MATLAB EXPO 2019

What’s New in MATLAB and Simulink

Mehernaz Savai
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
Using MATLAB & Simulink to Build Machine Learning Models

[Diagram showing the process of using data for machine learning and deep learning to design a model and derive outputs.]
Using Apps for Ground Truth Labeling

Image and Video Data

Computer Vision Toolbox
Using Apps for Ground Truth Labeling
Signal Data

![Labeling Screen](image)

<table>
<thead>
<tr>
<th>Label Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhaType</td>
</tr>
<tr>
<td>MoenRegions</td>
</tr>
<tr>
<td>TriRegions</td>
</tr>
<tr>
<td>TriPeaks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Labeled Signal Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>whale1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>whale2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>whale1</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Using Apps for Ground Truth Labeling
Audio Data
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: Inf
- Number of peaks: 1
- Minimum frequency gap: 0.001
- Peak excursion tolerance: 0
- Modal coefficients
- Band power

Predictive Maintenance Toolbox
Identifying the Useful Data
Using Apps for Designing Deep Learning Networks
Using Transfer Learning with Pre-trained Models

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

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

- Keras-Tensorflow
- Caffe
- PyTorch
- ONNX
- CNTK
- 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

→

Design

Machine Learning

Deep Learning

→

Outputs

Model

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

Find out more:

2:30 PM: Deep Learning and Reinforcement Learning Workflows in A.I.

3:30 PM: AI Techniques in MATLAB for Signal, Time-Series, and Text Data

4:30 PM: Deploying Deep Neural Networks to Embedded GPUs and CPUs
Using MATLAB & Simulink to Build Algorithms in Everything
# Working with Data

## Business and Transactional Data

### Repositories
- Databases (SQL/NoSQL)
- Hadoop

### File I/O
- Text
- Spreadsheet

### Web Sources
- RESTful/SOAP
- JSON
- HTML/XML
- Mapping
- Financial datafeeds
- FTP

## Recent Additions

### File I/O
- PDF
- Microsoft Word
- Parquet
- Vector BLF
- STL (Stereolithography)

### Web Sources
- Amazon Web Services
- Azure Blob Storage
- ThingSpeak

### Engineering, Scientific and Field Data

### File I/O
- CDF/HDF
- Audio/Image/Video
- Geospatial
- Microarrays
- CAD Models
- MDF

### Communication Protocols
- CAN (Controller Area Network)
- DDS (Data Distribution Service)
- OPC (OLE for Process Control) (e.g. PI)
- XCP (eXplicit Control Protocol)
- TCP/IP
- Serial/Bluetooth/USB

### Real-Time Sources
- Sensors/Instrumentation/Cameras
- GPS
- Communication systems
- Machines (embedded systems)
- Robot Operating System (ROS)

The above list is not all-inclusive, but is intended for guidance only.
Live Editor Tasks

Data Analytics - Load Forecasting Case Study

Load messy data

```
load LELdata.mat
head(ny1so)
```

Missing Data

```
table

<table>
<thead>
<tr>
<th>Date</th>
<th>Capital</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/01/2007</td>
<td>801.0000</td>
<td>1.571</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>891.5000</td>
<td>1.558</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>950.1000</td>
<td>1.569</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>888.7000</td>
<td>1.599</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>868.5000</td>
<td>1.558</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>948.0000</td>
<td>1.564</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>941.0000</td>
<td>1.535</td>
</tr>
<tr>
<td>05/01/2007</td>
<td>838.4000</td>
<td>1.557</td>
</tr>
</tbody>
</table>
```
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
    end
    if(inNormalRegion)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
```

Diagram:
- **Normal**:
  - `y = u`;
  - Transition: `[count(u<t1) >= N1]` to `Abnormal`;
  - Transition: `[count(u>=t2) >= N2]` to `Abnormal`;
- **Abnormal**:
  - `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

- Automatic Port Creation
- Edit on Block Icon
- Block Parameter Autocomplete
- Predictive Quick Insert
Controlling the Execution of Model Components

Schedulable Rate-Based Model

Export Function Model
Controlling the Execution of Model Components
More Ways to Componentize Your Design
Simplifying Blockset Creation and Sharing

Blockset Designer

Ideas → Code Templates → Design → Build → Test → Simulink Blocks → Packaged Toolbox → Organization

Development → Build → Test → Documentation → Publish
Viewing Generated Code Alongside the Model
Sharing Live Scripts
Deploying Web Apps
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
Evaluating Architectures

Inputs → Architecture → Design → Outputs

MATLAB & SIMULINK®
Designing System and Software Architectures

- Mechanical System View
- Power System View
- Control System View
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
Integrating with Third-party Requirements Tools

External Requirements

- .doc
- .xls

Requirements Management Tools

Simulink Requirements

- External Requirements
- Authored Requirements

R2019a

Import
Edit
Export

ReqIF
Include Custom Code in Test & Verification

Simulink

Stateflow

C/C++

Simulink Design Verifier

Test & Verification

- ✔

- ✗
Validating Function Arguments

```matlab
% Error check required input arguments
if nargin < 1
    error("rectangle requires width and height values");
elseif ~isnumeric(width) || ~isscalar(width)
    error("width must be a scalar numeric value")
elseif ~isnumeric(height) || ~isscalar(height)
    error("height must be a scalar numeric value")
end

% Process optional inputs xStart and yStart
xStart = 0;
if nargin > 2 && isnumeric(varargin{1}) && isscalar(varargin{1})
    xStart = varargin{1};
end
yStart = 0;
if nargin > 3 && isnumeric(varargin{2}) && isscalar(varargin{2})
    yStart = varargin{2};
end
```

Arguments:
- `width` (1,1) double `{mustBeNumeric}`
- `height` (1,1) double `{mustBeNumeric}`
- `xStart` (1,1) double `{mustBeNumeric}` = 0; % optional
- `yStart` (1,1) double `{mustBeNumeric}` = 0; % optional
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 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></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_MultiClassFaultDetectionUsingKafkaProducer.java</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></td>
</tr>
<tr>
<td>Main_ExampleWorkflow.m</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></td>
</tr>
</tbody>
</table>
Parallel Simulations in Simulink
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

Find out more:

2:30 PM: Wired Communications Systems Modeling and Analysis.

3:30 PM: Top-Down Modeling and Analysis of Analog Mixed-Signal Systems

4:30 PM: Understanding and Modeling the 5G NR Physical Layer
Developing Autonomous Systems

- Perception
- Planning
- Control
Developing Autonomous Systems

Find out more:

12:00 PM: Design and Test of Automated Driving Algorithms

4:00 PM: Sensor Fusion and Tracking for Autonomous Systems
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

Test & Verification → Collaboration → Scaling
### Attend Sessions this Afternoon

<table>
<thead>
<tr>
<th>Track 1: Siskiyou/Donner</th>
<th>Track 2: Sierra</th>
<th>Track 3: San Jose/Santa Clara</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>11:00 a.m.</strong></td>
<td><strong>11:30 a.m.</strong></td>
<td><strong>12:00 p.m.</strong></td>
</tr>
<tr>
<td>All Marjannejad, University of Southern California</td>
<td>Jonggab Ki, Intel</td>
<td>Matt Brown, Avnet</td>
</tr>
<tr>
<td><strong>12:30 p.m.</strong></td>
<td><strong>1:30 p.m.</strong></td>
<td><strong>2:00 p.m.</strong></td>
</tr>
<tr>
<td>Lunch and Technology Showcase: Bayshore Foyer and Cascade</td>
<td>Insights into MATLAB — Memory Handling and Datatypes</td>
<td>Adopting Model-Based Design for FPGA, ASIC, and SoC</td>
</tr>
<tr>
<td>Women in Tech Ignite Lunch and Networking: Carmel/Monterrey</td>
<td>Loren Shure, MathWorks</td>
<td>Robert Anderson, MathWorks</td>
</tr>
<tr>
<td><strong>2:30 p.m.</strong></td>
<td><strong>3:00 p.m.</strong></td>
<td><strong>3:30 p.m.</strong></td>
</tr>
<tr>
<td>Deep Learning and Reinforcement Learning Workflows in AI</td>
<td>Wired Communications Systems Modeling and Analysis</td>
<td>Planning Simulink Model Architecture and Modeling Patterns for ISO 26262 Compliance</td>
</tr>
<tr>
<td>Abhijit Bhattacharjee, MathWorks</td>
<td>Barry Katz, MathWorks</td>
<td>David Hoadley, MathWorks</td>
</tr>
<tr>
<td><strong>4:00 p.m.</strong></td>
<td><strong>4:30 p.m.</strong></td>
<td><strong>5:00 p.m.</strong></td>
</tr>
<tr>
<td>Sensor Fusion and Tracking for Autonomous Systems</td>
<td>Deploying Deep Neural Networks to Embedded GPUs and CPUs</td>
<td>Digital Twins for Smart Manufacturing</td>
</tr>
<tr>
<td>Rick Gentle, MathWorks</td>
<td>Abhijit Bhattacharjee, MathWorks</td>
<td>Pallavi Kar, MathWorks</td>
</tr>
</tbody>
</table>
Read the Release Notes

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 simulate Simulink models.

Stateflow Onramp
Learn to create, edit, and simulate state machines.

Deep Learning Onramp
Learn to use deep learning techniques in MATLAB.