MATLAB EXPO 2019

What’s New in MATLAB and Simulink

Cynthia Cudicini
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
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

Machine Learning

Data → COMPUTER → Model

Output
Artificial Intelligence

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

Data → Machine Learning → Deep Learning → Model

Inputs → Design → Outputs

MATLAB® & SIMULINK®
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

Deep Learning Toolbox
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
- Xception

- MobileNetV2
Using Models from Other Frameworks

MATLAB

Keras-Tensorflow

Caffe

PyTorch

ONNX

Caffe2

MXNet

Core ML

(...)

Deep Learning Toolbox
Deploying Deep Learning Applications

- Pre-processing
- Post-processing
- Deep Learning Networks

Coder Products

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

MATLAB Coder
GPU Coder
Using MATLAB and Simulink for Reinforcement Learning

Inputs: Data

Machine Learning
Deep Learning

Design

Model

Outputs

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

Data ➔ Machine Learning ➔ Deep Learning ➔ Model

Inputs ➔ Design ➔ Outputs

Reinforcement Learning Toolbox

© 2019 The MathWorks, Inc.
Using MATLAB and Simulink for Reinforcement Learning

Generate Data
- Scenario Design
- Simulation-based data generation

Machine Learning
Deep Learning

Model

Inputs
Design
Outputs

MATLAB & SIMULINK

Simulink
Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning
Using MATLAB & Simulink to Build Algorithms in Everything
Working with Text Data

Vehicle_Repairs.csv

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/20/2015</td>
<td>12:00:00</td>
<td>AM,14073,118743,14,04</td>
<td>DRIVER'S REPORT, &quot;PM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE WHEN DRIVING&quot;, 493.85, 0.493.85</td>
</tr>
<tr>
<td>10/21/2015</td>
<td>12:00:00</td>
<td>AM,14232,230593,13,08</td>
<td>PM SERVICE, &quot;SERVICEREG, EXT, 5604&quot;, 38.866999999999997, 0, 38.866999999999997</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14006,1243,116,04</td>
<td>DRIVER'S REPORT, NEED 4 PLOW PINS, 45, 0.45</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14114,B39109,178,04</td>
<td>DRIVER'S REPORT, INSTALL SPINNER ASSY, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14163,574950,215,13</td>
<td>SNOW BREAKDOWN, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14163,B00413,288,04</td>
<td>DRIVER'S REPORT, DOG BONE PIN BROKEN, 20, 0, 20</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14000,766153,248,08</td>
<td>PM SERVICE, &quot;NEED SERVICE, CHECK BRAKES&quot;, 387.17, 0, 387.17</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14155,525670,232,04</td>
<td>DRIVER'S REPORT, HYD CAP CHECK ENGINE LIGHT ON, 12.95, 0.12.95</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14157,621909,213,40</td>
<td>NEGLIGENCE, TARP VALVE STICKING RIGHT SIDE MIRROR BRACKET BROKEN, 50.02, 0.50.02</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14164,1226,117,13</td>
<td>SNOW BREAKDOWN, HANDLES IN CAB LOOSE, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14185,525999,114,04</td>
<td>DRIVER'S REPORT, NO FLOW LIGHTS, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14172,B34632,276,10</td>
<td>ROADCALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14174,1469,122,10</td>
<td>ROADCALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14175,68932,147,10</td>
<td>ROADCALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14176,68933,148,10</td>
<td>ROADCALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14177,621907,208,10</td>
<td>ROADCALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14181,337657,218,04</td>
<td>DRIVER'S REPORT, CONVEYOR NOT WORKING, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14182,D-1520,164,10</td>
<td>ROADCALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14183,525998,217,10</td>
<td>ROADCALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14184,526000,225,10</td>
<td>ROADCALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14185,621921,214,04</td>
<td>DRIVER'S REPORT, CONVEYOR NOT WORKING, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14185,8.1469,201,04</td>
<td>DRIVER'S REPORT, needs def/jim f, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14190,337656,219,04</td>
<td>DRIVER'S REPORT, NEEDS FLOOR MATS, 65.06999999999999, 0.65.06999999999993</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14191,B34632,276,10</td>
<td>ROADCALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14196,1222,118,04</td>
<td>DRIVER'S REPORT, HARDWARE FOR REAR SPRINGS, 14.32, 0.14.32</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14199,525655,626,04</td>
<td>DRIVER'S REPORT, WASHER FLUID DEF, 28.60, 0.28.60</td>
</tr>
<tr>
<td>11/1/2015</td>
<td>12:00:00</td>
<td>AM,14107,1467,121,08</td>
<td>PM SERVICE, &quot;REMOVE &amp; REPLACE REAR SPRINGS, CHECK COOLANT TUBES PM SERVIE&quot;, 4697.55, 0.0, 4697.55</td>
</tr>
</tbody>
</table>
Working with Text Data

```matlab
% Read text data from a file
filename = 'example_text_file.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;DON'T 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;HYD 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;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;CONVEYORY NOT WORKING&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
</tbody>
</table>
Working with Text Data

Deep Learning Toolbox
Statistics and Machine Learning Toolbox
Text Analytics Toolbox
MATLAB
Working with Text Data

Nouns

Adjectives

Nouns

Adjectives
Creating Your Own Data
Identifying the Useful Data

Acquire Data → Preprocess Data → Identify Condition Indicators → Train Model → Deploy & Integrate

- Visualize data
- Extract Features
- Select the most useful features
Identifying the Useful Data
Identifying the Useful Data

Signal Features
- Generate statistics from signals

Rotating Machinery Features
- Generate features from rotating machinery signals

Nonlinear Features
- Generate nonlinear features from signals

Spectral Features
- Condition variables: faultCode
- Computation mode: use full signal
- Spectral peaks
  - Peak amplitude
  - Peak frequency
- Peak value lower threshold: $\infty$
- Number of peaks: 1
- Minimum frequency gap: 0.001
- Peak excursion tolerance: 0
- Modal coefficients
- Band power

Predictive Maintenance Toolbox
Identifying the Useful Data
Designing Decision Logic with Stateflow

```matlab
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
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
```

© 2019 The MathWorks, Inc.
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

Simulink Coder

Column-Major

```c
#include "rtwdemo_row_lutcol2row_workflow_rowrow.h"

/* Block parameters (default storage) */

PrtP = {
    /* Variable: Tbl_1
        * 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
Viewing Generated Code Alongside the Model

Embedded Coder
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.

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

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

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.xlsx');
exoplanets(star);
```

**How Far Away Are these Planets?**

There are 90 exoplanets within 50 light-years of earth and 460 exoplanets within 200 light-years.

```matlab
histogram(exoplanets.st_distance, 50); %50 bins %show the x-axis
xlabel('Number of Planets'); %label the x-axis
ylabel('Light Years from Earth'); %label the y-axis
```

**Where is the nearest exoplanet?**

```matlab
idx = find(exoplanets.st_distance == min(exoplanets.st_distance));
name = char(exoplanets(star, idx_name));
```
Sharing Live Scripts

![MATLAB Live Script Example](image-url)

- **P**: 1:40
- **Slider**: 350
- **Drop down** "carbon dioxide"

**Graph**:
- **Title**: carbon dioxide @ 350 Kelvin
- **Y-axis**: Compressibility Factor, Z
- **X-axis**: 0.92 to 1.00
Creating Apps

Plate Browser  Summary Tables

Select Files  Current File: microtiter_data0001.csv

Microplate Plot

EC50 Curves

% Signal

Log [Compound]

<table>
<thead>
<tr>
<th>File</th>
<th>Compound Nr</th>
<th>NegControl</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_d</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>96.7684</td>
<td>97.1532</td>
<td>97.1910</td>
<td>97.1940</td>
</tr>
<tr>
<td>microtiter_d</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_d</td>
<td>3</td>
<td>0.0554</td>
<td>-0.4702</td>
<td>3.1908</td>
<td>52.9969</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_d</td>
<td>4</td>
<td>0.1096</td>
<td>0.2325</td>
<td>0.2335</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_d</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>96.2335</td>
<td>100.4717</td>
<td>100.5601</td>
<td>100.5700</td>
</tr>
</tbody>
</table>
Deploying Web Apps

MATLAB Web Apps

Transient Heat Conduction

Initial and Boundary Conditions
- Initial T (°C) 10
- Top T (°C) 0
- Bottom T (°C) 50
- Left T (°C) 25
- Right T (°C) 25

Geometry
- x (m) 0.05
- y (m) 0.05
- dx (m) 0.0025
- dy (m) 0.0025

Note: Numerical stability requires F, Current F = 0.0000

Thermal Diffusivity
- Alpha (m²/s) 1e-4

Time and Convergence
- dt (s) 0.01
- Total Time (s) 50
- Convergence Criterion: 1e-4

Start
Stop

Time = 35 s

MATLAB Compiler

© 2019 The MathWorks, Inc.
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs

MATLAB & Simulink®
Evaluating Architectures

Inputs → Architecture → Design → Outputs

MATLAB & SIMULINK®
Evaluating Architectures

Inputs → Architecture → Design → Outputs

MATLAB & SIMULINK®
Designing System and Software Architectures
Designing System and Software Architectures
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
Using MATLAB & Simulink to Build Algorithms in Everything
Integrating with Third-party Requirements Tools

External Requirements

- .doc
- .xls
- Database

Requirements Management Tools

Simulink Requirements

- External Requirements
- Authored Requirements

ReqIF

R2019a

Import
Edit
Export
Include Custom Code in Test & Verification
Using the MATLAB Unit Test Framework

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   [1x1 struct]
'test_Predictions/Test_Prediction'       false    true        true    0.11542   [1x1 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
- Sonargraph Integration
- JQueue
- Analysis Model API

MATLAB

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

MATLAB 1.0.0
Minimum Jenkins requirement: 2.7.3
ID: matlab

- Installs: No usage data available
  - GitHub →
  - Last released: 2 days ago

Maintainers
- MathWorks

Dependencies
- bouncycaesle API v.2.16.0
- Command Agent Launcher v.1.0
- JDK Tool v.1.0
- JAXB v.2.3.0

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>Test</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_MultiClassFaultDetectionUsing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>genPumpData.m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>javasetup.m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main_ExampleWorkflow.mx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLModels.mat</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rawdata.mat</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>README.md</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parallel Simulations in Simulink

Simulation Manager

batchsim

MATLAB Desktop

Simulation Jobs

Simulation Results

Simulink
Parallel Computing Toolbox
Scaling Computations on Clusters and Clouds

MATLAB Parallel Server

Parallel Computing Toolbox

Cloud

GPU

Multi-core CPU
Using MATLAB & Simulink to Build Algorithms in Everything
Specialized Tools for Building Algorithms in Everything

Communications

Physical interconnects

Analog Mixed-Signal

5G Toolbox

SerDes Toolbox

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
Using MATLAB & Simulink to Build Algorithms in Everything

 Inputs                      Architecture                      Design                      Outputs

---×---                      ---✓---                          ---✓---                      ---✓---

Test & Verification          Collaboration                    Scaling
Read the Release Notes
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.