MATLAB EXPO 2019

What’s New in MATLAB and Simulink
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 → COMPUTER → Output

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

Machine Learning

- Data
- Output
- Model
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

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

Deep Learning Toolbox
Using Models from Other Frameworks

MATLAB → Keras-Tensorflow → Caffe → ONNX → PyTorch → Caffe2 → MXNet → Core ML

(…)

Deep Learning Toolbox
Deploying Deep Learning Applications

Pre-processing → Deep Learning Networks → Post-processing → Coder Products → Intel MKL-DNN Library

Coder Products → NVIDIA TensorRT & cuDNN Libraries

Coder Products → ARM Compute Library

MATLAB Coder
GPU Coder
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
Using MATLAB and Simulink for Reinforcement Learning

Inputs

Machine Learning

Deep Learning

Outputs

Data

Design

Model

Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning

Generate Data

- Scenario Design
- Simulation-based data generation

Inputs

Machine Learning

Deep Learning

Design

Model

Outputs
Using MATLAB and Simulink for Reinforcement Learning
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs

MATLAB & SIMULINK®

© 2019 The MathWorks, Inc.
### Working with Text Data

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Dept, JobNo, VehicleId, UnitNo, Reason, Notes</th>
<th>CostParts, CostLabor, CostTotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/10/2015</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>02/10/2015</td>
<td>12:00:00</td>
<td>AM, 14212, 230973, 13, 08</td>
<td>PM SERVICE &quot;SERVICERGB, EXT, 5604&quot;, 38.869999999999997, 0, 38.869999999999997</td>
</tr>
<tr>
<td>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14006, 1243, 116, 04</td>
<td>DRIVER'S REPORT, NEED 4 FLOW PINS, 45, 0, 45</td>
</tr>
<tr>
<td>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14140, 39109</td>
<td>178, 04 DRIVER'S REPORT, INSTALL SPINNER ASSY, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</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>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14168, 109413</td>
<td>293, 04 DRIVER'S REPORT, DENT ON TOP, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14000, 761153, 248, 08</td>
<td>PM SERVICE &quot;NEED SERVICE, CHECK BRAKES&quot;, 387.17, 0, 387.17</td>
</tr>
<tr>
<td>02/11/2015</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>02/11/2015</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>02/11/2015</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>02/11/2015</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>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14172, 34632</td>
<td>276, 10 ROAD CALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</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>02/11/2015</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>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14176, 68933, 148, 10</td>
<td>ROAD CALL, WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</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>02/11/2015</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>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14182, 3-920</td>
<td>164, 10 ROAD CALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</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>02/11/2015</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>02/11/2015</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>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14186, 304169</td>
<td>201, 04 DRIVER'S REPORT, needs def/jim f, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14190, 337656, 219, 04</td>
<td>DRIVER'S REPORT, NEEDS FLOOR MATS, 65.069999999999993, 0, 65.069999999999993</td>
</tr>
<tr>
<td>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14191, 34632</td>
<td>ROAD CALL, DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>02/11/2015</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>02/11/2015</td>
<td>12:00:00</td>
<td>AM, 14199, 52565, 626, 04</td>
<td>DRIVER'S REPORT, WASHER FLUID DEF, 28.66, 0, 28.66</td>
</tr>
<tr>
<td>02/11/2015</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 SERVICED&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>
</table>
| "04" DRIVER'S REPORT | "PM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE WHEN DRIVING"
| "08" PM SERVICE   | "SERVICEROB,EXT,5604"
| "04" DRIVER'S REPORT | "NEED 4 PLOW PINS"
| "04" DRIVER'S REPORT | "INSTALL SPINNER ASSY"
| "13" SNOW BREAKDOWN | "DON'T START"
| "04" DRIVER'S REPORT | "DOG BONE PIN BROKEN"
| "08" PM SERVICE   | "NEED SERVICE, CHECK BRAKES"
| "04" DRIVER'S REPORT | "HYD CAP CHECK ENGINE LIGHT ON"
| "40" NEGLIGENCE    | "TARP VALVE STICKING RIGHT SIDE MIRROR BRACKET BROKEN"
| "13" SNOW BREAKDOWN | "HANDLES IN CAB LOOSE"
| "04" DRIVER'S REPORT | "NO PLOW LIGHTS"
| "10" ROADCALL    | "WILL NOT START"
| "10" ROADCALL    | "WILL NOT START"
| "10" ROADCALL    | "WILL NOT START"
| "10" ROADCALL    | "WILL NOT START"
| "04" DRIVER'S REPORT | "CONVEYOR NOT WORKING"
| "10" ROADCALL    | "DON'T START"
| "10" ROADCALL    | "DON'T START"
| "10" ROADCALL    | "DON'T START"
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

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
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**
- Spectrum: Vibration_tsa_ps/Data
- 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**

Predictive Maintenance Toolbox

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

```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
```

- **Normal**
  - \([\text{count}(u<t1)\geq N1]\)
  - \(y = u;\)

- **Abnormal**
  - \([\text{count}(u\geq t2)\geq N2]\)
  - \(y = 0;\)
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

```c
#include "rtwdemo_rowlutcol2row_workflow_rowrow.h"

/* Block parameters (default storage) */
PTX_Param = {
  /* 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

Fuel Rate Control Subsystem

- sensors
- EngSensors D2
- control_logic
- single D2 (g/s)
- fuel_mode
- airflow Calc
- fuel_rate
- est_airflow (g/s)
- C2_normal
- fb_correction
- fuel_mode
- single D2 (g/s)
- firn_mode
- validate_sample_time
- boolean D2
- single D2 (g/s)
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);
```

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, binWidth, 50); xtick([10 50 100 500 1000 10000]) ylabel('Number of Planets') xtick([10 50 100 500 1000 10000]) xlabel('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

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>-.0741</td>
<td>.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_data0001</td>
<td>2</td>
<td>-.0143</td>
<td>-.5044</td>
<td>-.5044</td>
<td>-.5044</td>
<td>-.5044</td>
<td>-.5044</td>
<td>-.5044</td>
<td>.1704</td>
<td>.7461</td>
</tr>
<tr>
<td>microtiter_data0001</td>
<td>3</td>
<td>.0054</td>
<td>-.4702</td>
<td>3.1998</td>
<td>52.9904</td>
<td>97.5746</td>
<td>100.5608</td>
<td>100.6068</td>
<td>100.6068</td>
<td>100.6068</td>
</tr>
<tr>
<td>microtiter_data0001</td>
<td>4</td>
<td>.1066</td>
<td>.2325</td>
<td>.2385</td>
<td>.3712</td>
<td>.3233</td>
<td>.4118</td>
<td>.94.7343</td>
<td>100.6591</td>
<td>100.9487</td>
</tr>
<tr>
<td>microtiter_data0001</td>
<td>5</td>
<td>.0572</td>
<td>.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

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): 0.25
- y (m): 0.25
- dx (m): 0.0025
- dy (m): 0.0025

Note: Numerical stability requires Fp < 0.0003
Current Fp = 0.0033

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

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

Deploying Web Apps using MATLAB Compiler
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
Designing System and Software Architectures
Designing System and Software Architectures
Designing **Beyond** System and Software Architectures

- Systems and Software
  - System Composer

- SoC Hardware and Software
  - SoC Blockset

- AUTOSAR Software
  - AUTOSAR Blockset
Using MATLAB & Simulink to Build Algorithms in Everything
Using MATLAB & Simulink to Build Algorithms in Everything

- Inputs
- Architecture
- Design
- Outputs

Test & Verification  Collaboration  Scaling
Integrating with Third-party Requirements Tools

**External Requirements**
- .doc
- .xls
- Database

**Simulink Requirements**
- External Requirements
- Authored Requirements

**ReqIF**
- Import
- Edit
- Export

**Simulink Requirements Management Tools**
Include Custom Code in Test & Verification

Simulink

C/C++

Stateflow

C/C++

Simulink Design Verifier

- ✗

- ✔
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

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
- |Query
- Analysis Model API

MATLAB

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

MATLAB 10.0

Minimum Jenkins requirement: 2.7.3
ID: matlab

Maintainers
MathWorks

Dependencies
- bouncyCastle API v.2.16.0 (implies) (what's this?)
- Command Line 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
Parallel Simulations in Simulink

Simulation Manager

batchsim

MATLAB Desktop

Simulation Jobs

Simulation Results

Worker

Worker

Worker

Head Worker

Simulink
Parallel Computing Toolbox

© 2019 The MathWorks, Inc.
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
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

Automated Driving Toolbox
Design Lane-following and Spacing Control Algorithms
Developing Autonomous Systems

Lidar Processing & Tracking
Computer Vision Toolbox

HERE HD Maps & OpenDRIVE Roads
Automated Driving Toolbox

UAV Algorithms
Robotics System Toolbox
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

Test & Verification
Collaboration
Scaling
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

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