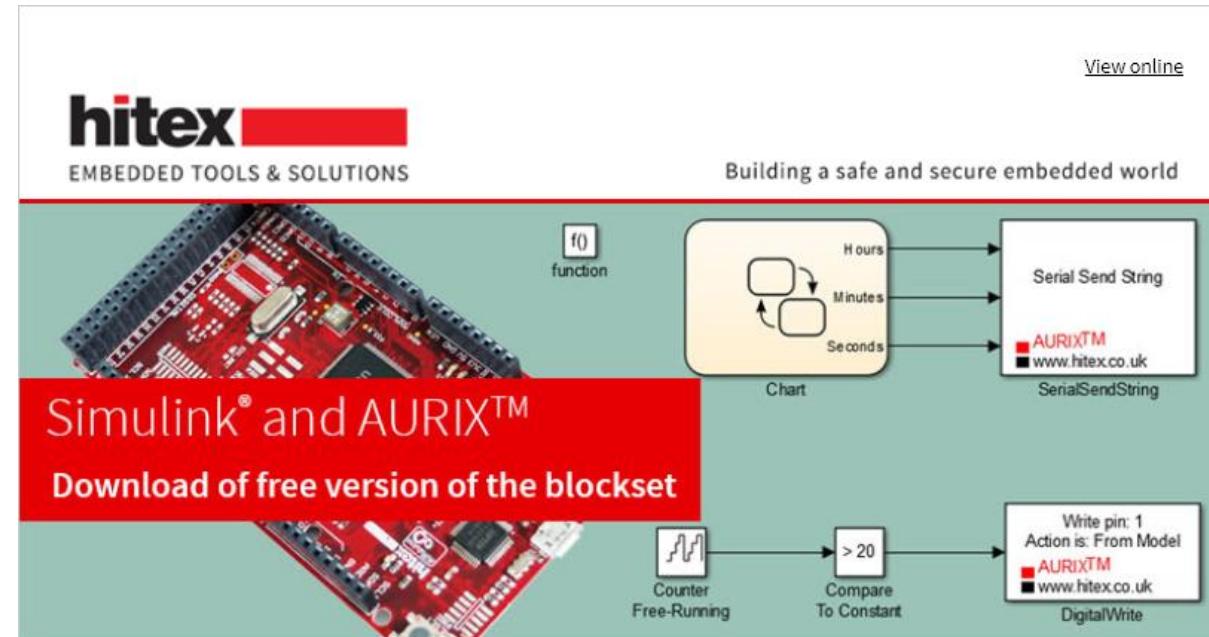
seongyon@gmail.com

1) 소프트웨어 및 라이센스
받을 이메일 계정 입력

Submit

2) 제출

제출한 메일 계정으로 다운로드 및 문서 자료 링크 수신



thank you for your interest in our free blockset for Simulink® and AURIX™. Please click on the download links below.

[Download AURIX Simulink Free Version](#)

[» Download AURIX Simulink
Free Version](#)

[Download AURIX Simulink Presentation](#)

[» Download AURIX Simulink
Presentation](#)

AURIX Simulink Blockset 설치 방법

- 메일을 통해 받은 zip 파일 압출 풀면 Aurix Simulink blockset 설치 폴더와 매뉴얼(2장 참조)



- 매뉴얼 2장 가이드라인대로 설치 한 후, MAC address 송부하여 메일로 라이센스 송부 받아야함

Firstly you will need to submit information regarding your PC's physical address by running **LicGetMAC.exe**. This application may be found in the License folder of your block installation. Running it will show a screen similar to this:

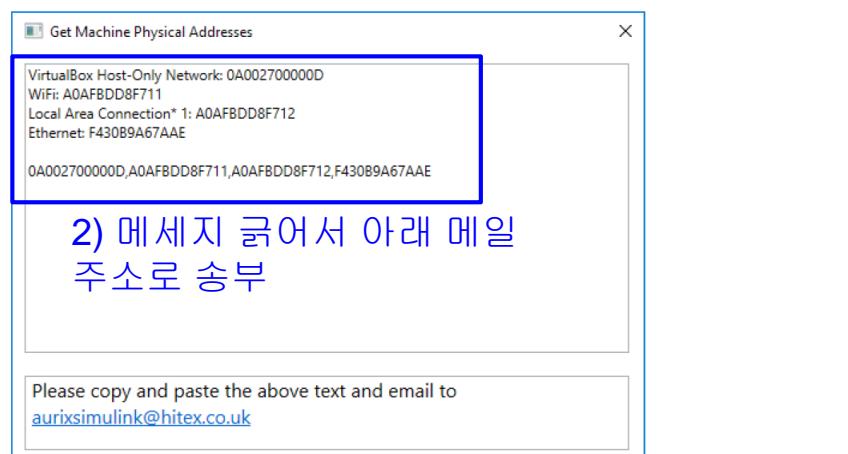
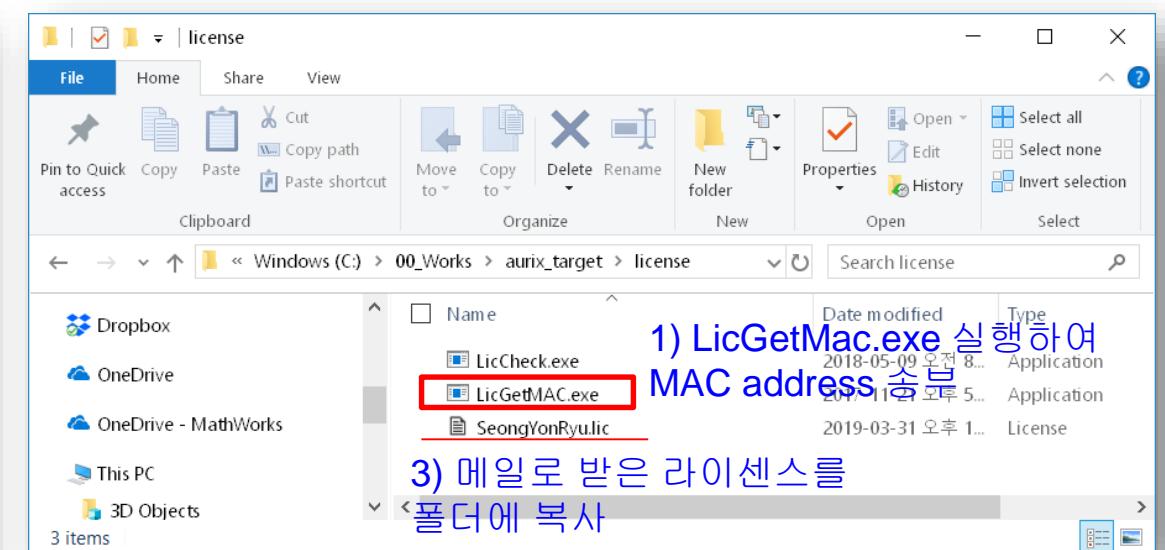
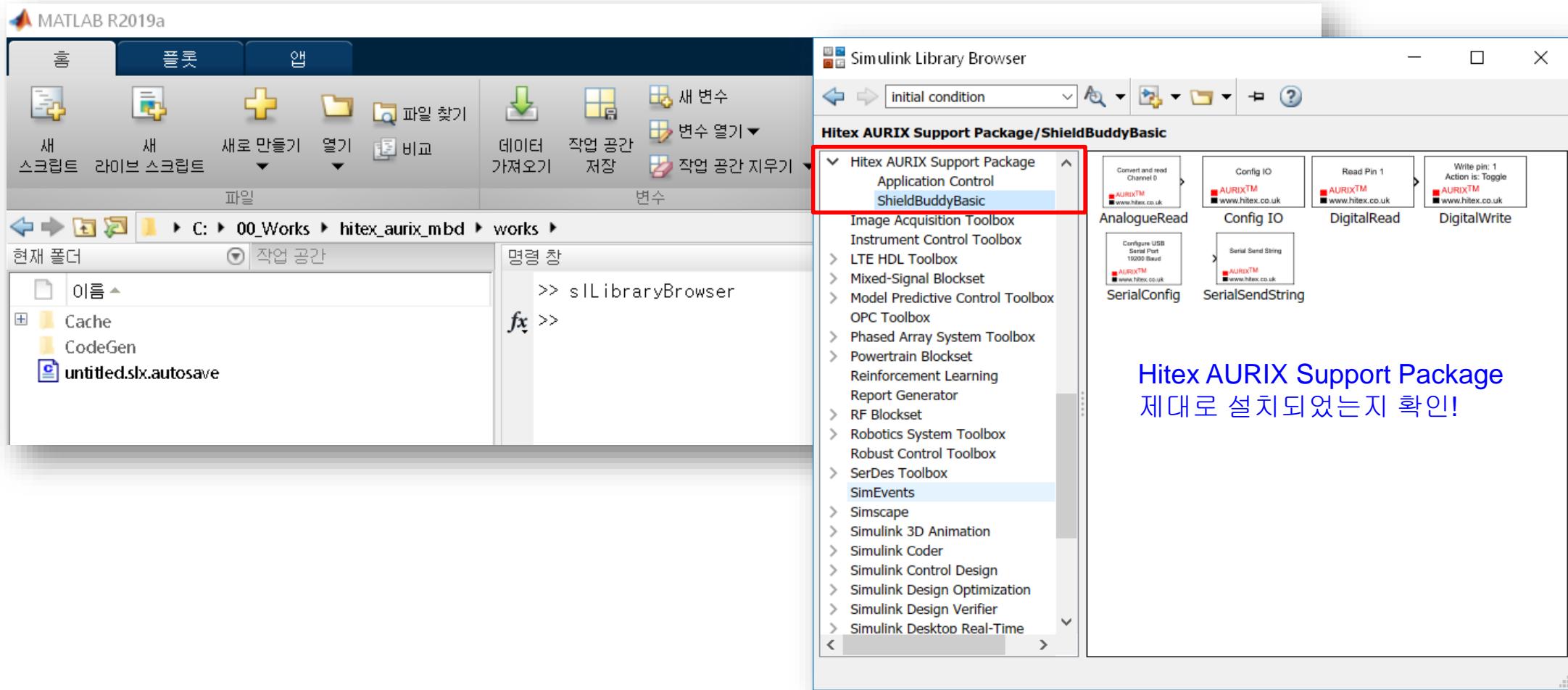


Figure 1 Acquiring Machine Physical Address



Simulink Library Browser 설치 확인

>> sLibraryBrowser



Demo & Simulink Setting

aurix_mbd_test - Simulink

모델 완성되면 빌드만 하면 끝!

Core0_Init

Core0_run

Initialise Function on Core 0
Pre Interrupt Enable
AURIX™
www.hitex.co.uk

Run Function on Core 0
Schedule: 0.005s
AURIX™
www.hitex.co.uk

function()

function()

Ready 146% FixedStepAuto

Configuration Parameters: aurix_mbd_test/Configuration (Active)

Search

Solver
Data Import/Export
Math and Data Types
▶ Diagnostics
Hardware Implementation
Model Referencing
Simulation Target
▼ Code Generation
Optimization

Target selection
System target file: aurix_sys_trgt.tlc

Language: C
Description: AURIX Target

Build process
 Generate code only

Configuration Parameters: aurix_mbd_test/Configuration (Active)

Search

AURIX Target Board Type: ShieldBuddy_TC275
 Generate debug information
 Generate map file
Optimization level: SIZE1
Additional compiler flags: <empty>
Additional linker flags: <empty>
Compiler Location (Folder containing the Bin folder): C:\HIGHTEC\toolchains\tricore\v4.9.1.0-infineon-2.0
 Select Compiler Location

Automatically download to target after Build
Memtool Location: C:\Program Files (x86)\Infineon\Memtool 4.7
 Select Memtool Location

OK Cancel Help Apply

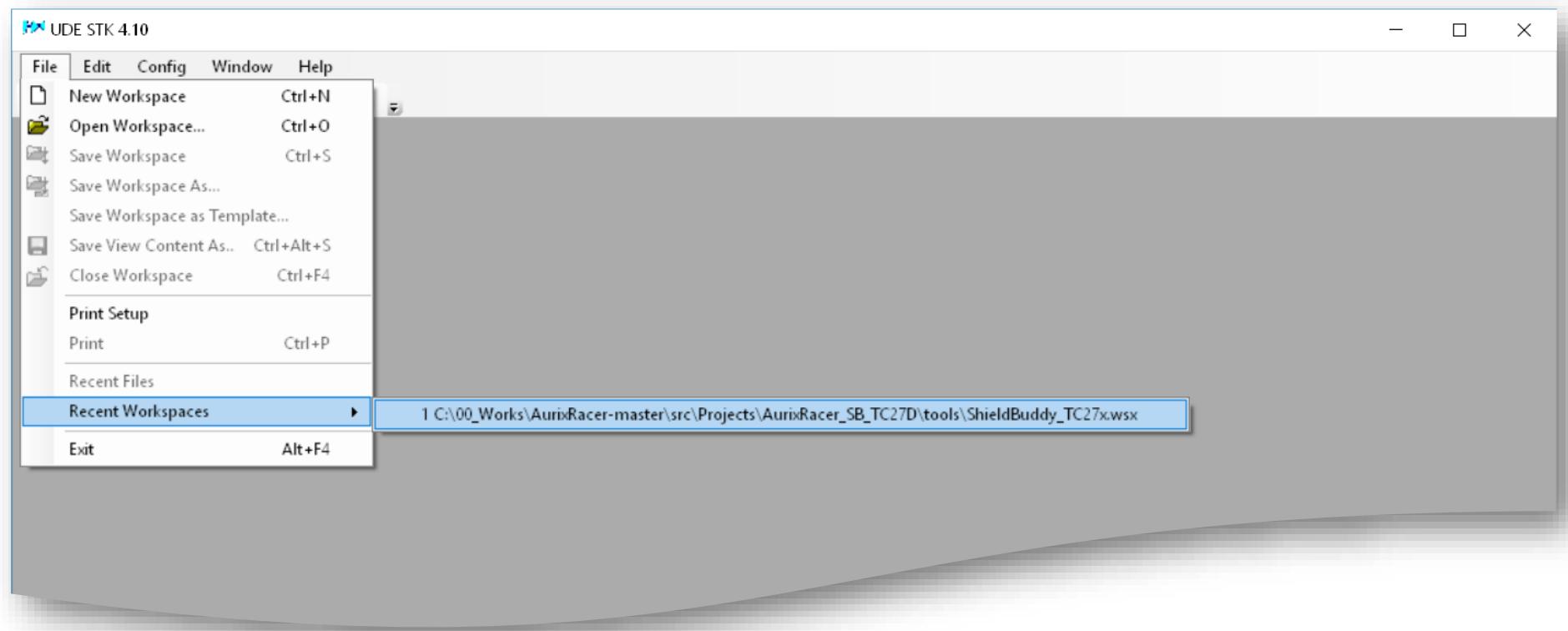
Simulink에서 바로 다운로드가 안되면 체크 해제하여 사용할 것

Red boxes highlight the 'function()' blocks in the model, the 'System target file' field, the 'Compiler Location' field, and the 'Automatically download to target after Build' checkbox.

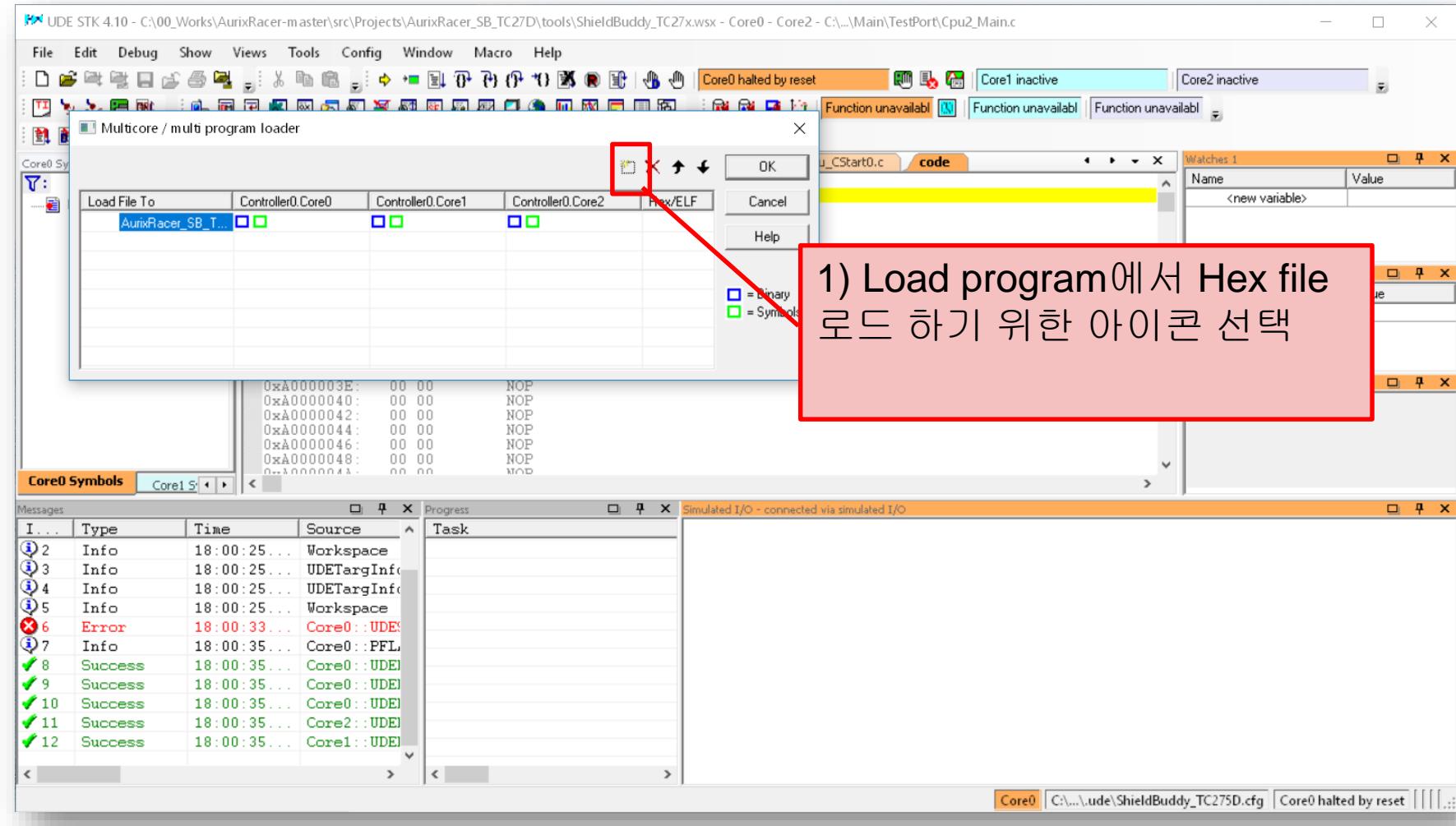
Orange arrows point from the 'Select Compiler Location' button to the 'Compiler Location' field, and from the 'Select Memtool Location' button to the 'Memtool Location' field.

AURIX Simulink Blockset 사용한 경우 Hex 다운로드

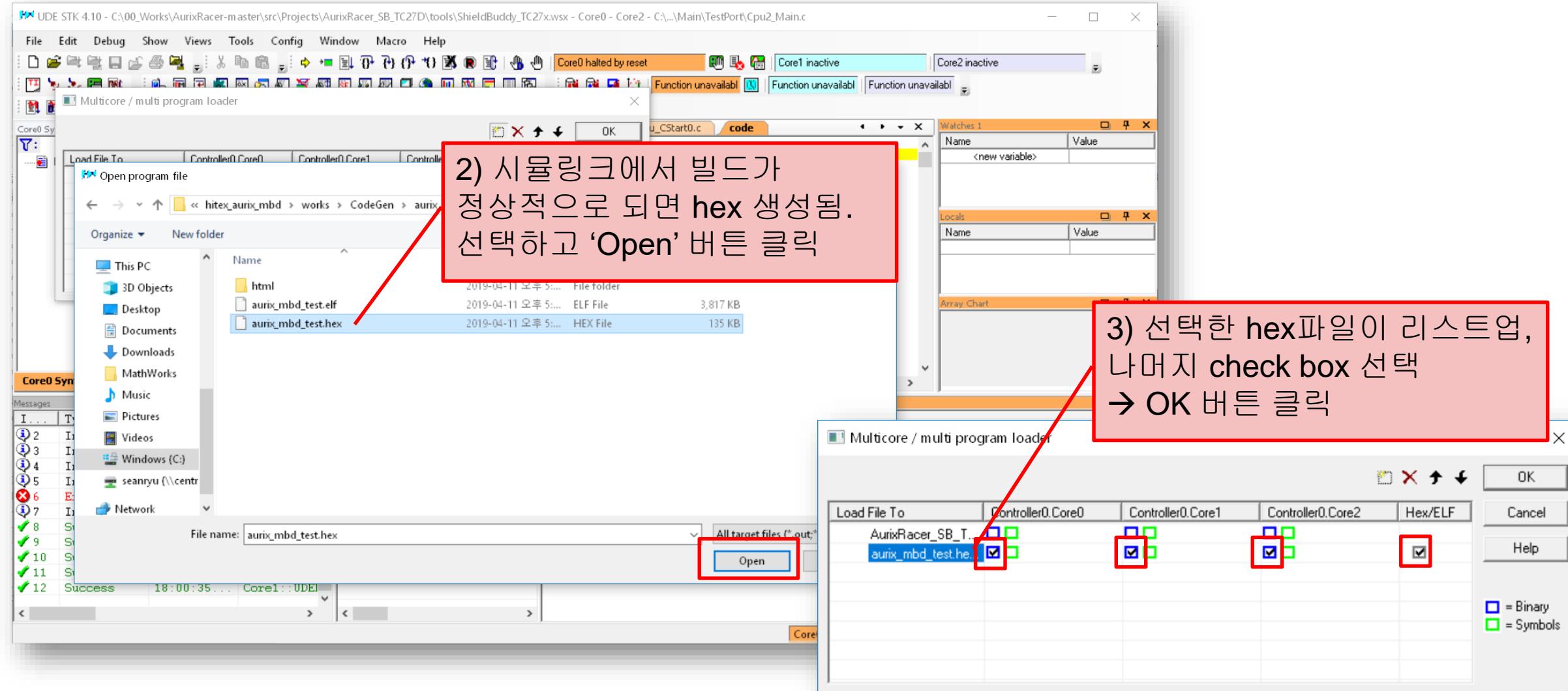
- UDEVisualPlatform.exe 실행
 - AurixRacer 사용위한 세팅 환경 그대로 사용



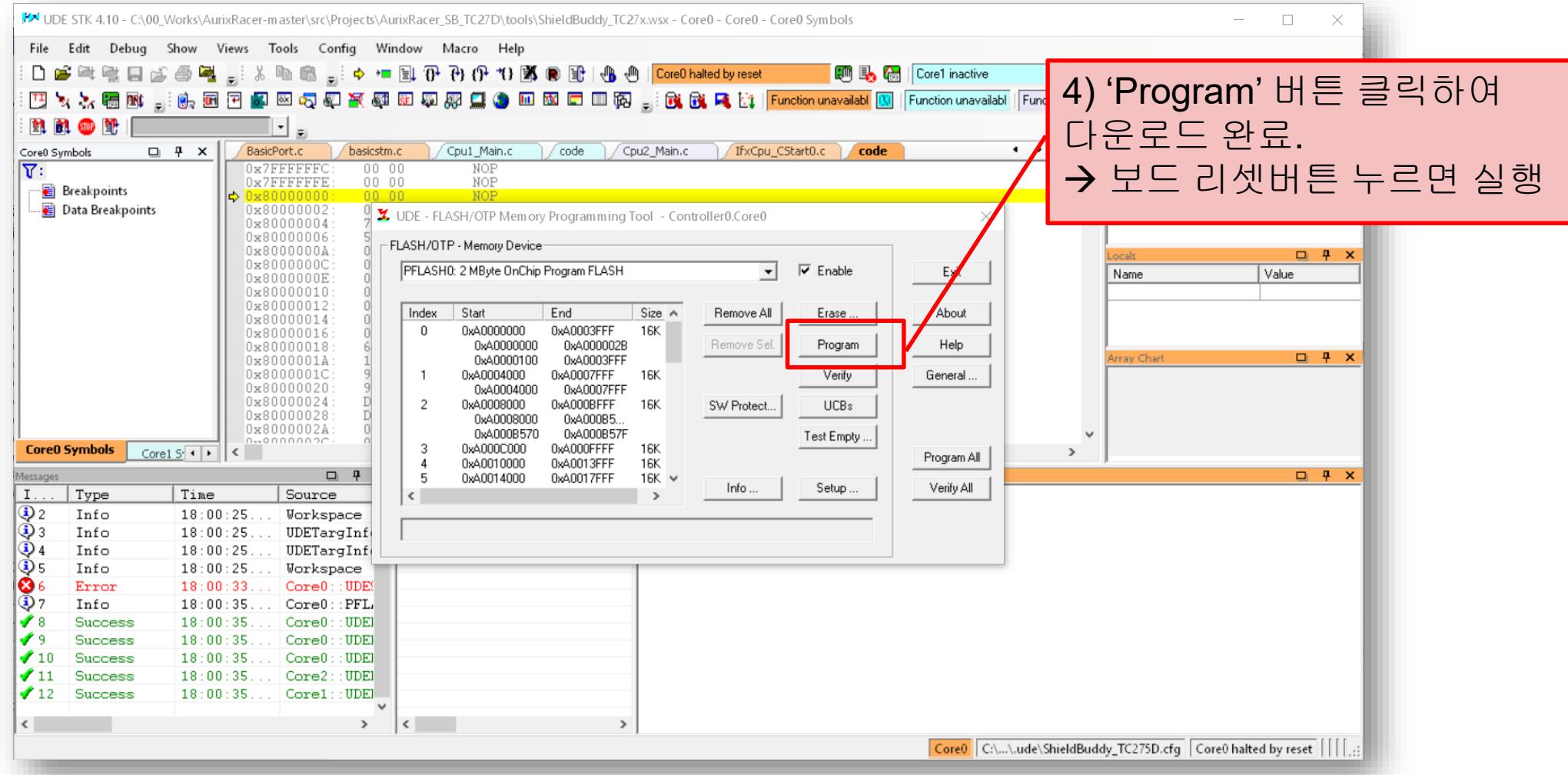
AURIX Simulink Blockset 사용한 경우 Hex 다운로드



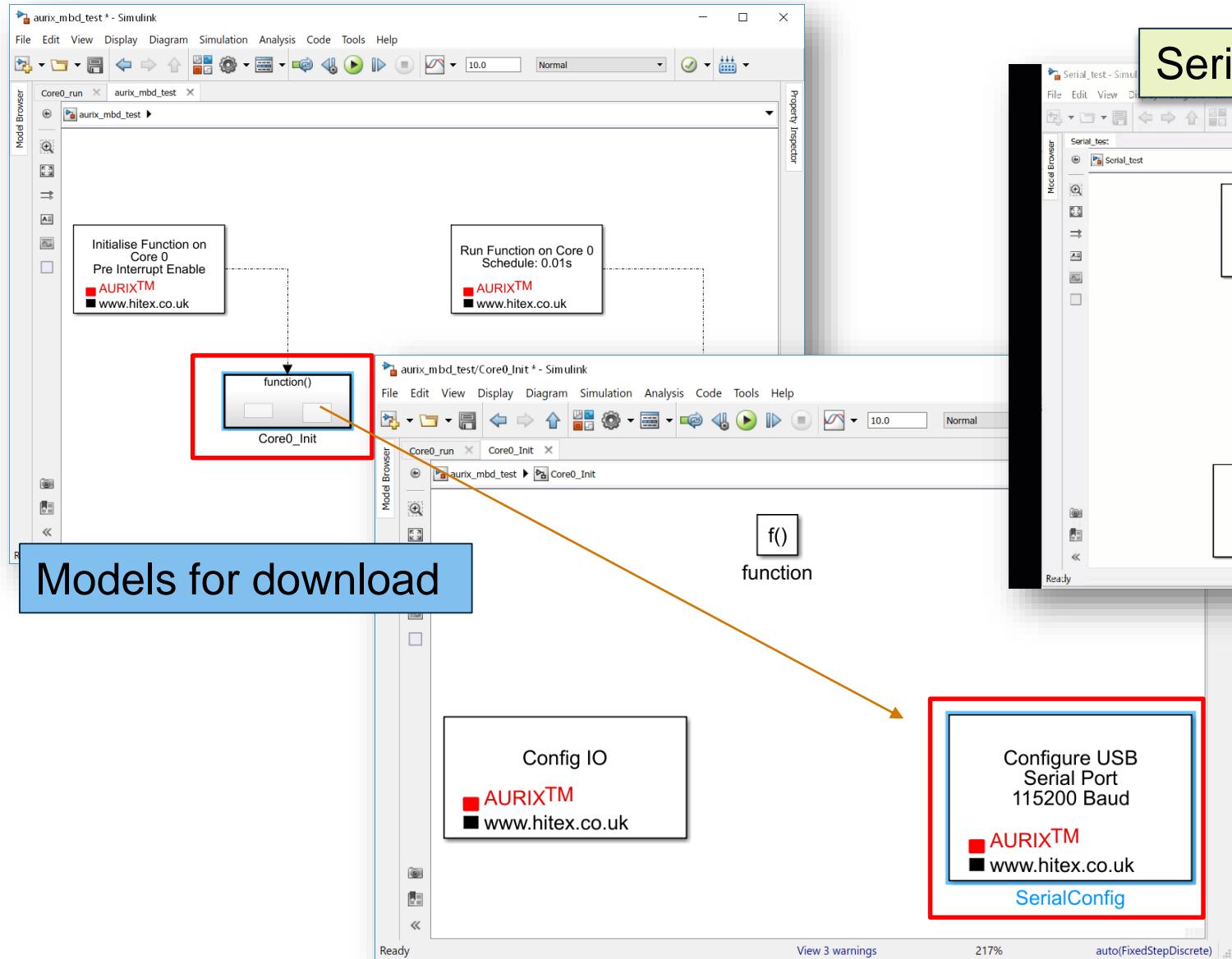
AURIX Simulink Blockset 사용한 경우 Hex 다운로드



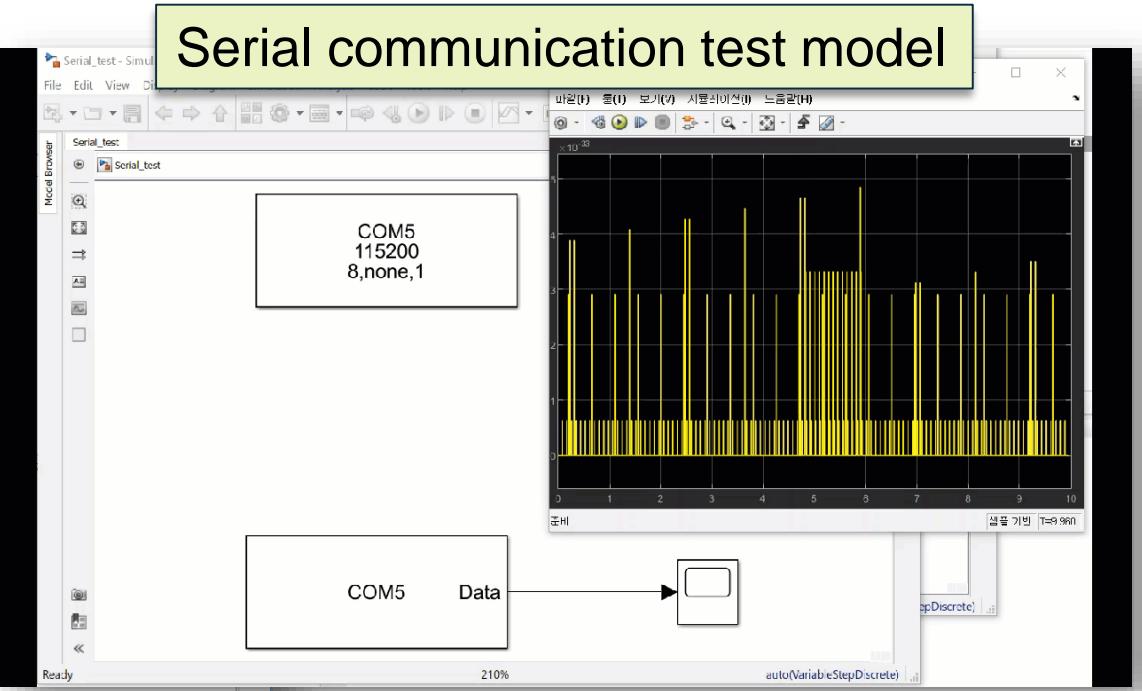
AURIX Simulink Blockset 사용한 경우 Hex 다운로드



Demo: Serial Communication



Serial communication test model



**Data logging using
Instrument Control Toolbox
ex. Data acquisition from
line scan camera testing**

Instrument Control Toolbox for Serial Communication

도움말

Interface-Based Instrument Communication - MathWorks 한국

Documentation All Examples Functions Blocks Apps R2019a 문서 검색

도움말 평가판 제품 업데이트 Translate This Page R2019a

도움말 목차 달기

« Documentation Home « Instrument Control Toolbox

Getting Started with Instrument Control Toolbox

Instrument Connection and Communication

Interface-Based Instrument Communication

- Interface-Based Communication
- Bluetooth Communication
- I2C Communication
- SPI Communication
- MODBUS Communication
- TCP/IP and UDP Interface
- Serial Port Interface
- GPIB Interface
- VISA Interface (Includes VXI, PXI, USB)
- Driver-Based Instrument Communication
- Direct Interface Communication in Simulink
- Supported Hardware
- Troubleshooting in Instrument Control Toolbox

Interface-Based Instrument Communication

Low-level protocols such as Bluetooth®, TCP/IP and UDP, I2C, SPI, MODBUS, Serial Port, GPIB, and VISA

With interface-based instrument communication, there are common steps that comprise the basic workflow. You create an object, open a connection, write data, and disconnect and clean up. For an outline of the basic steps, see [Interface-Based Communication](#).

Frequently Viewed Topics

- Configuring Bluetooth Communication
- Transmitting Data Over the Bluetooth Interface
- Create a TCP/IP Object
- VISA Overview

Interface-Based Communication

Communication using interfaces

Bluetooth Communication

Communication using the Bluetooth interface

I2C Communication

Communication using the I2C interface

SPI Communication

Communication using the SPI interface

MODBUS Communication

Communication using the MODBUS interface

TCP/IP and UDP Interface

Communication using the TCP/IP clients and servers or UDP protocol

Serial Port Interface

Communication using the serial port interface

GPIB Interface

Communicate with instruments that support the GPIB interface

VISA Interface (Includes VXI, PXI, USB)

Communicate with instruments that support the VISA standard

Simulink Library Browser

Instrument Control Toolbox

- Communications Toolbox HDL Support
- Computer Vision Toolbox
- Control System Toolbox
- Data Acquisition Toolbox
- Deep Learning Toolbox
- DSP System Toolbox
- DSP System Toolbox HDL Support
- Embedded Coder
- Fuzzy Logic Toolbox
- HDL Coder
- HDL Verifier
- Hitex AURIX Support Package
- Image Acquisition Toolbox
- Instrument Control Toolbox
- LTE HDL Toolbox
- Mixed-Signal Blockset
- Model Predictive Control Toolbox
- OPC Toolbox
- Phased Array System Toolbox
- Powertrain Blockset
- Reinforcement Learning
- Report Generator
- RF Blockset
- Robotics System Toolbox
- Robust Control Toolbox

Query Instrument

Serial Configuration

Serial Receive Data → Data Serial Send

Serial Receive Data → Data Serial Send

TCP/IP Client Receive Data → Data TCP/IP Client Send

TCP/IP Receive Data → Data TCP/IP Send

To Instrument

UDP Receive Data → Data UDP Send

UDP Receive Data → Data UDP Send

UDP Send

DSP System Toolbox

Design and simulate streaming signal processing systems

DSP System Toolbox™ provides algorithms, apps, and scopes for designing, simulating, and analyzing signal processing systems in MATLAB® and Simulink®. You can model real-time DSP systems for communications, radar, audio, medical devices, IoT, and other applications.

With DSP System Toolbox you can design and analyze FIR, IIR, multirate, multistage, and adaptive filters. You can stream signals from variables, data files, and network devices for system development and verification. The Time Scope, Spectrum Analyzer, and Logic Analyzer let you dynamically visualize and measure streaming signals. For desktop prototyping and deployment to embedded processors, including ARM® Cortex® architectures, the toolbox supports C/C++ code generation. It also supports bit-accurate fixed-point modeling and HDL code generation from filters, FFT, IFFT, and other algorithms.

Algorithms are available as MATLAB functions, System objects, and Simulink blocks.

Getting Started

Learn the basics of DSP System Toolbox

Signal Generation, Manipulation, and Analysis

Create, import, export, display, and manage signals

Filter Design and Analysis

FIR, IIR, frequency transformations

Filter Implementation

Single-rate, multirate, and adaptive filters

Transforms and Spectral Analysis

FFT, DCT, spectral analysis, linear prediction

Statistics and Linear Algebra

Measurements, statistics, matrix math, linear algebra

Fixed-Point Design

Floating-point to fixed-point conversion, fixed-point algorithm design.

Code Generation

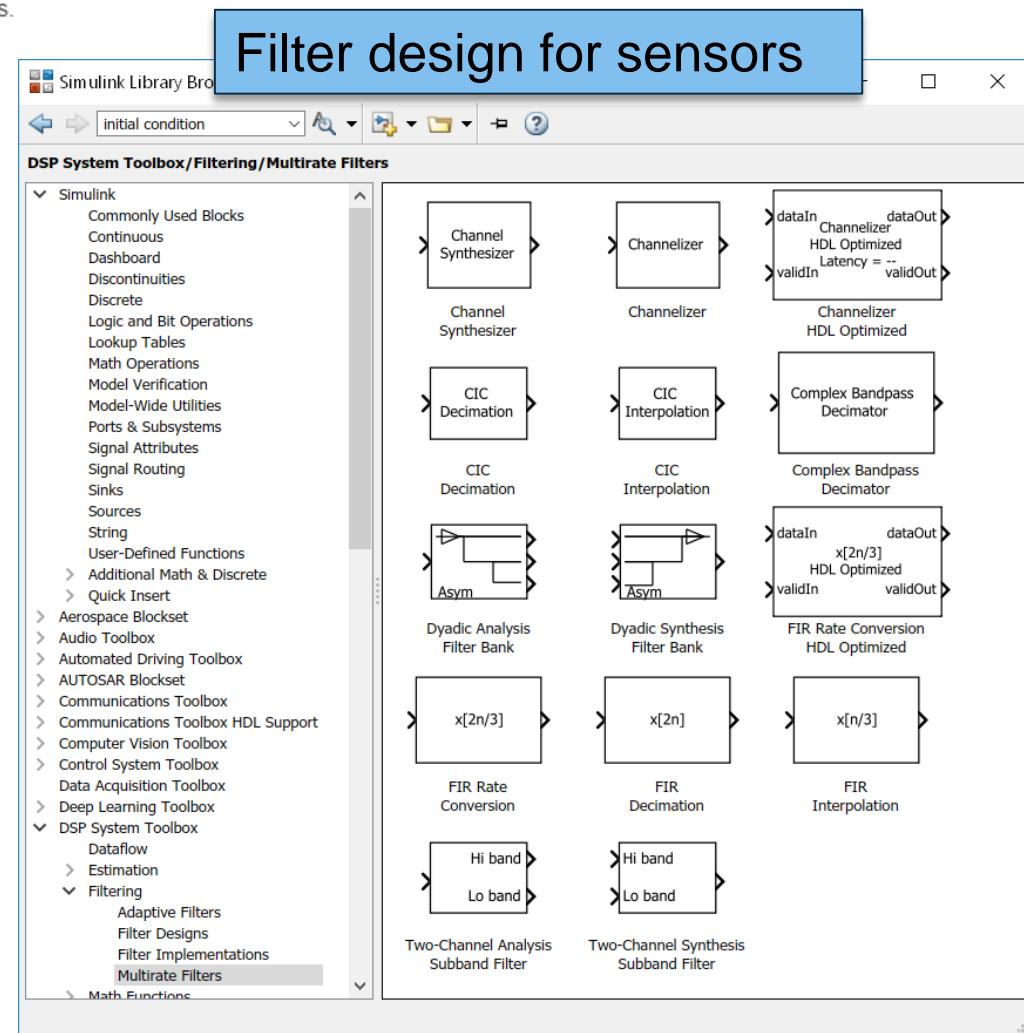
Simulation acceleration, code generation, optimization for ARM Cortex-M Processors and ARM Cortex-A Processors

Applications

Simulate radar, communication, and biomedical systems

Supported Hardware

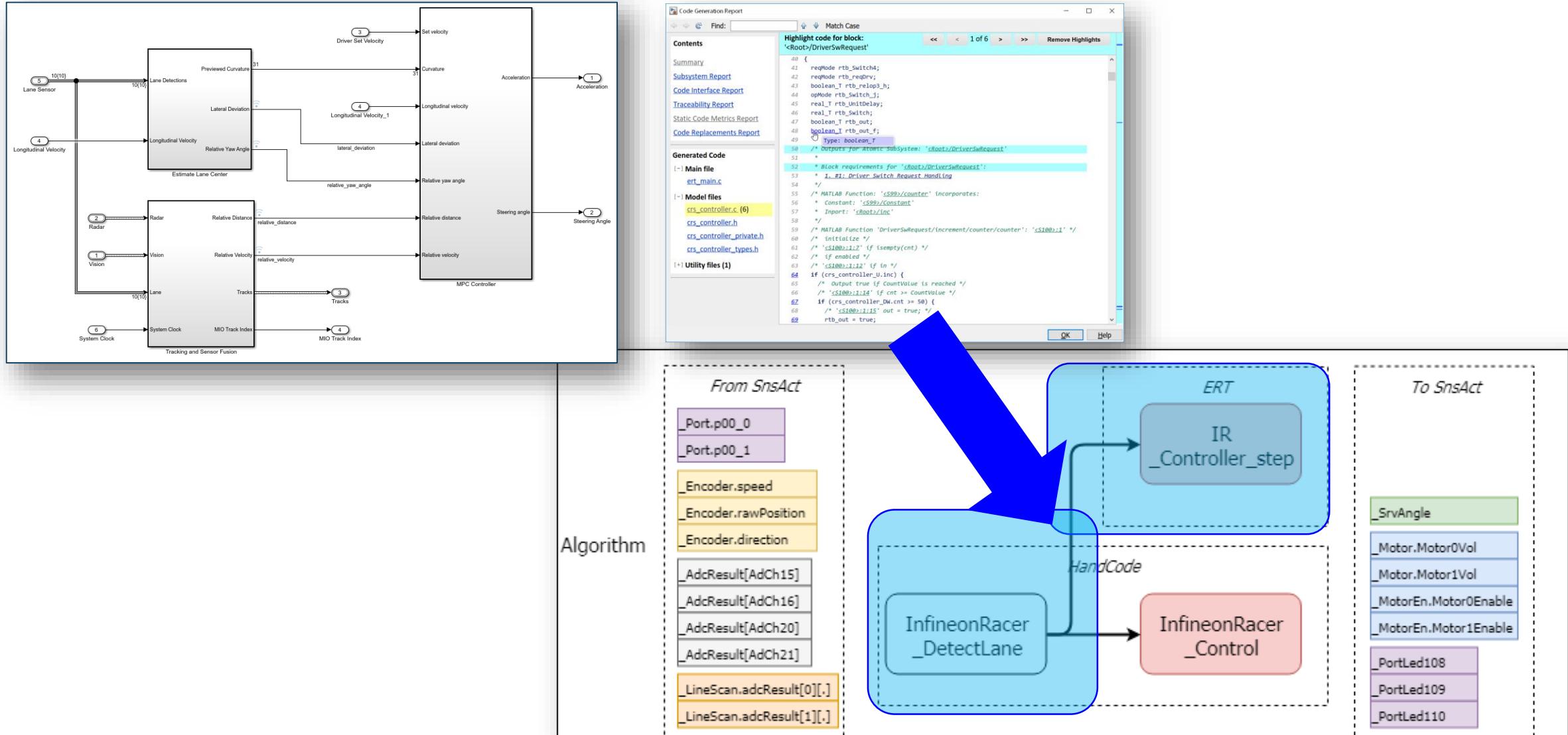
Support for third-party hardware



Getting Started from Simulink

- AurixRacer Platform

Remind from Software Integration...



Simulink Resource for SMCC

[Overview](#)

Repositories 2

Projects 0

Stars 0

Followers 0

Following 0

Pinned

Customize your pins

 AurixRacer_Simulink

Handed code와 Simulink를 이용한 스마트카 경진대회용 소프트웨어 개발 프로젝트

 C HitexAurixSimulink

HITEX Aurix Simulink Blockset 을 이용하여 Simulink로만 개발하는 경우 사용

 HTML

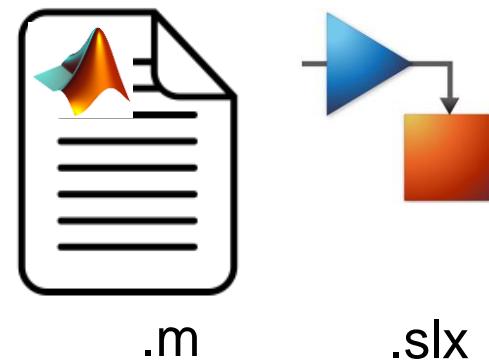
From <https://github.com/realsosy/AurixRacer/>
And modified for Simulink modeling in R2019a

Newly created for
HITEX Aurix Simulink Blockset



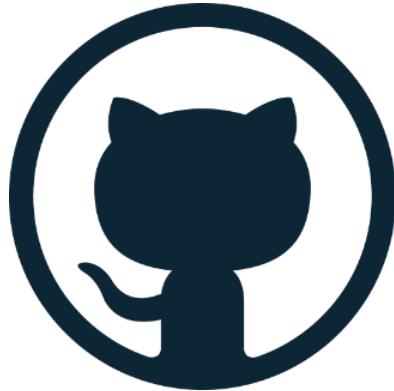
Successful collaborative development requires...

1. Same source code, tests, doc, requirements, ...



Successful collaborative development requires...

1. Same source code, tests, doc, requirements, ...
2. Integration with source control



Successful collaborative development requires...

1. Same source code, tests, doc, requirements, ...
2. Integration with source control
3. Consistent, shared environment

MATLAB Path

Workspace Data

C Compiler

Apps & Toolboxes

...

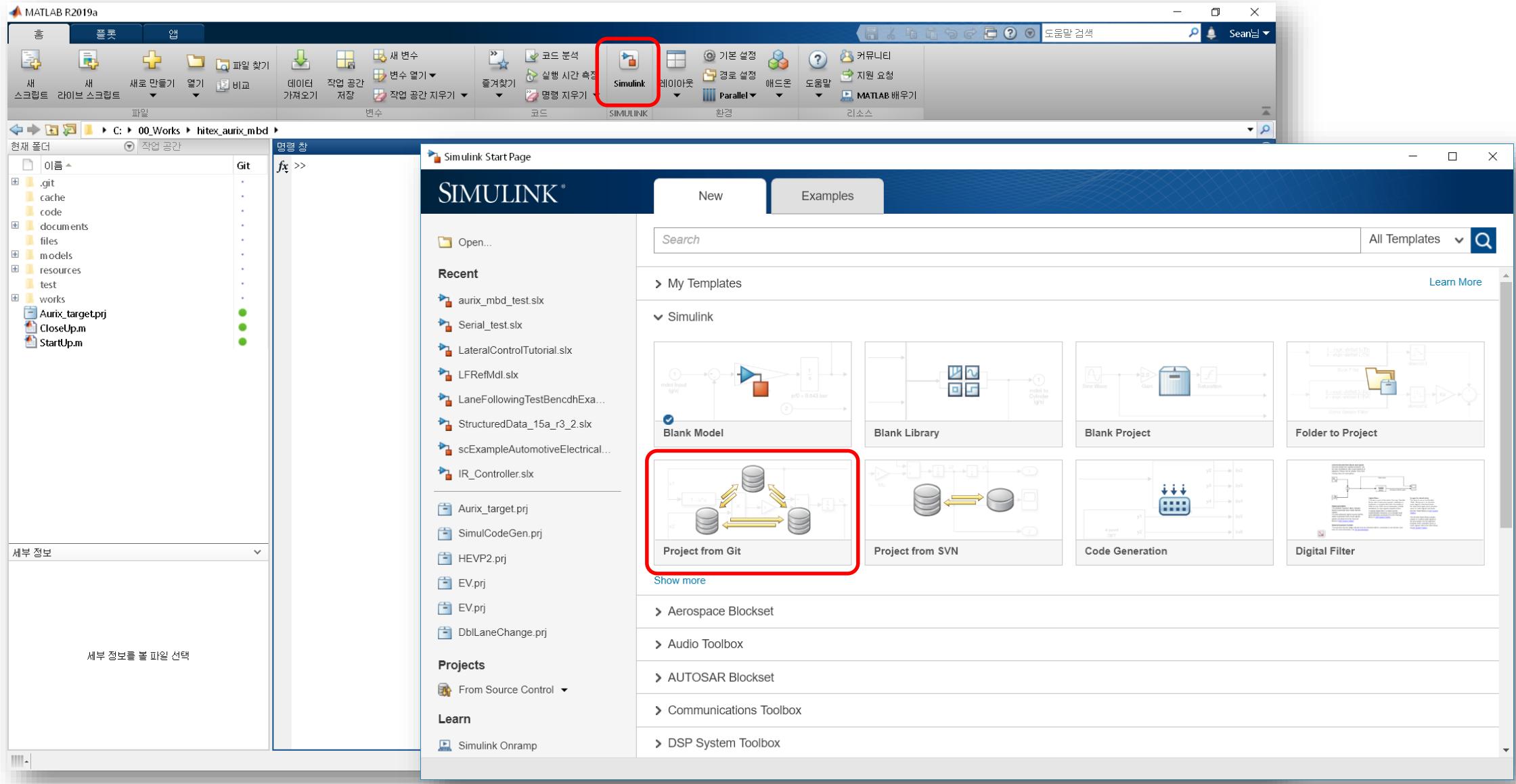
Successful collaborative development requires...

1. Same source code, tests, doc, requirements, ...
2. Integration with source control
3. Consistent, shared environment

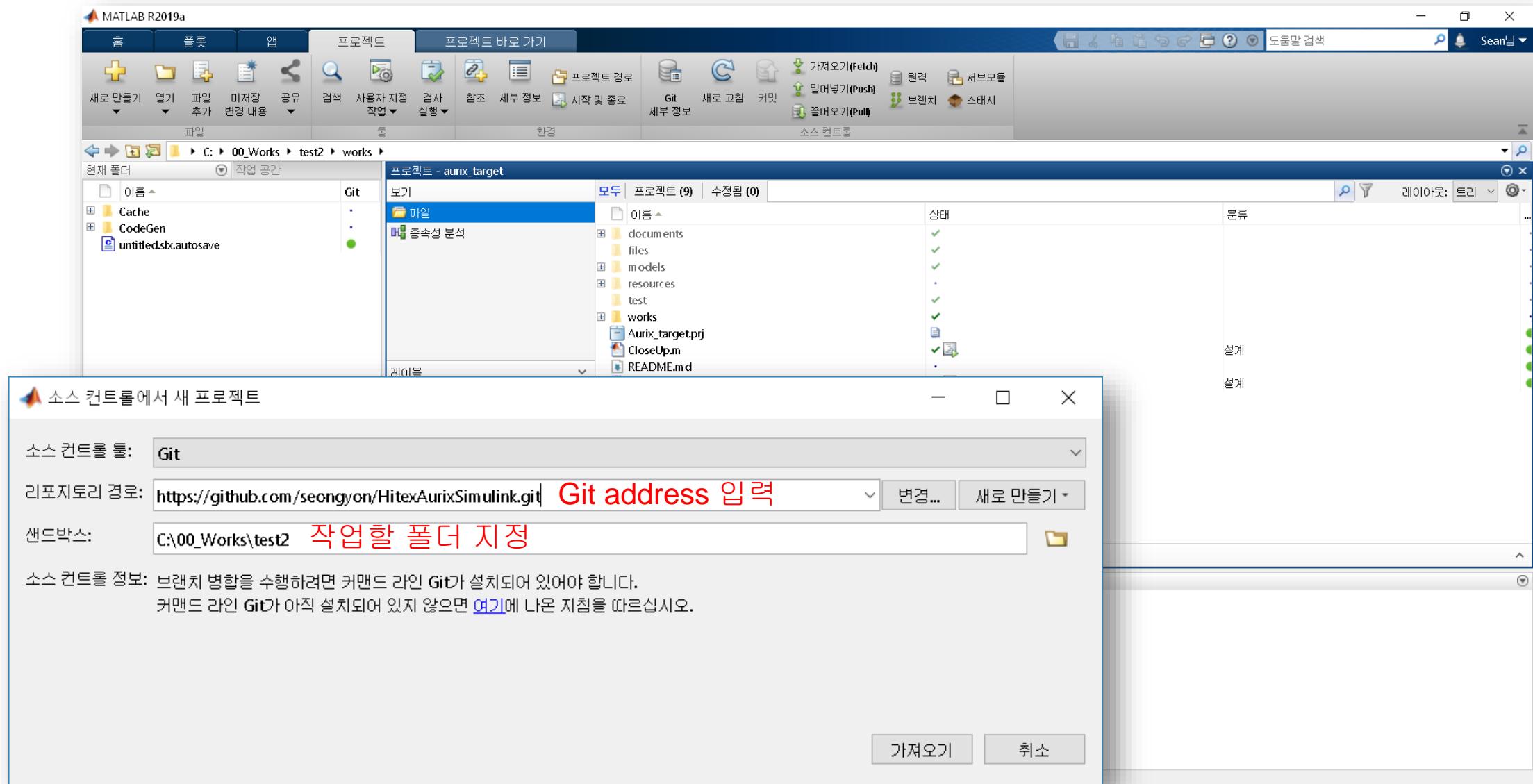


“Just works”

Let's Practice...



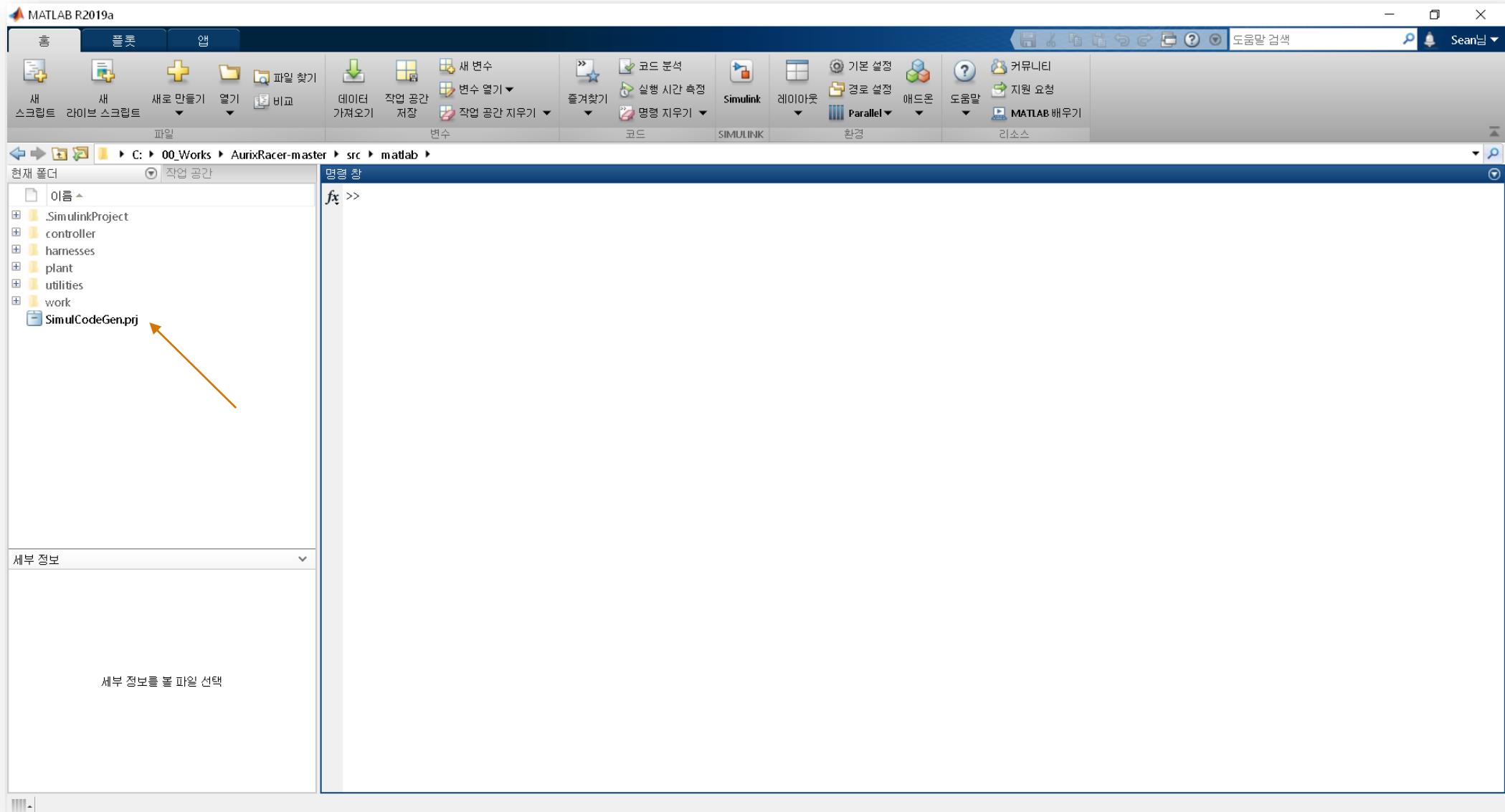
Project from Git: HITEX Aurix Simulink Blockset



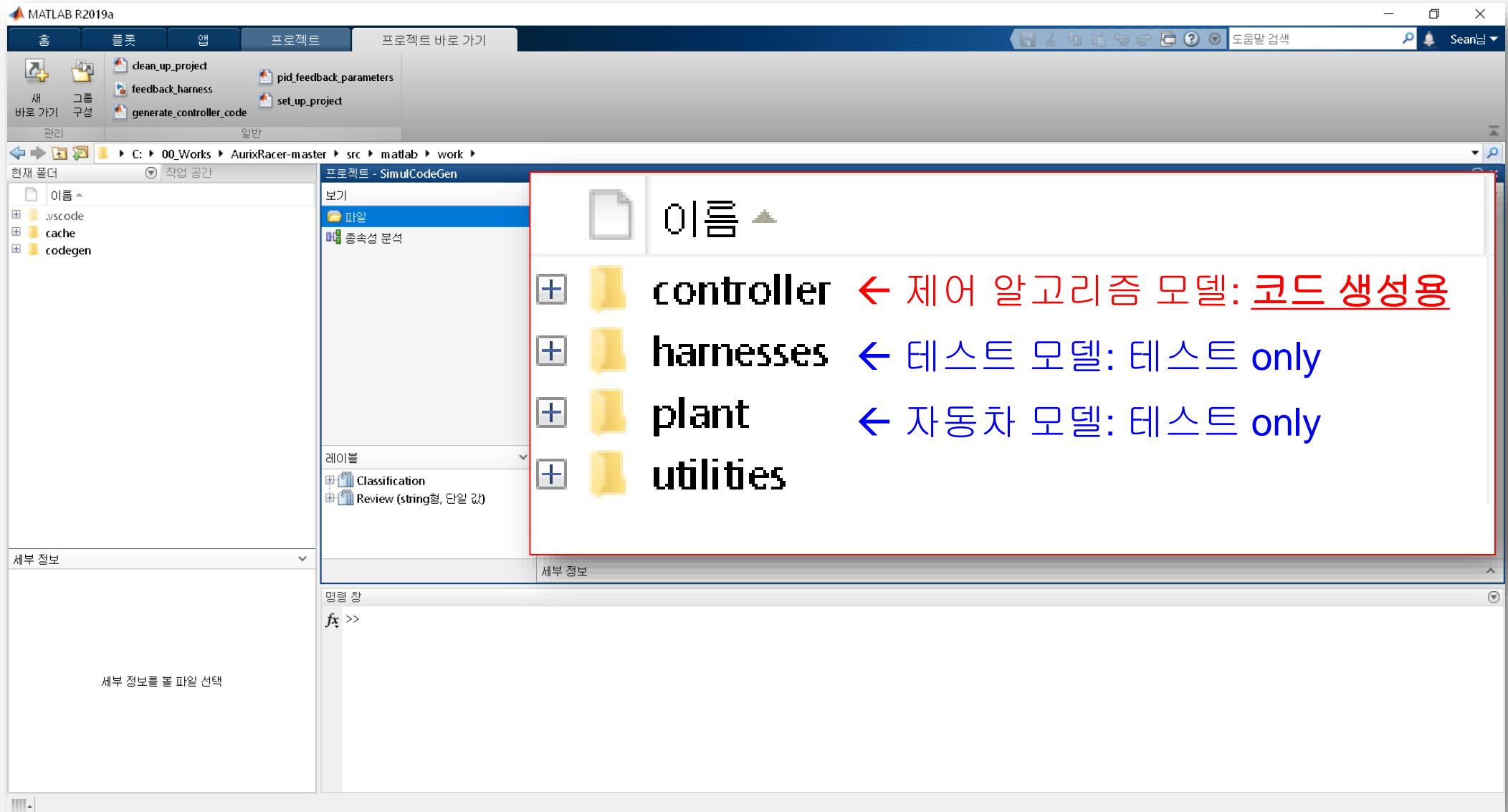
<https://github.com/seongyon/HitexAurixSimulink.git>

AurixRacer for Simulink

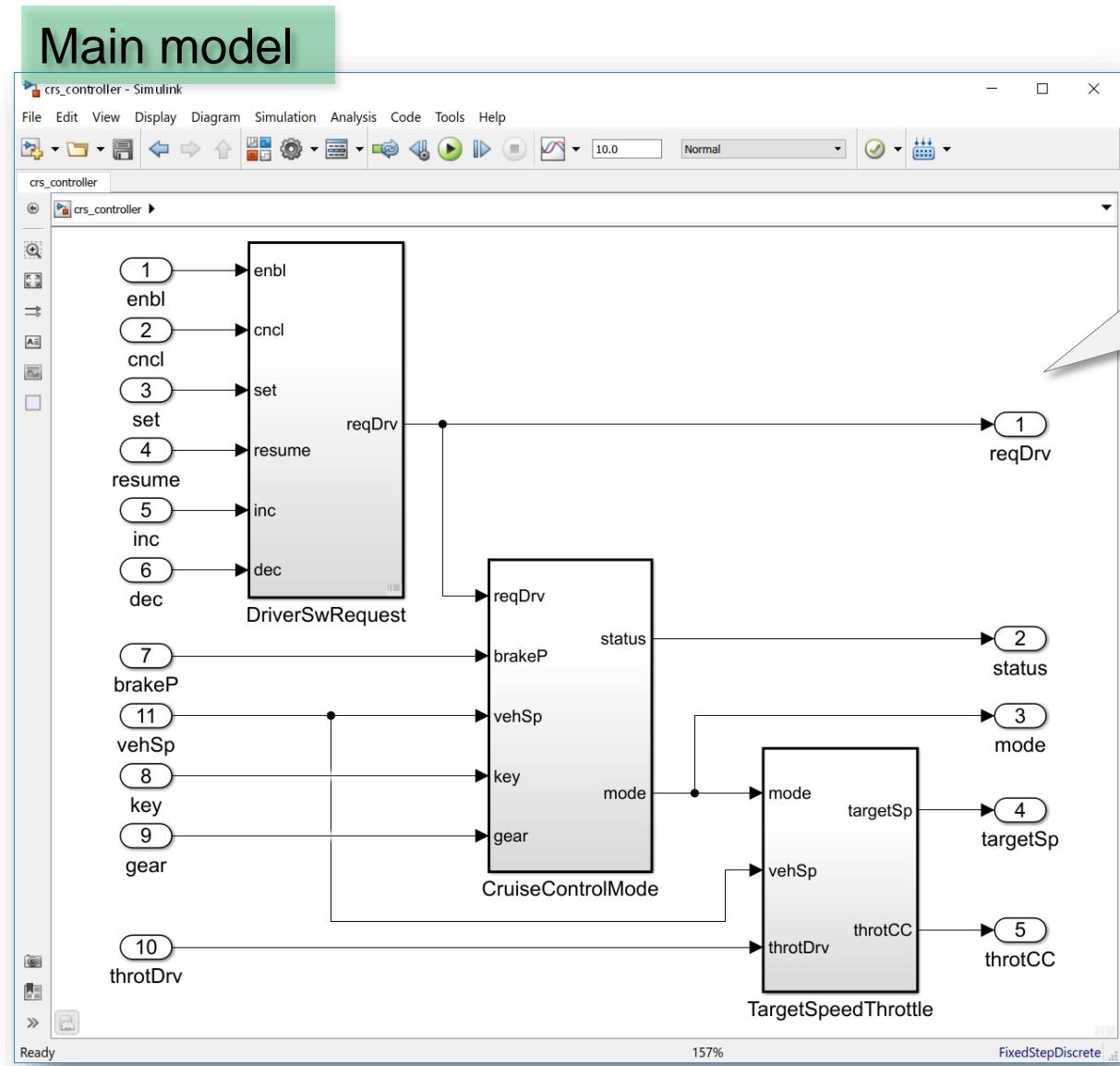
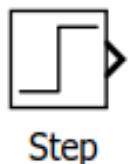
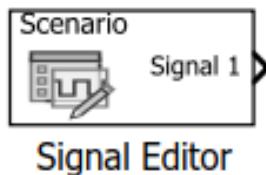
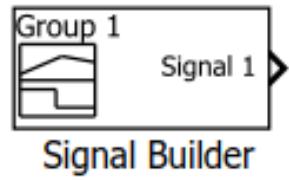
Starting Simulink Project



Folder Structure



How to Test Your Model...?



Do you
need to
modify it for
testing?



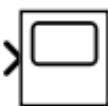
Gauge



Rotary Switch



Slider



Scope



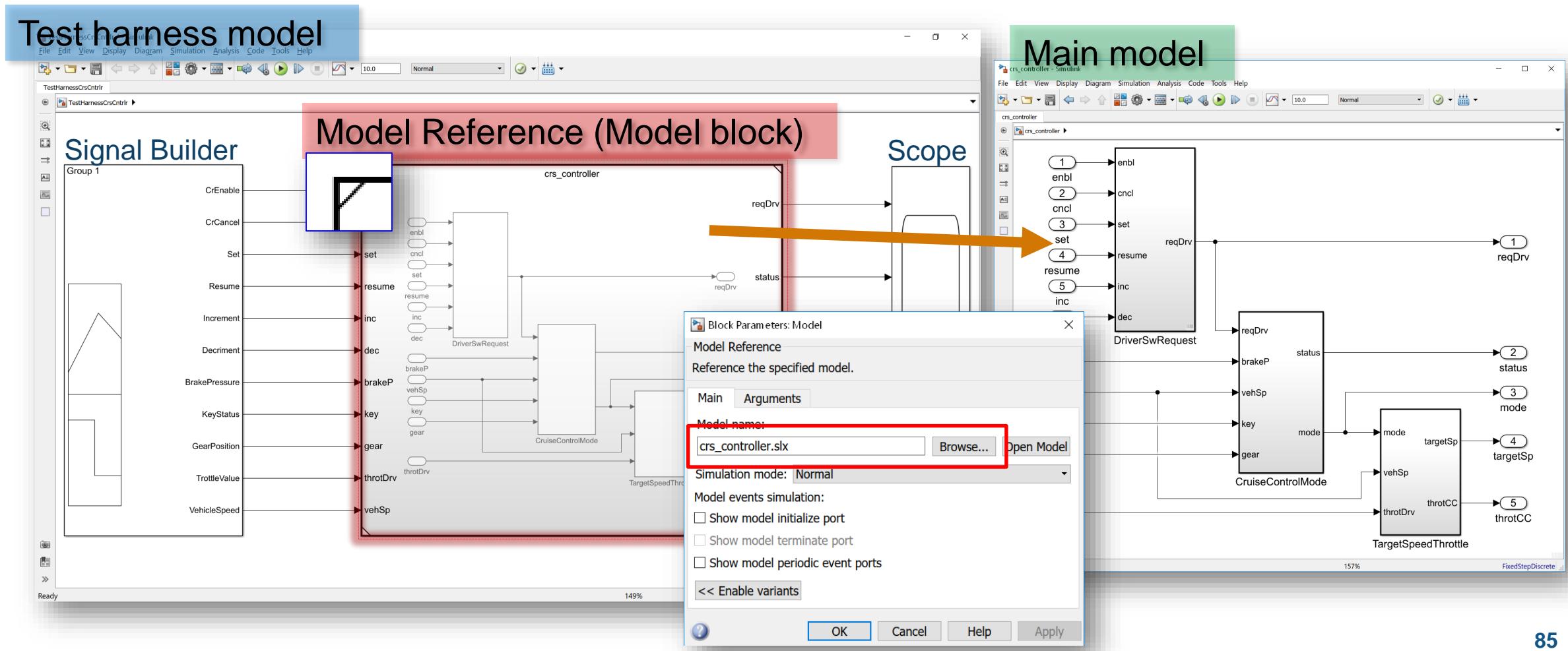
Test Condition



Verification Subsystem

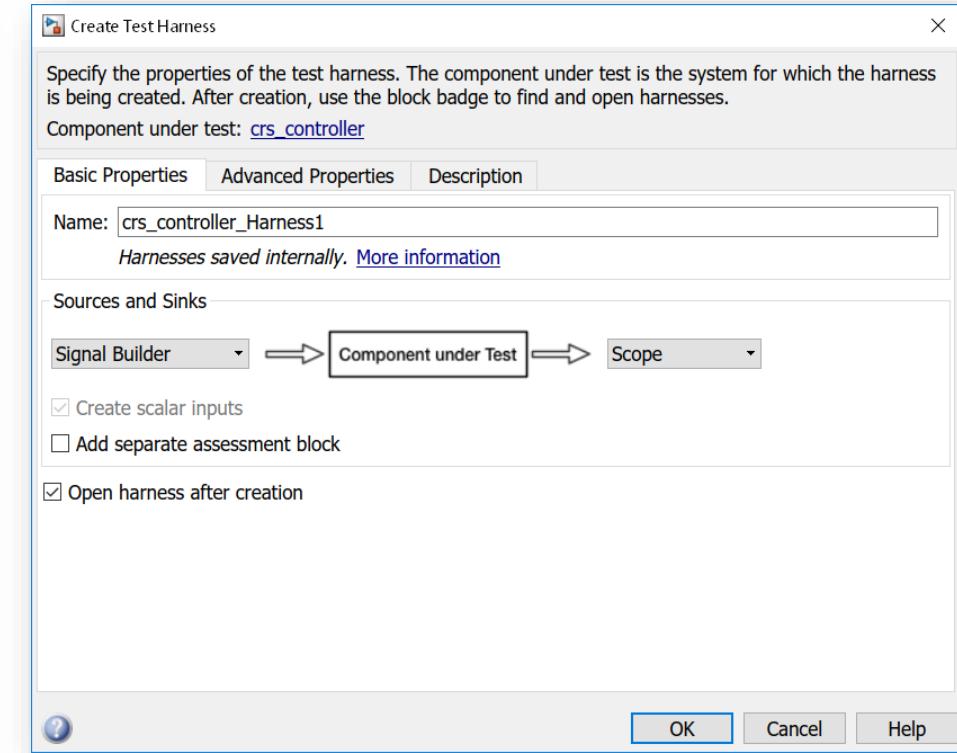
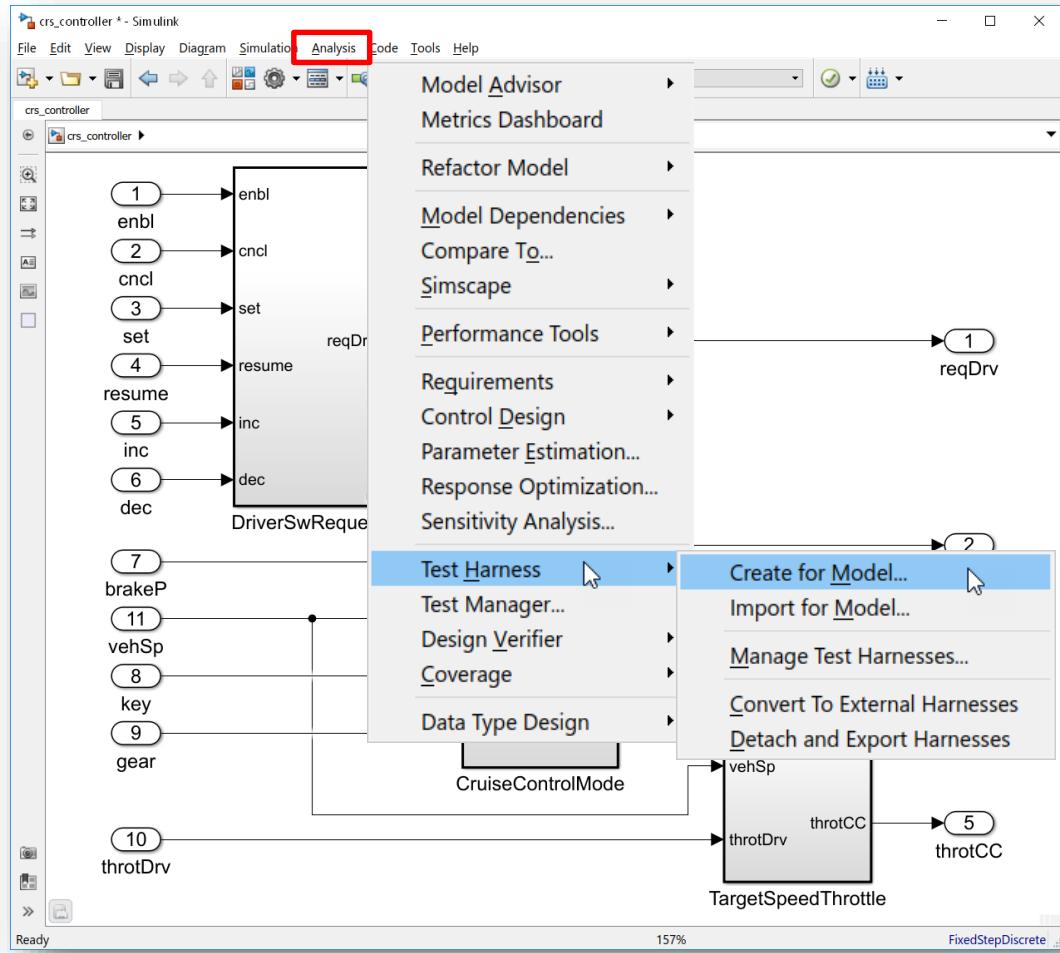
Building Test Harness Model using Model Reference

- Separated model not for code generation but only for testing



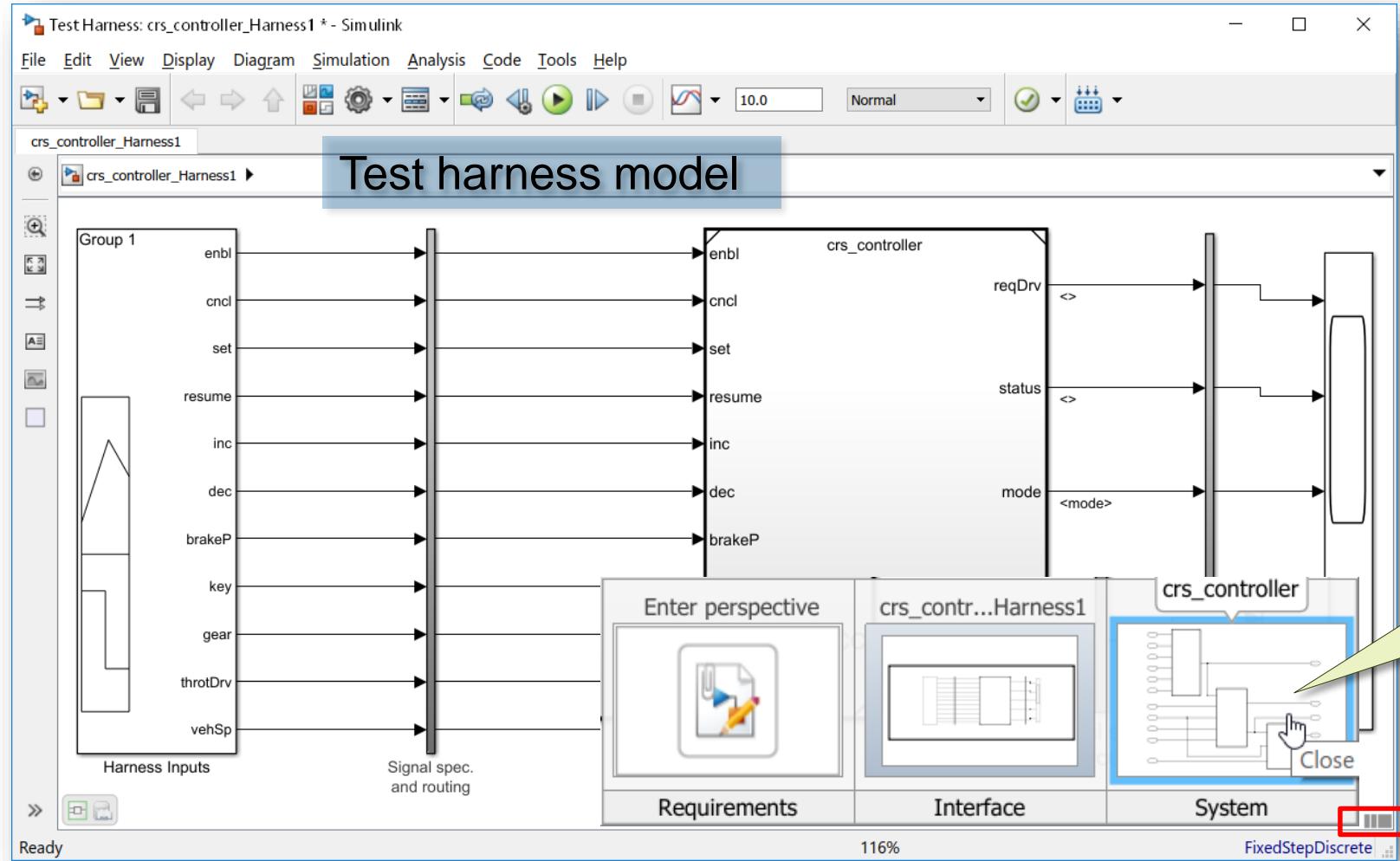
Test Harness Model

- Test harness model can be included in your original models



Test Harness Model

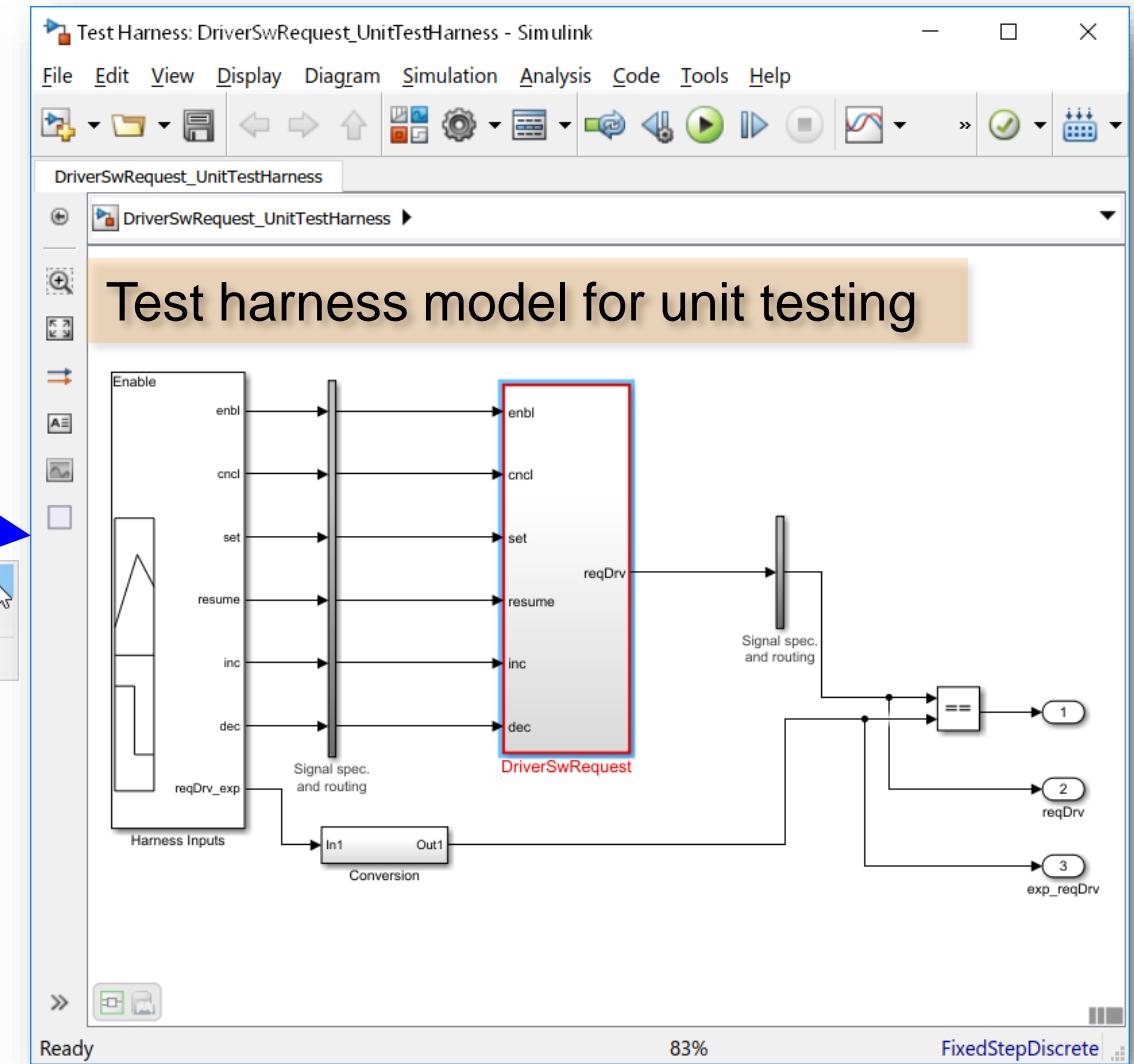
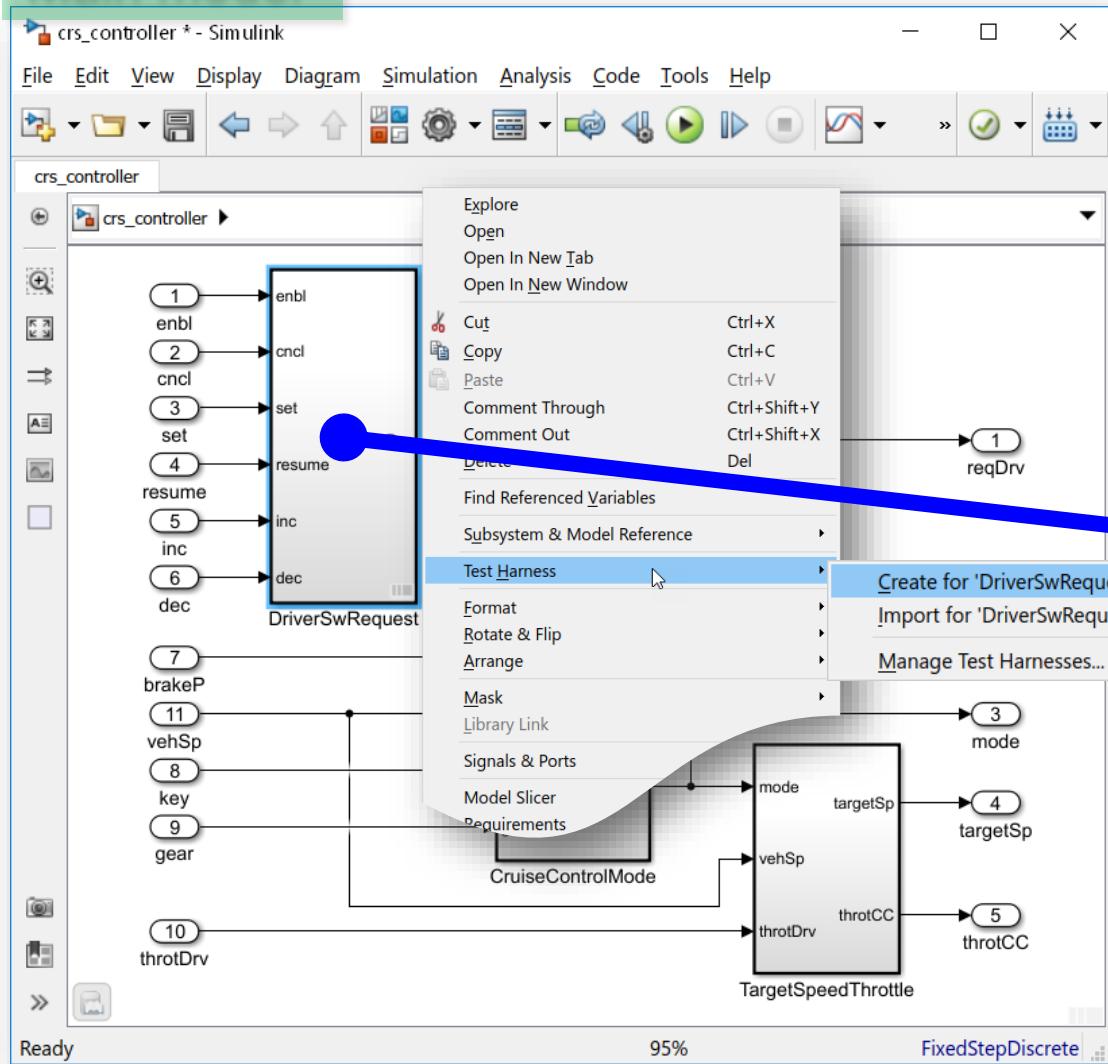
- Test harness model can be included in your original models



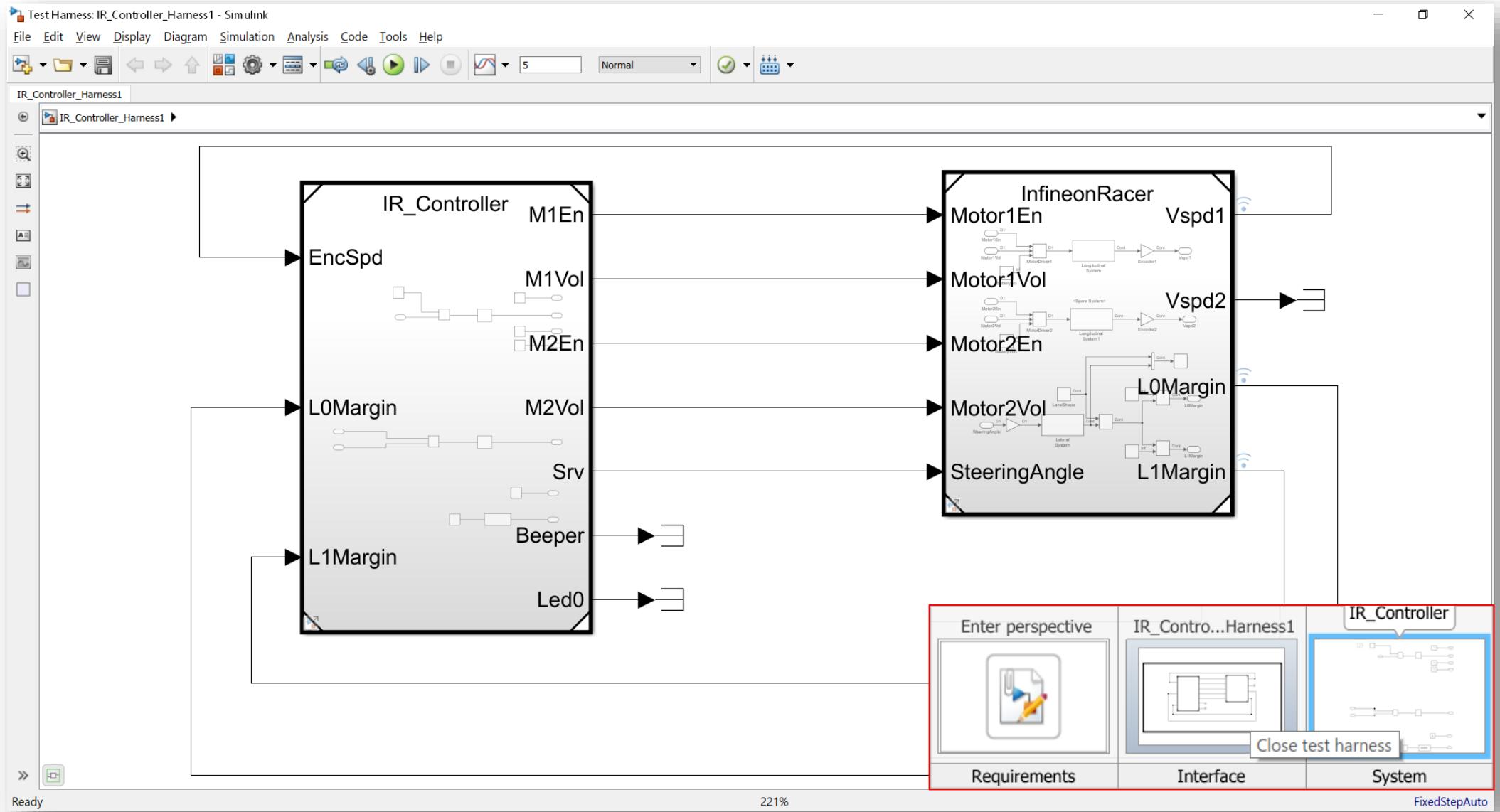
Click! to go back
to the
Main model

Unit Test using Test Harness Model

Main model

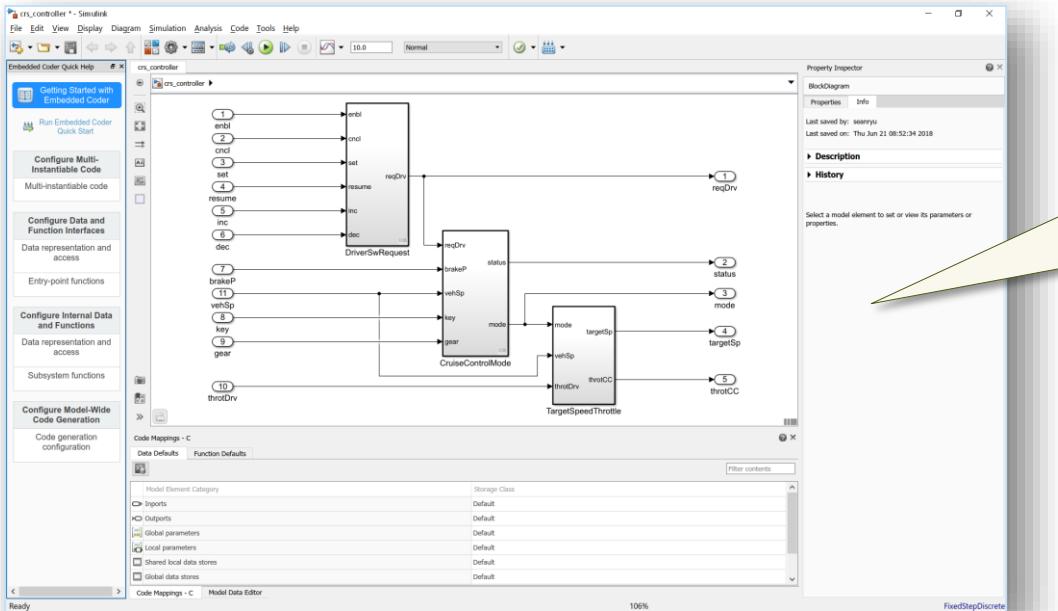


Let's Practice...



Code Perspective & Embedded Coder Dictionary

- Effective customization for code generation



Code Perspective

- Easy configuration for generated code into any C/C++ SW framework

Embedded Coder Dictionary: Controller

DICTIONARY

Name	Storage Type	Data Sc...	Header File	Definition File	Data Ini...	Memory S...	Source
ExportedGlobal	Unstructured	Exported	---	---	Auto	None	Built-in
ImportedExtern	Unstructured	Imported	---	---	Auto	None	Built-in
ImportedExternPointer	Unstructured	Imported	---	---	Auto	None	Built-in
BitField	FlatStructure	Exported	---	---	Auto	None	Simulink package
Const	Unstructured	Exported	<Instance specific>	<Instance specific>	Auto	MemConst	Simulink package
ConstVolatile	Unstructured	Exported	<Instance specific>	<Instance specific>	Auto	MemConstVolatile	Simulink package
Define	Unstructured	Exported	<Instance specific>	---	Macro	None	Simulink package
ImportedDefine	Unstructured	Imported	<Instance specific>	---	Macro	None	Simulink package
ExportToFile	Unstructured	Exported	<Instance specific>	<Instance specific>	Auto	None	Simulink package
ImportFromFile	Unstructured	Imported	<Instance specific>	---	Auto	None	Simulink package
FileScope	Unstructured	File	---	---	Auto	None	Simulink package
AutoScope	Unstructured	Auto	---	---	Auto	None	Simulink package
Struct	FlatStructure	Exported	---	---	Auto	None	Simulink package
GetSet	AccessFunction	Imported	<Instance specific>	---	Auto	None	Simulink package
CompilerFlag	Unstructured	Imported	---	---	Macro	None	Simulink package
Reusable	Unstructured	<Instance sp...	<Instance specific>	<Instance specific>	Dynamic	None	Simulink package

PROPERTY INSPECTOR

Embedded Coder Dictionary

- GUI for custom code definitions
 - Function template
 - Storage class
 - Memory section

Code Perspective

Edit-time checking

Navigate to code

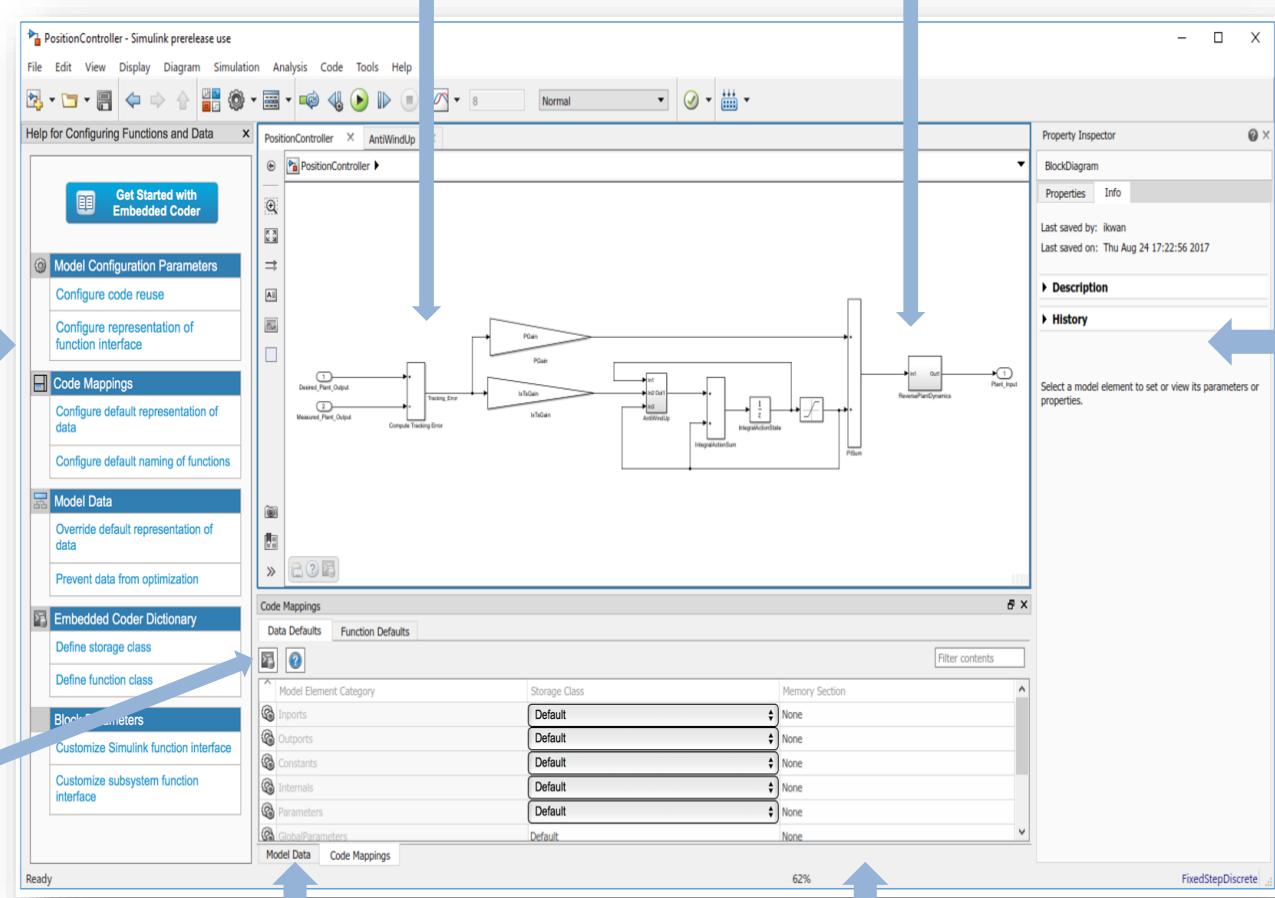
Help

Property Inspector

Code Dictionary

Model Data

Code Mapping



Embedded Coder Dictionary

1. Storage classes

- Control the code generated for model data (I/O, signals, data stores, states, parameters)

Storage allocation and scope
(ex, global, extern, static, register, pointer ...)

Bitfield, Constant, Pre-processor, ...

Export to or import from external files, ...

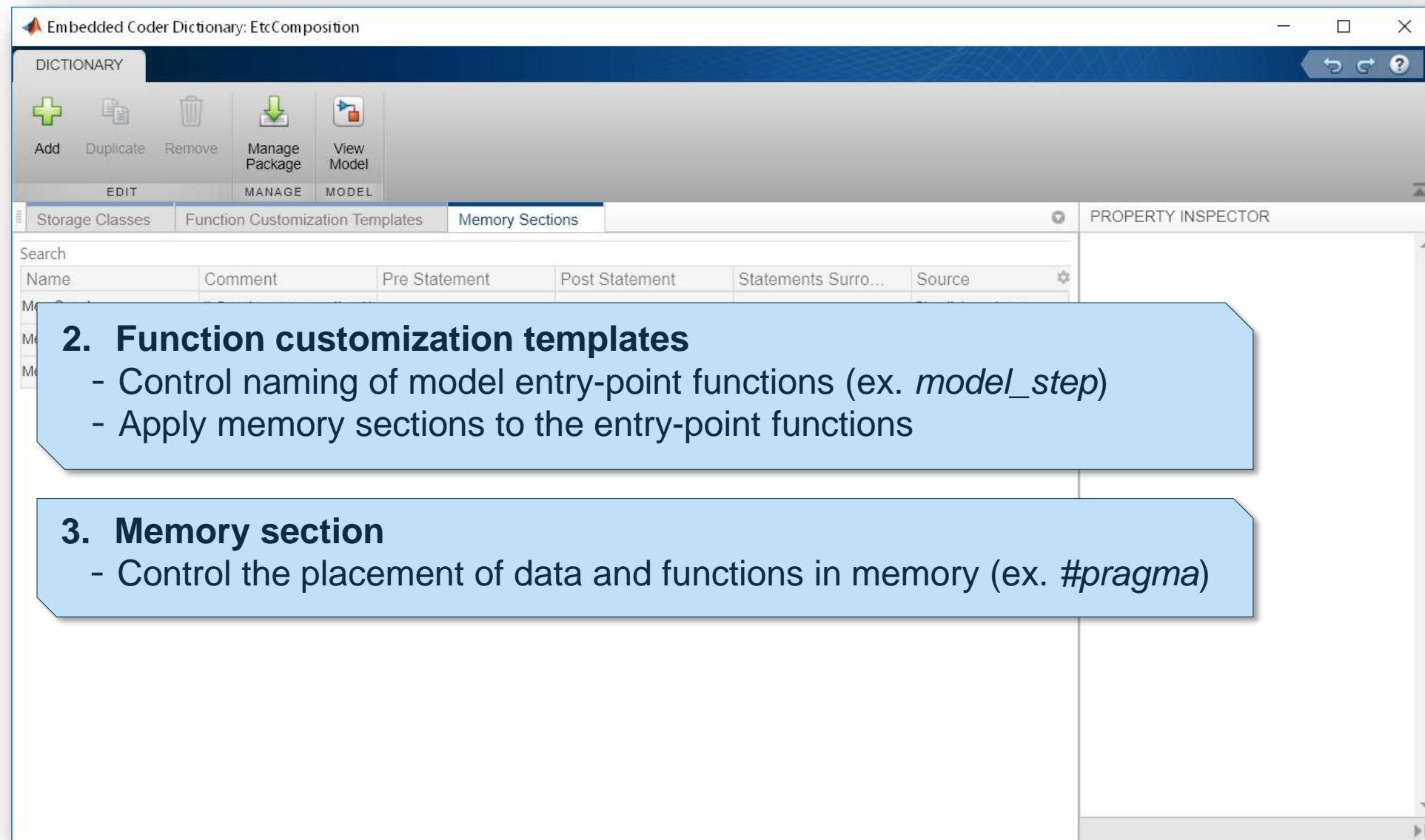
Etc.: Structure type, Get/Set APIs, ...

Defining storage classes

The screenshot shows the 'Storage Classes' tab of the Embedded Coder Dictionary. A red box highlights the row for 'MyParam'. The 'PROPERTY INSPECTOR' panel on the right shows the following settings for 'MyParam':

Name	Description	Source	Storage Type	Data Scope	Header File	Definition File	Data Initialization	Memory Section	Source
MyParam	Description	EtcComposition	Structured	Exported	\$N_Param.h	\$N_Param.c	Static	None	Simulink package

Embedded Coder Dictionary



Let's Practice...

IR_Controller - Simulink

File Edit View Display Diagram Simulation Analysis Code Tools Help

IR_Controller x Disabled x

IR_Controller ▶

Property Inspector

BlockDiagram

Properties Info Execution

Last saved by: seanryu
Last saved on: Fri Apr 12 23:09:23 2019

Description

Model information

Select a model element to set or view its parameters or properties.

Model Data Editor

Inputs/Outputs Signals Data Stores States Parameters

Code

Source	#	Signal Name	Resolve	Storage Class	Header File	Definition File	Get Function	Set Function	Struct Name
EncSpd	1	EncSpeed	<input type="checkbox"/>	GetSet	Basic.h		IR_get\$N	IR_set\$N	
L0Margin	2	Ls0Margin	<input type="checkbox"/>	GetSet	InfineonRacer.h		IR_get\$N	IR_set\$N	
L1Margin	3	Ls1Margin	<input type="checkbox"/>	GetSet	InfineonRacer.h		IR_get\$N	IR_set\$N	
M1En	1		<input type="checkbox"/>	Auto					
M1Vol	2		<input type="checkbox"/>	Auto					
M2En	3		<input type="checkbox"/>	Auto					

Filter contents

Property Inspector Code

FixedStepDiscrete

Model Data Editor Code Mappings - C

Ready 52%

Automated Driving Testing using Simulink

Challenges

- Algorithm Design
 - Lane Detection
 - Steering Control (Lateral Motion Control)
 - Velocity Control (Longitudinal Motion Control)
- How can I Test Algorithm in Simulink?
 - Vehicle model
 - Closed Loop Simulation

Lane Detection Processing

Developing Active Safety Systems

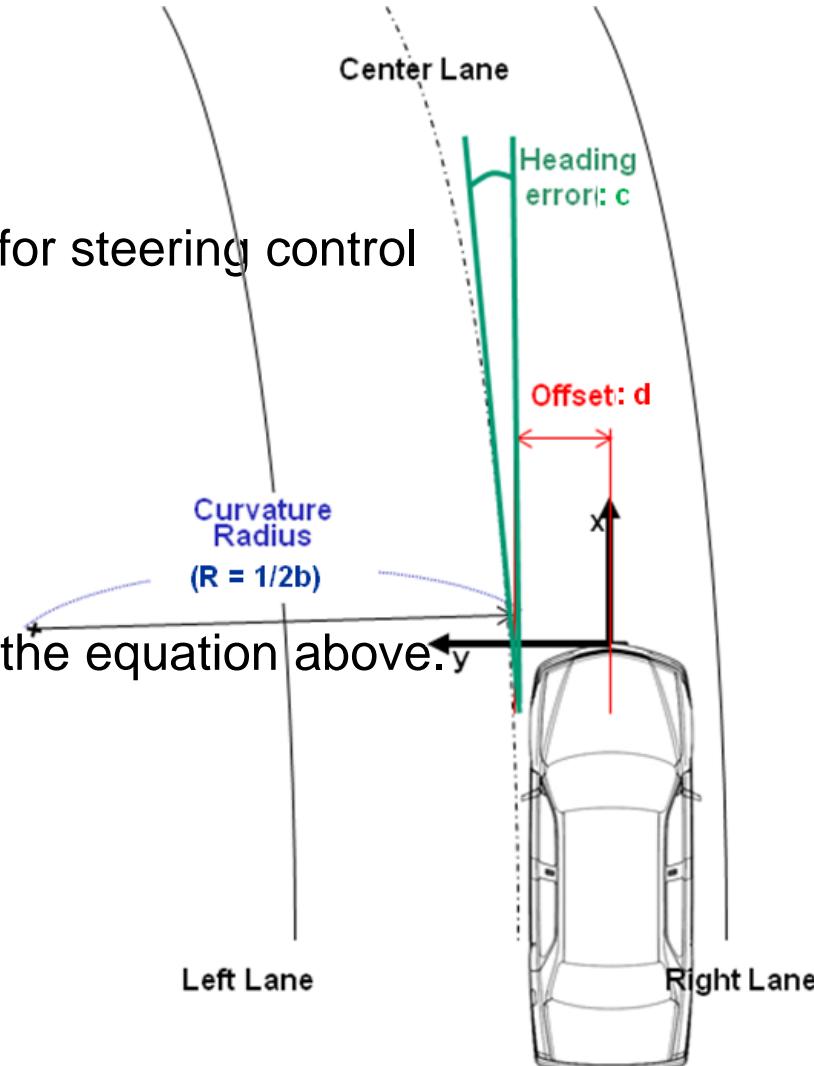
Developing the Vision Algorithm Part

- Goal of Lane Detection Algorithm

- Obtaining the 3rd order polynomial road model for steering control

- $$y = ax^3 + bx^2 + cx + d$$

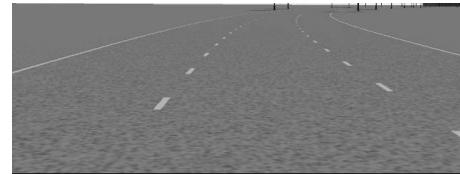
- But, only information using line scanner is **d** in the equation above.
 - If using two sensor, you can calculate **c**



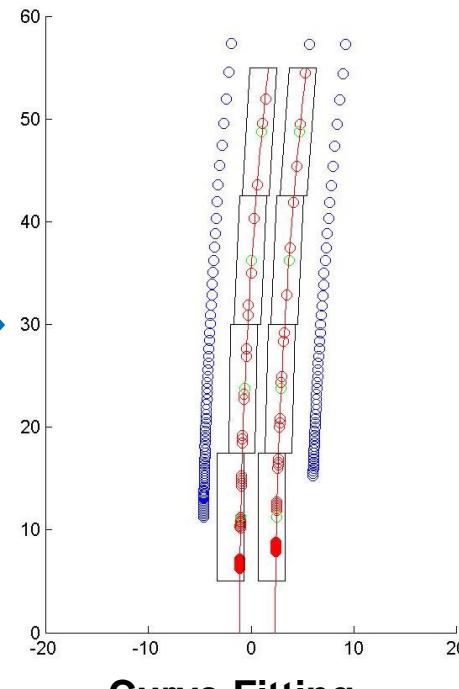
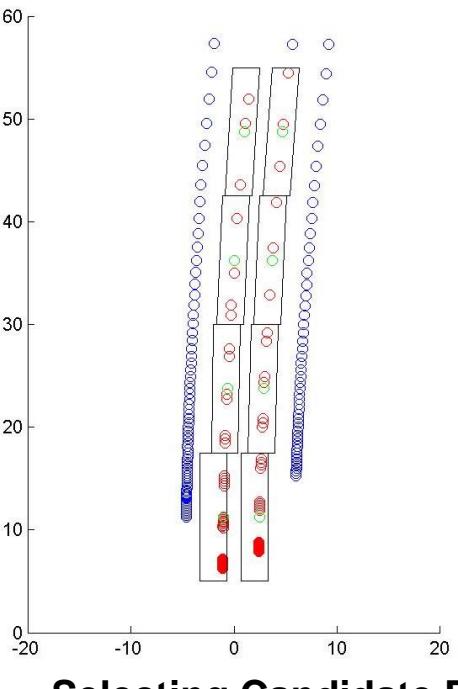
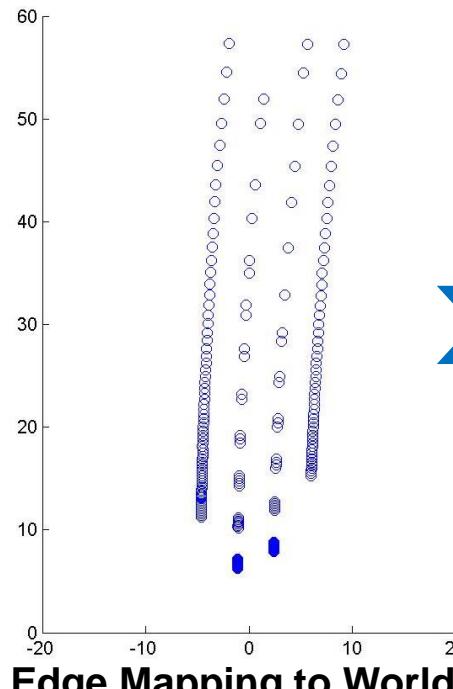
Developing Active Safety Systems

Developing the Vision Algorithm Part

ROI (Region Of Interest)



Captured Image

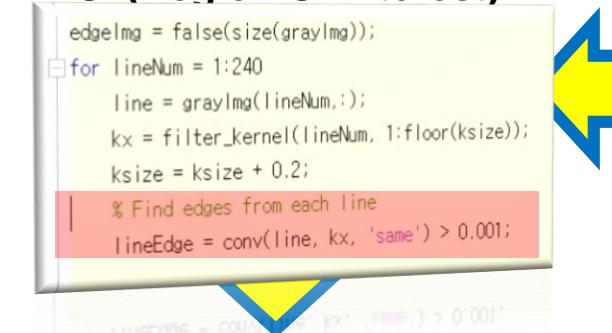
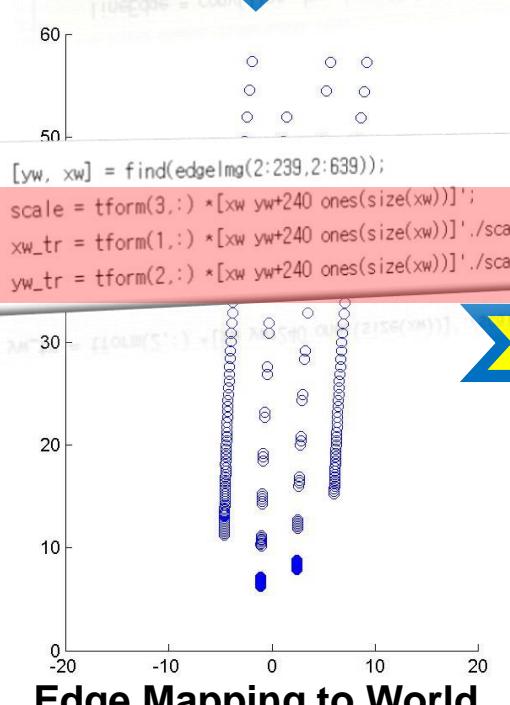


Developing Active Safety Systems

Developing the Vision Algorithm Part

ROI (Region Of Interest)

```
edgelng = false(size(grayImg));
for lineNum = 1:240
    line = grayImg(lineNum,:);
    kx = filter_kernel(lineNum, 1:floor(ksize));
    ksize = ksize + 0.2;
    % Find edges from each line
    lineEdge = conv(line, kx, 'same') > 0.001;
```



```
%>>> ROI (Region of Interest), Cropped Image
```



Captured Image

```
for lane = 1:2
    if NSupportPts(lane) > 3
        Eq(lane,2:4) = polyfit(Xcenter(lane,1:NSupportPts(lane)), Ycenter(lane,1:NSupportPts(lane)),2);
        ycenter_init(lane) = polyval(Eq(lane,:), 11.25);%Ycenter(lane,1);
        slope_init(lane) = Eq(lane,1)*11.25*11.25*3+Eq(lane,2)*11.25*2+Eq(lane,3);
    end
```

success = 1;

pts1 = pts(:,inlier1);

ptsX = pts1(:,1) - mean(pts1(:,1));

ptsY = pts1(:,2) - mean(pts1(:,2));

[u, s, v] = svd([ptsX; ptsY]', 0);

d1 = v(:,1);

```
q1 = A*inv(B)*d1;
```

for i = 1:100

if abs(q1) < 0.001

break

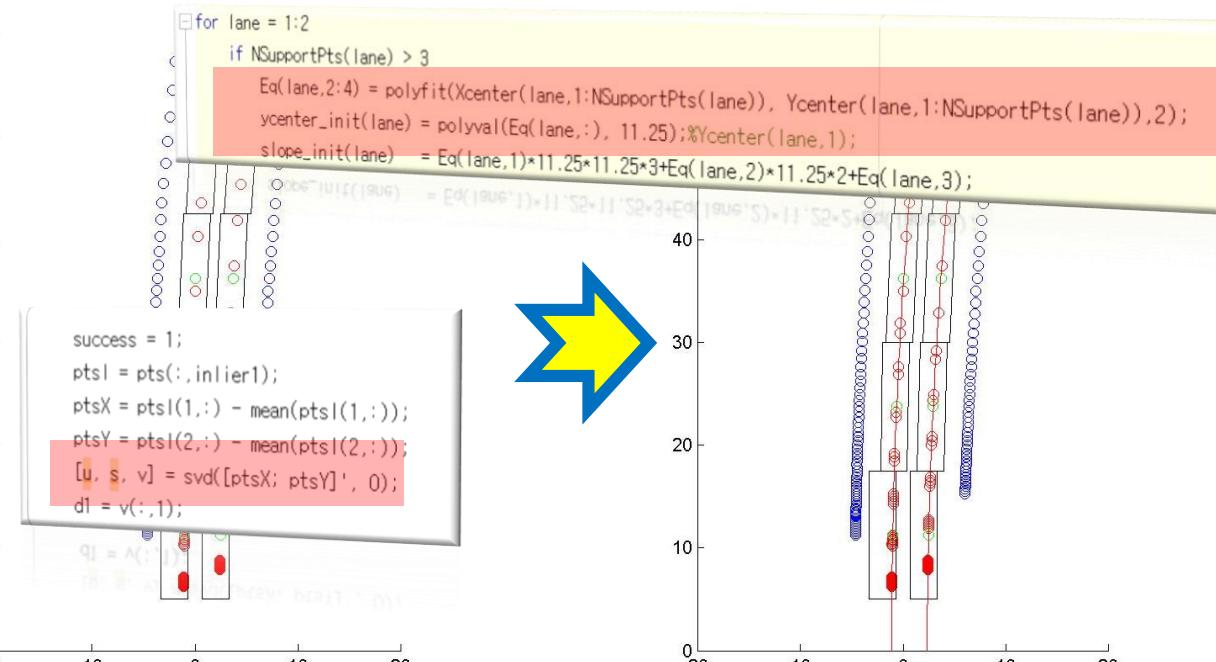
else

q1 = A*inv(B)*d1;

d1 = d1 + 0.001;

end

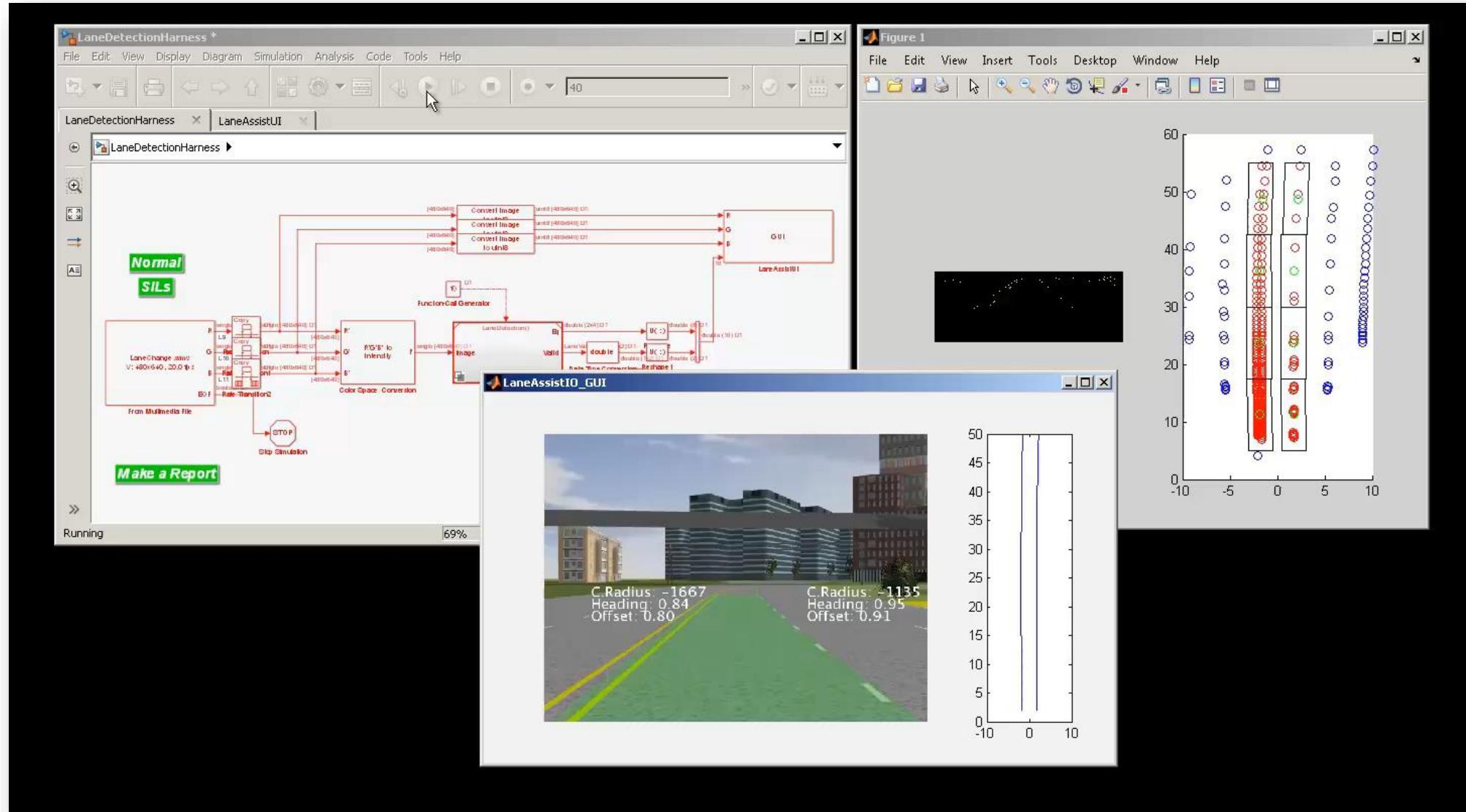
Selecting Candidate Pts



Curve Fitting

Developing Active Safety Systems

Developing the Vision Algorithm Part



Computer Vision System Toolbox Examples

MATLAB Examples

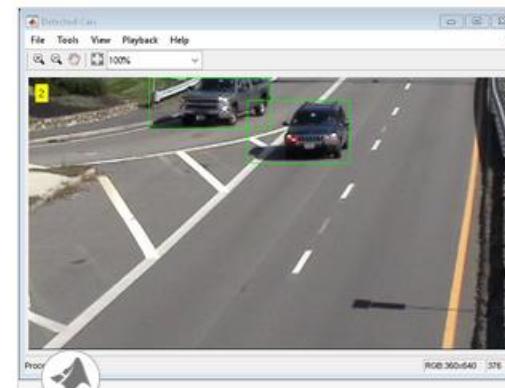
Object Detection and Recognition



Automatically Detect and Recognize Text in Natural Images

Detect regions in an image that contain text. This is a common task performed on unstructured scenes. Unstructured scenes are images that

[Open Live Script](#)



Detecting Cars Using Gaussian Mixture Models

Detect and count cars in a video sequence using foreground detector based on Gaussian mixture models (GMMs).

[Open Live Script](#)



Object Detection in a Cluttered Scene Using Point Feature Matching

Detect a particular object in a cluttered scene, given a reference image of the object.

[Open Script](#)



Object Detection Using Deep Learning

Train an object detector using deep learning and R-CNN (Regions with Convolutional Neural Networks).

[Open Live Script](#)

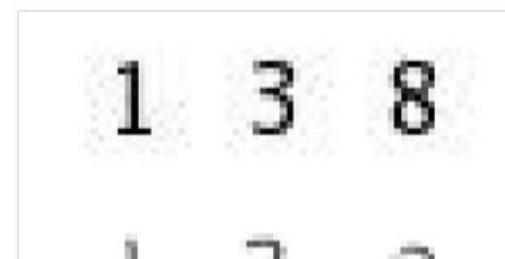


Image Processing Toolbox Examples

Deep Learning



Single Image Super-Resolution Using Deep Learning

Train a Very-Deep Super-Resolution (VDSR) neural network, then use a VDSR network to estimate a high-resolution image from a single low-

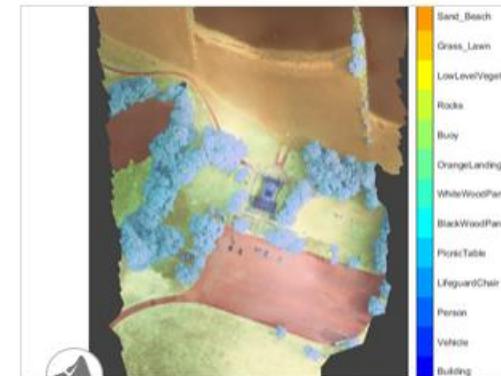
[Open Live Script](#)



JPEG Image Deblocking Using Deep Learning

Train a denoising convolutional neural network (DnCNN), then use the network to reduce JPEG compression artifacts in an image.

[Open Live Script](#)

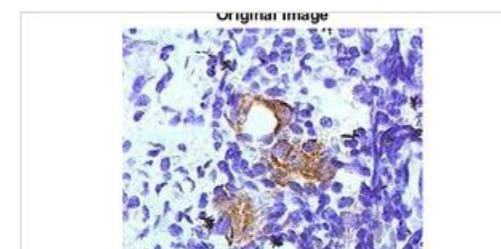


Semantic Segmentation of Multispectral Images Using Deep Learning

Train a U-Net convolutional neural network to perform semantic segmentation of a multispectral image with seven channels: three

[Open Live Script](#)

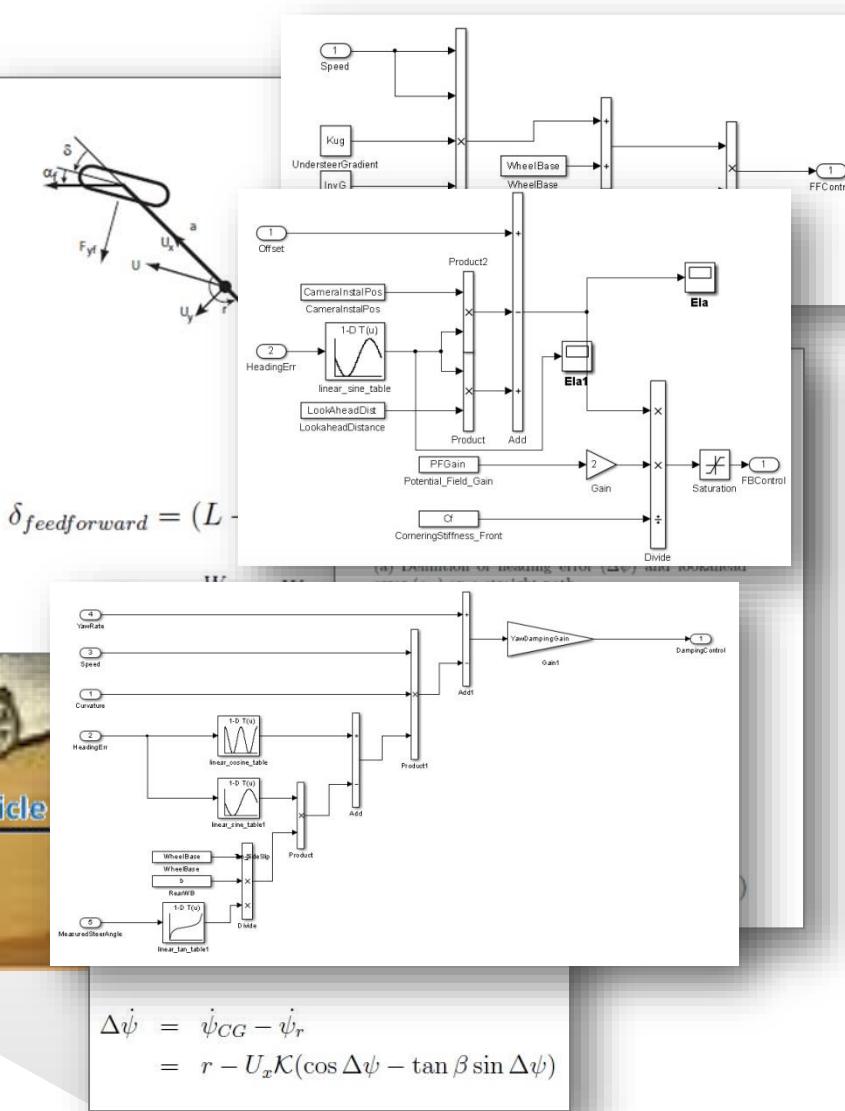
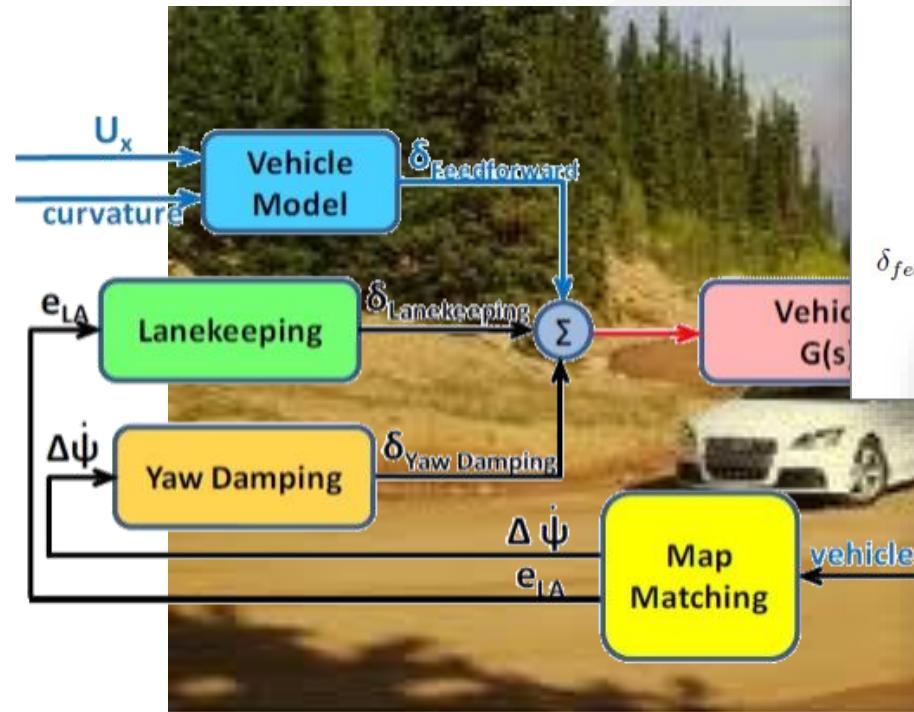
Deblurring



Control Part

Developing Active Safety Systems

Steering angle compensator

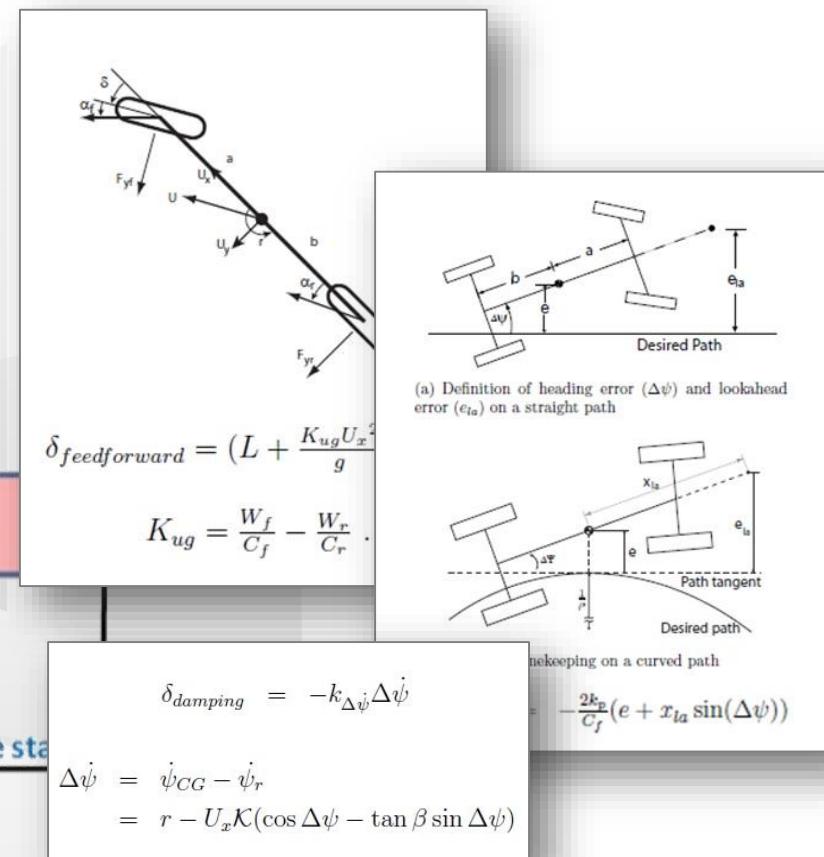
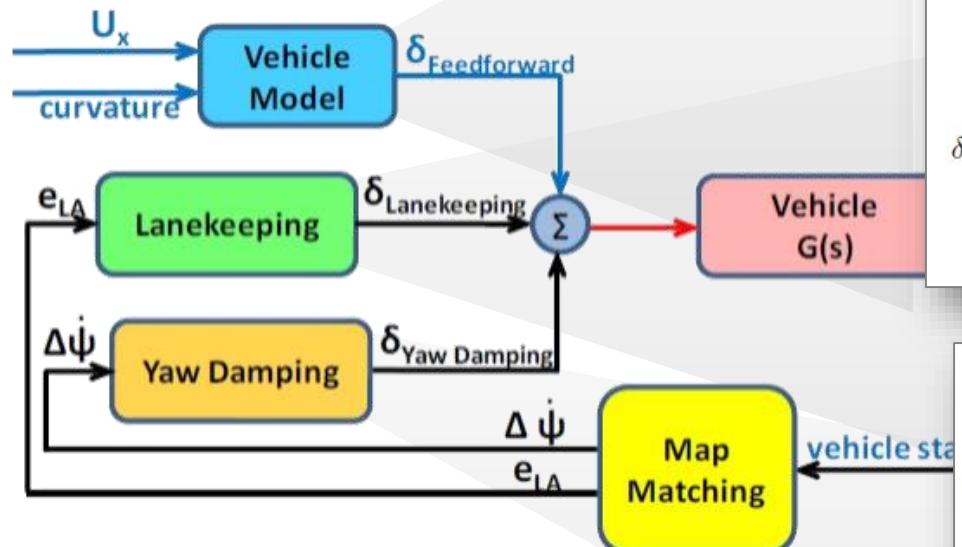


Krisada (Mick) Kritayakirana, From 'AUTONOMOUS VEHICLE CONTROL AT THE LIMITS OF HANDLING',
PhD DISSERTATION, Stanford Univ, June 2012

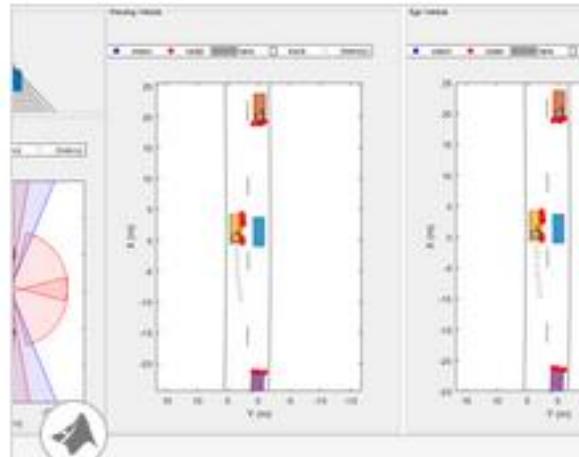
Developing Active Safety Systems

Steering angle compensator

- Curvature information is absent when using line scanner.
- Applying feedforward term is impossible.



Examples – Sensor Fusion



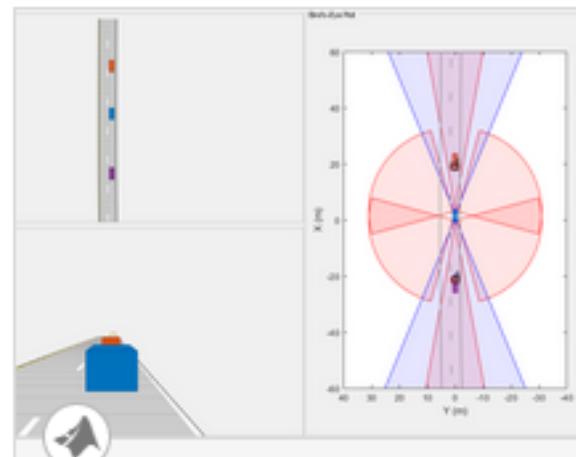
Extended Object Tracking

Track objects whose dimensions span multiple sensor resolution cells.

[Open Script](#)

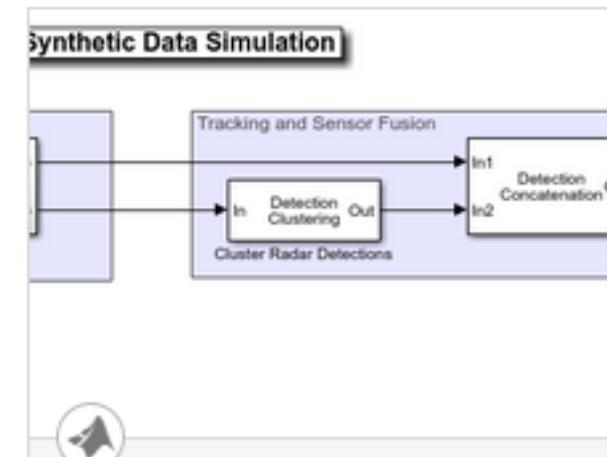
Visual-Inertial Odometry Using Synthetic Data

Estimate the pose (position and orientation) of a vehicle by using an inertial measurement unit (IMU) and a monocular camera.

[Open Script](#)

Sensor Fusion Using Synthetic Radar and Vision Data

Generate a scenario, simulate sensor detections, and use sensor fusion to track simulated vehicles.

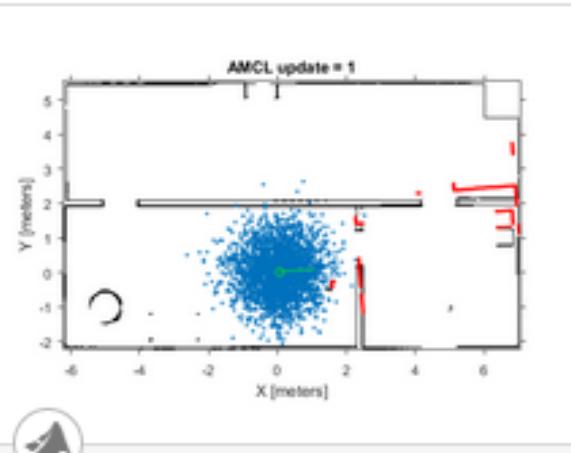
[Open Script](#)

Sensor Fusion Using Synthetic Radar and Vision Data in Simulink

Implement a synthetic data simulation for tracking and sensor fusion in Simulink with Automated Driving System Toolbox.

[Open Model](#)

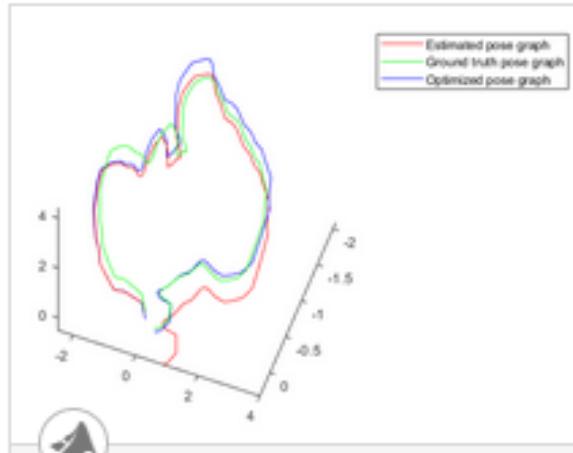
Examples – SLAM and Localization



Localize TurtleBot Using Monte Carlo Localization

Demonstrates an application of the Monte Carlo Localization (MCL) algorithm on TurtleBot® in simulated Gazebo® environment.

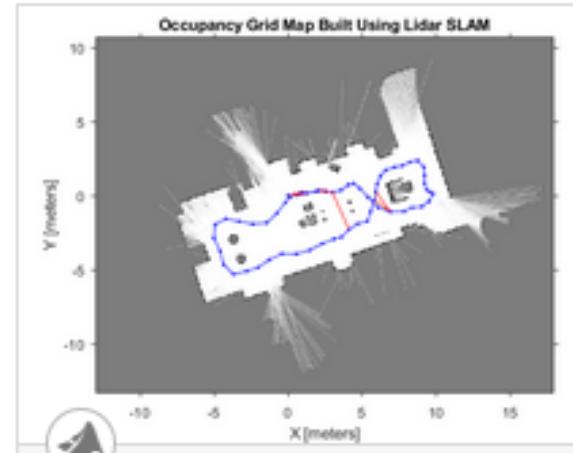
[Open Live Script](#)



Reduce Drift in 3-D Visual Odometry Trajectory Using Pose Graphs

Reduce the drift in the estimated trajectory (location and orientation) of a monocular camera using 3-D pose graph optimization.

[Open Live Script](#)



Implement Simultaneous Localization And Mapping (SLAM) with Lidar Scans

Demonstrates how to implement the Simultaneous Localization And Mapping (SLAM) algorithm on a collected series of lidar scans using

[Open Live Script](#)

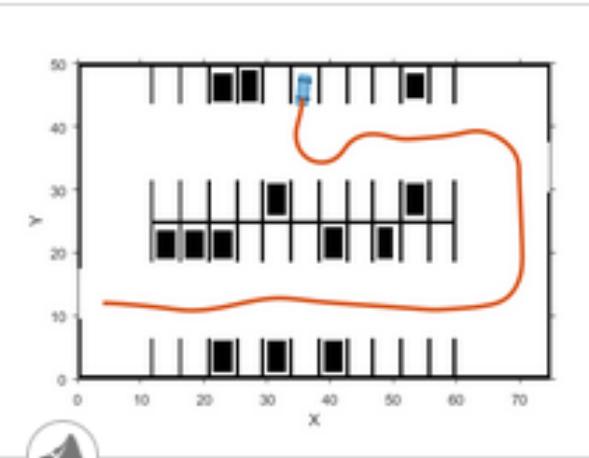


Create Occupancy Grid Using Monocular Camera and Semantic...

Estimate free space and create an occupancy grid using semantic segmentation and deep learning.

[Open Live Script](#)

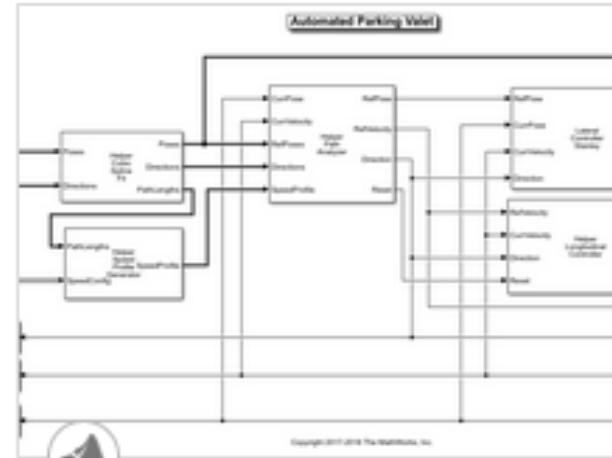
Examples – Path Planning and Following



Automated Parking Valet

Construct an automated parking valet system using path planning, trajectory generation, and vehicle control techniques.

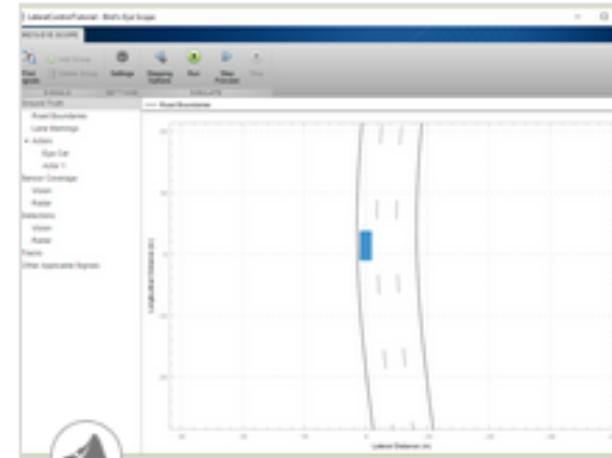
[Open Script](#)



Automated Parking Valet in Simulink

Construct an automated parking valet system in Simulink with Automated Driving System Toolbox.

[Open Model](#)



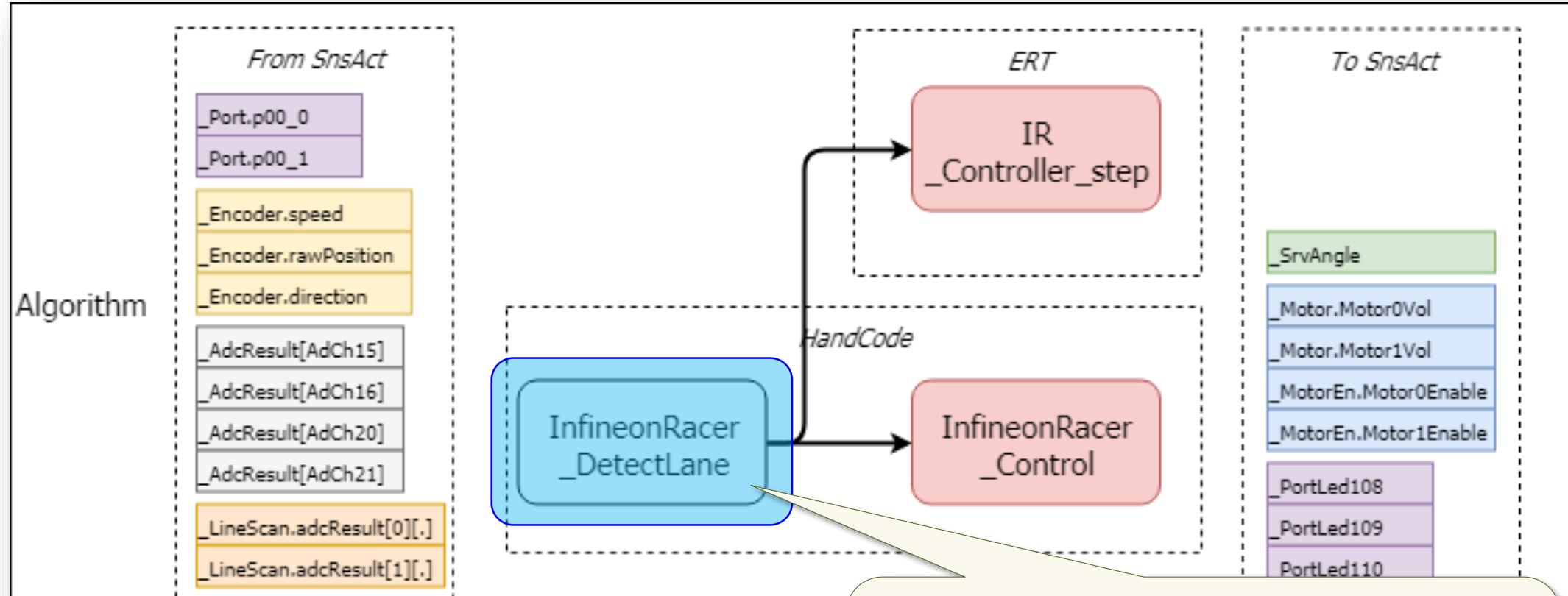
Lateral Control Tutorial

Control the steering angle of a vehicle following a planned path and perform lane changing.

[Open Model](#)

Automated Driving System Toolbox

Example for SMCC



Lane Following Control with Sensor Fusion and Lane Detection

This example shows how to simulate and generate code for an automotive lane-following controller.

In this example, you will:

1. Review a control algorithm that combines sensor fusion, lane detection, and a lane following controller from the Model Predictive Control Toolbox™.
2. Test the control system in a closed-loop Simulink® model using synthetic data generated by the Automated Driving Toolbox™.
3. Configure the code generation settings for software-in-the-loop simulation and automatically generate code for the control algorithm.

Introduction

A lane following system is a control system that keeps the vehicle traveling within a marked lane of a highway, while maintaining a user-set velocity or safe distance from the preceding vehicle. A lane following system includes combined longitudinal and lateral control of the ego vehicle:

- Longitudinal control - Maintain a driver-set velocity and keep a safe distance from the preceding car in the lane by adjusting the longitudinal acceleration.
- Lateral control - Keep the ego vehicle travelling along the centerline of its lane by adjusting the steering of the ego vehicle.

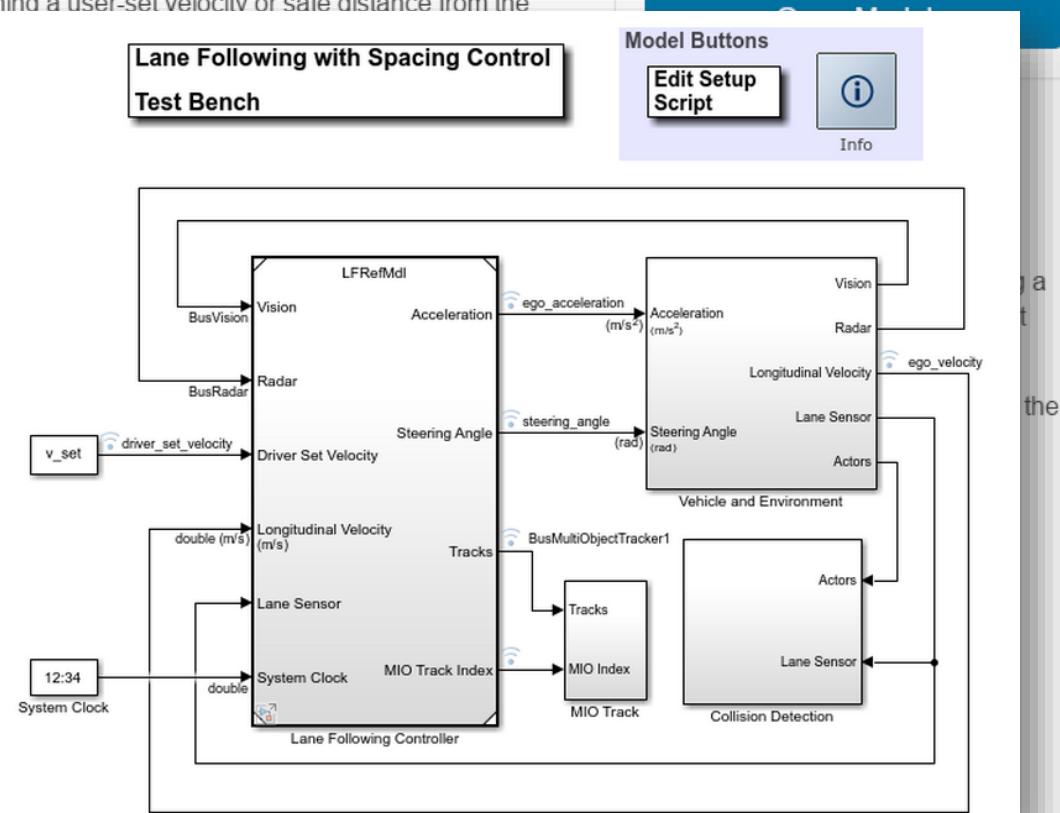
The combined lane following control system achieves the individual goals for longitudinal and lateral control. Further, the lane keeping assist (LKA) system with lane detection, see [Lane Keeping Assist with Lane Detection](#) (Model Predictive Control Toolbox™), consider surrounding vehicles.

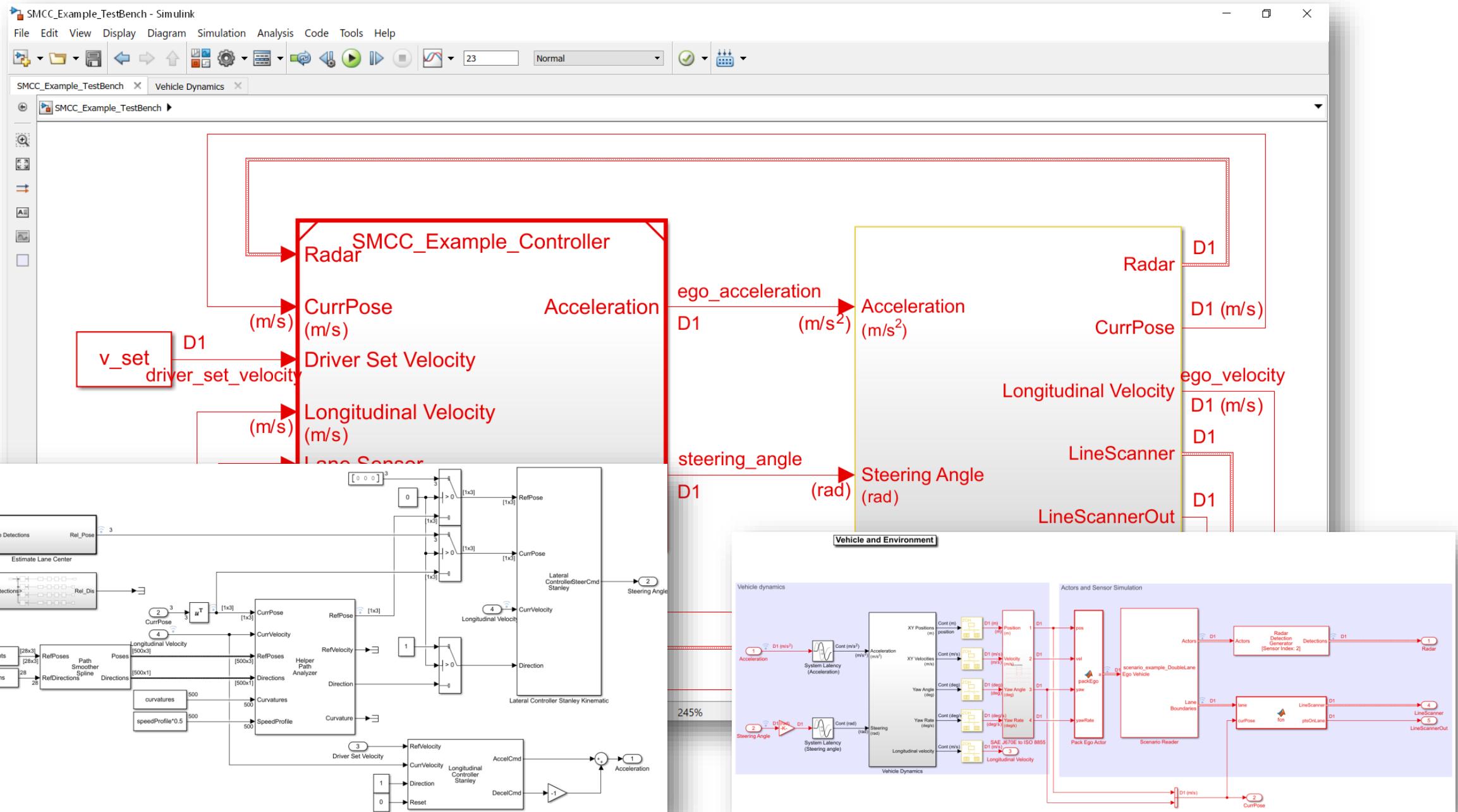
For an example of longitudinal control using adaptive cruise control (ACC) with sensor fusion, see [Adaptive Cruise Control with Sensor Fusion](#). For an example of a lane keeping assist (LKA) system with lane detection, see [Lane Keeping Assist with Lane Detection](#) (Model Predictive Control Toolbox™). In this example, both lane detection and surrounding cars are considered. The lane following system synthesizes data from the longitudinal acceleration and steering angle of the ego vehicle.

Add example file folder to MATLAB® path.

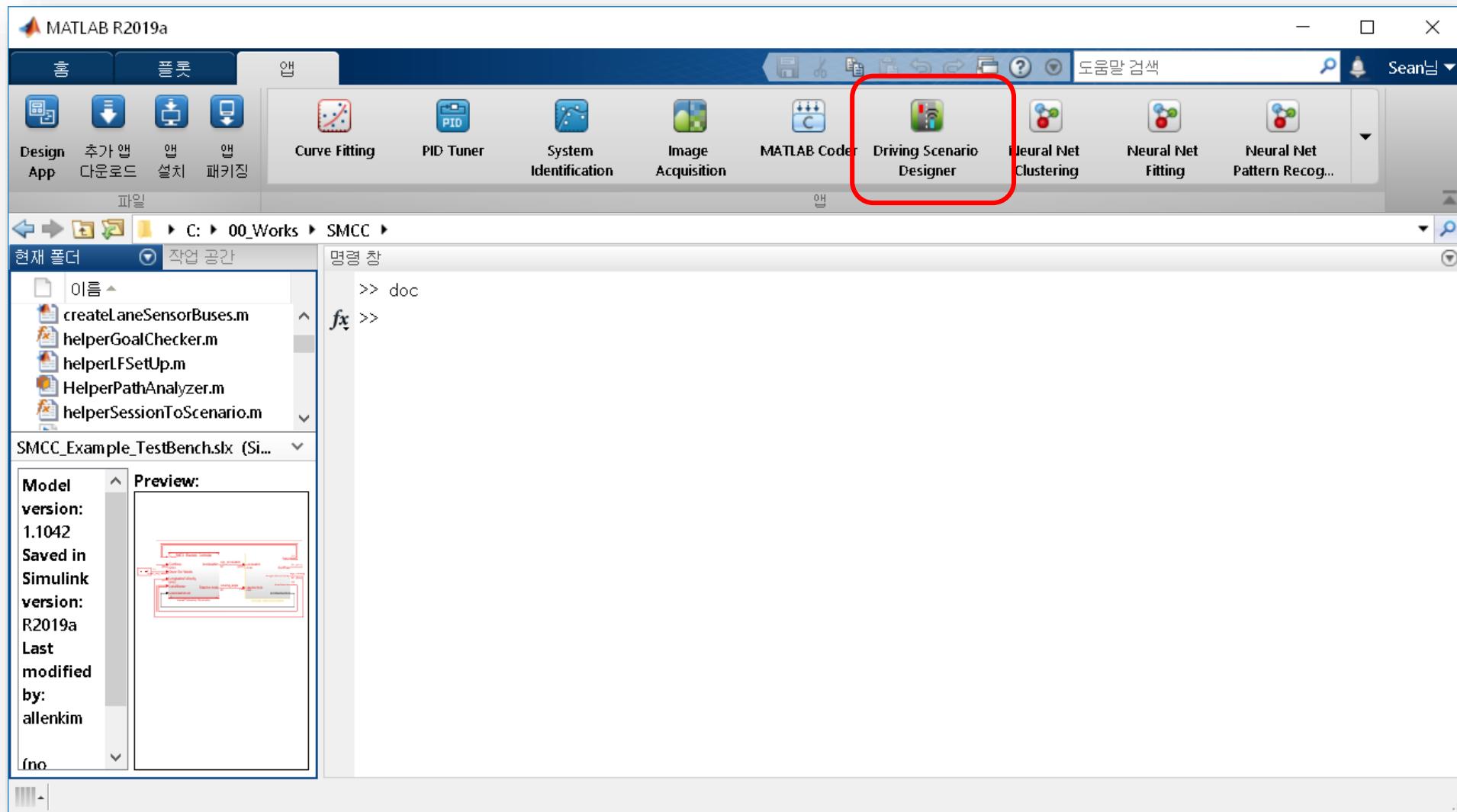
```
addpath(fullfile(matlabroot, 'examples', 'mpc', 'main'));
```

This example uses:
[Automated Driving Toolbox](#)
[Embedded Coder](#)
[Model Predictive Control Toolbox](#)
[Simulink Control Design](#)
[Simulink](#)

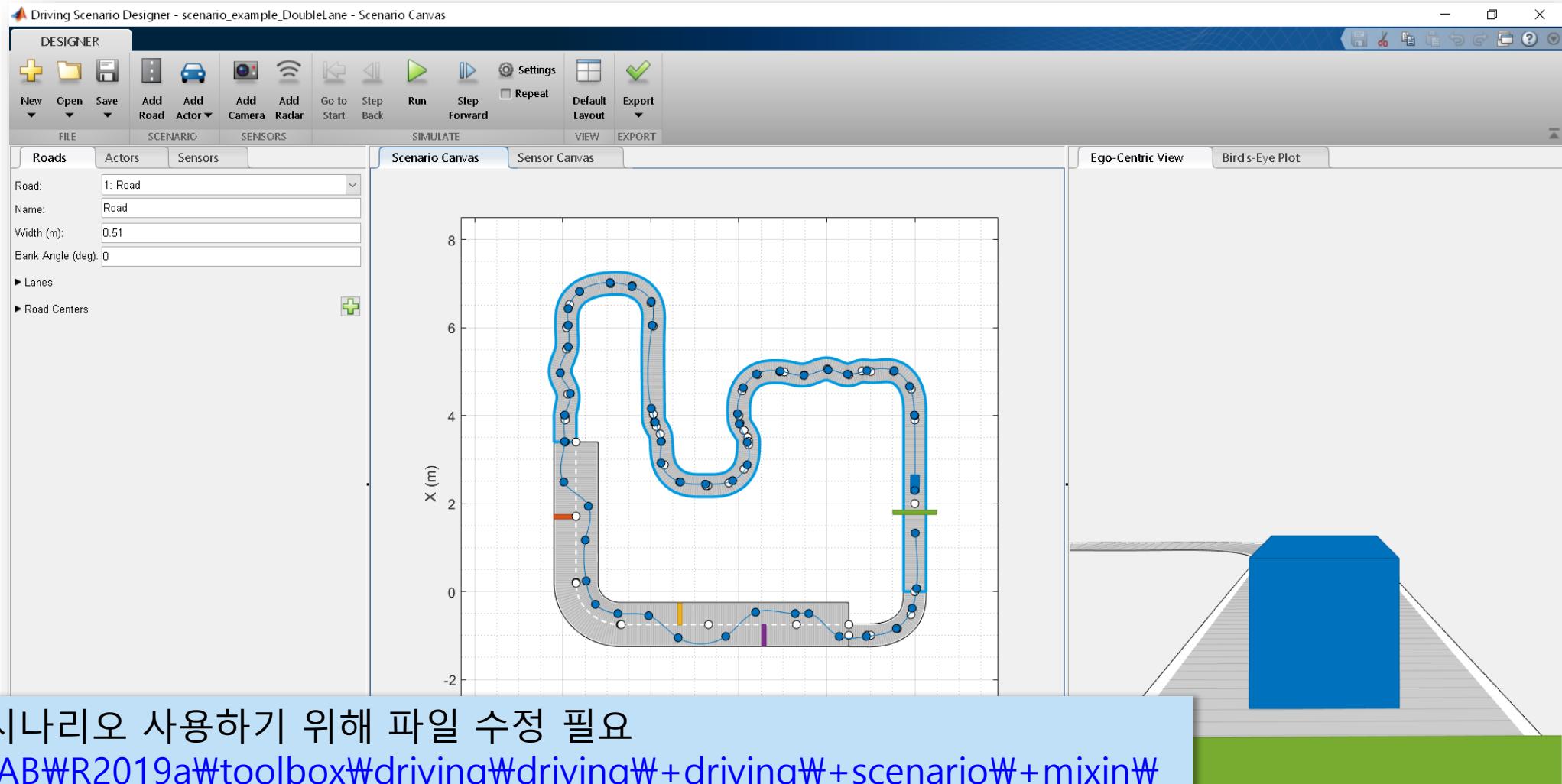




Driving Scenario Designer



Driving Scenario Example



*대회용 시나리오 사용하기 위해 파일 수정 필요

C:\MATLAB\R2019a\toolbox\driving\driving\scenario\mixin

1) RoadNetwork.m 파일 수정

2) Line 20: TileLength = 2 → 0.05로 변경

Control System Design with Simulink®

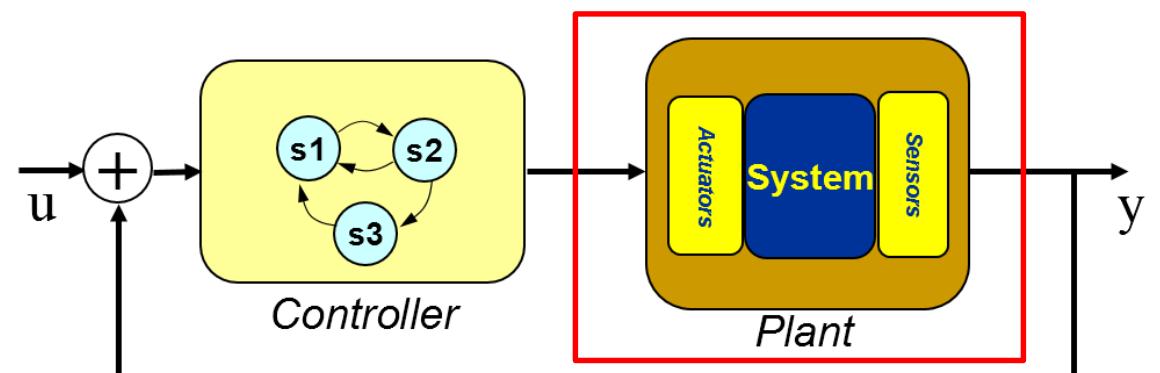
Control Design Process

- **Modeling:**

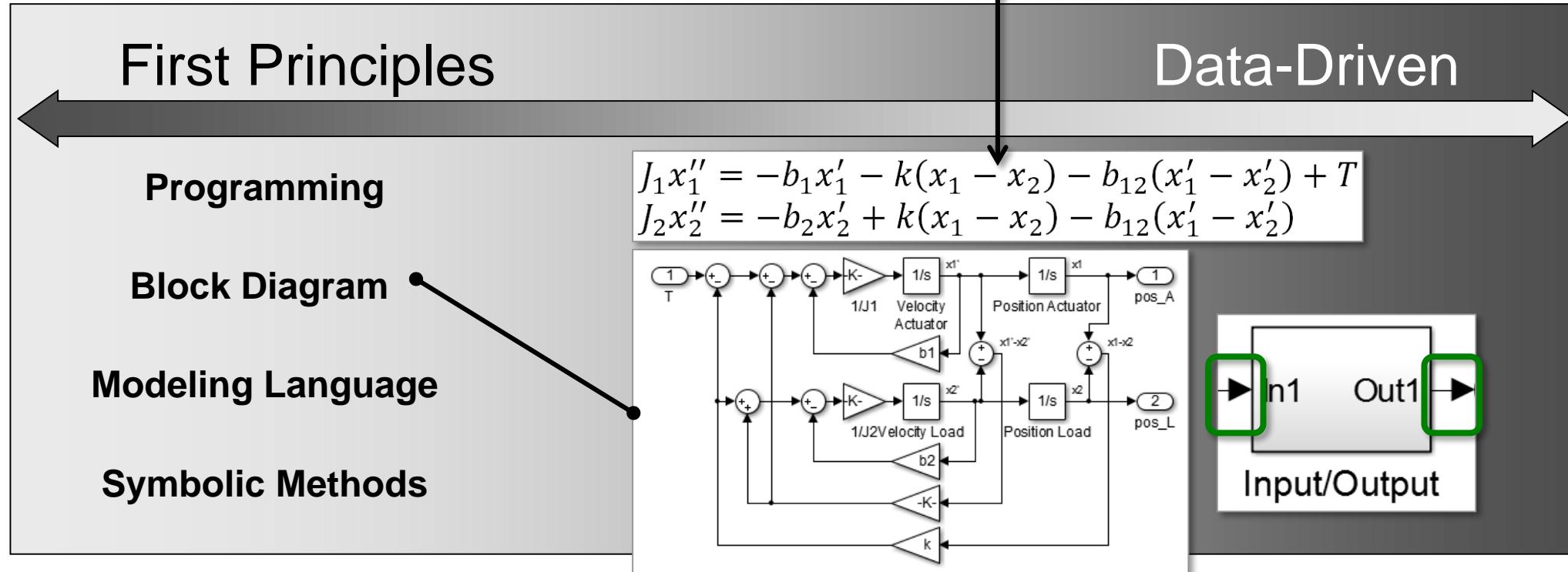
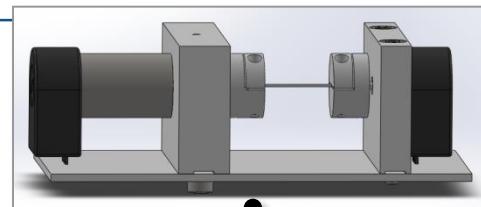
understanding your design problem

- **Control Design:**

solving your design problem

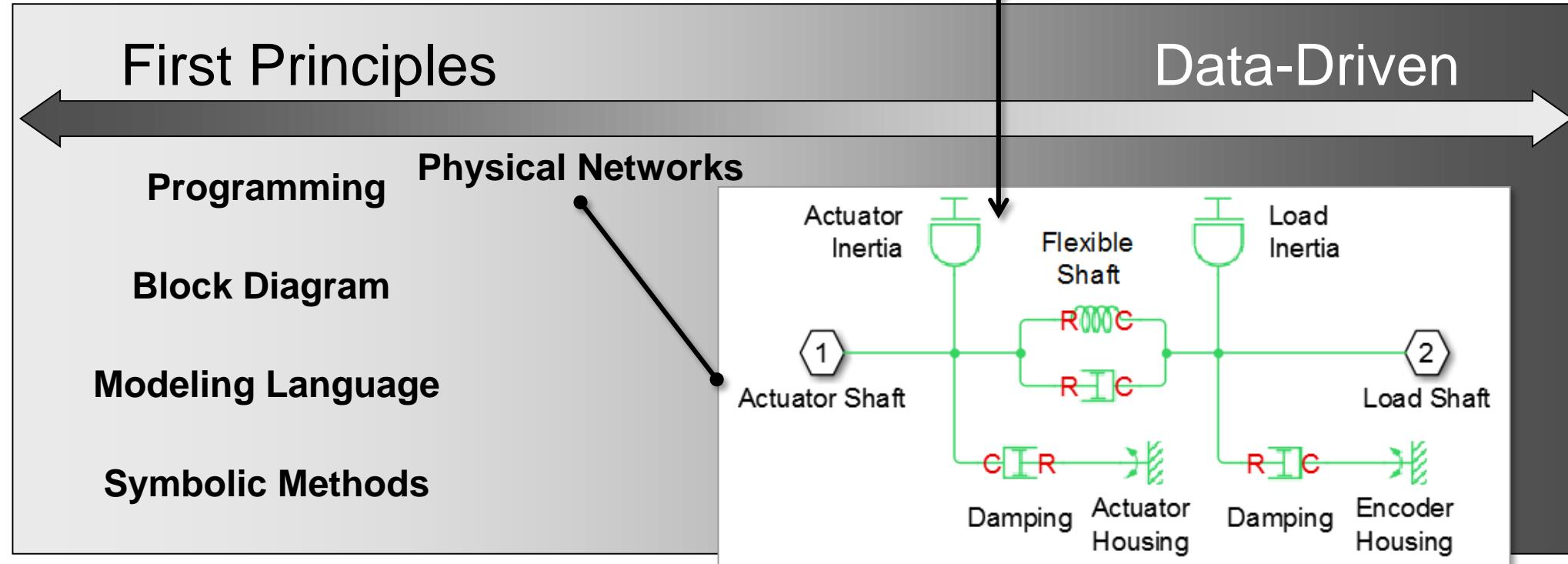
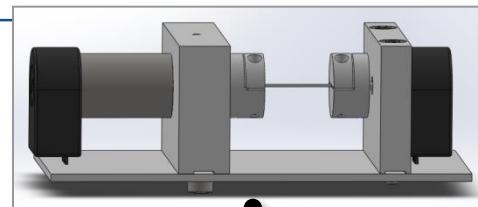


Modeling Approaches

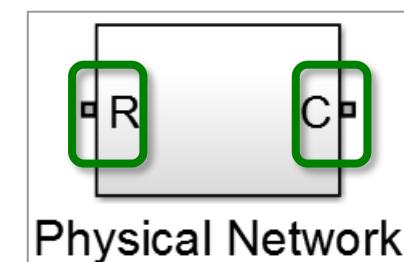


- Purpose: Explore design or physical parameters
- Requirements:
 - Physics of system are well-known
 - System-level equations can be derived and implemented

Modeling Approaches



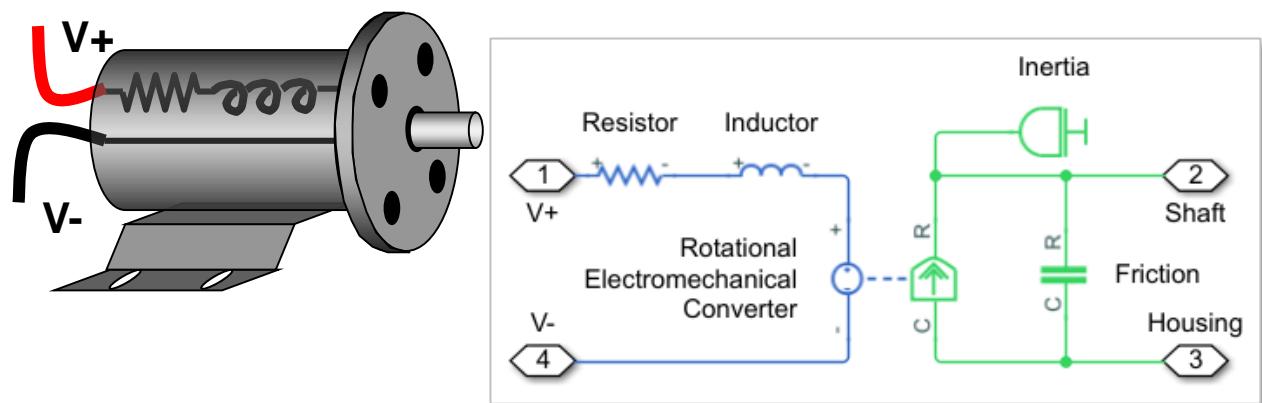
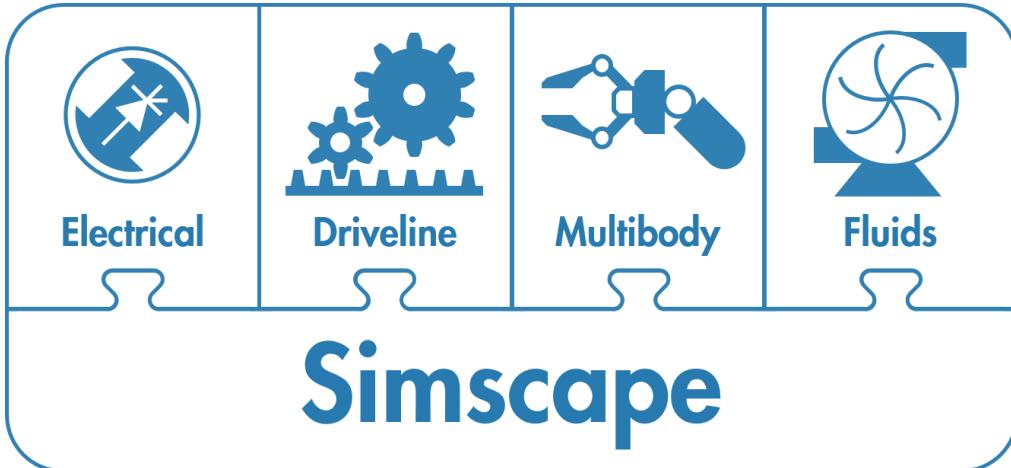
- Purpose: Explore design or physical parameters
- Requirements:
 - Physics of system are well-known
 - Component-level models exist or can be created



Simscape

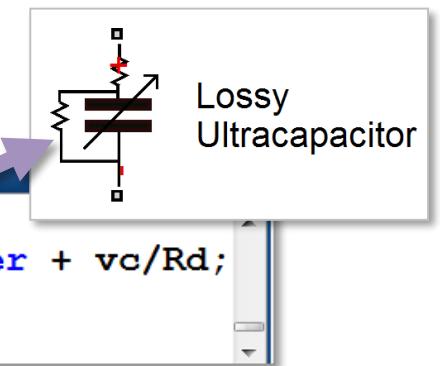
Overview

- Enables physical modeling (acausal) of multidomain physical systems
 - Assemble a schematic
 - Equations derived automatically
 - Leverage MATLAB and Simulink
- With Simscape you can:
 - Refine requirements for system
 - Discover integration issues early
 - Design control systems and logic
 - Optimize system-level performance
 - Test embedded software without hardware prototypes



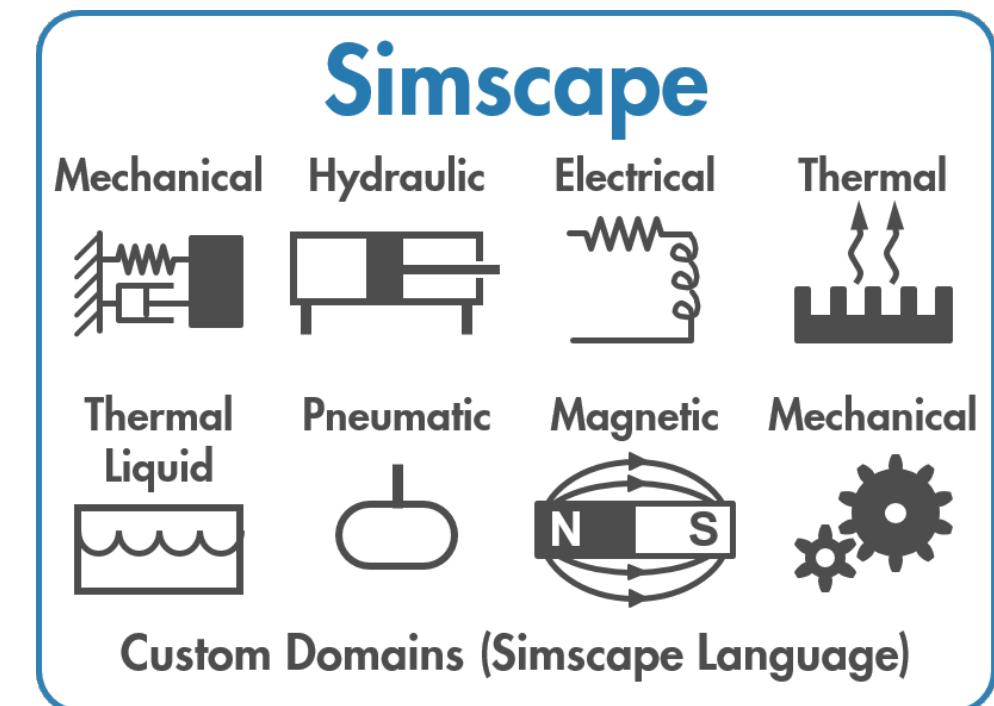
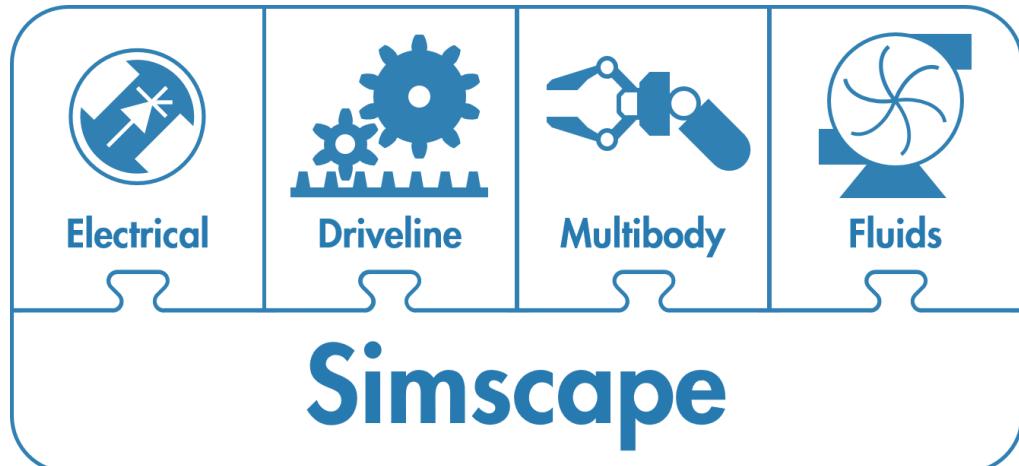
$$i = (C_0 + C_v v) \frac{dv}{dt} + \frac{v}{r_d}$$

```
Editor - C:\+MyComponents\LossyUltraCapacitor.ssc
40 equations
41 i == (C0 + Cv*vc)*vc.der + vc/Rd;
42 v == vc + i*R;
43 end
```



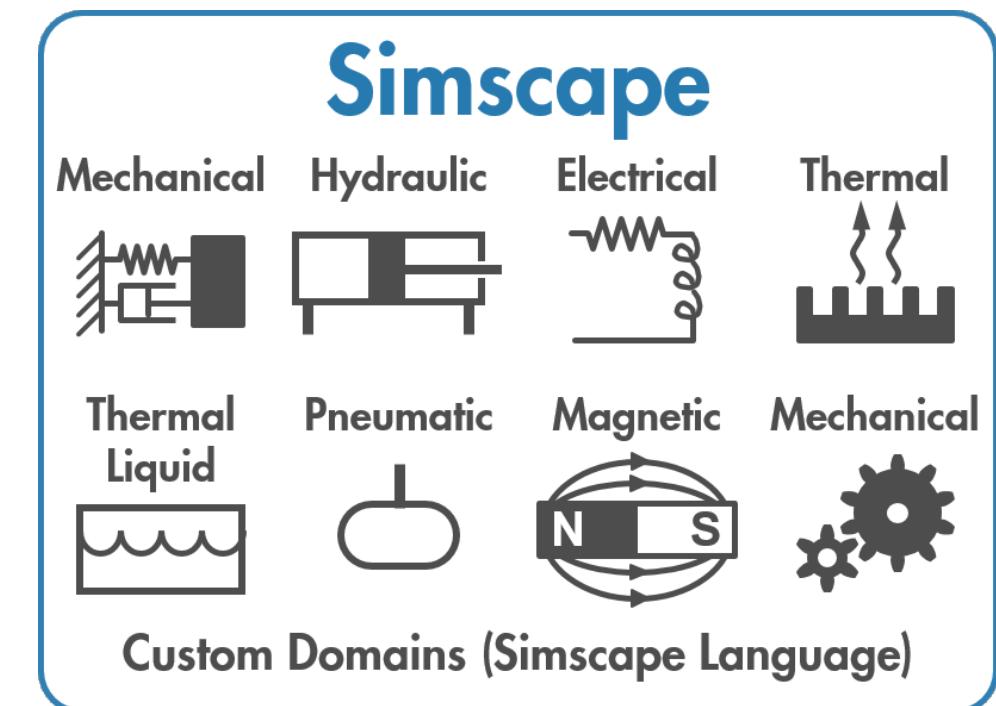
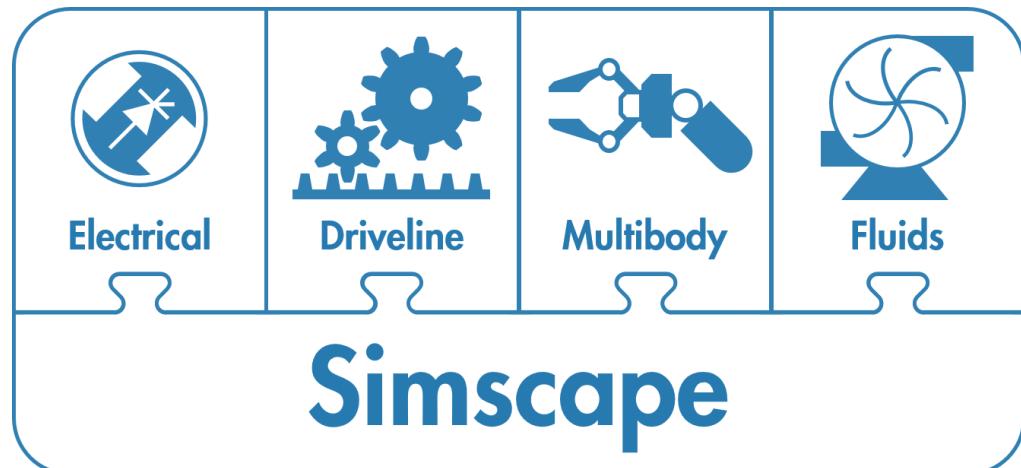
Simscape Products

- Simscape platform
 - Foundation libraries in 8 domains
 - Language for defining custom blocks
 - Extension of MATLAB
 - Simulation engine and custom diagnostics
- Simscape add-on libraries
 - Extend foundation domains with components, effects, parameterizations
 - Models can be converted to C code

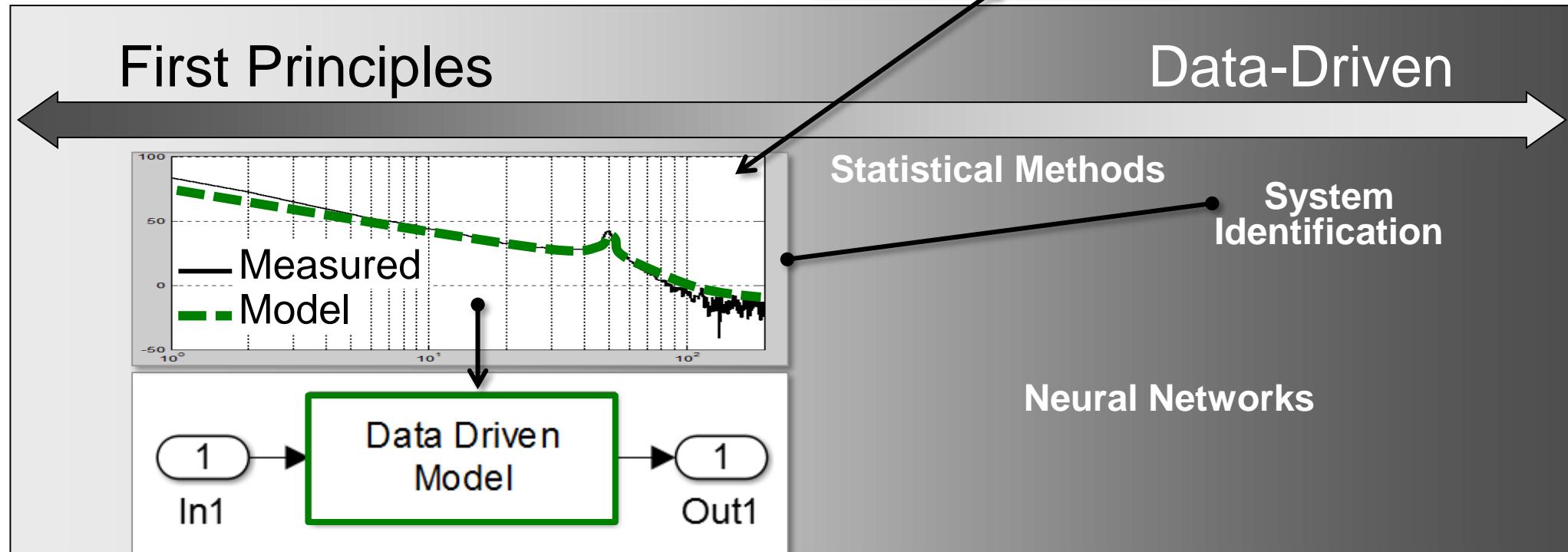
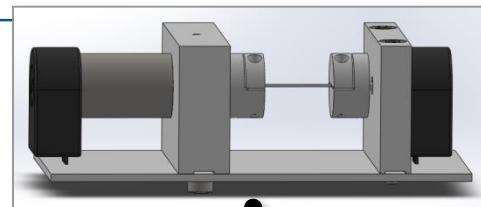


Simscape Add-on Libraries

- Simscape Electricals
 - Actuators, sensors, and semiconductors
 - Three-phase electrical networks
- Simscape Fluids
 - Pumps, actuators, pipelines, valves, tanks
- Simscape Multibody
 - Multibody systems: joints, bodies, frames
- Simscape Driveline
 - Gears, leadscrew, clutches, tires, engines



Modeling Approaches



- Purpose: Model an existing design (real or virtual)
- Requirements:
 - Relevant set of measured data is available
 - Design and physical parameters will not be changed

Modeling Approach for unknown Actuator

Data-Driven Approach : System Identification

Input



Voltage

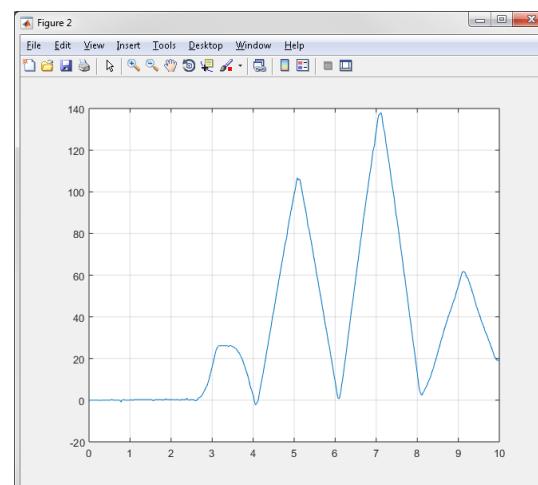
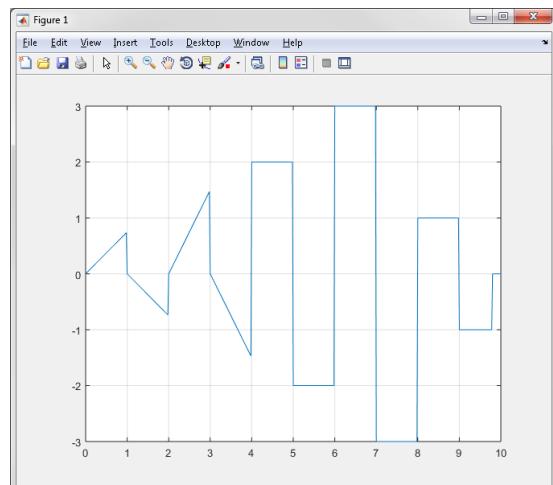


Output



Angle or
Angular Velocity

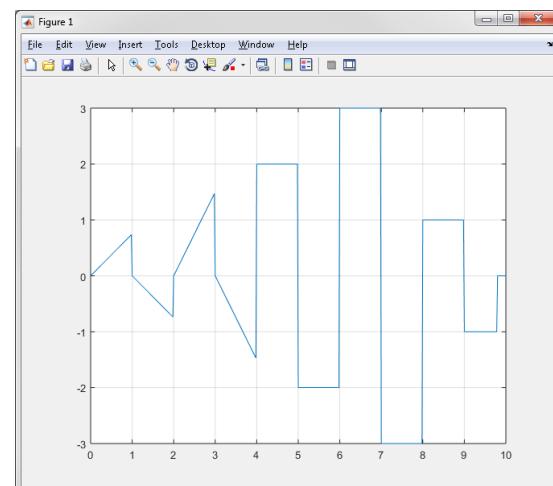
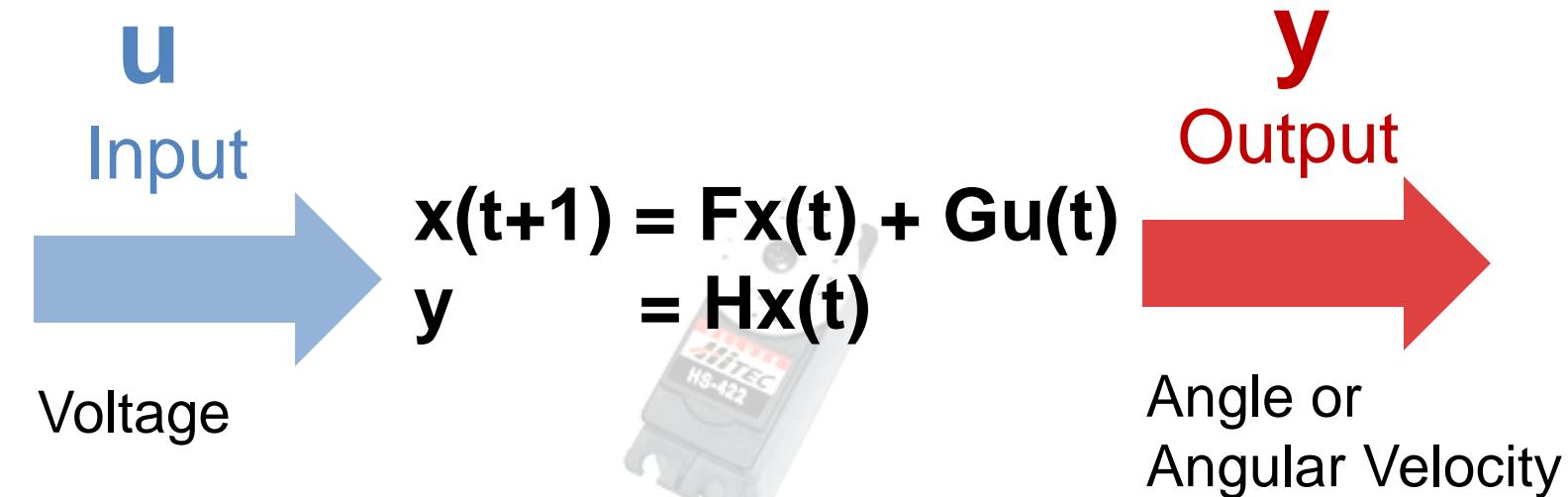
System



>>load measuredData.mat

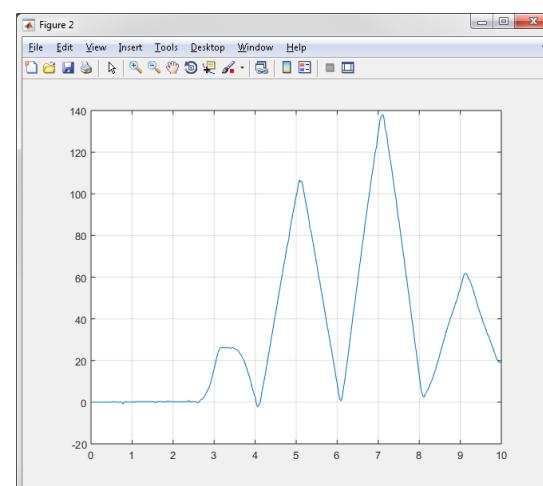
Modeling Approach for unknown Actuator

Data-Driven Approach : System Identification



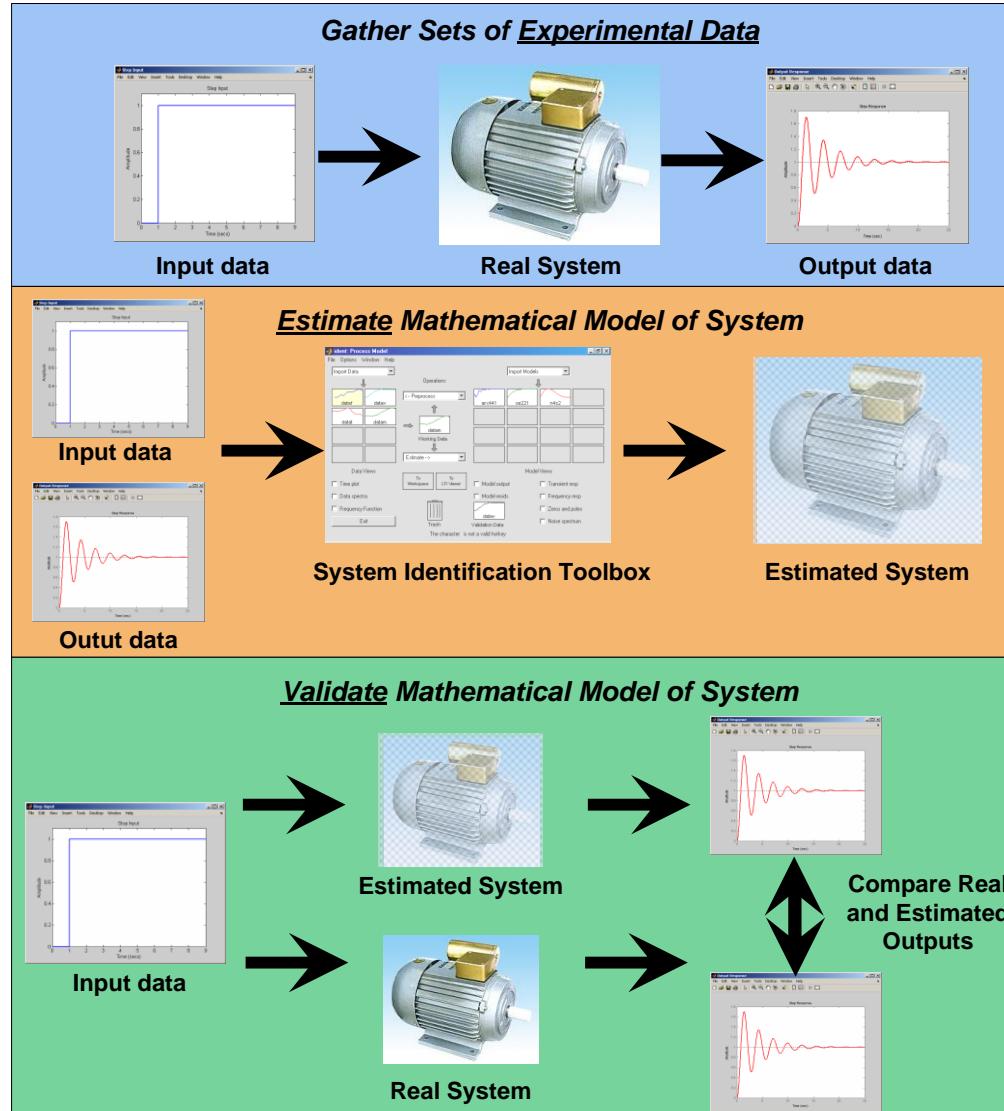
Model

u, y : measured time or frequency domain signals



System Identification Process

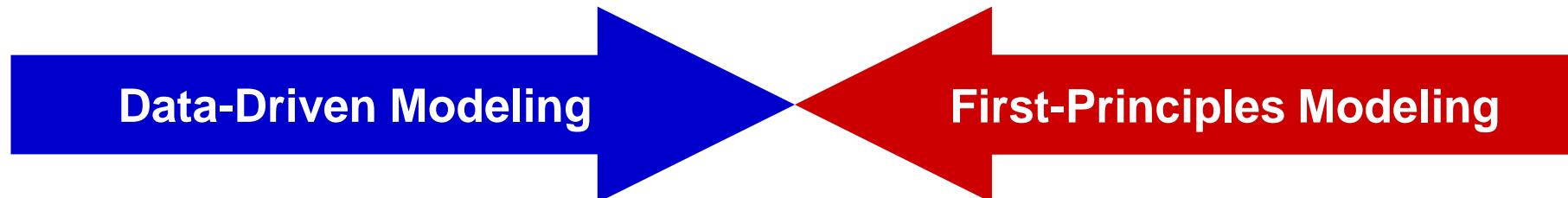
1. Gather experimental data.



2. Estimate model from data.

3. Validate model with independent data.

Both have advantages & disadvantages



Advantages:

- Fast
- Accurate

Disadvantages:

- Requires plant
- Requires data acquisition system

First-Principles Modeling

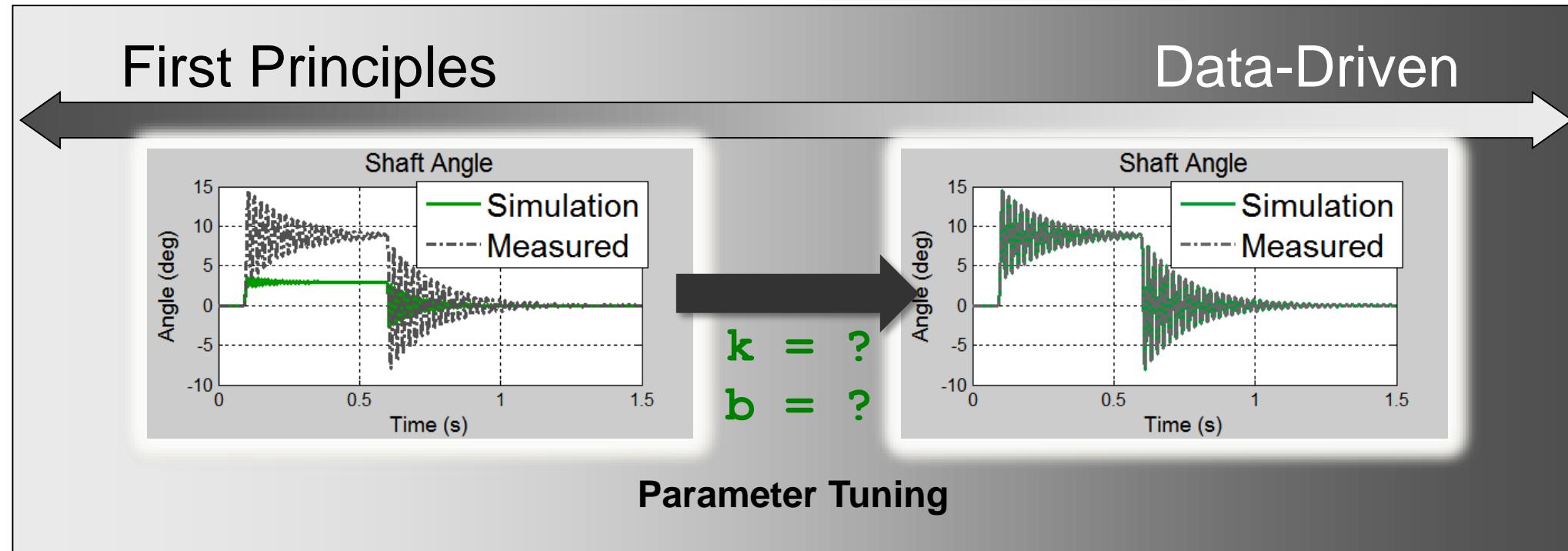
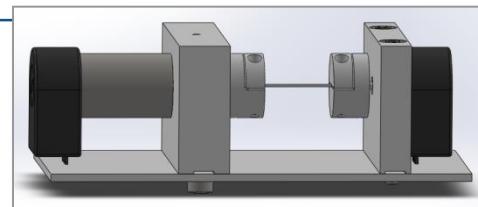
Advantages:

- Insight in behavior
- Physical parameters

Disadvantages:

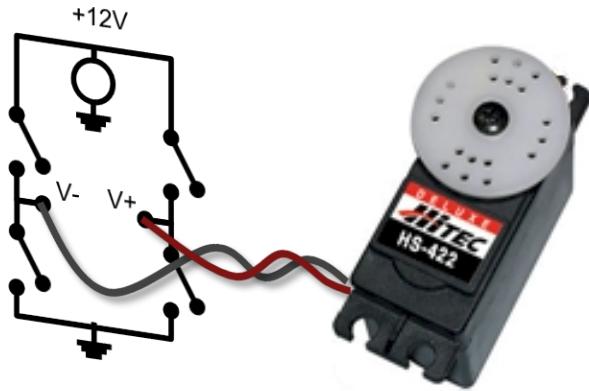
- Time-consuming
- Requires expertise

Modeling Approaches



- Purpose: Ensuring parameter values are accurate
- Requirements:
 - Relevant set of measured data is available
 - Physically meaningful parameters can be automatically tuned

Validate your models against experimental data using automatic parameter estimation



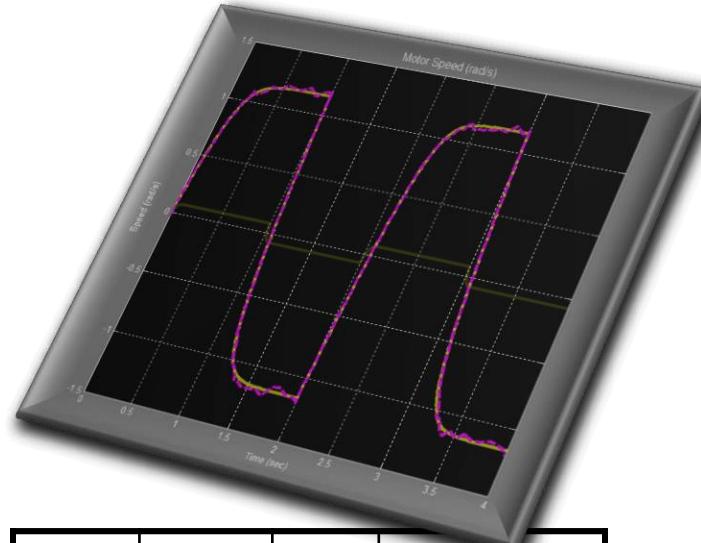
R = Resistance

L = Inductance

J = Inertia

B = Friction

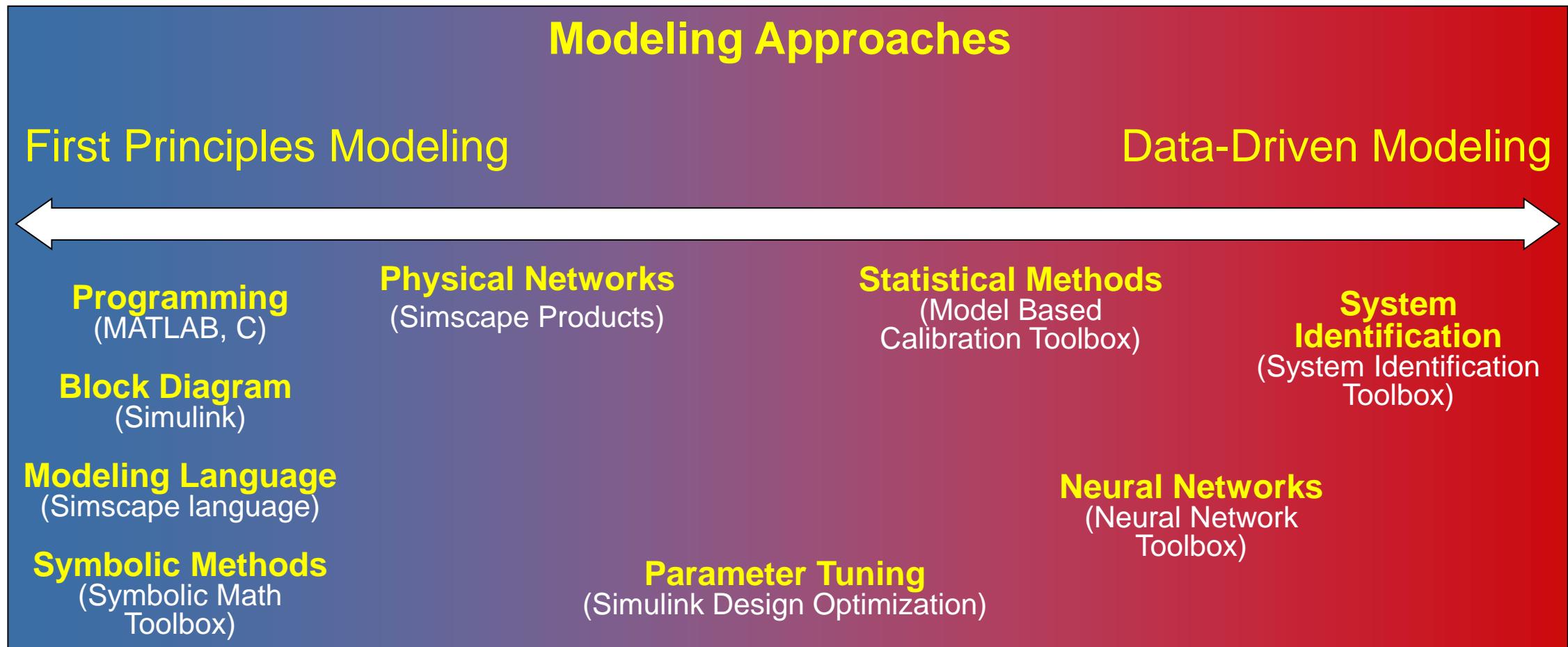
K = Back EMF Constant



R	L	J	K	B
4.03	1e-4	0.11	0.45	1.07

Use **Simulink Design Optimization** to tune model parameters by taking advantage of the capabilities in the **Optimization Toolbox** directly in Simulink

Modeling Physical Systems With MathWorks Products



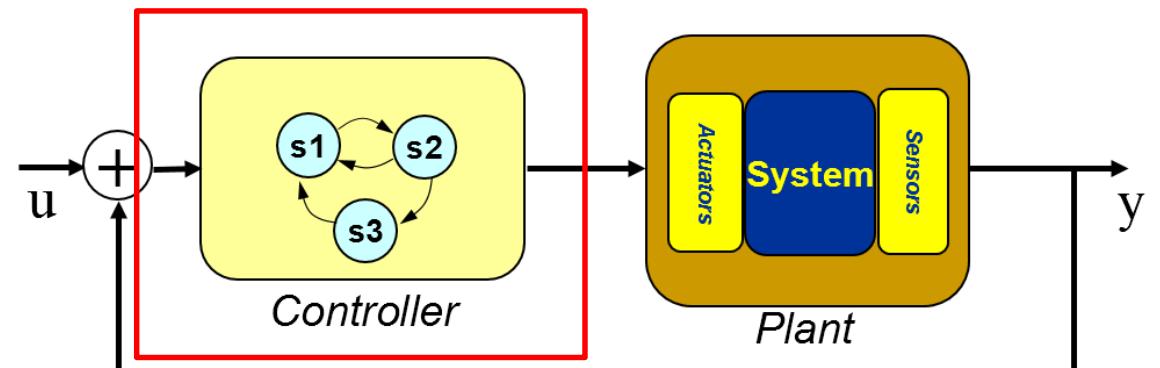
Control Design Process

- **Modeling:**

understanding your design problem

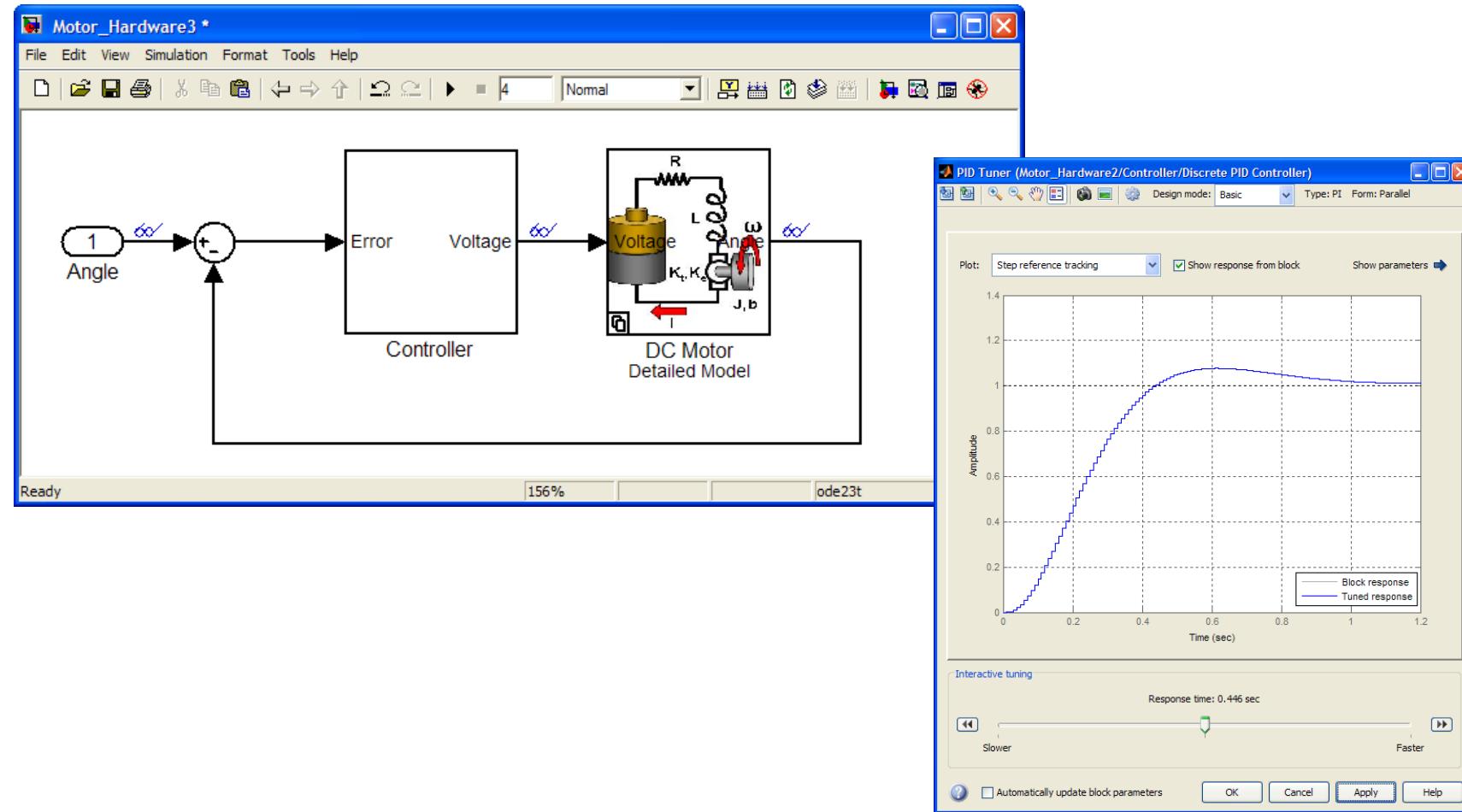
- **Control Design:**

solving your design problem

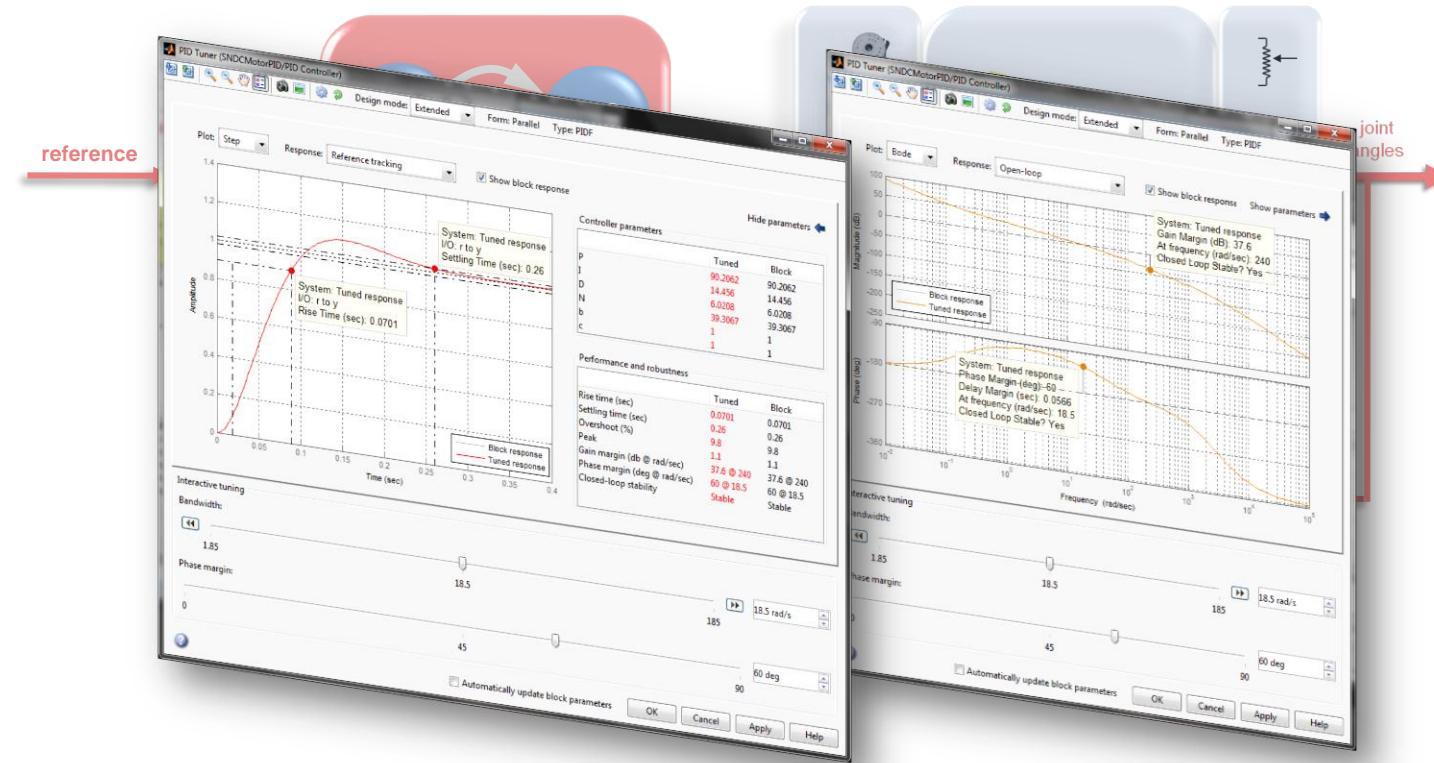


DC Motor Controller Design

PID block with Simulink Control Design



Automatic PID tuning

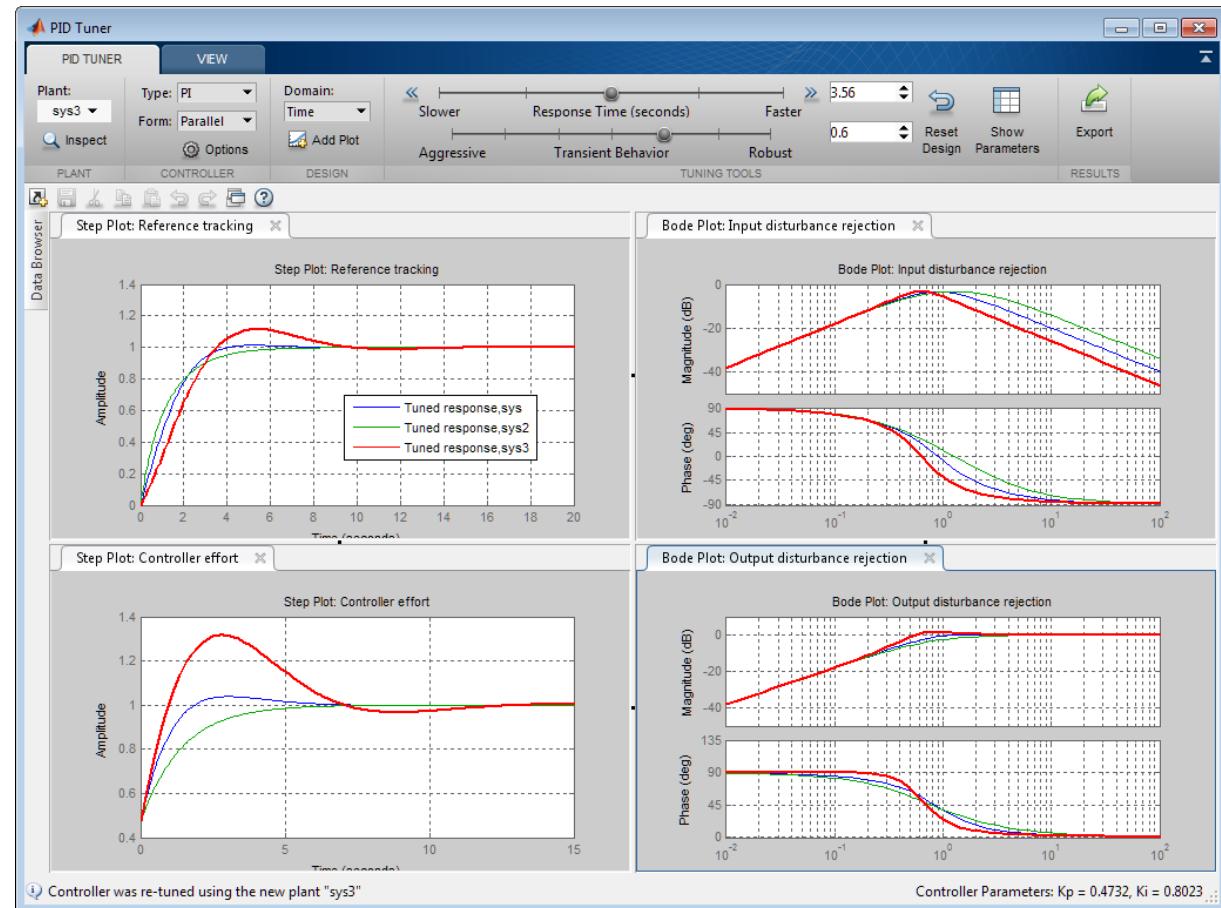


Use **Simulink Control Design** and the **Control System Toolbox** to automatically linearize the plant, design and tune your PID controllers

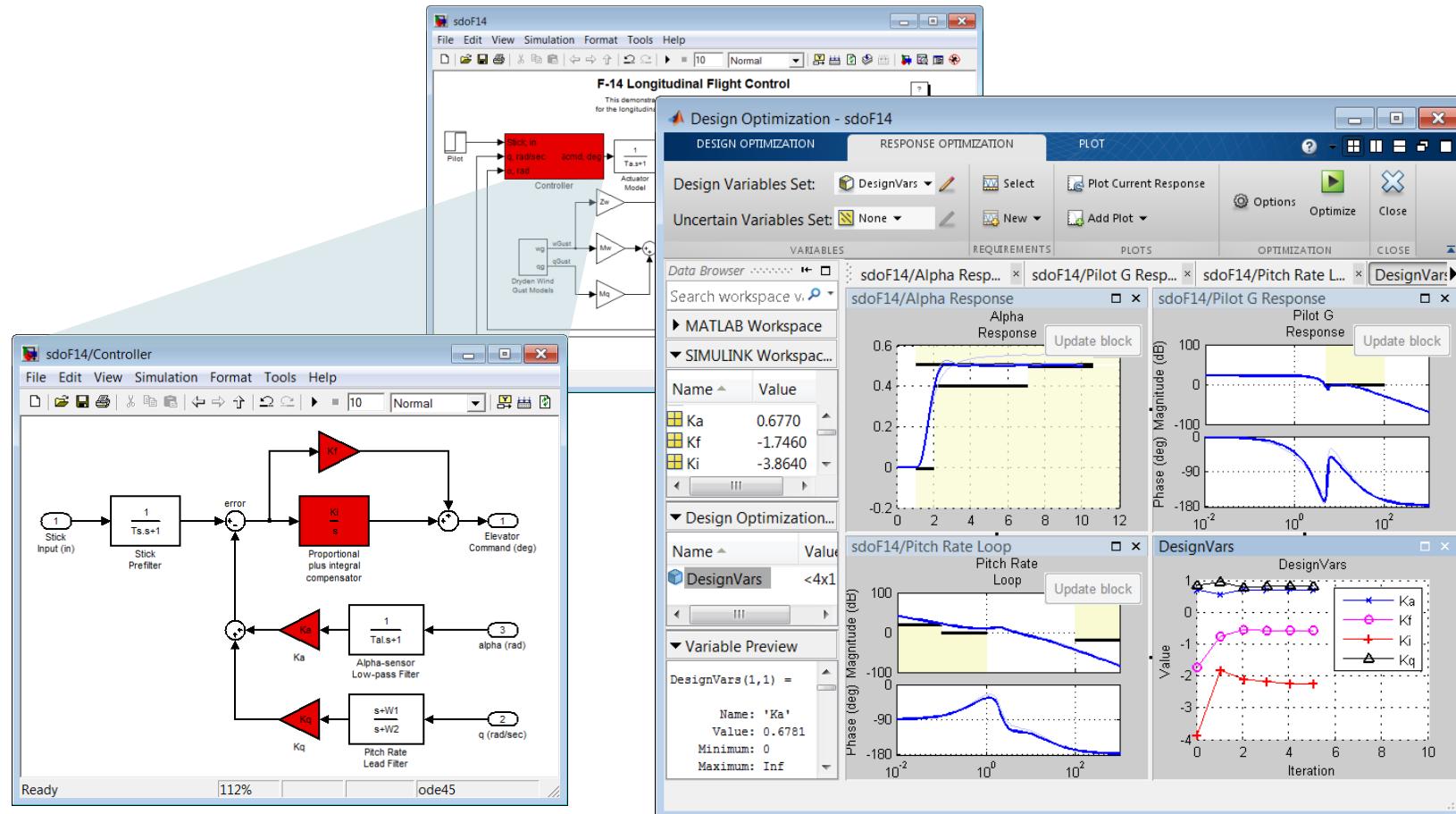
PID Tuner App

Interactive tuning of PID controllers

- Automatically finds the design that balances performance and robustness
- Lets you easily try different controller structures
- Provides two sliders for fine-tuning the design
- Several response plots can be displayed simultaneously

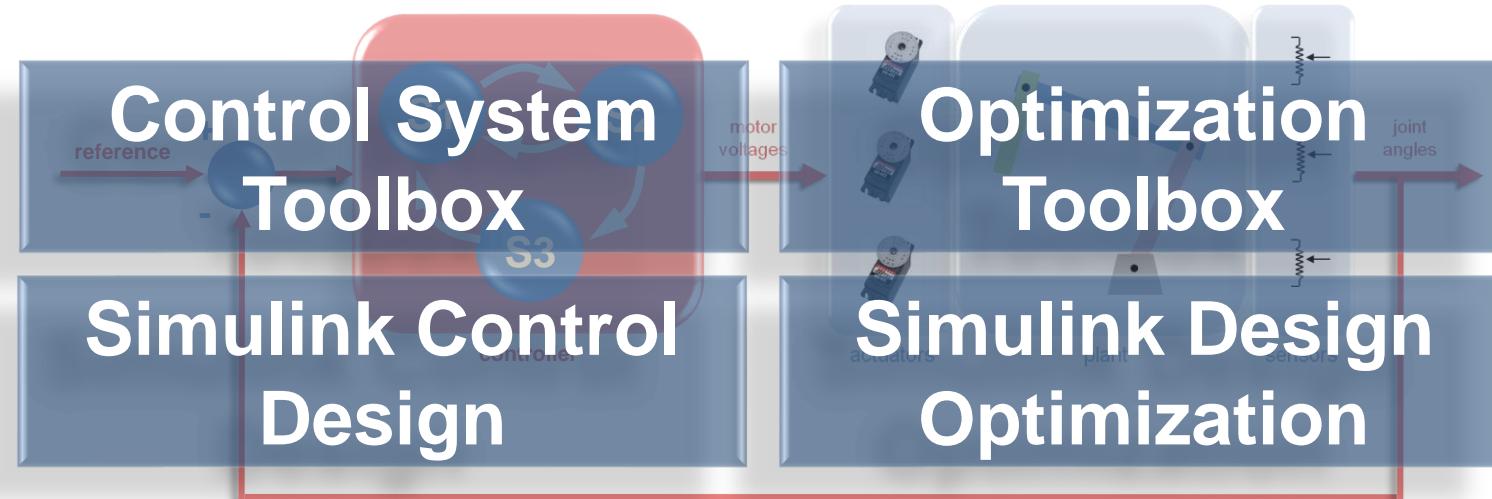


Fine tune controller gains using response optimization



Use **Simulink Design Optimization** to optimize overall system response against requirements in the time and frequency domain

Base/recommended control design tools



CONTROL TUTORIALS FOR MATLAB® & SIMULINK®

INTRODUCTION CRUISE CONTROL MOTOR SPEED MOTOR POSITION SUSPENSION INVERTED PENDULUM AIRCRAFT PITCH BALL & BEAM

SYSTEM

MODELING

ANALYSIS

CONTROL

PID

ROOT LOCUS

FREQUENCY

STATE-SPACE

DIGITAL

SIMULINK

MODELING

CONTROL



CONTROL TUTORIALS FOR MATLAB® & SIMULINK®

Welcome to the Control Tutorials for MATLAB and Simulink (CTMS): They are designed to help you learn how to use MATLAB and Simulink for the analysis and design of automatic control systems. They cover the basics of MATLAB and Simulink and introduce the most common classical and modern control design techniques.



About the Authors: These tutorials were originally developed by Prof. Bill Messner at Carnegie Mellon and Prof. Dawn Tilbury at the University of Michigan with funding from NSF. With further support from the MathWorks in 2011 and 2017, Prof. Messner, Assoc. Prof. Rick Hill (Detroit Mercy), and PhD Student JD Taylor (CMU), expanded the tutorials, completely redesigned the web interface, and

All contents licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



Offline Control Tutorials for ML & SL



제품 솔루션 아카데미아 지원 커뮤니티 이벤트 안내

MATLAB 구매



MATLAB Courseware

MathWorks.com내 검색



"Control Tutorials for MATLAB and Simulink" Courseware

Thank you for your interest in *Control Tutorials for MATLAB and Simulink* courseware:

[Download course materials \(.zip\)](#)



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Additional Resources

- [Explore course web page](#)
- [Browse all MATLAB Courseware](#)

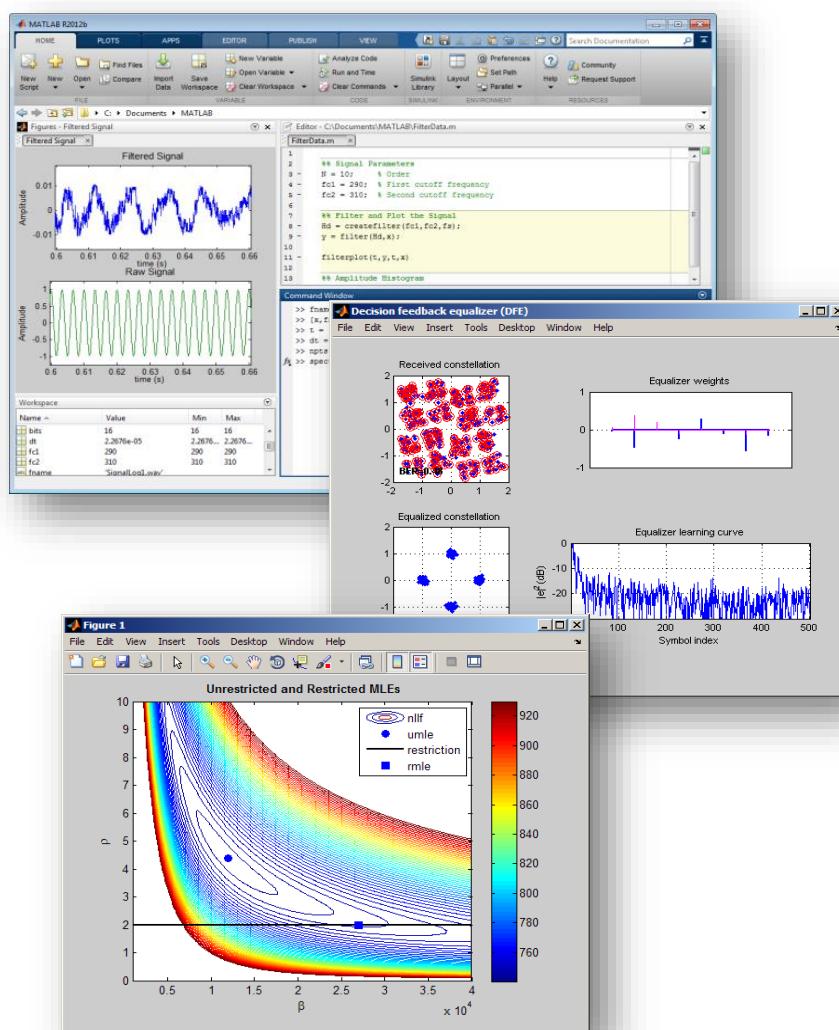
Introduction to MATLAB®

Agenda

- What is MATLAB?
- Getting started with MATLAB
- Data Generation
- Data Indexing
- Calculation
- Visualization
- Programming Keywords
- Script
- Function

What is MATLAB?

- The leading environment for technical computing
- High-level language
 - Native support for vector and matrix operations
 - Built-in math and visualization functions
- Interactive development environment
 - Interactive and easy to get started
 - Ideal for iterative exploration and design
- Used for:
 - Numerical computation
 - Data analysis and visualization
 - Algorithm development and programming
 - Application development and deployment



MATLAB
is the **easiest** and
most productive environment
for engineers and scientists

How do you feel to learn programming language?



How to Make Program about Drawing Sine Wave

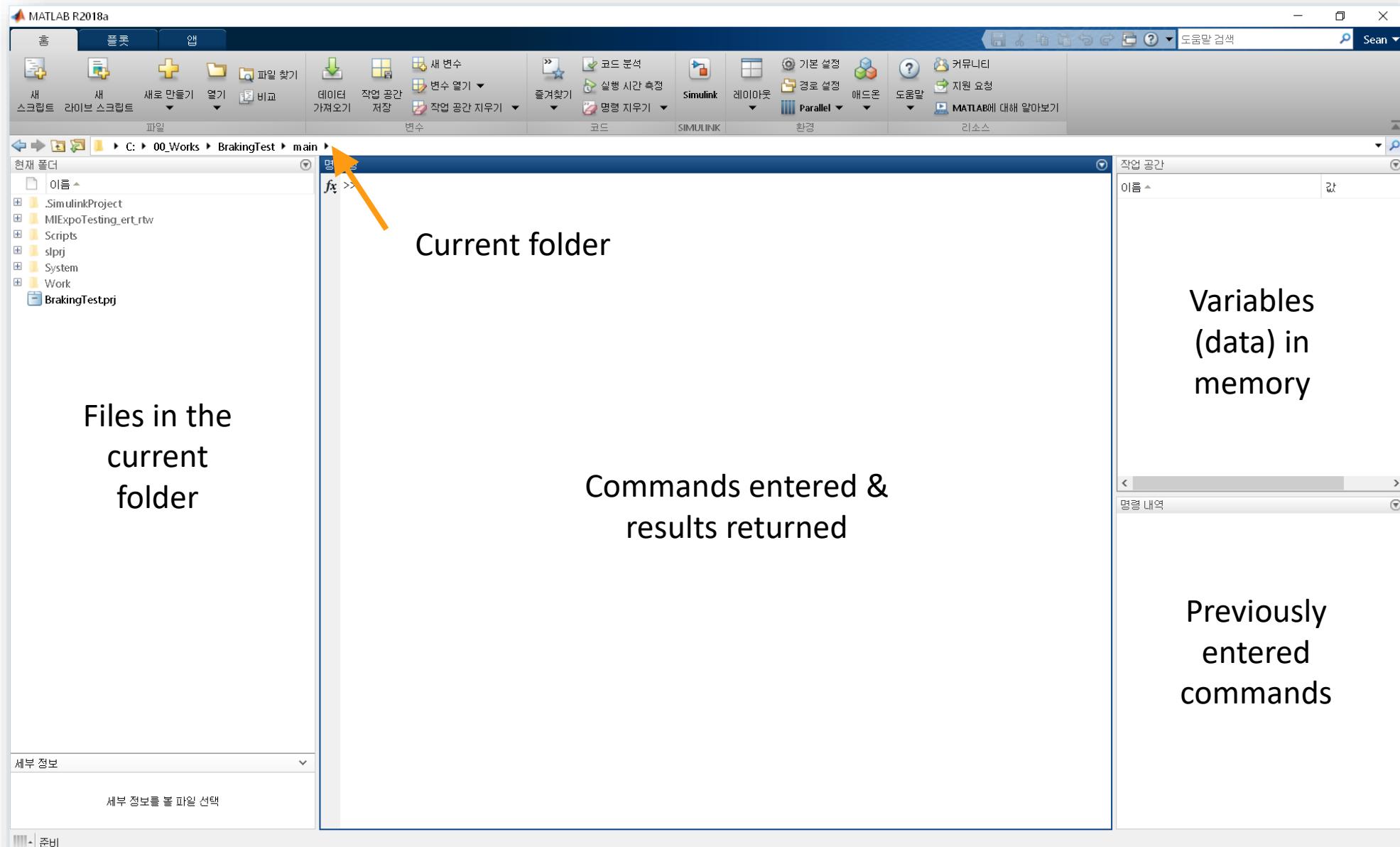
```
void mysin(double y[63])
{
    int i0;
    static const double x[63] = { 0.0, 0.1, 0.2, 0.3000000000000004, 0.4, 0.5,
        0.6000000000000009, 0.7000000000000007, 0.8, 0.9, 1.0, 1.1,
        1.2000000000000002, 1.3, 1.4000000000000001, 1.5, 1.6, 1.7000000000000002,
        1.8, 1.9000000000000001, 2.0, 2.1, 2.2, 2.3000000000000003,
        2.4000000000000004, 2.5, 2.6, 2.7, 2.8000000000000003, 2.9000000000000004,
        3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.3, 4.4,
        4.5, 4.6, 4.7, 4.8, 4.9, 5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9,
        6.0, 6.1000000000000005, 6.2 };

    for (i0 = 0; i0 < 63; i0++) {
        y[i0] = sin(x[i0]);
    }
}
```

Graph?

```
>> t=0:0.1:2*pi;
>> y=sin(t);
>> plot(t,y)
fx >>
```

The MATLAB® Desktop



Files in the
current
folder

Current folder

Commands entered &
results returned

Variables
(data) in
memory

Previously
entered
commands

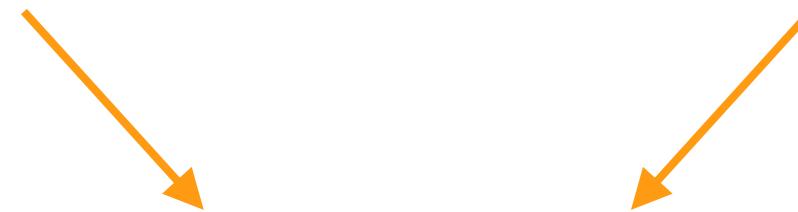
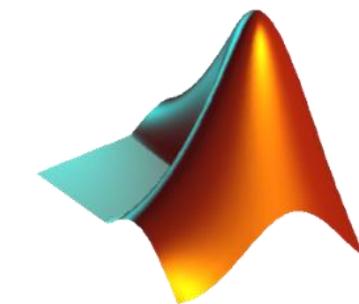
Data Generation

Generating Data in MATLAB®

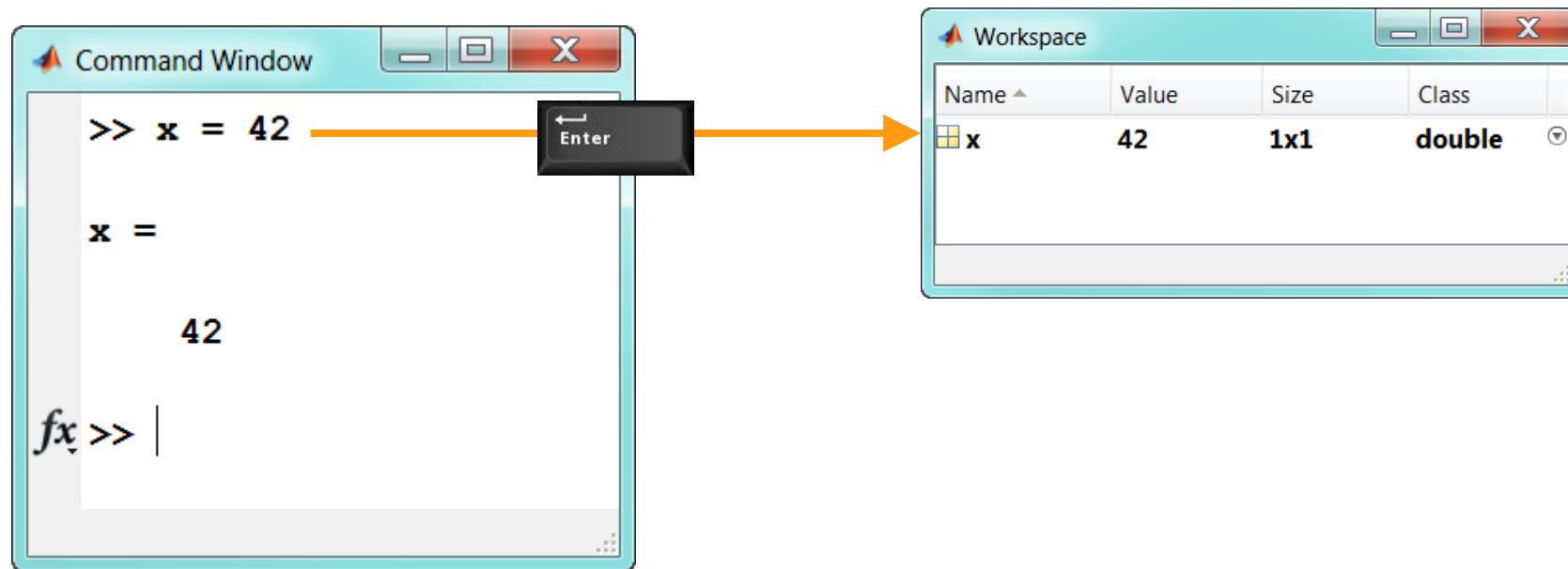


Generating data
from external source

Generating data
from user in ML



Creating Numeric Data from Commands



`>> variable_name1=number or calculable equations;`

Creating Scalar, Vectors, and Matrices

1. Scalar

```
>> variable_name=num; or variable_name=[num];
```

3.1

2. Vector

```
>> variable_name=[num1; num2];
```

3.1

```
>> variable_name=[num1, num2];
```

4.2

```
>> variable_name=a:dx:b; or variable_name=linspace(a,b,n);
```

9.0

6	8	10	12	14	16
---	---	----	----	----	----

3. Matrix

```
>> variable_name=[num1, num2; num3 num4];
```

1	0	0
0	1	0
0	0	1

```
>> variable_name=function(num1,num2);
```

Concatenating Arrays

A	<table border="1"><tr><td>2</td><td>3</td></tr><tr><td>5</td><td>7</td></tr><tr><td>11</td><td>13</td></tr></table>	2	3	5	7	11	13
2	3						
5	7						
11	13						

B	<table border="1"><tr><td>-1</td><td>1</td></tr><tr><td>1</td><td>-1</td></tr></table>	-1	1	1	-1
-1	1				
1	-1				

C	<table border="1"><tr><td>0</td></tr><tr><td>8</td></tr><tr><td>0</td></tr></table>	0	8	0
0				
8				
0				

```
>> X = [A;B]
```

2	3
5	7
11	13
-1	1
1	-1

```
>> Y = [A,C]
```

2	3	0
5	7	8
11	13	0

```
>> Y = cat(2, A, C)
```

2	3	0
5	7	8
11	13	0

Importing Supported File Data from External

```
>> docsearch('Supported File Formats for Import and Export')
```



Audio



Excel



Text



Video

...

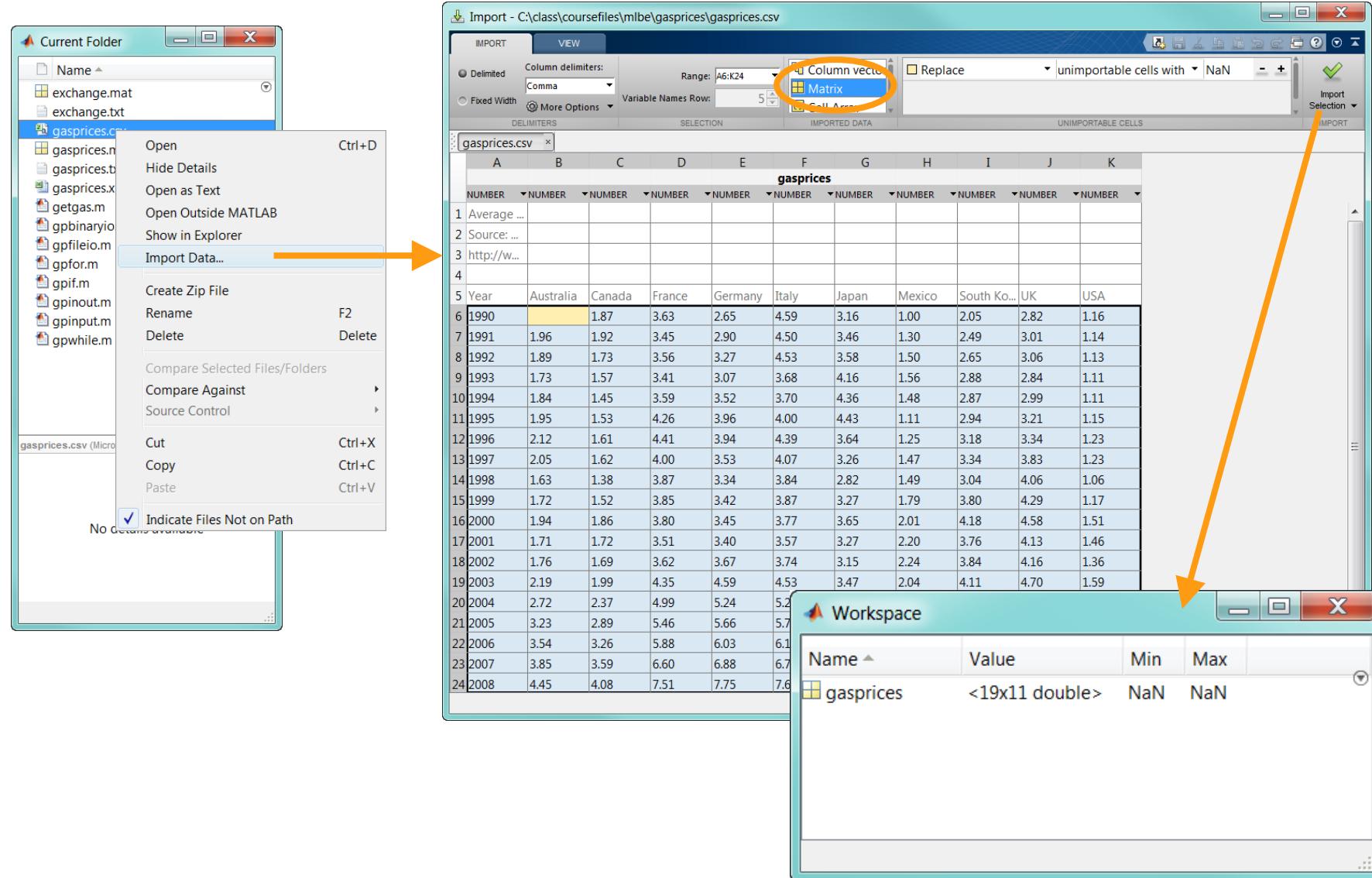
`data = xyzread('myfile')`

Ex) `audioread()`
`xlsread()`
`dlmread()`
`imread()`

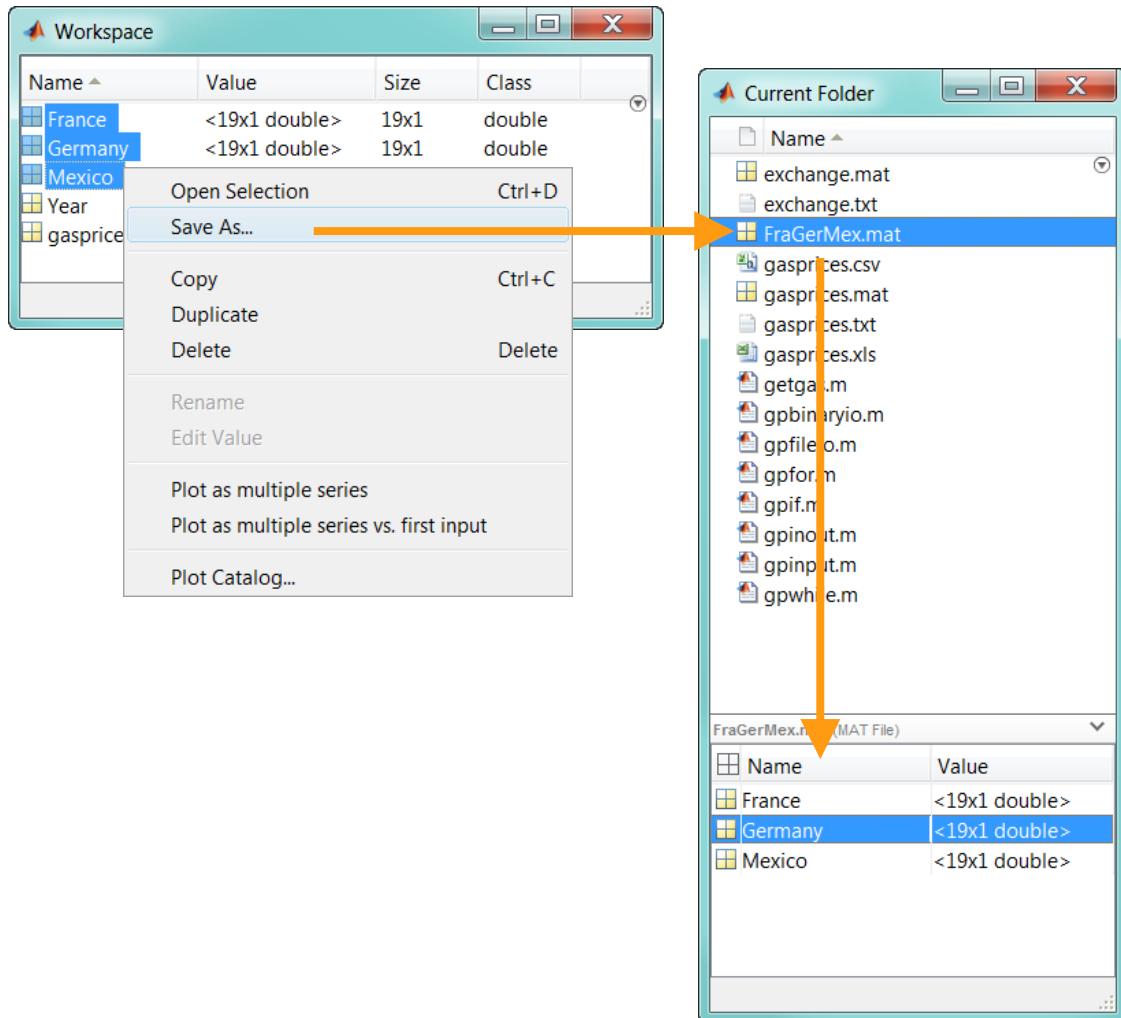


Name	Size	Class
Australia	19x1	double
Canada	19x1	double
France	19x1	double
Germany	19x1	double
Italy	19x1	double
Japan	19x1	double
Mexico	19x1	double
SouthKorea	19x1	double
UK	19x1	double
USA	19x1	double
Year	19x1	double

Interactive Importing



Saving and Loading Variables



```
>> save filename variable_name1 variable_name2  
>> load filename.mat
```

Data Indexing

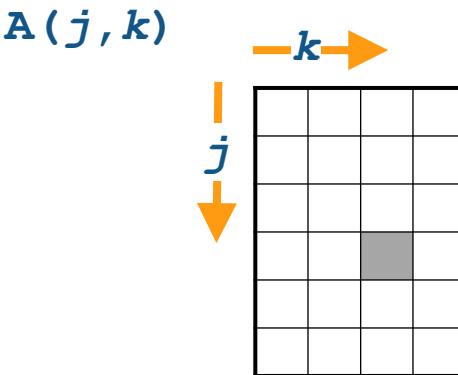
Row-Column Indexing

```
>> A = [1,2;3,4];  
A =  
    1 (1,1) 2 (1,2)  
    3 (2,1) 4 (2,2)
```

```
>> x = A(1 Row Indices ,2 Column Indices )  
x = 2
```

```
>> Y = A(1:2,2)  
Y =  
    2  
    4
```

```
>> z = A(end,:)  
z = 3 4
```



Linear Indexing

```
>> A = magic(5)
```

```
A =
```

1	17	6	24	11	1	16	8	21	15
2	23	7	5	12	7	17	14	22	16
3	4	8	6	13	13	18	20	23	22
4	10	9	12	14	19	19	21	24	3
5	11	10	18	15	25	20	2	25	end 9

Indices Data

```
>> x = A(20)
```



Calculation

Array Operations

1. +, -, *, ^, / (A/B=A*inv(B)), \ (A\b=inv(A)*B)

2. .* , .^ , ./ (A elements/B elements), .\ (B elements\A elements)

$$\gg C = A + B \qquad \gg C = A + B$$

$$\begin{pmatrix} 3 \\ 5 \\ 4 \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \textcolor{blue}{+} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \begin{pmatrix} 4 \\ 5 \\ 3 \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \textcolor{blue}{+} \begin{pmatrix} 2 \end{pmatrix}$$

$$\gg C = A * B \qquad \gg C = A \backslash B$$

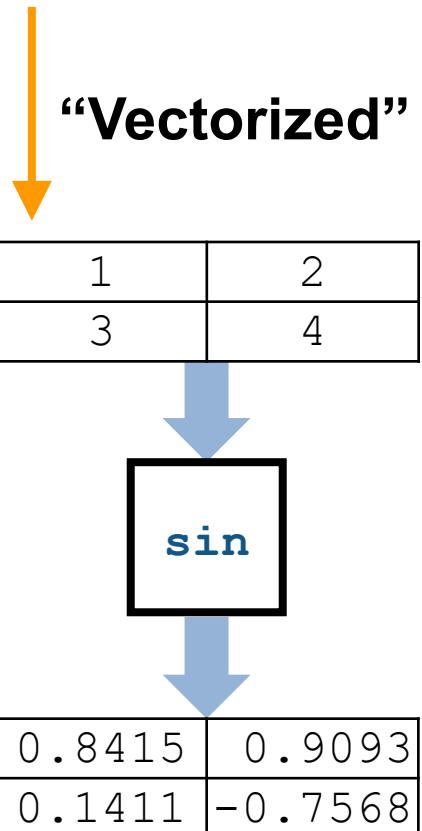
$$\text{Error} \equiv \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \textcolor{blue}{\times} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \begin{pmatrix} -1 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix} \textcolor{blue}{\backslash} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\gg C = A .* B \qquad \gg C = A ./ B$$

$$\begin{pmatrix} 2 \\ 6 \\ 3 \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \textcolor{blue}{\cdot} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \begin{pmatrix} 2 \\ 1.5 \\ 0.33 \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \textcolor{blue}{./} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

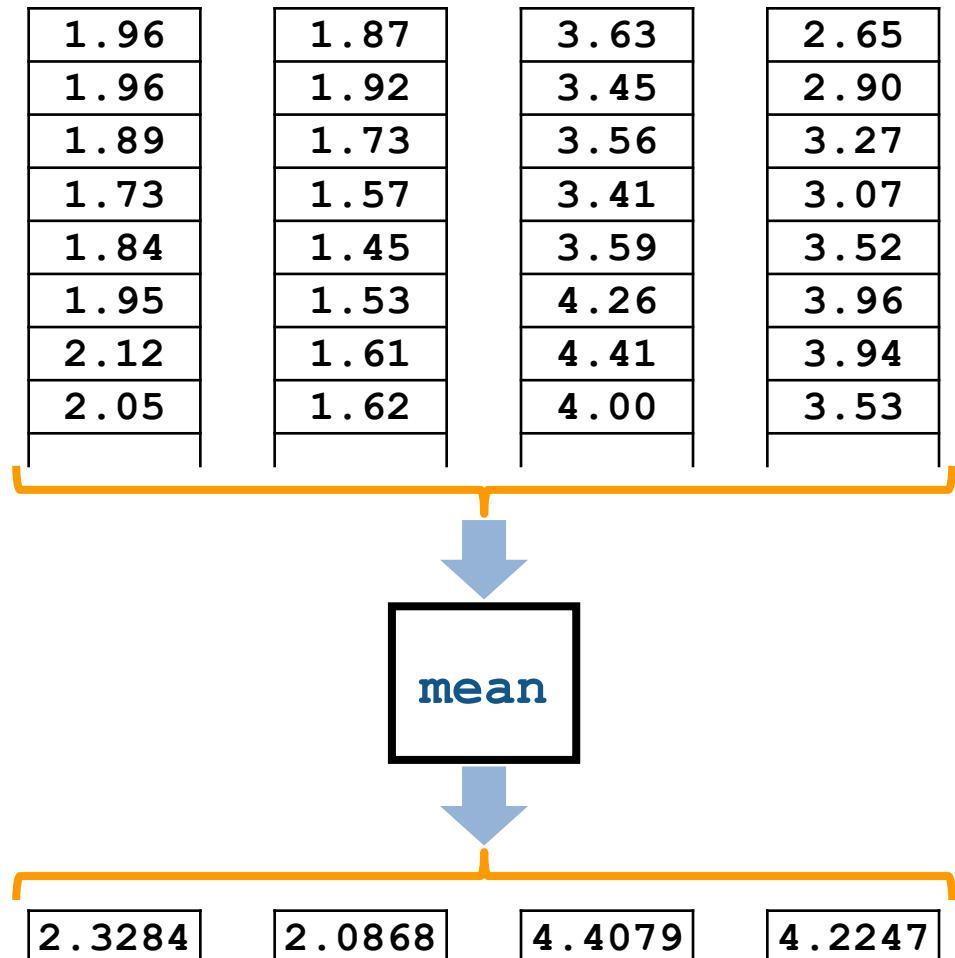
Mathematical functions

```
>> B = sin(A);
```



sin
sind
sinh
asin
exp
log
log2
log10
sqrt
nthroot
abs
angle
floor
ceil
round
mod

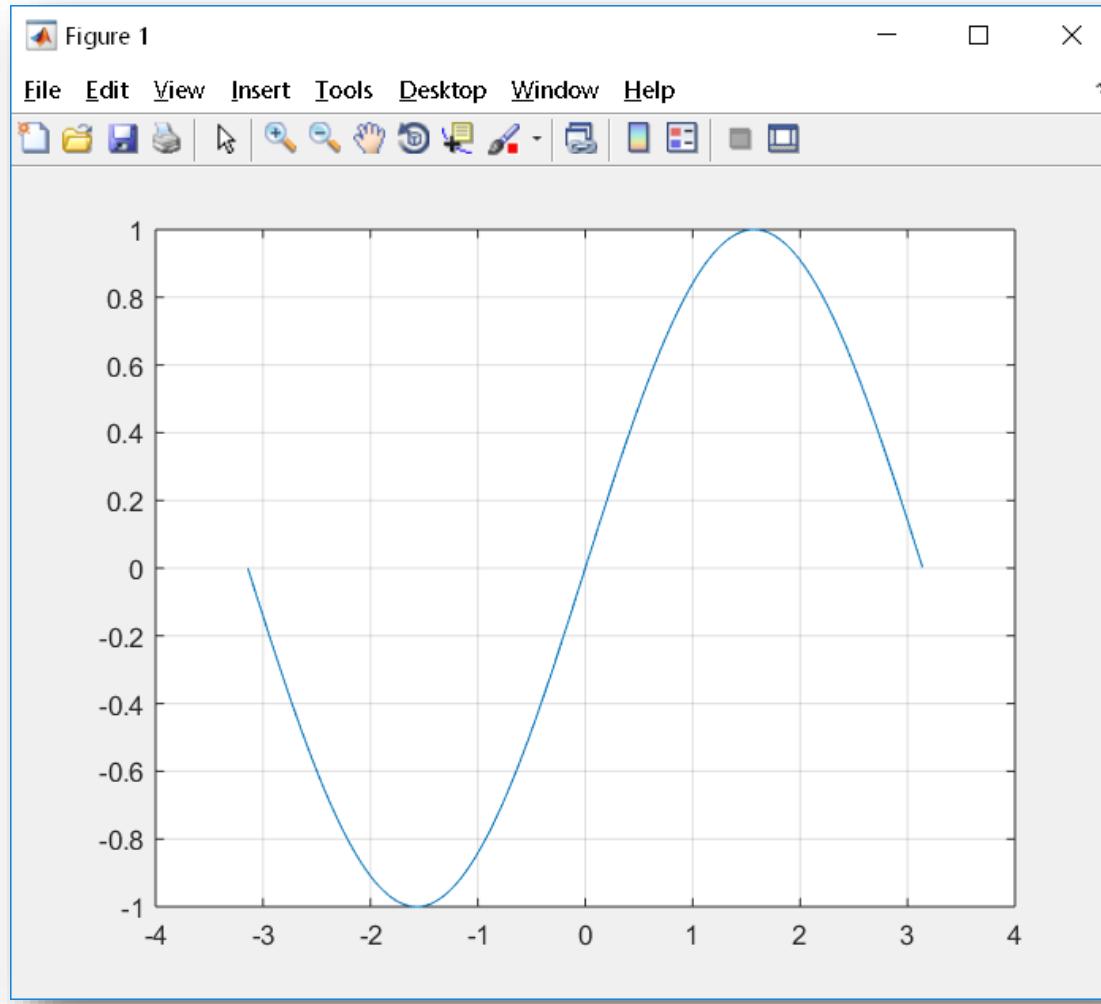
Statistical functions



max
min
mean
median
std
sum
prod
diff
gradient
cumsum
cumprod
corrcoef
cov

Visualization

Plot Function



Plot Sine Function

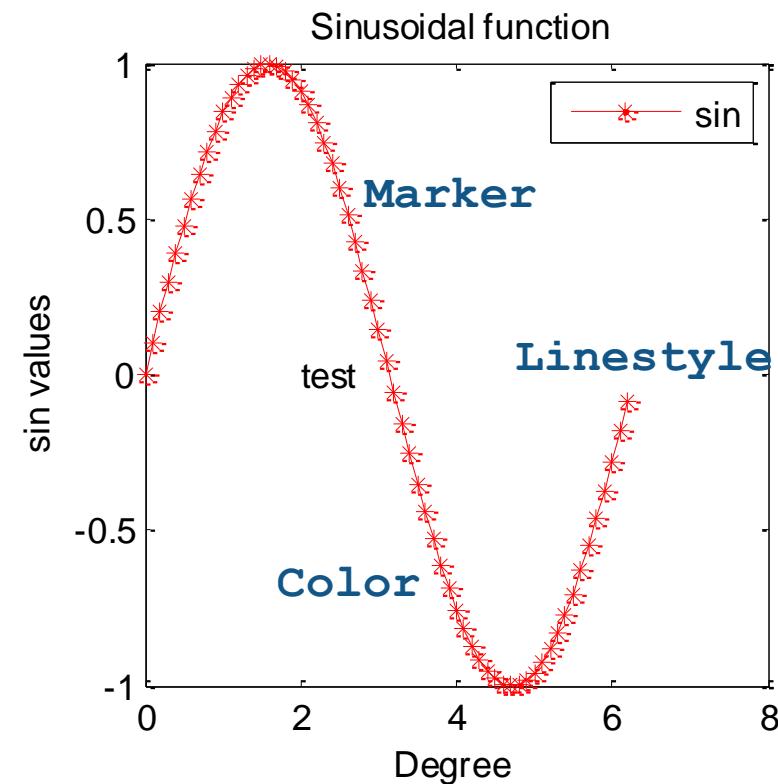
Plot the sine function over the domain $-\pi \leq x \leq \pi$

```
x = -pi:0.01:pi;
y = sin(x);
plot(x,y), grid on
```

Plot Options

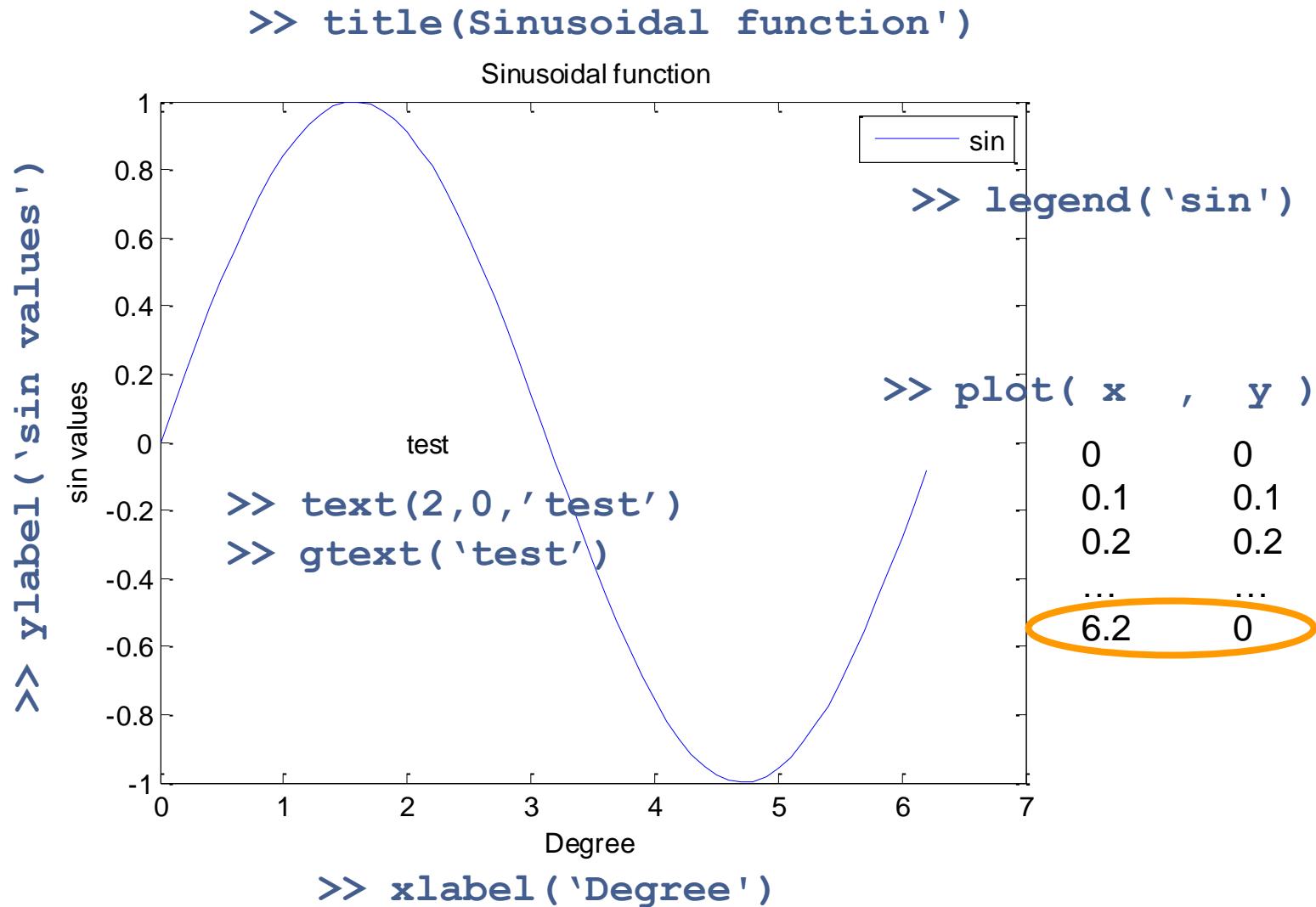
Color	Marker	Line Style
b blue	.	solid
g green	o	dashed
r red	x	dotted
c cyan	+	dash-dot
m magenta	*	
y yellow	s	
k black	d	
w white	v	
	^	
	<	
	>	
	p	
	h	

`>> plot(x,y,'PropertyName',Value)`

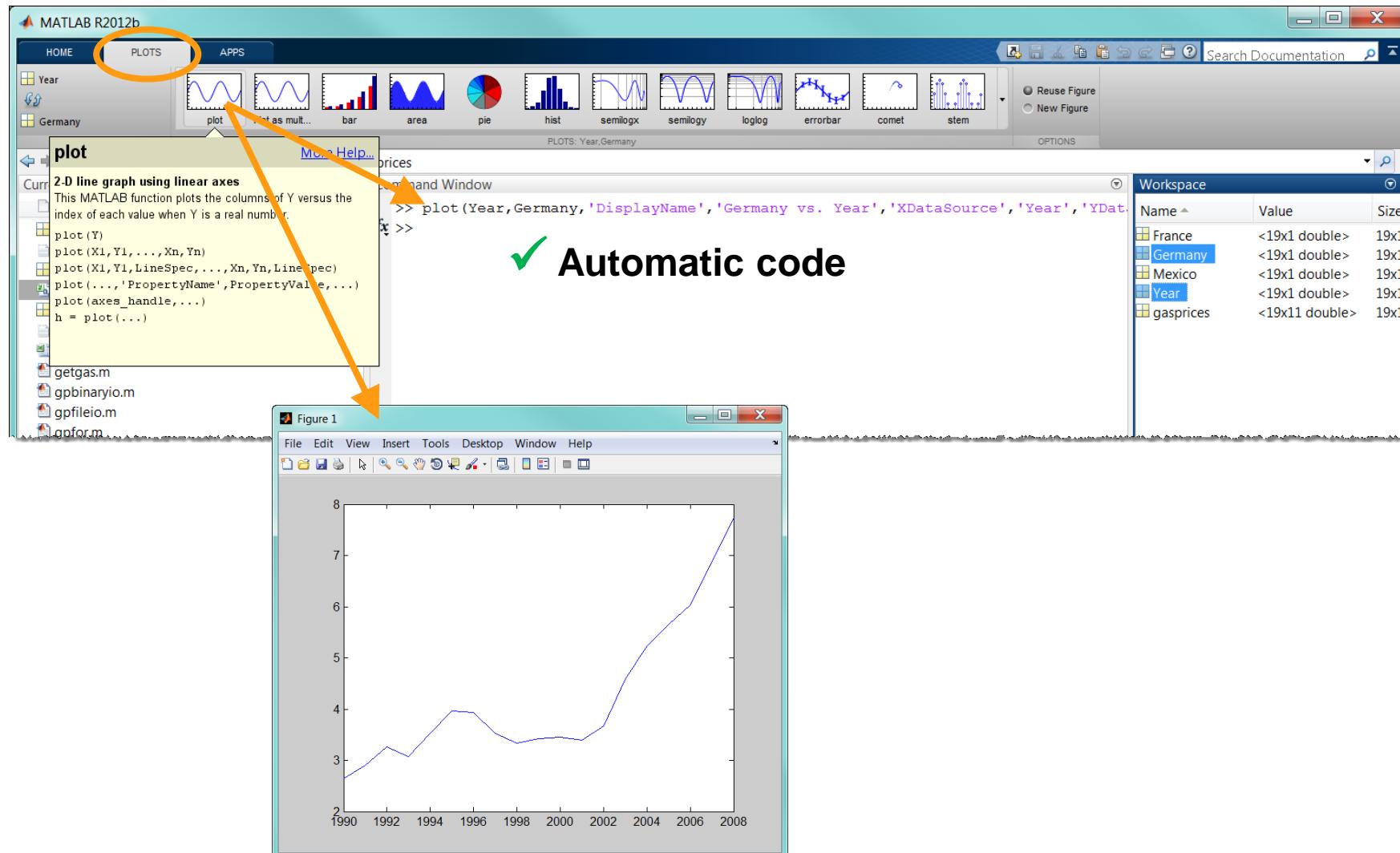


`>> plot(x,y,'r*-')`

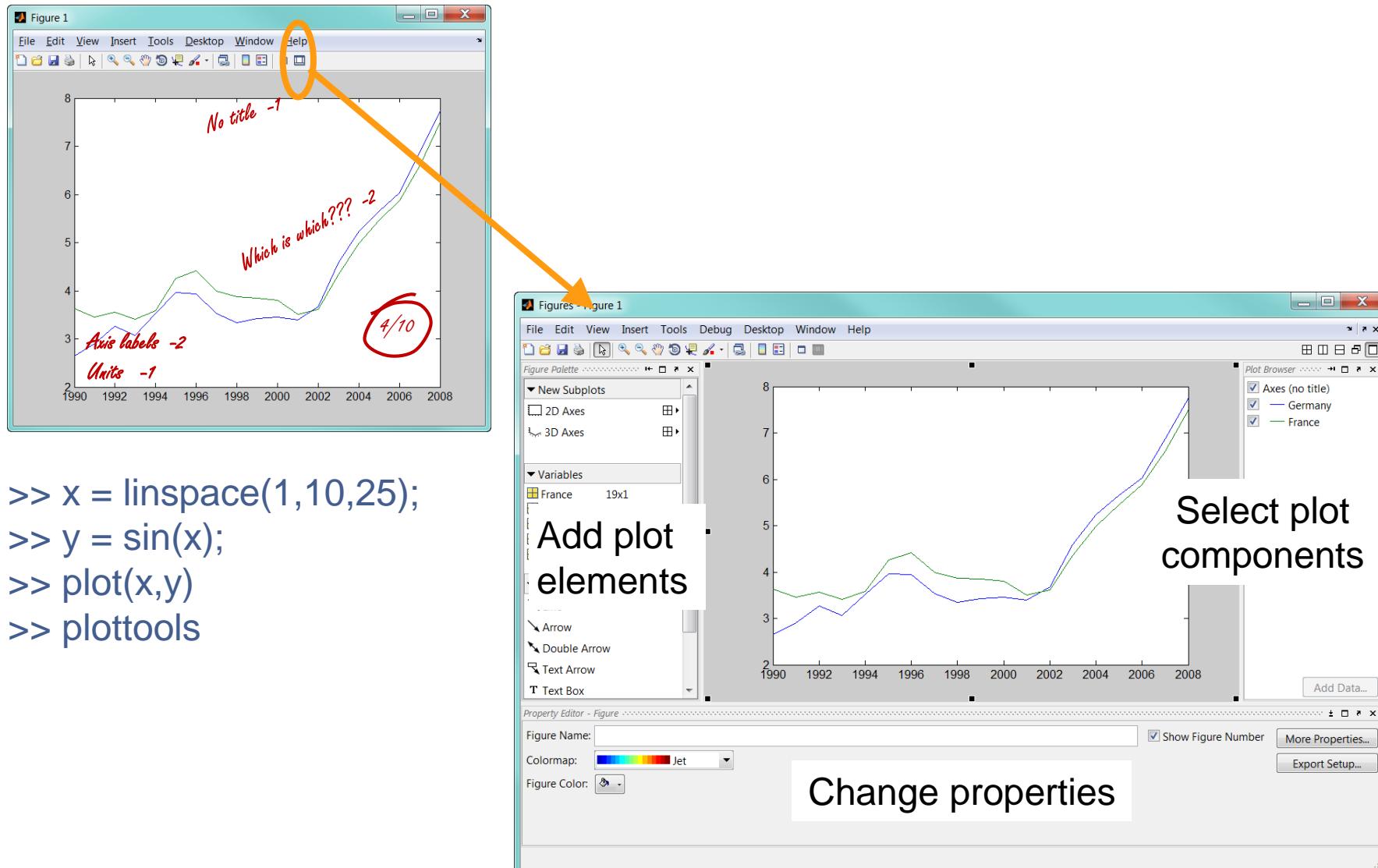
Annotating Plots



Plotting the Data

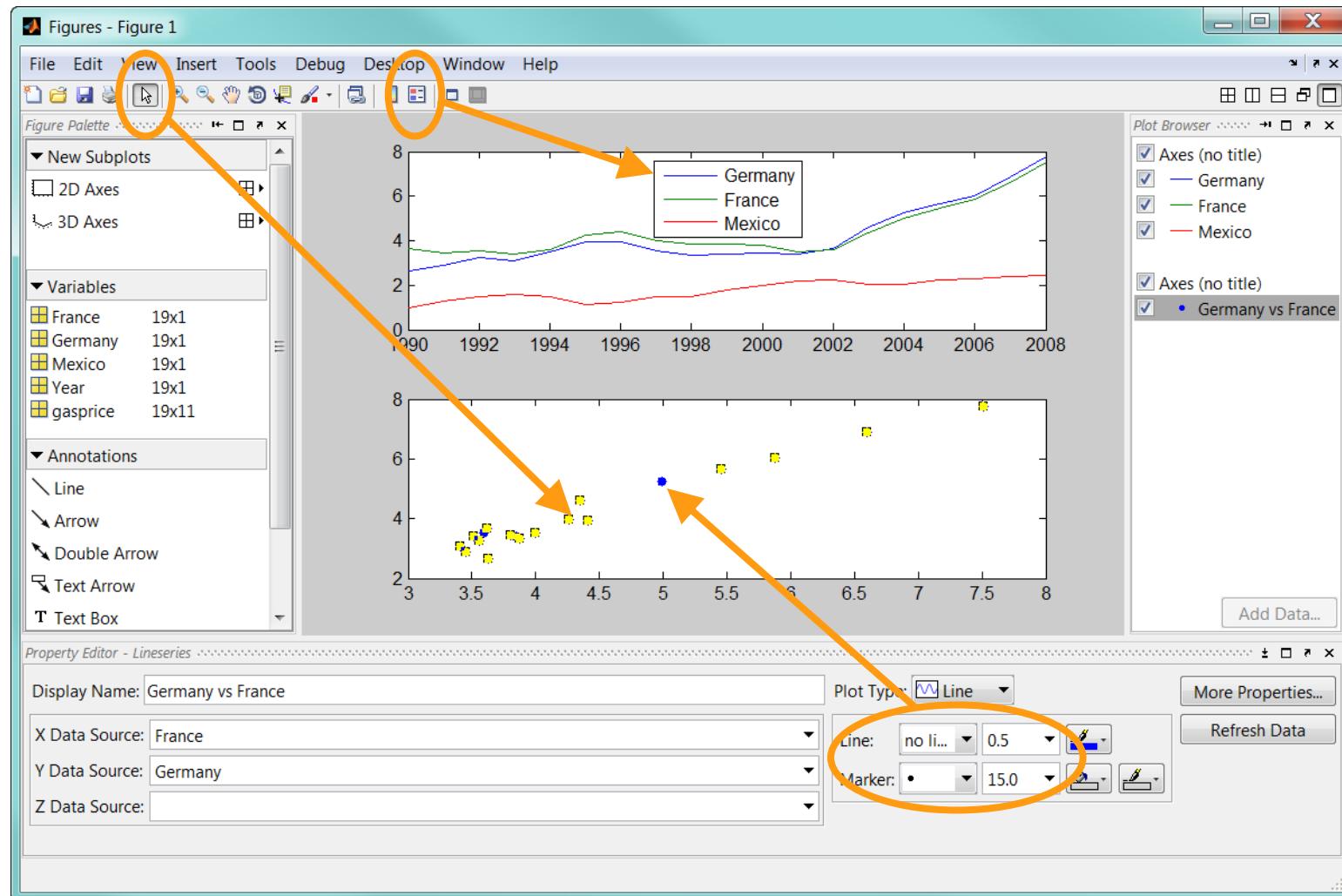


Plot Tools



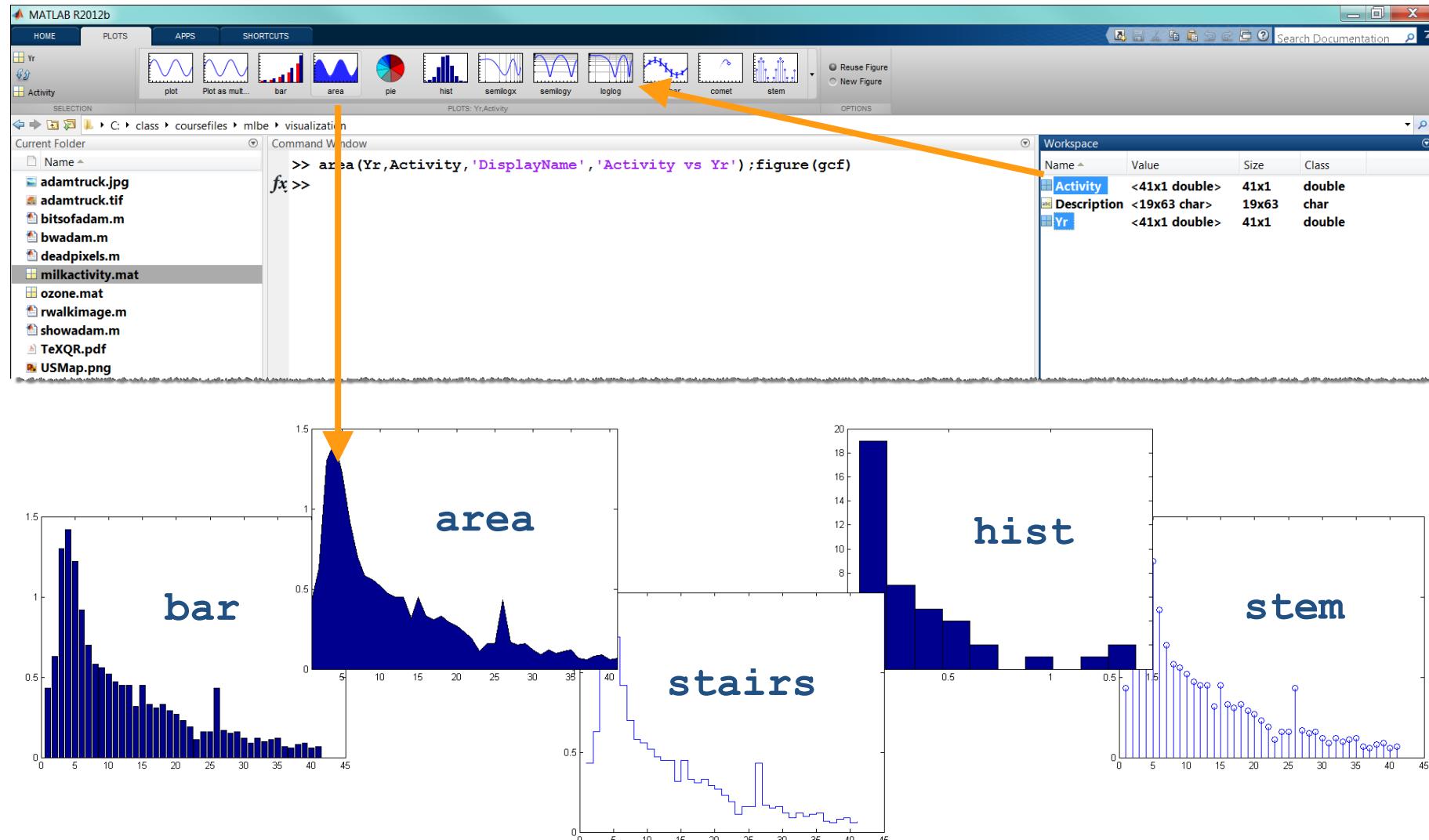
```
>> x = linspace(1,10,25);
>> y = sin(x);
>> plot(x,y)
>> plottools
```

Formatting the Plot



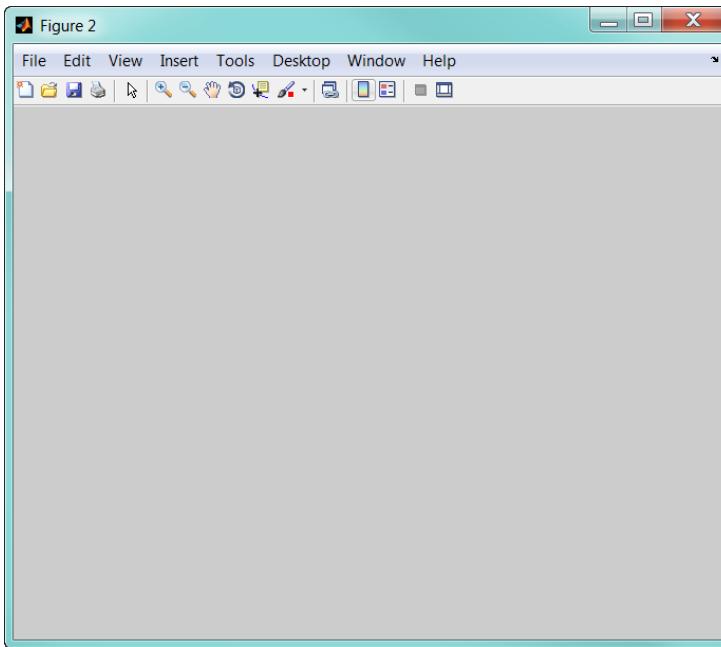
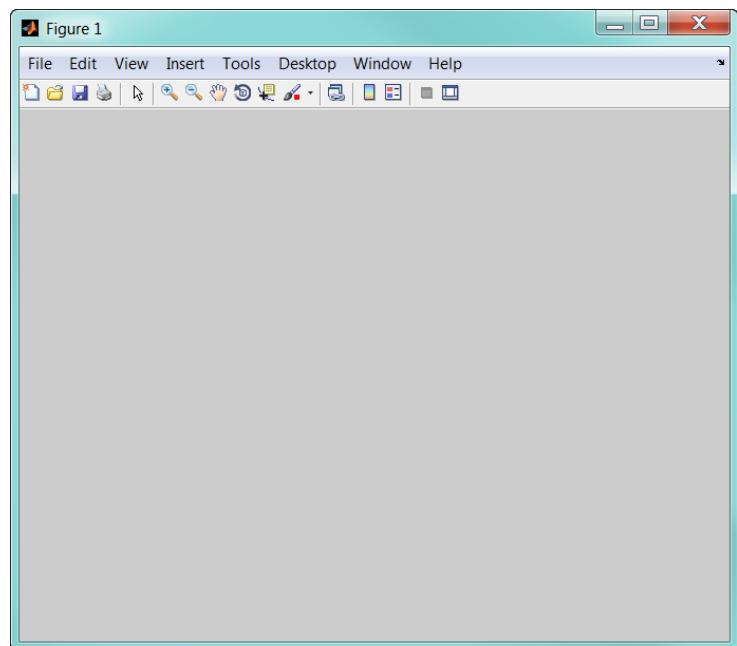
`web(fullfile(docroot, 'matlab/creating_plots/plotting-tools-interactive-plotting.html'))`

Additional Vector Plot Types



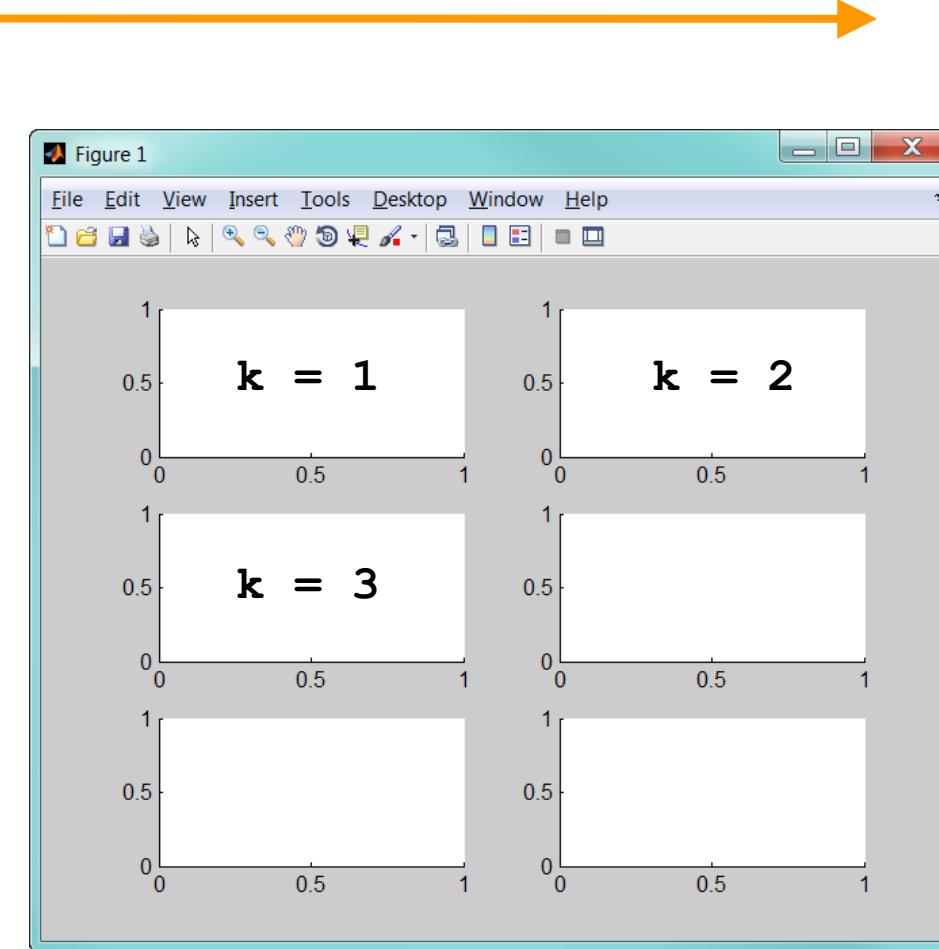
Multiple Figures

```
>> figure  
>> figure(n)
```



Multiple Axes

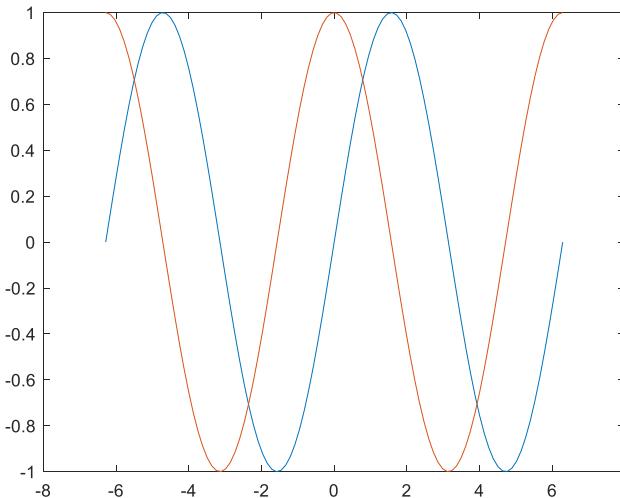
```
>> subplot(3,2,k)
```



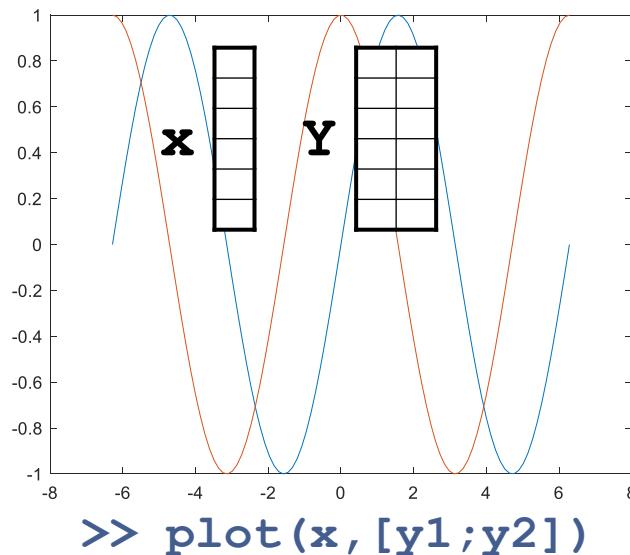
Multiple Plots

Plot Multiple Lines

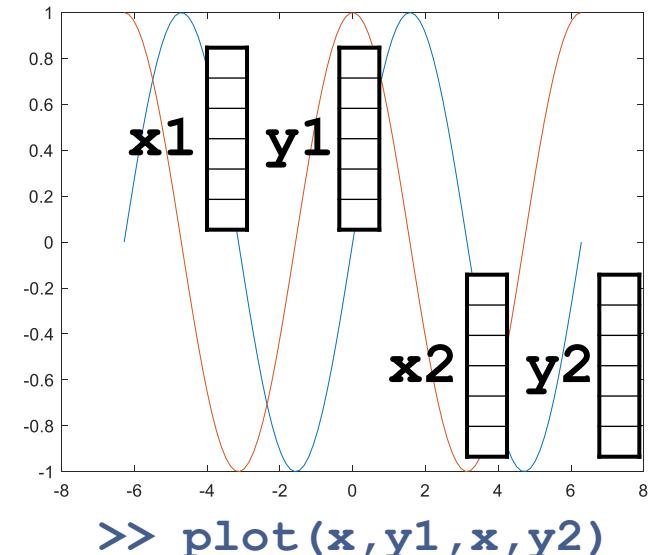
```
x = linspace(-2*pi,2*pi);  
y1 = sin(x);  
y2 = cos(x);
```



```
>> plot(x,y1)  
>> hold on;  
>> plot(x,y2)
```

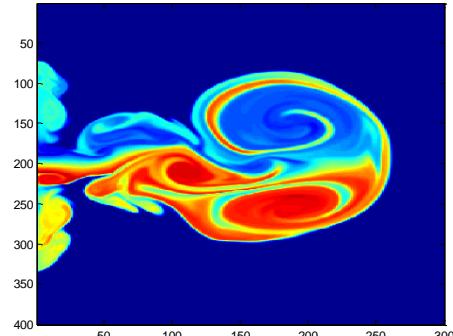


```
>> plot(x,[y1;y2])
```



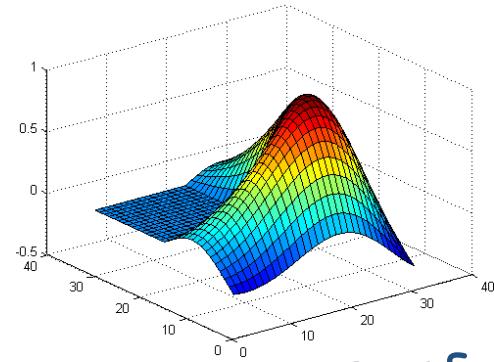
```
>> plot(x,y1,x,y2)
```

Additional Visualization



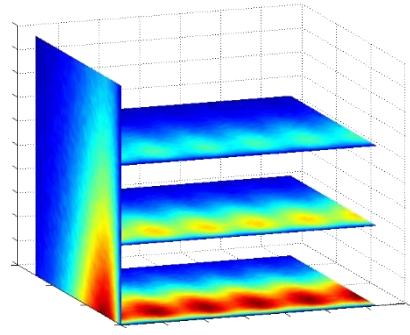
imagesc

`web(fullfile(docroot, 'matlab/ref/imagesc.html'))`



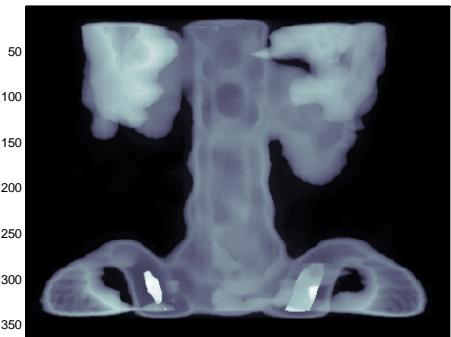
surf

`web(fullfile(docroot, 'matlab/ref/surf.html'))`



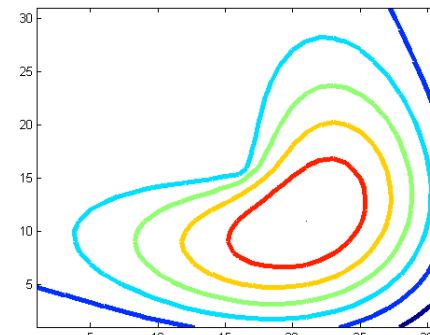
slice

`web(fullfile(docroot, 'matlab/ref/slice.html'))`



colormap

`web(fullfile(docroot, 'matlab/ref/colormap.html'))`



contour

`web(fullfile(docroot, 'matlab/ref/contour.html'))`

Programming Keywords

Programming Keywords

```
for index= first:increment:last  
    statements  
end
```

```
>> a=0;  
for i=1:10  
    a=a+i;  
end  
disp(a)
```

Programming Keywords

```
while condition  
    statements  
end
```

```
>> a=0; i=1;  
while i<11  
    a=a+i;  
    i=i+1;  
end  
disp(a)
```

Programming Keywords

```
if logical test1
    statement1
elseif logical test2
    statement2
else
    statement3
end
```

```
>> a=0; b=0; i=1;
while i<11
if mod(i,2)==1
a=a+i;
else
b=b+i;
end
i=i+1;
end
disp([a b])
```

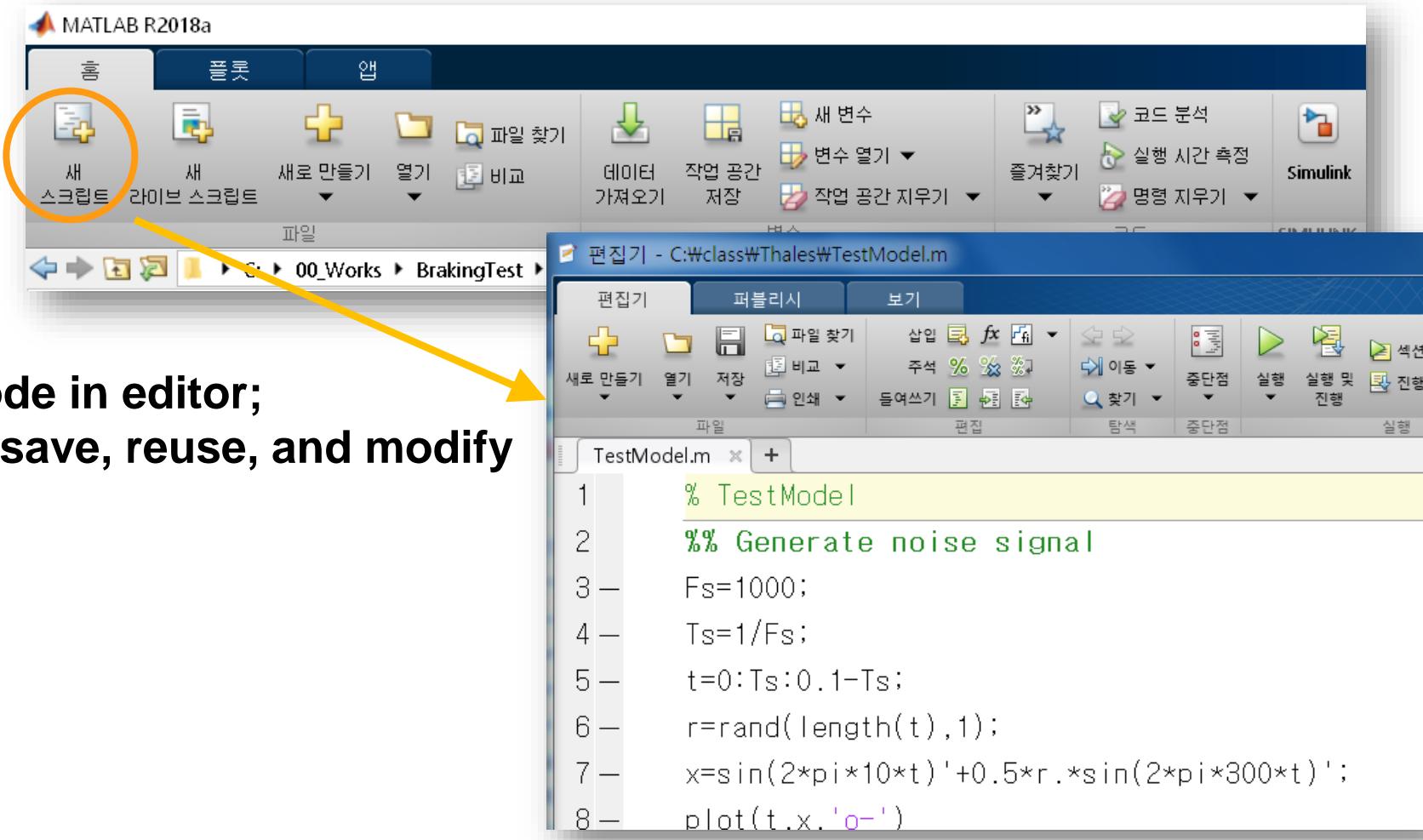
Programming Keywords

```
switch expression  
case value1  
    statement1  
case value2  
    statement2  
otherwise  
    statement3  
end
```

```
>> a=0; b=0;  
for i=1:10  
switch i  
case {1 3 5 7 9}  
a=a+i;  
case {2 4 6 8 10}  
b=b+i;  
otherwise  
error('quit');  
end  
end  
disp([a b])
```

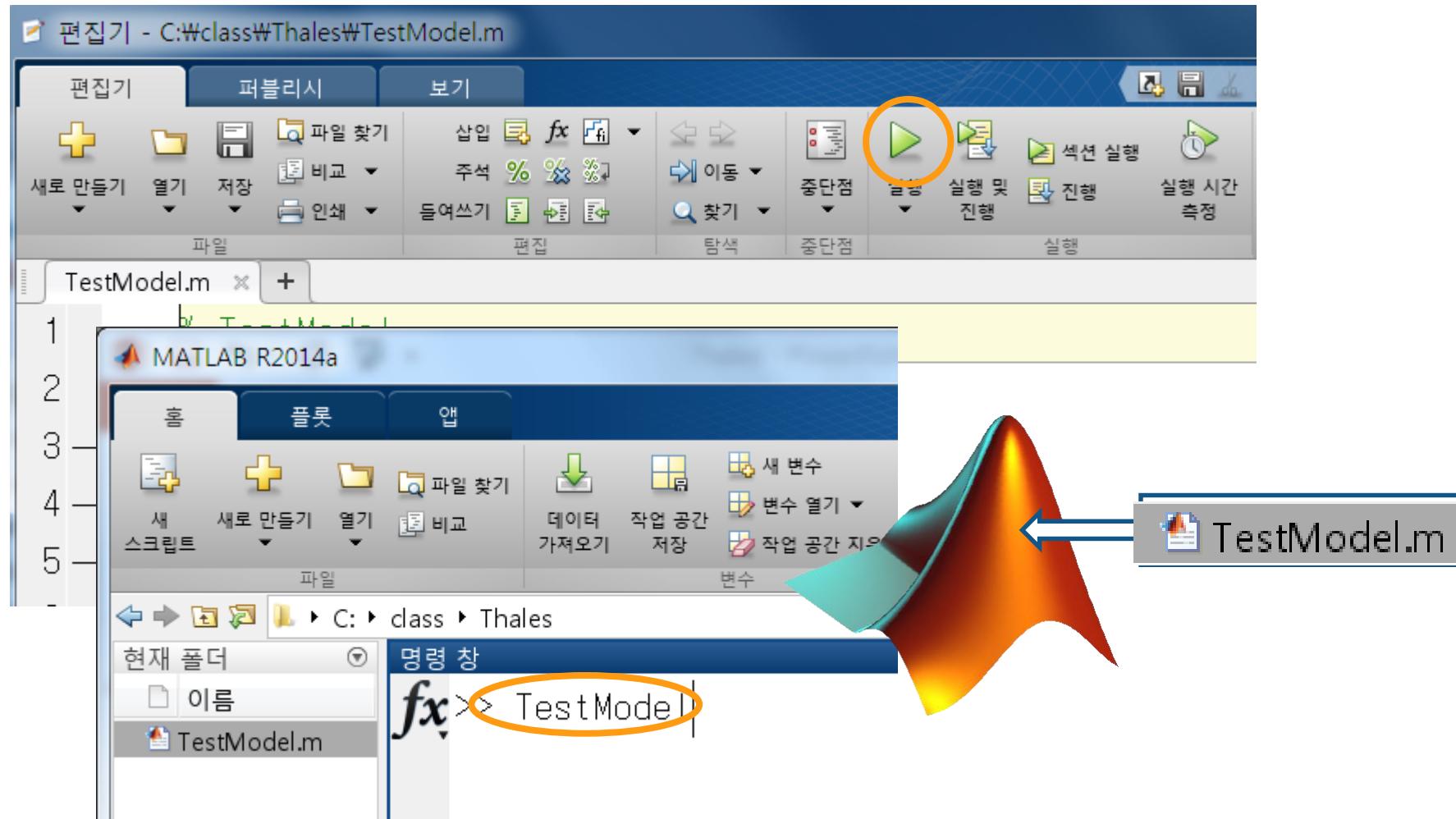
Script

Script File



Running a Script

Press F5



Comment and Code Section

```
H1 line % CALLMODEL Models a blue whale B call.  
%  
% Uses a model of the form y = A.*y0  
% where A = A0*exp(-B*t).*sin(2*pi*fm*t)  
% and y0 is a sum of harmonics  
% yn = sin(2*pi*n*f0*t)  
  
Code section %% MATLAB Code  
% Create the time base for the signal.  
fs = 4000;  
t = 0:(1/fs):1.5;  
  
% Set the fundamental frequency of the call.  
f0 = 175;  
  
% Create the harmonics.  
y0 = sin(2*pi*f0*t) + sin(2*pi*2*f0*t) + sin(2*pi*3*f0*t);
```

Code

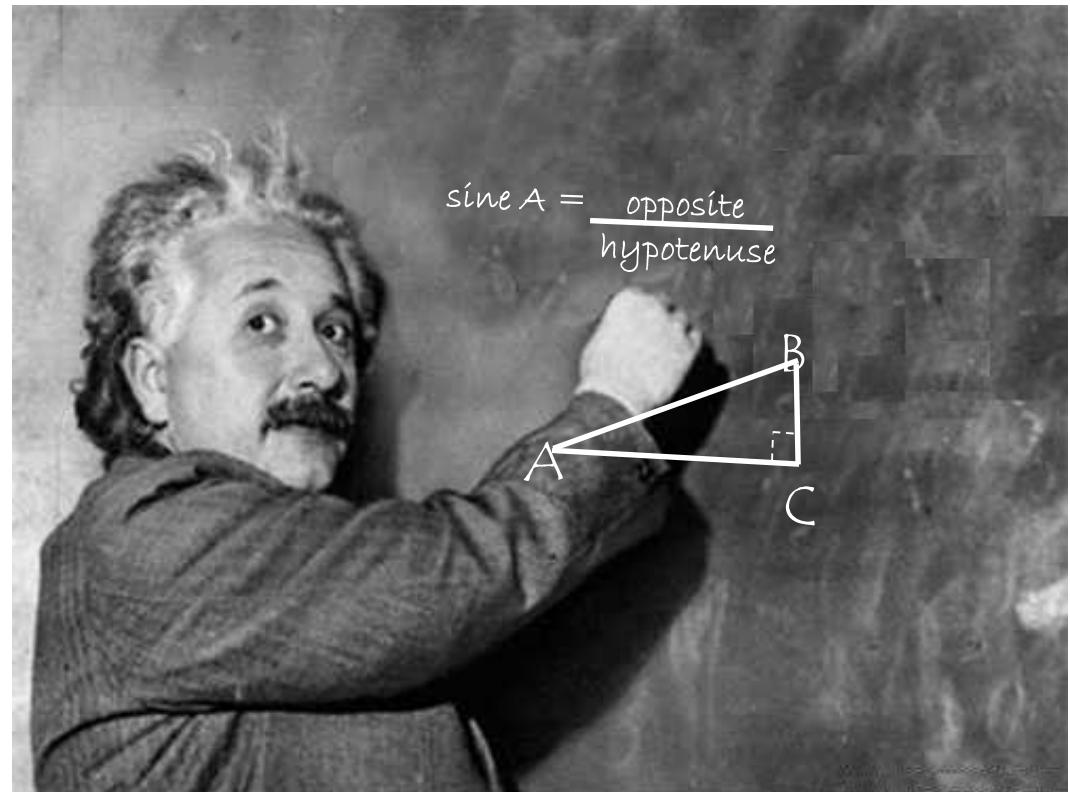
Help

Comments

Function

What is a MATLAB® function?

- MATLAB provides elementary mathematical and statistical functions:
[abs](#), [sqrt](#), [exp](#), [mean](#), [std](#), [sin](#), [cos](#), etc.
- MATLAB also provides advanced mathematical functions:
matrix inverse, matrix eigenvalues,
Bessel and gamma functions, fast Fourier transforms, etc.



How to Create and Use Function

function [arg1, arg2, ...]= func_name(arg1, arg2, ...)

```
% TestModel
%% Generate noise signal
Fs=1000;
Ts=1/Fs;
t=0:Ts:0.1-Ts;
r=rand(length(t),1);
x=sin(2*pi*10*t)+0.5*r.*sin(2*pi*300*t)';
plot(t,x,'o-')
m=20;
s=smoothing(x,m); %<-- Circled by orange oval
hold on; plot(t,s,'r*-')

%% FFT analysis
X=fft(x);
S=fft(s);
n=length(x);
f=(0:n-1)*(Fs/n);
z=conj(X).*X/n;
zz=conj(S).*S/n;
figure; stem(f,[z,zz])
```

```
function s=smoothing(x,m)
w=ones(m,1)/m;
s=conv(x,w,'same');
```

**Easy to reuse and maintain
Shorten programming code**

>> arg1=func_name(arg1,arg2)

Create MATLAB function

```
function [p, m] = plusmin(a,b)
    p = a + b;
    m = a - b;
end
```

Save as plusmin.m

At Command Window

```
>>[a b] = plusmin(5,3)
a =
    8
b =
    2
```

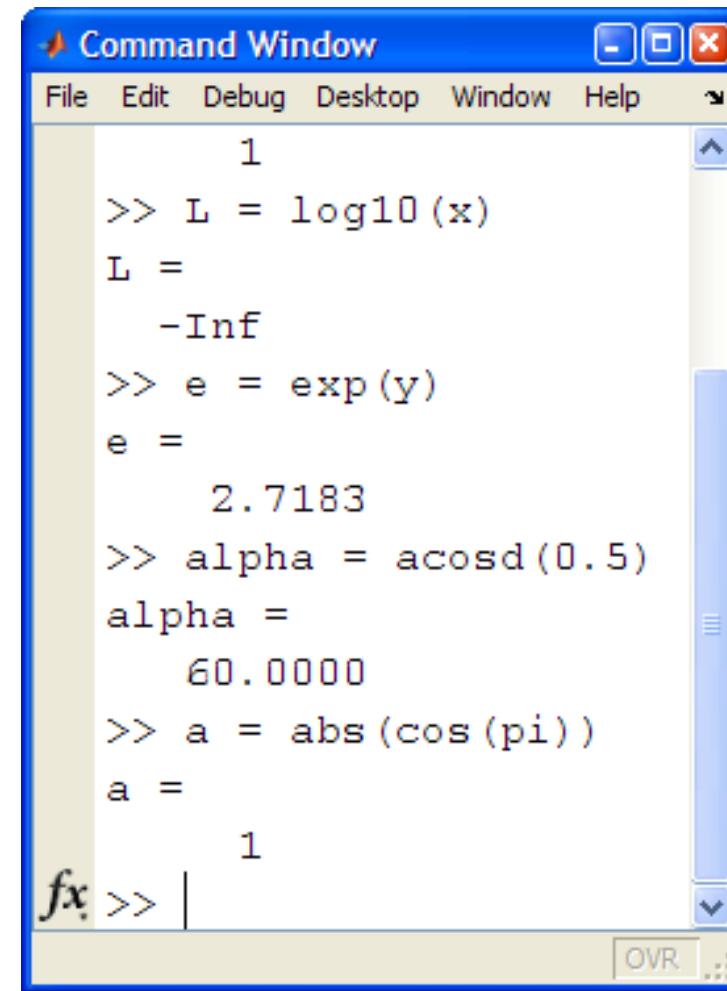
Basic Function Calling Syntax

```
output = functionName(input)
```

x = sin(0)

- The result of a function is assigned to the output variable.
- The input value is always entered in parenthesis after the function name.

How do I use basic MATLAB functions?



The screenshot shows the MATLAB Command Window with the following session history:

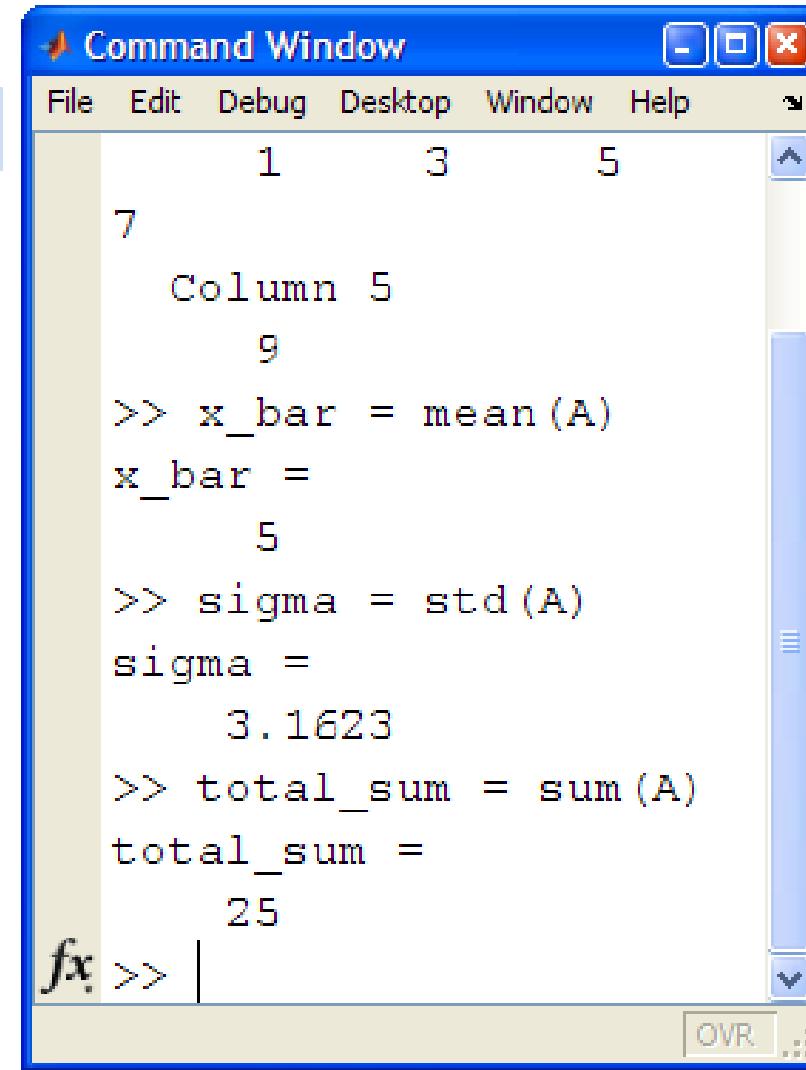
```
1
>> L = log10(x)
L =
-Inf
>> e = exp(y)
e =
2.7183
>> alpha = acosd(0.5)
alpha =
60.0000
>> a = abs(cos(pi))
a =
1
fx >> |
```

Basic Function Calling Syntax

```
output = functionName(input)
```

x_bar = mean(A)

Functions can operate
on a vector or matrix.



The image shows a screenshot of the MATLAB Command Window. The window title is "Command Window". The menu bar includes File, Edit, Debug, Desktop, Window, and Help. The command history shows the following code execution:

```
1   3   5
7
    Column 5
    9
>> x_bar = mean(A)
x_bar =
      5
>> sigma = std(A)
sigma =
      3.1623
>> total_sum = sum(A)
total_sum =
      25
fx >> |
```

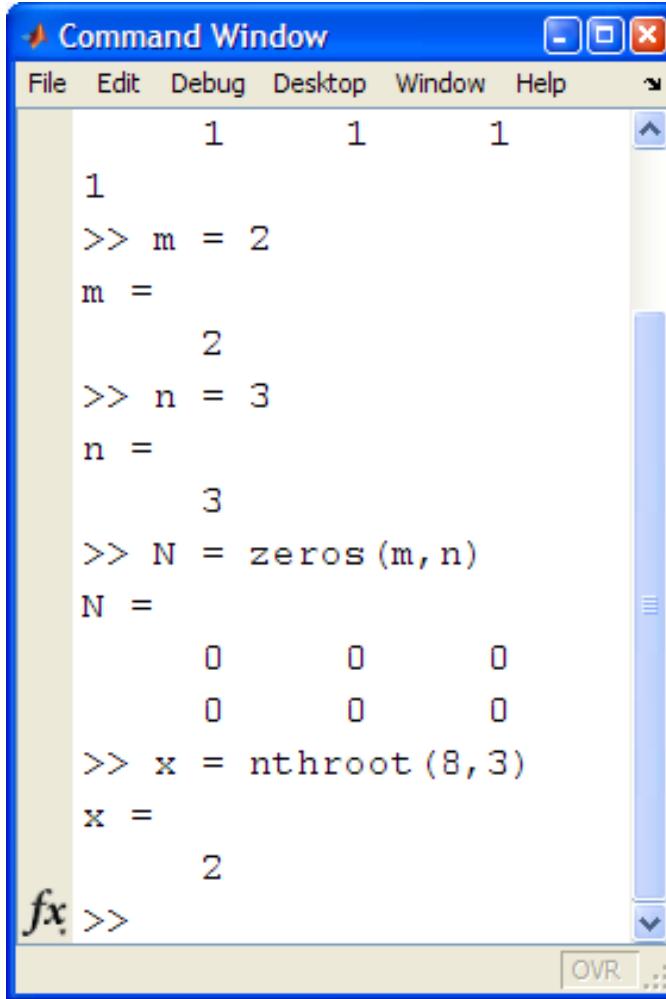
The output pane displays the results of the commands: the mean of the vector [1, 3, 5] is 5, the standard deviation of the vector is 3.1623, and the sum of the vector is 25. The cursor is positioned at the end of the command line.

Function Calling Syntax

```
output = functionName(input1, input2)
```

M = ones(2, 4)

Functions can have multiple input variables



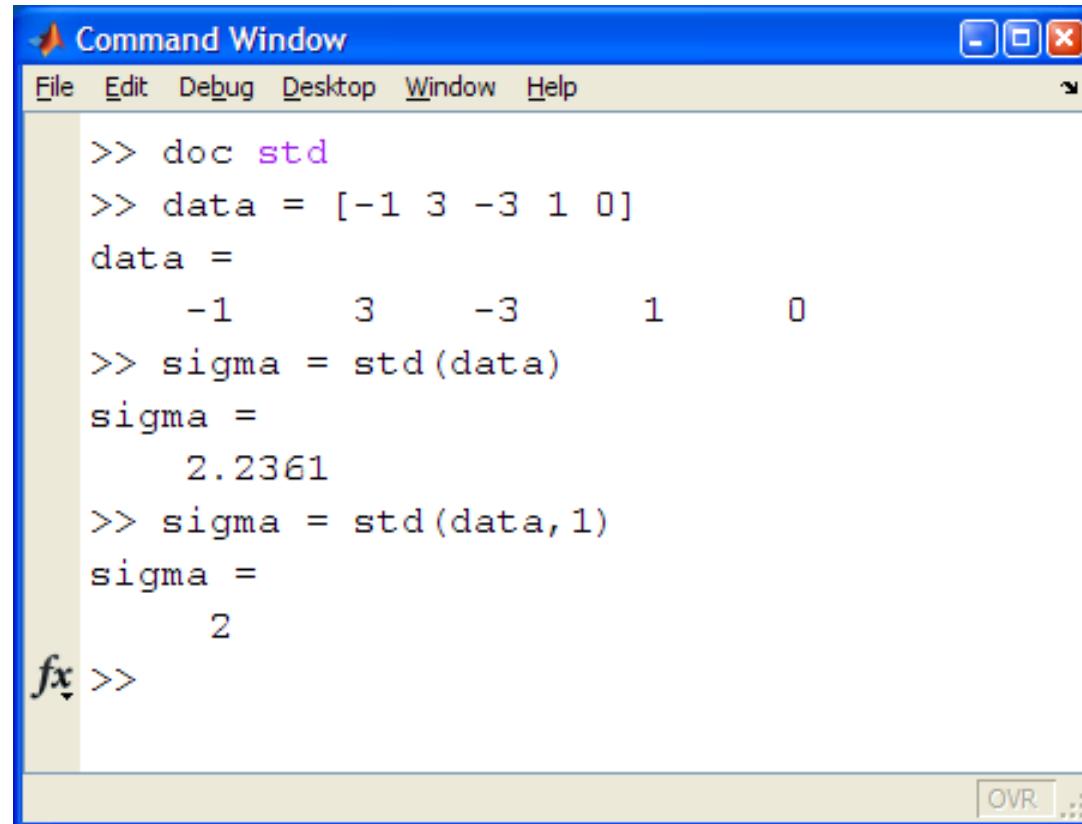
The image shows a screenshot of the MATLAB Command Window. The window title is "Command Window". The menu bar includes File, Edit, Debug, Desktop, Window, and Help. The command history and workspace are visible. The workspace pane shows variables m and n with values 2 and 3 respectively. The command history pane shows the following entries:

```
1 1 1  
1  
>> m = 2  
m =  
2  
>> n = 3  
n =  
3  
>> N = zeros(m, n)  
N =  
0 0 0  
0 0 0  
>> x = nthroot(8, 3)  
x =  
2  
fx >>
```

MATLAB Documentation

- The same function can be used in different ways.
- The documentation lists all the options for each function.

>> doc std



The image shows a screenshot of the MATLAB Command Window. The window title is "Command Window". The menu bar includes File, Edit, Debug, Desktop, Window, and Help. The command history shows the following code execution:

```
>> doc std
>> data = [-1 3 -3 1 0]
data =
    -1      3     -3      1      0
>> sigma = std(data)
sigma =
    2.2361
>> sigma = std(data, 1)
sigma =
    2
fx >>
```

The output shows the computation of standard deviation for the vector [-1 3 -3 1 0] using both the default behavior (std) and specifying the sample size (std, 1). The standard deviation is calculated as approximately 2.2361 for the first case and 2 for the second case.

Getting Started with MATLAB

The screenshot shows the MATLAB Documentation interface. The title bar reads "Getting Started with MATLAB". The left sidebar has a "CONTENTS" section with "Getting Started with MATLAB" expanded, showing sub-sections like "Language Fundamentals", "Mathematics", "Graphics", etc. Other sections include "Examples", "Functions", "Release Notes", and "PDF Documentation". The main content area is titled "Getting Started with MATLAB" and includes sections for "MATLAB Product Description", "System Requirements", "Tutorials" (with "Desktop Basics" and "Matrices and Arrays" examples), "Online Learning" (with a screenshot of the MATLAB Onramp course), "Videos" (with a screenshot of a MATLAB video player), and "Help and Documentation" (with a screenshot of the MATLAB command window).

Documentation

CONTENTS Close

< Documentation Home

< MATLAB

Getting Started with MATLAB

- Language Fundamentals
- Mathematics
- Graphics
- Data Import and Analysis
- Programming Scripts and Functions
- App Building
- Advanced Software Development
- Desktop Environment
- Supported Hardware

Examples

Functions

Release Notes

PDF Documentation

Getting Started with MATLAB

MATLAB Product Description

System Requirements

Tutorials

Desktop Basics

Enter statements at the command line and view results.

Matrices and Arrays

MATLAB® operates primarily on arrays and matrices, both in whole and in part. A matrix is a two-dimensional array often used for linear algebra.

Array Indexing

Variables in MATLAB are typically arrays that can hold many numbers. When you want to access selected elements of an array, use indexing.

Workspace Variables

The workspace contains variables that you create within or import into MATLAB from data files or other programs.

Text and Characters

When you are working with text, enclose sequences of characters in single quotes.

Calling Functions

MATLAB provides a large number of functions that perform computational tasks. To call a function, enclose its input arguments in parentheses.

2-D and 3-D Plots

Graphics functions include 2-D and 3-D plotting functions to visualize data and communicate results.

Programming and Scripts

The simplest type of MATLAB program is called a script. A script contains a sequence of commands and function calls.

Help and Documentation

All functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.

Search Help

Online Learning

MATLAB Onramp

Free two-hour online MATLAB course

Videos

Getting Started with MATLAB

Get an overview of MATLAB, the language of technical computing.

MATLAB Download & Installation

아카데미아

MathWorks.com내 검색



학생 대상 교육자 대상 연구원 대상

MATLAB 및 Simulink 교육과 학습

전 세계 5,000개 이상의 대학에서 사용하고 있습니다.

학생용 소프트웨어
받기

기본 배우기



강의 준비하기



프로젝트 시작하기



▶ 1:30

학생용 소프트웨어 받기

학생용 소프트웨어를 사용하여 학습 과정 및 연구를 성공적으로 이행하고, 취업 및 실무에 해당 지식을 활용하십시오.

[MATLAB Student 구매](#)[Looking for a campus-wide option?](#)[Check out MATLAB for your campus](#)[Which license is right for you?](#)[Explore all license options](#)



Academia

MathWorks.com내 검색



학생 홈 학생용 MATLAB 예제 학생 경진 대회 서적 하드웨어 지원

MATLAB Onramp
무료 인터랙티브 강의로 MATLAB을 경험하고 학습합니다.

[자세히 보기](#)

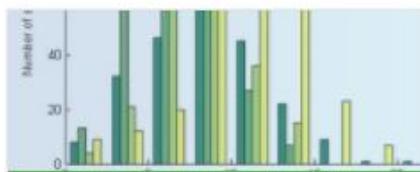
●○○
a = log10(b);
y = f(x);
x = 0:pi/30:2*pi;

학생

기술적인 영감을 주는 툴을 강의실은 물론, 업계 전반에서 활용할 수 있습니다. MATLAB 및 Simulink 기술과 함께하는 밝은 미래가 여러분을 기다립니다.



튜토리얼

[» 양방향 수업 둘러보기](#)

MATLAB 및 Simulink 예제

[» 예제 보기](#)

경진대회 및 콘테스트

[» 소프트웨어 지원 및 도움말](#)

학생용 MATLAB

[» MATLAB 구매](#)

최우수 학생

[» 학생들의 프로젝트 둘러보기](#)

학생 대상 MATLAB 및 Simulink 혼란지

[» 자세히 보기](#)



Automotive

- American Solar Challenge
- BAJA SAEINDIA
- EcoCAR Mobility Challenge
- EducEco
- Formula SAE Brasil
- Formula SAE Japan
- Formula SAE® Michigan and Formula SAE® Lincoln
- Formula Student China
- Formula Student Germany
- Formula Student Spain
- Formula Student UK
- Range Extended Electric Vehicle (REEV)
- SAE AutoDrive Challenge
- SUPRA SAEINDIA



Robotics



- BEST Robotics
- Coupe de France de Robotique Challenge
- Droid Racing Challenge
- European Rover Challenge
- High School Autonomous Vehicle Challenge
- Intelligent Ground Vehicle Competition
- Micromouse Contest
- Pan-African Robotics Competition
- ROBOCON India
- ROBO-ONE
- RoboCup
 - RoboCupJunior
- RoboNation Competitions
- RoboRace
- Smart Model Car Contest
- Turtlebot3 AutoRace
- VEX Robotics
- World Robot Summit
- Zero Robotics



The Winner's Circle



See how students are winning competitions worldwide with MATLAB and Simulink

» Explore student projects

MATLAB and Simulink Student Challenges



» Learn more

Academia

MathWorks.com내 검색



Student Home | MATLAB Student ▾ | MATLAB Examples | Student Competitions ▾ | 서적 | 하드웨어 지원

지능형 모델 자동차 경진대회

지능형 모델 자동차 경진대회는 학생 팀들이 안티록 브레이크와 컴퓨터식 엔진 관리를 위한 안전 및 효율 시스템 개발에 사용되는 것과 동일한 기술인 ICT(Intelligent Car Technology)를 사용하여 모델 자동차를 설계, 제작하여 직접 경주해보는 대회입니다. 여러분의 팀은 MATLAB 및 Simulink를 활용한 모델 기반 설계를 적용함으로써 보다 효율적인 제어, 신호 처리, 이미지 및 비디오 처리, 메카트로닉스와 신뢰성 있는 임베디드 시스템으로 주어진 도전 과제를 수행해 낼 수 있습니다.

MathWorks는 지능형 모델 자동차 경진대회 참가 학생들에게 MATLAB 및 Simulink 소프트웨어를 제공합니다. 학생들이 MATLAB 및 Simulink를 모델 자동차 개발에 어떻게 활용하는지에 대해 보고서를 제출하면, MathWorks에서 타임랩(time-lap)과 보고서 점수를 기반으로 하여 수상자 세 팀을 선정합니다.



무료 소프트웨어

MathWorks에서는 이번 대회를 위해 무료 소프트웨어를 제공합니다. 대회에 참가하는 팀이고 소프트웨어가 필요한 경우, 소프트웨어 신청 양식을 작성하십시오.

[소프트웨어 신청](#)MATLAB 및
Simulink
레이싱 라운지

계속해서 개발되는 이 비디오 시리즈를 통해 MATLAB 및 Simulink의 기본 사항과 자동차 개발에 대한 자세한 정보를 확인하십시오.

[비디오 보기](#)

학생경진대회 소프트웨어 신청방법

 Offer of Complimentary Software License(s)

MathWorks is pleased to sponsor Smart Car. MathWorks will provide software, training, and access to MathWorks engineering mentors and technical support to teams who have completed the Student Competition Software Request Form.

Software Request Form

Here are the products we will be providing that you may have access to:

Complimentary Software for

- MATLAB
- Simulink
- Control System Toolbox
- DSP System Toolbox
- Embedded Coder
- Filter Design HDL Coder
- Fixed-Point Designer
- Global Optimization Toolbox
- HDL Coder
- MATLAB Coder
- MATLAB Report Generator
- Neural Network Toolbox
- Optimization Toolbox
- Signal Processing Toolbox
- Simscape Driveline
- Simscape Electronics
- Simscape Multibody
- Simscape Power Systems
- Simscape
- Simulink Coder
- Simulink Control Design
- Simulink Design Optimization
- Simulink Design Verifier
- Simulink Report Generator
- Simulink PLC Coder
- Simulink Verification and Validation
- Stateflow
- System Identification Toolbox

Type: Designated Computer
License option: Individual
Platform: All

Your team leader or faculty advisor should review and complete the Student Competition Software Request Form to take advantage of our software offer.

Software Request Form

Best of luck in your preparation!

Lauren Tabolinsky
MathWorks Student Competition Program

학생경진대회 소프트웨어 신청방법

The screenshot shows the MathWorks Academic software interface for competition registration. At the top, there's a navigation bar with links for 'Home', '한국' (Korean), '언락처' (Unlocker), '구입 방법' (Purchase Method), 'Search MathWorks', 'Hwang Eunju | My account', and '로그아웃' (Logout). Below the navigation bar, there are tabs for '제품' (Products), '솔루션' (Solutions), '대학 커리큘럼' (Academic Curriculum) (which is selected), '지원' (Support), 'User Community', '이벤트' (Events), and '회사소개' (About Company). A sub-menu for 'Academia' is open, showing a '피드백' (Feedback) button.

학생 경진 대회 소프트웨어 신청 양식

신청을 희망하는 팀의 리더 혹은 학과 담당자가 아래의 양식을 기입하여 주시기 바랍니다.

"How will you use MathWorks software?"라는 질문에는 "Academic Use"를 선택하여 주십시오. 본 신청양식을 기입함으로써 학생경진대회를 위한 소프트웨어 신청이 이루어지게 됩니다. 신청에 대한 답변은 매스웍스에서 business hour 72시간 안에 답변 드릴 것입니다. 본 신청이 승인되면, 소프트웨어 다운로드를 위한 가이드 및 안내를 받게 됩니다.

만약에 매스웍스의 계정이 없으시다면, 지금 바로 매스웍스 홈페이지 (www.mathworks.com)에서 계정을 생성하여 주십시오. 본 신청은 귀하의 이메일 주소를 통해 가입 확인 절차가 이루어지게 됩니다.

* Indicates required information

안녕하십니까 Eunju Hwang

Team Leader or Faculty Advisor

*Team Leader/Faculty Advisor Confirmation

팀 리더/학과 담당자임을 확인합니다. (아래 체크하여 주십시오)

신청자 영문 정보

*회사학교명(영문)

회사 혹은 학교의 공식 명칭을 입력해 주십시오.

Competition Information - Complete in English

*참가를 희망하는 경진대회 이름 (ex. Smart Car Contest)

*대학교 이름

*팀 이름

*총 팀원수 – 매스웍스 소프트웨어를 사용하게 될 인원 수

MATLAB R2019a 설치 및 실행

