MATLAB EXPO 2019

Novedades de las últimas versiones de MATLAB y Simulink

Paz Tárrega y David Pérez
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
Using MATLAB and Simulink to Build Deep Learning Models

Inputs

Data

Machine Learning

Deep Learning

Outputs

Model
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
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
- ShuffleNet
- SqueezeNet

2016 2017 2018 2019
Using Models from Other Frameworks

MATLAB 

Keras-Tensorflow

Caffe

PyTorch

ONNX

Caffe2

MXNet

Core ML

(...)

Deep Learning Toolbox
Deploying Deep Learning Applications

Pre-processing → Deep Learning Application → 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

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

Inputs

Design

- Machine Learning
- Deep Learning

Outputs

Model

MATLAB & Simulink

Simulink
Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>AM/PM</th>
<th>ID</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14009</td>
<td>DRIVER'S REPORT</td>
<td>PM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE WHEN DRIVING, 493.85, 0.493.85</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14232</td>
<td>DRIVER'S REPORT</td>
<td>PM SERVICE</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14006</td>
<td>DRIVER'S REPORT</td>
<td>NEED 4 FLOW PINS, 45, 0.45</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14140</td>
<td>DRIVER'S REPORT</td>
<td>INSTALL SPINNER ASSY, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14163</td>
<td>SNOW BREAKDOWN</td>
<td>DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14000</td>
<td>DRIVER'S REPORT</td>
<td>DOG BONE PIN BROKEN, 20, 0, 20</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14156</td>
<td>PM SERVICE</td>
<td>**&quot;NEED SERVICE, CHECK BRAKES&quot;, 387.17, 0, 387.17</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14155</td>
<td>DRIVER'S REPORT</td>
<td>HYD CAP CHECK ENGINE LIGHT ON, 12.95, 0, 12.95</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14157</td>
<td>NEGLIGENCE</td>
<td>TARP VALVE STICKING RIGHT SIDE MIRROR BRACKET BROKEN, 50, 0.0, 50.02</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14164</td>
<td>SNOW BREAKDOWN</td>
<td>HANDLES IN CAB LOOSE, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14185</td>
<td>DRIVER'S REPORT</td>
<td>NO FLOW LIGHTS, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14172</td>
<td>ROADCALL</td>
<td>WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14174</td>
<td>ROADCALL</td>
<td>WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14160</td>
<td>ROADCALL</td>
<td>WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14181</td>
<td>ROADCALL</td>
<td>WILL NOT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14182</td>
<td>ROADCALL</td>
<td>CONVEYOR NOT WORKING, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14183</td>
<td>ROADCALL</td>
<td>DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14184</td>
<td>ROADCALL</td>
<td>DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14185</td>
<td>ROADCALL</td>
<td>DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14186</td>
<td>ROADCALL</td>
<td>CONVEYOR NOT WORKING, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14190</td>
<td>DRIVER'S REPORT</td>
<td>NEEDS FLOOR MATS, 65.06999999999993, 0.65.06999999999993</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14191</td>
<td>ROADCALL</td>
<td>DONT START, 0, 0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14196</td>
<td>ROADCALL</td>
<td>HARDWARE FOR REAR SPRINGS, 14.32, 0, 14.32</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14159</td>
<td>ROADCALL</td>
<td>WASHER FLUID DEF, 28, 0.0, 28.00</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14107</td>
<td>ROADCALL</td>
<td>PM SERVICE, REMOVE &amp; REPLACE REAR SPRINGS, 0, 0.0, 0</td>
</tr>
<tr>
<td>2015-01-02</td>
<td>12:00:00</td>
<td>AM</td>
<td>14167</td>
<td>PM SERVICE</td>
<td>*** &quot;REMOVE &amp; REPLACE REAR SPRINGS, CHECK COOLANT TUBES/PM SERVICE&quot;, 4697.55, 0.0, 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>&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</td>
<td>&quot;SERVICEROB,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</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

Deep Learning Toolbox
Statistics and Machine Learning Toolbox
Text Analytics Toolbox
MATLAB
Creating Your Own Data
Identifying the Useful Data

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

Identify the useful data by acquiring and preprocessing data, then identifying condition indicators. Extract features and select the most useful ones. Train a model using machine learning techniques and deploy the solution for integration.
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

**Design**

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
```

Stateflow Diagram:
- **Normal** state:
  - Formula: \( \text{count}(u<t1) \geq N1 \)
  - Action: \( y = u \)

- **Abnormal** state:
  - Formula: \( \text{count}(u \geq t2) \geq N2 \)
  - Action: \( y = 0 \)
Using Stateflow in MATLAB

Stateflow
MATLAB

© 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
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) */
 PertP = {
    /* Variable: Tbl_1 */
    * Referenced by: '<Root>/2-D Lookup Table'
    */
    Tbl_1 = {
        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

Fuel Rate Control Subsystem

- sensors
- single D2 (g/s)
- validate_sample_time
- EngSensors D2
- es_i
- es_o
- control_logic
- fuel_mode
- airflow_calc
- fb_correction
- single D2 (g/s)
- C2_normal
- sensors
- est_airflow (g/s)
- est_airflow (g/s)
- single D2 (g/s)
- fuel_rate
- fuel_rate (g/s)
- fuel_mode
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.

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

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

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

MATLAB

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(st(:,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, [0 1000]);
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, 'name'));
```
Sharing Live Scripts

![Live Editor with MATLAB code and output](image)

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

**Output**: Chart showing the compressibility factor, Z, of carbon dioxide at 350 Kelvin.
Creating Apps

Plate Browser  Summary Tables

Select Files  Current File: microtiter_data0001.csv

Microplate Plot

EC50 Curves

File  Compound Nr  NegControl  Conc1  Conc2  Conc3  Conc4  Conc5  Conc6  Conc7  Conc8
michrotier_d...  1  -0.9741  0.3564  9.8759  55.8743  91.7323  96.7684  97.1532  57.1910  57.1940
michrotier_d...  2  -0.0143  -0.5044  -0.5044  -0.5044  -0.5044  -0.5044  -0.5044  -0.5044  17.0436
michrotier_d...  3  0.0054  -0.4702  3.1908  52.9966  97.5746  100.5606  100.6086  100.6086  100.6086
michrotier_d...  4  0.1096  0.2325  0.2385  0.3712  3.2339  41.1680  94.7343  100.6591  100.9487
michrotier_d...  5  -0.0572  0.7481  1.7104  26.8872  84.5134  96.2335  100.4717  100.5601  100.5700
Deploying Web Apps
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs

MATLAB & SIMULINK®
Evaluating Architectures

Architecture

Inputs

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

Requirements Management Tools

Simulink Requirements
- External Requirements
- Authored Requirements

ReqIF
- Import
- Edit
- Export

R2019a

Test & Verification
Include Custom Code in Test & Verification

Simulink

C/C++

Stateflow

C/C++

Simulink Design Verifier

Test & Verification
Using the MATLAB Unit Test Framework

```matlab
>> result.table
ans =
  2x6 table

<table>
<thead>
<tr>
<th>Name</th>
<th>Passed</th>
<th>Failed</th>
<th>Incomplete</th>
<th>Duration</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>'test_Predictions/Test_ModelType'</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>0.12241</td>
<td>[1x1 struct]</td>
</tr>
<tr>
<td>'test_Predictions/Test_Prediction'</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>0.11542</td>
<td>[1x1 struct]</td>
</tr>
</tbody>
</table>
```
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.

Plugins Index

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

Jenkins

Find plugins

MATLAB 10.0
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
Parallel Simulations in Simulink

Simulation Manager

batchsim

Simulation Jobs

Simulation Results

Simulation Manager

Simulink

Parallel Computing Toolbox
Scaling Computations on Clusters and Clouds

Parallel Computing Toolbox

MATLAB Parallel Server

MATLAB

Cloud

GPU

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

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.