MATLAB EXPO 2019

MATLAB과 Simulink 제품군의 새로운 기능들 (R2018b/R2019a)

이영준
Algorithms in Everything
Using MATLAB & Simulink to Build Algorithms in Everything

Simplifying your work…

…often at higher levels of abstraction.
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
Artificial Intelligence

The capability of a machine to match or exceed intelligent human behavior by training a machine to learn the desired behavior.
There are two ways to get a computer to do what you want

Traditional Programming

- Data
- Program

COMPUTER

Output
There are two ways to get a computer to do what you want.
Artificial Intelligence

Data → Machine Learning → Deep Learning → Model
Using MATLAB and Simulink to Build Deep Learning Models

Inputs

Data

Machine Learning

Deep Learning

Model

Outputs
Using Apps for Ground Truth Labeling Image and Video Data

Computer Vision Toolbox
Using Apps for Ground Truth Labeling
Signal Data
Using Apps for Ground Truth Labeling
Audio Data
Using Apps for Designing Deep Learning Networks
Using Transfer Learning with Pre-trained Models

- Inception-v3
- ResNet-101
- VGG-16
- Inception-ResNet-v2
- ResNet-18
- GoogLeNet
- DenseNet-201
- VGG-19
- SqueezeNet
- AlexNet
- ResNet-50

Deep Learning Toolbox
Using Models from Other Frameworks

- Keras-Tensorflow
- PyTorch
- Caffe2
- ONNX
- Caffe
- CNTK
- MXNet
- Core ML

Deep Learning Toolbox
Deploying Deep Learning Applications

- Pre-processing
- Post-processing
- Coder Products

- Intel MKL-DNN Library
- NVIDIA TensorRT & cuDNN Libraries
- ARM Compute Library

Deep Learning Networks

MATLAB Coder
GPU Coder
Deploying Deep Learning Applications

Find out more:
임베디드 하드웨어로의 딥러닝 응용프로그램 배포

인공지능과 딥러닝 트랙 송완빈
Using MATLAB and Simulink for Reinforcement Learning

Inputs

Data

Design

Machine Learning

Deep Learning

Outputs

Model

Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning

Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning

Inputs

Data

Machine Learning

Deep Learning

Model

Outputs
Using MATLAB and Simulink for Reinforcement Learning

Generate Data

- Scenario Design
- Simulation-based data generation

Inputs

Machine Learning

Deep Learning

Design

Model

Outputs

Simulink Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning
Using MATLAB and Simulink for Reinforcement Learning

Find out more:
딥러닝과 강화학습
인공지능과 딥러닝 트랙
김종남
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
## Working with Text Data

### Example Table

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Equipment ID</th>
<th>Reason</th>
<th>Notes</th>
<th>Cost Parts</th>
<th>Cost Labor</th>
<th>Cost Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/20/2015</td>
<td>12:00 PM</td>
<td>14073</td>
<td>SERVICE: CHECK TURN SIGNAL</td>
<td>DRIVER'S REPORT, PM SERVICE</td>
<td>493.85</td>
<td>0.4935</td>
<td></td>
</tr>
<tr>
<td>10/20/2015</td>
<td>12:00 PM</td>
<td>14232</td>
<td>PM SERVICE</td>
<td>493.85</td>
<td>0.4935</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14006</td>
<td>NEED 4 FLOW PINS</td>
<td>DRIVER'S REPORT, INSTALL SPINNER ASSY</td>
<td>45.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14110</td>
<td>PM SERVICE</td>
<td>SNOW BREAKDOWN, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14165</td>
<td>PM SERVICE</td>
<td>DRIVER'S REPORT, DOG BONE PIN BROKEN</td>
<td>20, 0, 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14000</td>
<td>PM SERVICE</td>
<td>SNOW BREAKDOWN, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14155</td>
<td>PM SERVICE</td>
<td>HYD CAP CHECK ENGINE LIGHT ON</td>
<td>12.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14157</td>
<td>NEED SERVICE, CHECK BRAKES</td>
<td>DRIVER'S REPORT, HYD CAP CHECK ENGINE LIGHT ON</td>
<td>387.17, 0, 387.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14164</td>
<td>PM SERVICE</td>
<td>NEGLIGENCE, TARP VALVE STICKING RIGHT SIDE MIRROR BRACKET BROKEN</td>
<td>50.02, 0, 50.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14165</td>
<td>PM SERVICE</td>
<td>SNOW BREAKDOWN, HANDLES IN CAB LOOSE</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14172</td>
<td>PM SERVICE</td>
<td>DRIVER'S REPORT, NO FLOW LIGHTS</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14172</td>
<td>ROADCALL, WILL NOT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14174</td>
<td>ROADCALL, WILL NOT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14176</td>
<td>ROADCALL, WILL NOT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14177</td>
<td>ROADCALL, WILL NOT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14181</td>
<td>ROADCALL, WILL NOT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14182</td>
<td>ROADCALL, CONVEYOR NOT WORKING</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14183</td>
<td>ROADCALL, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14191</td>
<td>ROADCALL, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14190</td>
<td>ROADCALL, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14185</td>
<td>ROADCALL, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14188</td>
<td>ROADCALL, DONT START</td>
<td>0, 0, 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14191</td>
<td>ROADCALL, HARDWARE FOR REAR SPRINGS</td>
<td>14.32, 0, 14.32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14191</td>
<td>ROADCALL, HARDWARE FOR REAR SPRINGS</td>
<td>28.60, 0, 28.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14107</td>
<td>PM SERVICE</td>
<td>DRIVER'S REPORT, HARDWARE FOR REAR SPRINGS</td>
<td>14.32, 0, 14.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00 PM</td>
<td>14107</td>
<td>PM SERVICE</td>
<td>REMOVE &amp; REPLACE REAR SPRINGS, CHECK COOLANT TUBESPM SERVIVE</td>
<td>14.32, 0, 14.32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Working with Text Data

```matlab
% Read text file
filename = 'example.txt';
t = readtable(filename,'TextType','string');
disp(t(1:20,6:7));
```

<table>
<thead>
<tr>
<th>Reason</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;PM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE WHEN DRIVING&quot;</td>
</tr>
<tr>
<td>&quot;08 PM SERVICE&quot;</td>
<td>&quot;SERVICER08,EXT,5604&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;NEED 4 PLOW PINS&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;INSTALL SPINNER ASSY&quot;</td>
</tr>
<tr>
<td>&quot;13 SNOW BREAKDOWN&quot;</td>
<td>&quot;DONT START&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;DOG BONE PIN BROKEN&quot;</td>
</tr>
<tr>
<td>&quot;08 PM SERVICE&quot;</td>
<td>&quot;NEED SERVICE, CHECK BRAKES&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;HVD CAP CHECK ENGINE LIGHT ON&quot;</td>
</tr>
<tr>
<td>&quot;40 NEGLIGENCE&quot;</td>
<td>&quot;TARP VALVE STICKINGRIGHT SIDE MIRROR BRACKET BROKEN&quot;</td>
</tr>
<tr>
<td>&quot;13 SNOW BREAKDOWN&quot;</td>
<td>&quot;HANDLES IN CAB LOOSE&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;NO PLOW LIGHTS&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;CONVEYOR NOT WORKING&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DONT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DONT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DONT START&quot;</td>
</tr>
</tbody>
</table>
Working with Text Data
Creating Your Own Data
Identifying the Useful Data

1. Acquire Data
2. Preprocess Data
3. Identify Condition Indicators
4. Preprocess Data
5. Visualize data
6. Extract Features
7. Select the most useful features
8. Train Model
9. Machine Learning
10. Deploy & Integrate
Identifying the Useful Data
Identifying the Useful Data

Signal Features
Generate statistics
features from signals

Rotating Machinery Features
Generate features from
rotating machinery signals

Nonlinear Features
Generate nonlinear
features from signals

Predictive Maintenance Toolbox

Spectral Features
Condition variables: faultCode
Computation mode: use full signal

- Spectral peaks
  - Peak amplitude
  - Peak frequency

- Peak value lower threshold: Inf
- Number of peaks: 1
- Minimum frequency gap: 0.001
- Peak excursion tolerance: 0

- Modal coefficients

- Band power

© 2019 The MathWorks, Inc.
Identifying the Useful Data
Designing Decision Logic with Stateflow

```
inNormalRegion = true; counter = 0;
for i=1:length(inData)
    if(inNormalRegion)
        if(inData(i)<t1)
            counter = counter+1;
            if(counter>=N1)
                inNormalRegion = false;
        end
        else
            counter = 0;
        end
    else
        if(inData(i)>=t2)
            counter = counter+1;
            if(counter>=N2)
                inNormalRegion = true;
        end
        else
            counter = 0;
        end
    end
if(inNormalRegion)
    outData(i) = inData(i);
else
    outData(i) = 0;
end
```
Using Stateflow in MATLAB

```
% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        app.lanternLogic = blinkLanternLogic('app', app);
    end

    % Button pushed function: POWERButton
    function POWERButtonPushed(app, event)
        app.lanternLogic.powerButton();
    end

    % Button pushed function: COLORButton
    function COLORButtonPushed(app, event)
        app.lanternLogic.colorButton();
    end

    % Close request function: UIFigure
    function UIFigureCloseRequest(app, event)
        delete(app.lanternLogic);
        delete(app);
    end

    % Button pushed function: BLINKButton
    function BLINKButtonPushed(app, event)
        app.lanternLogic.blinkButton();
    end
end
```
Editing at the Speed of Thought
Editing at the Speed of Thought
Editing at the Speed of Thought
Editing at the Speed of Thought
Editing at the Speed of Thought
Controlling the Execution of Model Components

Schedulable Rate-Based Model

Export Function Model
Controlling the Execution of Model Components
Simplifying Integration with External C/C++ Code

```c
#include "rtwdemo_rowlutcol2row_workflow_rowrow.h"

/* Block parameters (default storage) */

PrtP = {
    /* Variable: Tbl_3 */
    /* Referenced by: '<Root>/2-D Lookup Table' */
    {
        1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0,
        11.0, 12.0, 13.0, 14.0,
        15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0,
        28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0,
        41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0, 50.0, 51.0, 52.0, 53.0,
        54.0, 55.0, 56.0, 57.0, 58.0, 59.0, 60.0
    }
};
```
Simplifying Integration with External C/C++ Code
Viewing Generated Code Alongside the Model
Sharing Live Scripts

Estimating Sunrise and Sunset

Using the latitude \((\phi)\), the sun's declination \((\delta)\) and the solar time correction \((SC)\) we can calculate sunrise and sunset times.

\[
sunrise = 12 - \frac{\cos^{-1} (-\tan \phi \tan \delta)}{15^\circ} - \frac{SC}{60}
\]

\[
sunset = 12 + \frac{\cos^{-1} (-\tan \phi \tan \delta)}{15^\circ}
\]

Refer to this page for background and details on the equations used.
Sharing Live Scripts

Exploring Exoplanets

In this example we will explore some data on exoplanets - planets outside our own solar system. The data used here is a subset of data from the NASA Exoplanet Archive. We will start by using the data to answer some questions about the set of exoplanets in the archive. Then we will do some calculations to try to identify planets in the archive that might be capable of supporting life.

```matlab
exoplanets = readtable('exoplanets.xls');
exoplanets(1:10,1)
```

How Far Away Are these Planets?

There are 50 exoplanets within 50 light-years of earth and 460 exoplanets within 200 light-years.

```matlab
histogram(x(exoplanets.st_distance<50),10); %50
xlabel('Number of Planets');
ylabel('Light Years From Earth');
```

Where is the nearest exoplanet?

```matlab
idx = find(exoplanets.st_distance == min(exoplanets.st_distance));
name = char(exoplanets(idx, 'st_name'))
```
Sharing Live Scripts
Creating Apps

Microplate Plot

EC50 Curves

<table>
<thead>
<tr>
<th>File</th>
<th>Compound Nr</th>
<th>Neg Control</th>
<th>Conc1</th>
<th>Conc2</th>
<th>Conc3</th>
<th>Conc4</th>
<th>Conc5</th>
<th>Conc6</th>
<th>Conc7</th>
<th>Conc8</th>
</tr>
</thead>
<tbody>
<tr>
<td>microtiter</td>
<td>1</td>
<td>-0.0741</td>
<td>0.3564</td>
<td>9.8759</td>
<td>55.8743</td>
<td>91.7323</td>
<td>96.7084</td>
<td>97.1532</td>
<td>57.1910</td>
<td>97.1940</td>
</tr>
<tr>
<td>microtiter</td>
<td>2</td>
<td>-0.0143</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>17.0436</td>
</tr>
<tr>
<td>microtiter</td>
<td>3</td>
<td>0.0054</td>
<td>-0.4702</td>
<td>3.1908</td>
<td>52.9906</td>
<td>97.5746</td>
<td>100.5606</td>
<td>100.6086</td>
<td>100.6086</td>
<td>100.6086</td>
</tr>
<tr>
<td>microtiter</td>
<td>4</td>
<td>0.1096</td>
<td>0.2325</td>
<td>0.2385</td>
<td>0.3712</td>
<td>3.2339</td>
<td>41.1680</td>
<td>94.7343</td>
<td>100.6591</td>
<td>100.9487</td>
</tr>
<tr>
<td>microtiter</td>
<td>5</td>
<td>-0.0572</td>
<td>0.7461</td>
<td>1.7104</td>
<td>26.8872</td>
<td>84.5134</td>
<td>98.2335</td>
<td>100.4717</td>
<td>100.5601</td>
<td>100.5700</td>
</tr>
</tbody>
</table>
Deploying Web Apps
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs

Inputs

Design

Outputs

MATLAB & SIMULINK®
Evaluating Architectures

Inputs

Architecture

Design

Outputs
Evaluating Architectures

Inputs → Architecture → Design → Outputs
Designing System and Software Architectures
Designing System and Software Architectures
요구사항부터 아키텍처 설계와 시뮬레이션까지 시스템 엔지니어링을 위한 방안

재어 및 임베디드 시스템 트랙
류성연
Designing Beyond System and Software Architectures

Systems and Software

SoC Hardware and Software

AUTOSAR Software

System Composer

SoC Blockset

AUTOSAR Blockset
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

Test & Verification
Collaboration
Scaling
Using MATLAB & Simulink to Build Algorithms in Everything

DEFINITION
OF READY
BACKLOG
SPRINT
DELIVERY
DEFINITION
OF DONE

TEST
INTEGRATION
DEVELOPMENT
Integrating with Third-party Requirements Tools

External Requirements

- .doc
- .xls

Requirements Management Tools

Simulink Requirements

- External Requirements

External Requirements

- Authored Requirements

ReqIF

Import
Edit
Export

R2019a

Simulink Requirements
Include Custom Code in Test & Verification

Simulink

Stateflow

Simulink Design Verifier

C/C++

C/C++
Include Custom Code in Test & Verification

Find out more:
모델기반설계를 이용한 요구사항 기반 검증의 단순화

제어 및 임베디드 시스템
홍혁기
Using the MATLAB Unit Test Framework

```matlab
>> result.table
ans =
    2×6 table
      Name                 Passed Failed  Incomplete  Duration     Details
    ____________________  ______  _______  _________  ________  ________________
    'test_Predictions/Test_ModelType' true     false   false      0.12241  [1×1 struct]
    'test_Predictions/Test_Prediction' false    true    true       0.11542  [1×1 struct]
```
Using the MATLAB App Testing Framework

testCase.press(myApp.checkbox)

testCase.choose(myApp.discreteKnob, "Medium")

testCase.drag(myApp.continuousKnob, 10, 90)

testCase.type(myApp.editfield, myTextVar)
Using the MATLAB Performance Testing Framework
Using Continuous Integration

Plugins Index

Discover the 1000+ community contributed Jenkins plugins to support building, deploying and automating any project.

Browse categories
- Platforms
- User interface
- Administration
- Source code management

New Plugins
- ORebel
- MATLAB
- MISRA Compliance Report
- Zoom
- VectorCAST Execution
- Knocwork Community
- jQuery
- Analysis Model API

MATLAB

https://plugins.jenkins.io/
Using Continuous Integration

The Jenkins plugin for MATLAB® enables you to easily run your MATLAB tests and generate test artifacts in formats such as JUnit, TAP, and Cobertura code coverage reports.

Features

- Support to run MATLAB tests, present in the Jenkins workspace automatically. (This also includes the tests present in .prj files)
- Generate tests artifacts in JUnit, TAP & Cobertura code coverage formats.
- Support to run tests using custom MATLAB command or custom MATLAB script file.
Using Projects in MATLAB

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Git</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>+Test</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACI</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dashboard</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elasticsearch</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MachineLearning</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATLAB_Kafka_Producer_Java</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mps_stream</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimExecutable</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DocExample_MultiClassFaultDetectionUsi...</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>genPumpData.m</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>javasetup.m</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>Main_ExampleWorkflow.mlx</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>MLModels.mat</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>rawdata.mat</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>README.md</td>
<td>✔️</td>
<td></td>
<td>Design</td>
</tr>
</tbody>
</table>
Parallel Simulations in Simulink

Simulation Manager

batchsim

Simulation Jobs
Simulation Results

MATLAB Desktop

Worker
Worker
Worker

Simulink
Parallel Computing Toolbox
Scaling Computations on Clusters and Clouds

MATLAB Parallel Server

Parallel Computing Toolbox

Cloud

GPU

Multi-core CPU
Team-Based Collaboration for Code Verification and Review

- Web browser-based view of results directly in the code
- Navigation tools for investigating code analysis and proving results
- Ability to triage, assign, and justify code analysis results
- Create and assign tickets in bug-tracking systems such as Jira
Team-Based Collaboration for Code Verification and Review

- Web browser-based view of results directly in the code
- Navigation tools for investigating code analysis and proving results
- Ability to triage, assign, and justify code analysis results
- Create and assign tickets in bug-tracking systems such as Jira

Find out more:

팀 협업을 통한 소프트웨어 안전성 및 보안성 확보 방안

재어 및 임베디드 시스템 트랙

Jay Abraham
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

- Test & Verification
- Collaboration
- Scaling
Specialized Tools for Building Algorithms in Everything

Communications

- 5G Toolbox

Physical interconnects

- SerDes Toolbox

Analog Mixed-Signal

- Mixed-Signal Blockset
Developing Autonomous Systems

Perception

Planning

Control
Evaluate Sensor Fusion Architectures
Simulate Path Planning Algorithms
Design Lane-following and Spacing Control Algorithms
Developing Autonomous Systems

Lidar Processing & Tracking

HERE HD Maps & OpenDRIVE Roads

UAV Algorithms

Computer Vision Toolbox

Automated Driving Toolbox

Robotics System Toolbox
Developing Autonomous Systems

Find out more:
MATLAB 및 Simulink를 이용한 자율주행 시스템 설계 및 시뮬레이션

제어 및 임베디드 시스템 트랙
김종현
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs ➔ Architecture ➔ Design ➔ Outputs

Test & Verification
Collaboration
Scaling
<table>
<thead>
<tr>
<th>Time</th>
<th>Session Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:40</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>12:10</td>
<td>Java, Python, C++ for Big Data, MathWorks Korea</td>
</tr>
<tr>
<td>12:30</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>13:00</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>13:30</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>14:00</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>14:30</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>15:00</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>15:30</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>16:00</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>16:30</td>
<td>Attend Sessions this Afternoon</td>
</tr>
<tr>
<td>17:00</td>
<td>Attend Sessions this Afternoon</td>
</tr>
</tbody>
</table>
Read the Release Notes

R2019a at a Glance

Explore What's New
Get more out of MATLAB and Simulink by downloading the latest release.

Download release now

R2019a

Release Highlights

Deep Learning
Develop controllers and decision making systems using reinforcement learning, train deep learning models on NVIDIA DGX and cloud platforms, and apply deep learning to 3-D data.

» Learn more

Automotive
Design and simulate AUTOSAR software, interface with HERE HD maps, and generate energy balance reports.

» Learn more

Systems Engineering
Design and analyze system and software architectures with System Composer.

» Learn more

Projects
Use projects in MATLAB and Simulink to organize, manage, and share your work.

» Learn more
Get Started

MATLAB Onramp
Quickly learn the essentials of MATLAB.

Simulink Onramp
Learn to create, edit, and troubleshoot Simulink models.

Deep Learning Onramp
Learn to use deep learning techniques in MATLAB for image recognition.
MATLAB EXPO 2019

데모 부스와 상담부스로 질문 하시기 바랍니다.

감사합니다