MATLAB EXPO 2019

What’s New in MATLAB and Simulink

Prashant.Rao@mathworks.in
@_prashantrao_
prashantrao
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 → COMPUTER → Output

Program
There are two ways to get a computer to do what you want

Machine Learning

Data -> COMPUTER -> Model

Output
Artificial Intelligence

Machine Learning
Deep Learning

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

Inputs → Data → Machine Learning → Deep Learning → Model → Outputs

Data

Inputs

Model

Outputs

Machine Learning

Deep Learning
Using Apps for Ground Truth Labeling
Image and Video Data
Using Apps for Ground Truth Labeling

Signal Data

Signal Processing Toolbox
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

- AlexNet
- VGG-16
- GoogLeNet
- Inception-v3
- DenseNet-201
- Xception
- NasNetLarge
- VGG-19
- ResNet-50
- ResNet-101
- Inception-ResNet-v2
- MobileNet-v2
- NasNetMobile
- ResNet-18
- Places365-GoogLeNet
- SqueezeNet
- ShuffleNet

Timeline:
- 2016
- 2017
- 2018
- 2019
Using Models from Other Frameworks

MATLAB

Keras-Tensorflow

Caffe

ONNX

PyTorch

Caffe2

MXNet

Core ML

(...)
Deploying Deep Learning Applications

Deep Learning Application

Pre-processing → Post-processing

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

Training

Machine Learning

Deep Learning

Outputs

Model

Reinforcement Learning Toolbox

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

Inputs

Design

Outputs

Generate Data

Scenario Design

Simulation-based data generation

Machine Learning

Deep Learning

Model

MATLAB & SIMULINK

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

Find out more:
Deep Learning and Reinforcement Learning Workflows in A.I.

Avi Nehemiah
Deep Learning & Autonomous Systems Track
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
## Working with Text Data

### Example Data from Vehicle Repairs CSV

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Repair ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/20/2015</td>
<td>12:00:00</td>
<td>AM,14073,118743</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>10/20/2015</td>
<td>12:00:00</td>
<td>AM,14232,230973</td>
<td>PM SERVICE, &quot;SERVICEROB,EXT, 5604&quot;, 38.869999999999997, 0, 38.869999999999997</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14006,1243,116</td>
<td>DRIVER'S REPORT, NEED 4 FLOW PINS, 45, 0, 45</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14140,B39109</td>
<td>DRIVER'S REPORT, INSTALL SPINNER ASSY, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</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>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14166,B00413</td>
<td>DRIVER'S REPORT, DOG BONE PIN BROKEN, 20, 0, 20</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14000,768153,248,08</td>
<td>PM Service, &quot;NEED SERVICE, CHECK BRAKES&quot;, 387.17, 0, 387.17</td>
</tr>
<tr>
<td>11/01/2015</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>11/01/2015</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>11/01/2015</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>11/01/2015</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>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14172,B34632</td>
<td>ROAD CALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14174,1469,122,10</td>
<td>ROAD CALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14176,68932,147,10</td>
<td>ROAD CALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14177,621907,208,10</td>
<td>ROAD CALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</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>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14182,D-1250</td>
<td>ROAD CALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14183,525998,217,10</td>
<td>ROAD CALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14184,526000,225,10</td>
<td>ROAD CALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14185,621901,214,04</td>
<td>DRIVER'S REPORT, CONVEYOR NOT WORKING, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14186,014169</td>
<td>201,04 DRIVER'S REPORT, needs def/jim f, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14190,337656,219,04</td>
<td>DRIVER'S REPORT, NEEDS FLOOR MATTS 65.069999999999993, 0, 65.069999999999993</td>
</tr>
<tr>
<td>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14191,B34632</td>
<td>ROAD CALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>11/01/2015</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>11/01/2015</td>
<td>12:00:00</td>
<td>AM,14199,525655,626,04</td>
<td>DRIVER'S REPORT, WASHED FLUID DEF, 28.56, 0, 28.56</td>
</tr>
<tr>
<td>11/01/2015</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&quot; PM SERVIVE&quot;, 4697.55, 0, 4697.55</td>
</tr>
</tbody>
</table>
### Working with Text Data

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

1. Acquire Data
2. Preprocess Data
3. Identify Condition Indicators
4. Train Model
5. Deploy & Integrate

- Visualize data
- Extract Features
- Select the most useful features

Machine Learning
Identifying the Useful Data
Identifying the Useful Data

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;
        else
            counter = 0;
        end
    end
    if(inNormalRegion)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
```

Diagram:
- **Normal y=u;**
  - Transition: `[count(u<t1)>=N1]`
  - Next State: Abnormal y=0;
- **Abnormal y=0;**
  - Transition: `[count(u>=t2)>=N2]`
  - Back to Normal y=u;
Using Stateflow in MATLAB

% Callbacks that handle component events

methods (Access = private)

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

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

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

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

% Button pushed function: BLINKButton
function BLINKButtonPushed(app, event)
app.lantern_logic.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

#include "rtwdemo_rowlutcol2row_workflow_rowrow.h"

/* Block parameters (default storage) */
PrtP = {
  /* Variable: Tbl_1 */
  /* Referenced by: 'Data/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
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.

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

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.xlsx');
exoplanets(1:10,1:2);
```

How Far Away Are these Planets?

There are 90 exoplanets within 50 light-years of earth and 465 exoplanets within 200 light-years.

```matlab
histogram(exoplanets.st_distance, 'BinWidth', 50)
xlabel('Number of Planets')
ylabel('Light Years From Earth')
```

Where is the nearest exoplanet?

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

![Live Script Example](image)

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

Graph: **carbon dioxide @ 350 Kelvin**
Creating Apps

Plate Browser  Summary Tables

Select Files  Current File:  microtiter_data0001.csv

Microplate Plot

EC50 Curves

<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_data0001</td>
<td>1</td>
<td>-0.0741</td>
<td>0.3564</td>
<td>9.8759</td>
<td>58.8743</td>
<td>91.7323</td>
<td>96.7084</td>
<td>97.1532</td>
<td>97.1910</td>
<td>97.1940</td>
</tr>
<tr>
<td>microtiter_data0001</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>-0.4544</td>
<td>17.0436</td>
</tr>
<tr>
<td>microtiter_data0001</td>
<td>3</td>
<td>0.0554</td>
<td>-0.4702</td>
<td>3.1908</td>
<td>52.9996</td>
<td>97.5746</td>
<td>100.5606</td>
<td>100.6006</td>
<td>100.6006</td>
<td>100.6006</td>
</tr>
<tr>
<td>microtiter_data0001</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>41.1660</td>
<td>94.7343</td>
<td>100.6591</td>
<td>100.9487</td>
</tr>
<tr>
<td>microtiter_data0001</td>
<td>5</td>
<td>-0.0572</td>
<td>0.7481</td>
<td>1.7104</td>
<td>28.8872</td>
<td>84.5134</td>
<td>98.2335</td>
<td>100.4717</td>
<td>100.5601</td>
<td>100.5700</td>
</tr>
</tbody>
</table>

Previous File  Next File  Clear selection
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 Fp < 0.0005

Current Fp = 0.0003

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

Material:
- Copper
- Water

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

Graphs showing temperature distribution and time evolution.
Using MATLAB & Simulink to Build Algorithms in Everything
Evaluating Architectures

Architecture

Inputs

Design

Outputs
Evaluating Architectures

Inputs → Architecture → Design → Outputs

MATLAB® & SIMULINK®
Designing System and Software Architectures
Designing System and Software Architectures
Find out more:
Systems Engineering: Requirements to Architecture to Simulation

Gaurav Dubey
Systems Modeling, Implementation, and Verification Track
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

R2019a

Import
Edit
Export

ReqIF
Include Custom Code in Test & Verification

Simulink

C/C++

Simulink Design Verifier

Stateflow

C/C++

Test & Verification
Include Custom Code in Test & Verification

Find out more:
Simplifying Requirements-Based Verification with Model-Based Design

Vamshi Kumbham
Systems Modeling, Implementation, and Verification Track
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
- QRebel
- MATLAB
- MISRA Compliance Report
- VectorCAST Execution
- Zoom
- JqQuery
- Klocwork Community
- Analysis Model API

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

MATLAB

Minimum Jenkins requirement: 2.7.3
ID: matlab

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

Maintainers
MathWorks

Dependencies
- bouncyCastle API v.2.16.0 (implied) (what's this?)
- Command Agent Launcher v.1.0 (implied) (what's this?)
- JDK Tool v.1.0 (implied) (what's this?)
- JAXB v.2.3.0 (implied) (what's this?)

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

#### Project Directory Tree

<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_KafkaProducer_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_MultiClassFaultDetectionUsingJava</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.mxml</td>
<td>✔</td>
<td></td>
<td>Design</td>
</tr>
<tr>
<td>MLMModels.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

Simulation Jobs
Simulation Results

MATLAB Desktop

Worker
Worker
Worker

batchsim

Simulation Jobs
Simulation Results

Simulink
Parallel Computing Toolbox
Scaling Computations on Clusters and Clouds

MATLAB

Parallel Computing Toolbox

MATLAB Parallel Server

Cloud

GPU

Multi-core CPU
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

Test & Verification → Collaboration → Scaling

© 2019 The MathWorks, Inc.
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

Sensor Fusion and Tracking Toolbox
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:
Automated Driving System Design and Simulation

Dr. Amod Anandkumar
Deep Learning and Autonomous Systems Track
Using MATLAB & Simulink to Build Algorithms in Everything

- Inputs
- Architecture
- Design
- Outputs

- Test & Verification
- Collaboration
- Scaling
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