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
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
Using Apps for Ground Truth Labeling
Image and Video Data
Using Apps for Ground Truth Labeling
Signal Data
Using Apps for Ground Truth Labeling
Audio Data

Audio Toolbox
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
- Inception-ResNet-v2
- MobileNet-v2
- NasNetMobile
- ResNet-101
- ResNet-18
- Places365-GoogLeNet
- ShuffleNet
- SqueezeNet

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

MATLAB

Keras-Tensorflow

PyTorch

ONNX

Caffe2

MXNet

Core ML

(...)

Caffe

CNTK

Deep Learning Toolbox
Deploying Deep Learning Applications

Matlab Coder
GPU Coder

Pre-processing → Deep Learning Application → Post-processing → Coder Products

Intel MKL-DNN Library
NVIDIA TensorRT & cuDNN Libraries
ARM Compute Library
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
Using MATLAB and Simulink for Reinforcement Learning

Data → Machine Learning → Deep Learning → Model

Inputs → Design → Outputs

Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning

- **Generate Data**
  - Scenario Design
  - Simulation-based data generation

- **Design**
  - Machine Learning
  - Deep Learning

- **Outputs**
  - Model

Inputs

- **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

MATLAB & SIMULINK®
# Working with Text Data

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Time Stamp</th>
<th>Provider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-06-02</td>
<td>12:00</td>
<td>AM, 14073,</td>
<td>118743,</td>
<td>DRIVER'S REPORT,&quot;FM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE WHEN DRIVING&quot;, 493.85, 0.493.85</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14232,</td>
<td>230973,</td>
<td>&quot;FM SERVICE&quot;</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14006,</td>
<td>1243, 116</td>
<td>&quot;DRIVER'S REPORT, NEED 4 FLOW PINS, 45.0, 45</td>
</tr>
<tr>
<td>2015-06-02</td>
<td>12:00</td>
<td>AM, 14140,</td>
<td>B39109,</td>
<td>178.0, 0, DRIVER'S REPORT, INSTALL SPINNER ASSY, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14163,</td>
<td>574950, 215</td>
<td>SNOW BREAKDOWN, DONT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14163,</td>
<td>B39109,</td>
<td>178.0, 0, DRIVER'S REPORT, DOG BONE PIN BROKEN, 20, 20</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14000,</td>
<td>766153, 248.08</td>
<td>&quot;FM SERVICE&quot;</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14157,</td>
<td>621909, 213</td>
<td>&quot;NEGLIGENCE, TARP VALVE STICKING RIGHT SIDE MIRROR BRACKET BROKEN, 50.02, 0, 50.02</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14164,</td>
<td>1226, 117</td>
<td>13, SNOW BREAKDOWN, HANDLES IN CAB LOOSE, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14165,</td>
<td>525999, 114</td>
<td>04, DRIVERS REPORT, NO FLOW LIGHTS, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14172,</td>
<td>B34632,</td>
<td>1276.10, ROADCALL, WILL NOT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14172,</td>
<td>1469, 122</td>
<td>10, ROADCALL, WILL NOT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14176,</td>
<td>68932, 147</td>
<td>10, ROADCALL, WILL NOT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14176,</td>
<td>68933, 148</td>
<td>10, ROADCALL, WILL NOT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14177,</td>
<td>621907, 208</td>
<td>10, ROADCALL, WILL NOT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14181,</td>
<td>337657, 218</td>
<td>04, DRIVER'S REPORT, CONVEYOR NOT WORKING, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14182,</td>
<td>15920, 164</td>
<td>10, ROADCALL, DONT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14183,</td>
<td>525998, 217</td>
<td>10, ROADCALL, DONT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14184,</td>
<td>526000, 225</td>
<td>10, ROADCALL, DONT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14185,</td>
<td>621921, 214</td>
<td>04, DRIVER'S REPORT, CONVEYOR NOT WORKING, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14186,</td>
<td>301469, 201</td>
<td>04, DRIVER'S REPORT, NEEDS DEF/JIM F, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14190,</td>
<td>337657, 219</td>
<td>04, DRIVER'S REPORT, NEEDS FLOOR MATTS, 65.0, 65.09999999999993, 0.65.06999999999993</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14191,</td>
<td>B34632,</td>
<td>1276.10, ROADCALL, DONT START, 0, 0</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14196,</td>
<td>1222, 118</td>
<td>04, DRIVER'S REPORT, HARDWARE FOR REAR SPRINGS, 14.32, 0, 14.32</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14199,</td>
<td>52565, 626</td>
<td>04, DRIVER'S REPORT, WASHING FLUID DEF, 28.60, 0, 28.60</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14107,</td>
<td>1487, 121</td>
<td>08, FM SERVICE</td>
</tr>
<tr>
<td>2015-06-03</td>
<td>12:00</td>
<td>AM, 14107,</td>
<td>1487, 121</td>
<td>08, FM SERVICE</td>
</tr>
</tbody>
</table>
Working with Text Data

```matlab
% Read a table from a 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;SERVICERO8,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;HYD CAP CHECK ENGINE LIGHT ON&quot;</td>
</tr>
<tr>
<td>&quot;40 NEGLIGENCE&quot;</td>
<td>&quot;TARP VALVE STICKING RIGHT 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;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
Working with Text Data

Text Analytics Toolbox
MATLAB
Creating Your Own Data
Identifying the Useful Data

Data Acquisition
- Acquire Data
- Preprocess Data

Data Analysis
- Identify Condition Indicators

Machine Learning
- Train Model

Deployment
- Deploy & Integrate

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

Machine Learning
- Identify Condition Indicators

Select the most useful features
Identifying the Useful Data
Identifying the Useful Data

Signal Features
- Generate statistics from signals
- Generate features 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
- Number of peaks
- Minimum frequency gap
- Peak excursion tolerance
- Modal coefficients
- Band power
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
    if(inNormalRegion)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
```

Normal: \( y = u; \)

Abnormal: \( y = 0; \)

Transition conditions:
- \([\text{count}(u < t1) >= N1]\)
- \([\text{count}(u >= t2) >= N2]\)
Using Stateflow in MATLAB

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

% Code that executes after component creation
function startupFcn(app)
    app.lanternlogic = blink.lanternLogic('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

Simulink Coder

Column-Major

```c
#include "rtwdemo_rowlutcol2row_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

Row-Major
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.

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

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(zScores);
```

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(x20*exoplanets.st_distance), hold on
scatter(x20*exoplanets.st_distance); hold off
xlabel('Number of Planets')
ylabel('Light Years From Earth')
```

Where is the nearest exoplanet?

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

![Live Script with MATLAB interface and graph](image)

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

**Graph**:
- Title: carbon dioxide @ 350 Kelvin
- X-axis: Viscosity Factor, Z
- Y-axis: 1.0 to 0.92

[Hide Code]
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>-0.9741</td>
<td>0.3564</td>
<td>9.8759</td>
<td>56.8743</td>
<td>91.7323</td>
<td>96.7684</td>
<td>97.1532</td>
<td>57.1910</td>
<td>97.1940</td>
</tr>
<tr>
<td>microtiter_d...</td>
<td>2</td>
<td>-0.0143</td>
<td>-0.5044</td>
<td>-0.5944</td>
<td>-0.5944</td>
<td>-0.5944</td>
<td>-0.5944</td>
<td>-0.5944</td>
<td>0.0436</td>
<td>17.0436</td>
</tr>
<tr>
<td>microtiter_d...</td>
<td>3</td>
<td>0.0054</td>
<td>-0.4702</td>
<td>3.1906</td>
<td>52.9962</td>
<td>97.5746</td>
<td>100.5606</td>
<td>100.6096</td>
<td>100.6096</td>
<td>100.6096</td>
</tr>
<tr>
<td>microtiter_d...</td>
<td>4</td>
<td>-0.1986</td>
<td>0.2325</td>
<td>0.2355</td>
<td>0.3712</td>
<td>3.2339</td>
<td>41.1610</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>96.2395</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)
- Top T (°C)
- Bottom T (°C)
- Left T (°C)
- Right T (°C)

Geometry
- x (m)
- y (m)

Time and Convergence
- dt (s)
- Total Time (s)
- Convergence Criterion

Note: Numerical stability requires ε
Current Pe = 6.0000

Thermal Diffusivity
- Alpha (m²/s)

Mat: Copper or Water

Start | Stop

Time: 35 s

Output
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
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
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

Inputs → Architecture → Design → Outputs

Test & Verification → Collaboration → Scaling
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++

Stateflow

C/C++

Simulink Design Verifier

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:
- ORebel
- MATLAB
- MISRA Compliance Report
- Zoom
- VectorCAST Execution
- Klocwork Community
- jQuery
- Analysis Model API

MATLAB

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

MATLAB 10.0
Minimum Jenkins requirement: 2.7.3
ID: matlab

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

Maintainers
MathWorks

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

<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>Design</td>
</tr>
<tr>
<td>javasetup.m</td>
<td>✓</td>
<td>•</td>
<td>Design</td>
</tr>
<tr>
<td>Main_ExampleWorkflow.m</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

Simulink
Parallel Computing Toolbox

batchsim

Simulation Jobs
Simulation Results

MATLAB Desktop

Worker
Worker
Worker

Head

Input

Output

Design

Scaling
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
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
Developing Autonomous Systems

Lidar Processing & Tracking
HERE HD Maps & Roads
UAV Algorithms

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

MATLAB & Simulink

© 2019 The MathWorks, Inc.
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.