MATLAB EXPO 2018

Predictive Maintenance

Using MATLAB and Simulink

Amit Doshi,
Senior Application Engineer – Data Analytics
MathWorks India

Why do well-designed engineering systems fail?

- Example: faulty braking system leads to windmill disaster
  - https://youtu.be/-YJuFvjtM0s?t=39s
- Systems like these cost millions of dollars
- Failures can be dangerous
- Maintenance also very expensive and dangerous

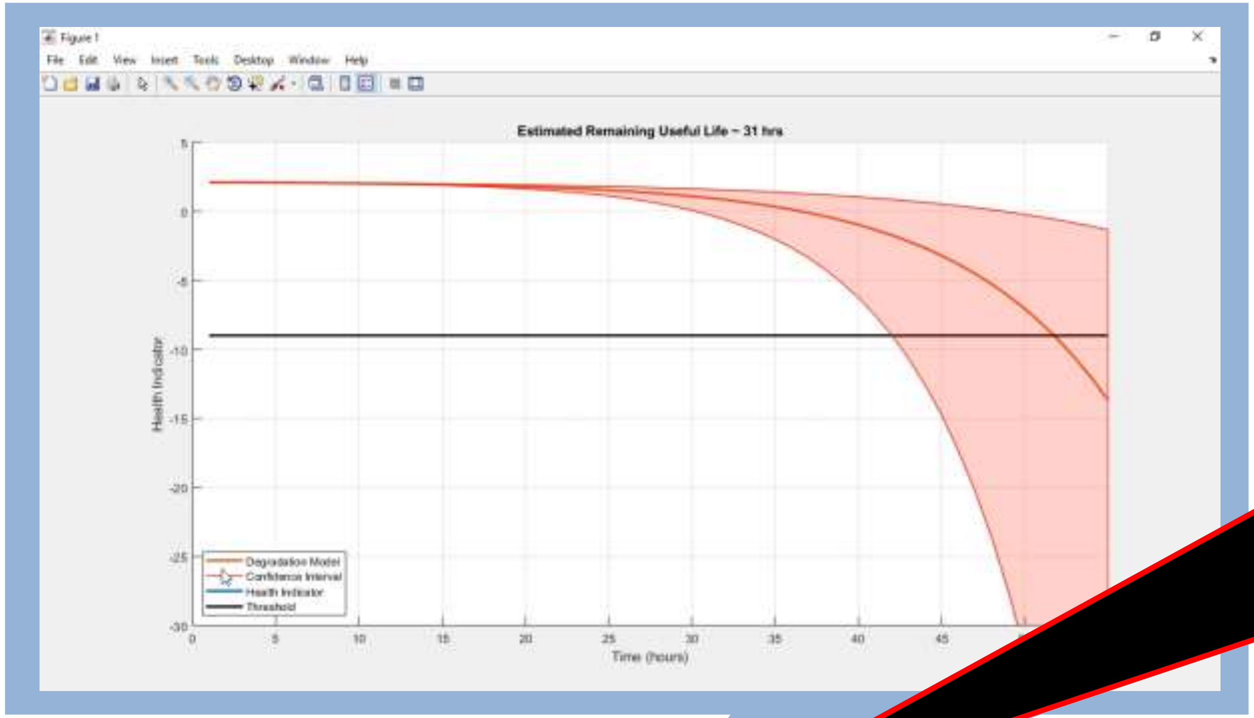
Types of Maintenance

- Reactive – Do maintenance once there’s a problem
  - Example: replace car battery **when it has a problem**
  - **Problem**: unexpected failures can be expensive and potentially dangerous

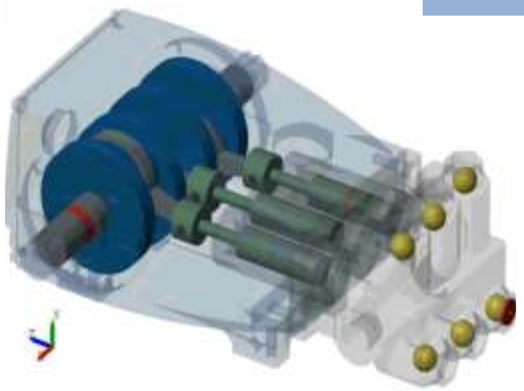
- Scheduled – Do maintenance at a regular rate
  - Example: change car’s oil every 5,000 miles
  - **Problem**: unnecessary maintenance **can be wasteful**; **may not eliminate** all failures

- Predictive – Forecast when problems will arise
  - Example: certain GM car models forecast problems with the battery, fuel pump, and starter motor
  - **Problem**: **difficult to make accurate** forecasts for complex equipment

Predictive Maintenance: Example



ALARM:
Pump-15, location- Rocky Mountain Site, needs Urgent maintenance.
Details: One of it's cylinders is blocked. It will shut down your line in 15 hours.



Site Engineer

Benefits of Predictive Maintenance

Increase “up time” and safety — **Reliability**

Minimize maintenance costs — **Cost of Ownership**

Optimize supply chain — **Reputation**

Cost of rig: >$1M
Repair cost: $100,000

Cost of valve: $200

Baker Hughes Develops Predictive Maintenance Software for Gas and Oil Extraction Equipment Using Data Analytics and Machine Learning

Truck with positive displacement pump.

Challenge

Develop a predictive maintenance system to reduce pump equipment costs and downtime

Solution

Use MATLAB to **analyze nearly one terabyte** of data and create a **neural network** that can predict machine failures before they occur

Results

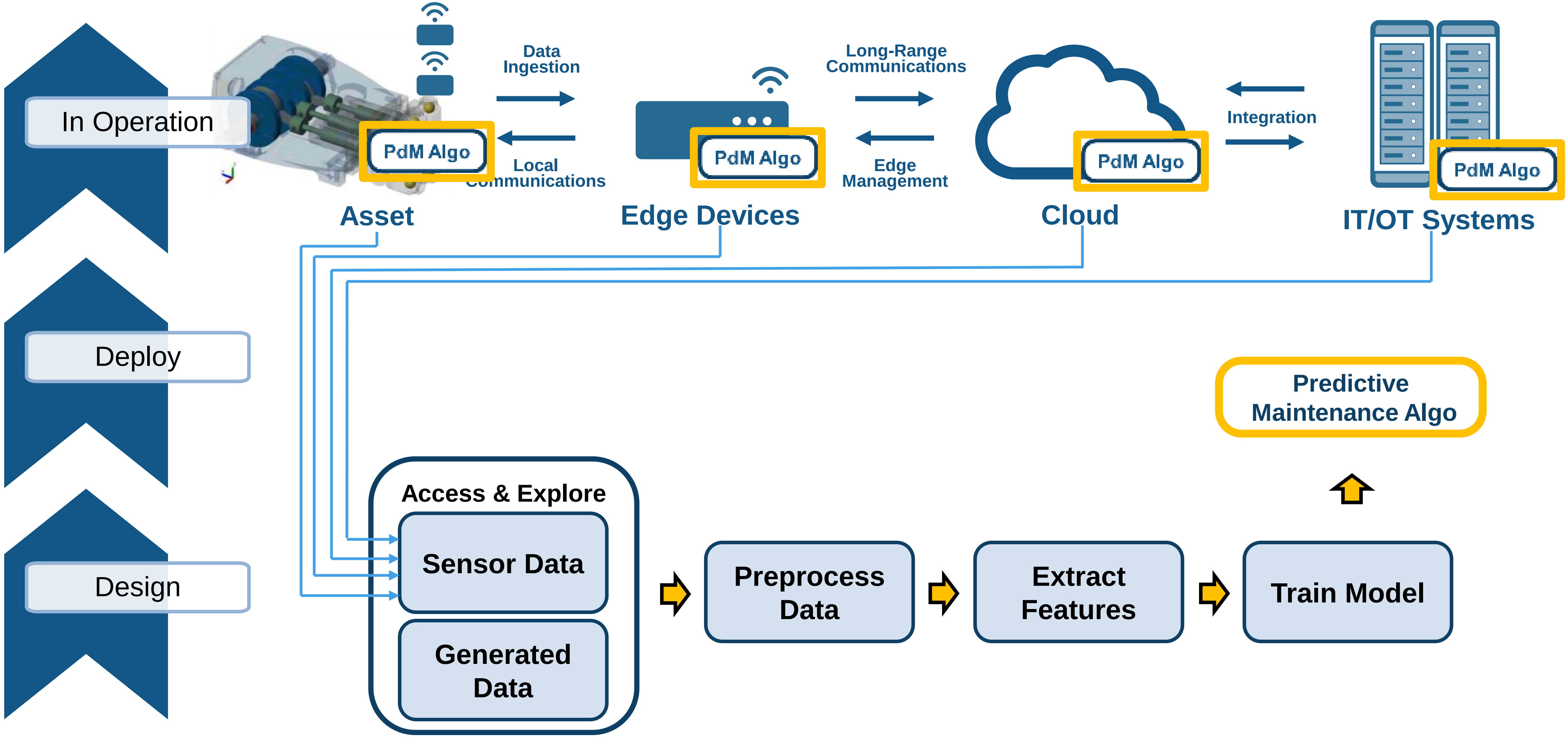
- Savings of more than **$10 million** projected
- Development time reduced tenfold
- Multiple types of data easily accessed

“MATLAB gave us the ability to convert previously unreadable data into a usable format; automate filtering, spectral analysis, and transform steps for multiple trucks and regions; and ultimately, apply machine learning techniques in real time to predict the ideal time to perform maintenance.”

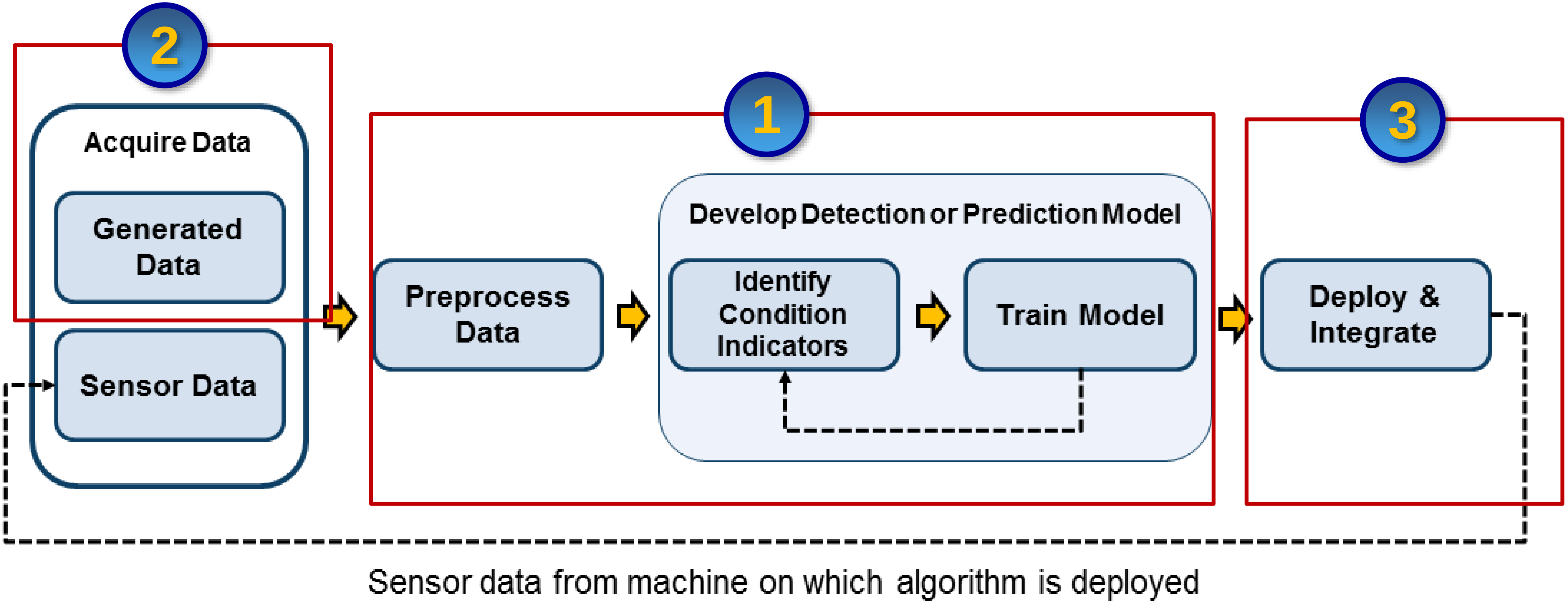
- Gulshan Singh, Baker Hughes

Link to user story

Predictive Maintenance Solution Framework - IIoT

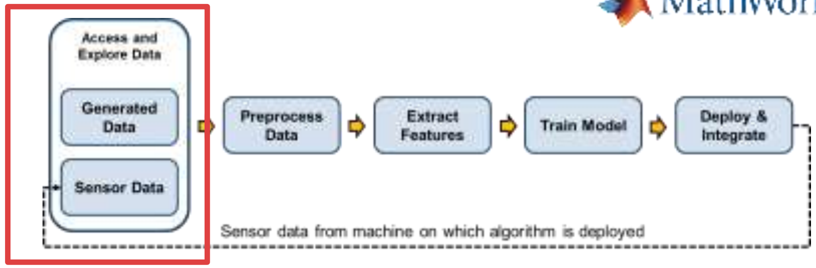


Workflow for developing a predictive maintenance algorithm



Access Data

Session at 4:30 pm



Datastore

Read large collections of data

The `datastore` function creates a datastore, which is a repository for collections of data that are too large to fit in memory. A datastore allows you to read and process data stored in multiple files on a disk, a remote location, or a database as a single entity. If the data is too large to fit in memory, you can manage the incremental import of data, create a `tall` array to work with the data, or use the datastore as an input to `mapreduce` for further processing. For more information, see Getting Started with Datastore.

Functions

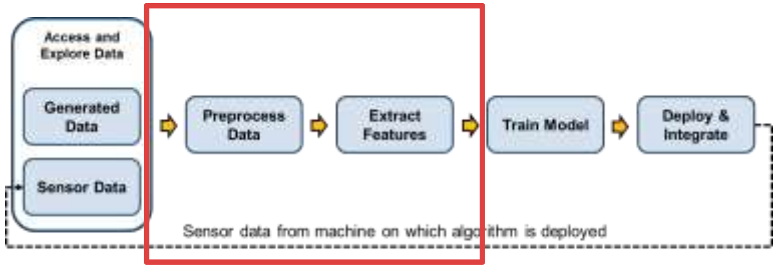
expand all

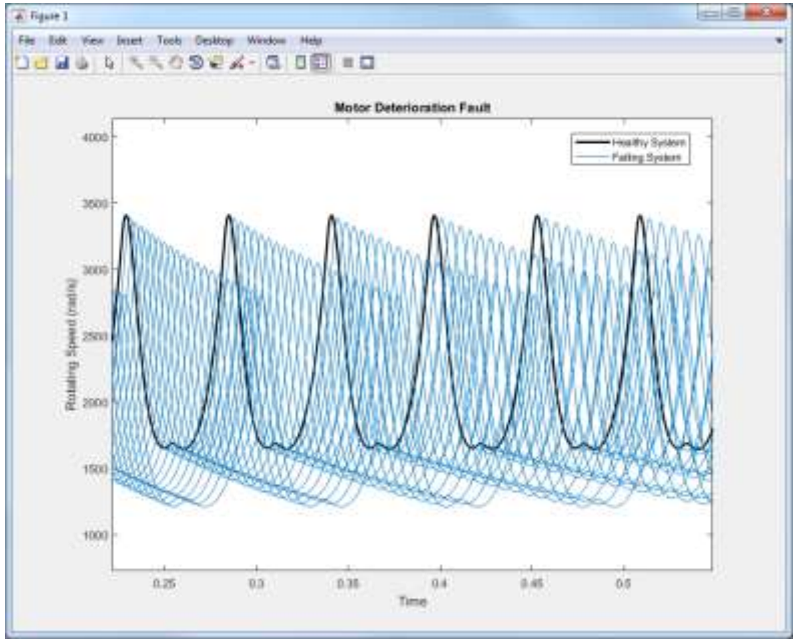
Create Datastore

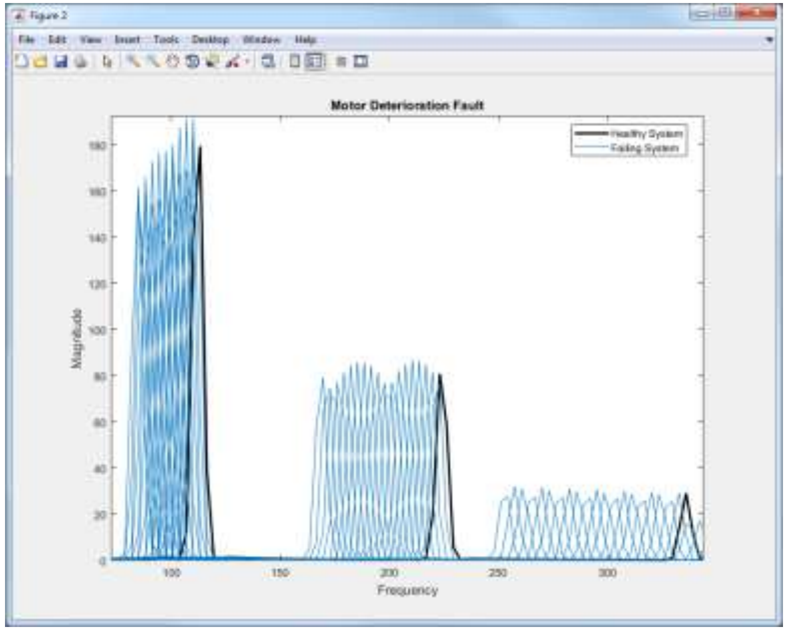
`datastore`	Create datastore for large collections of data
`TabularTextDatastore`	Datastore for tabular text files
`SpreadsheetDatastore`	Datastore for spreadsheet files
`ImageDatastore`	Datastore for image data
`FileDatastore`	Datastore with custom file reader

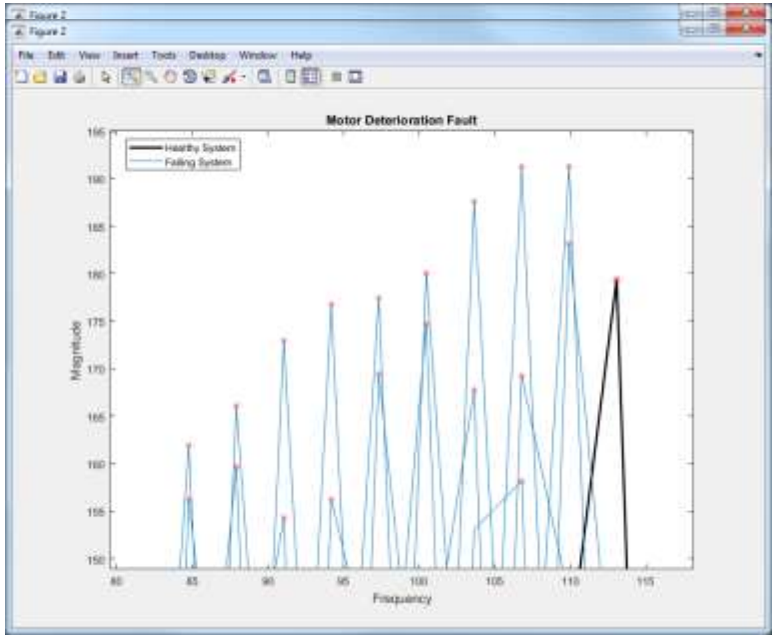
`datastore`

Preprocess Data & Extract Features









Failure Data (Sensors/Simulation)

Preprocessed Data

Health Indicators

MathWorks

Documentation

Search Help

CONTENTS

Identify Condition Indicators

Explore data to identify features that can indicate system state or predict future states

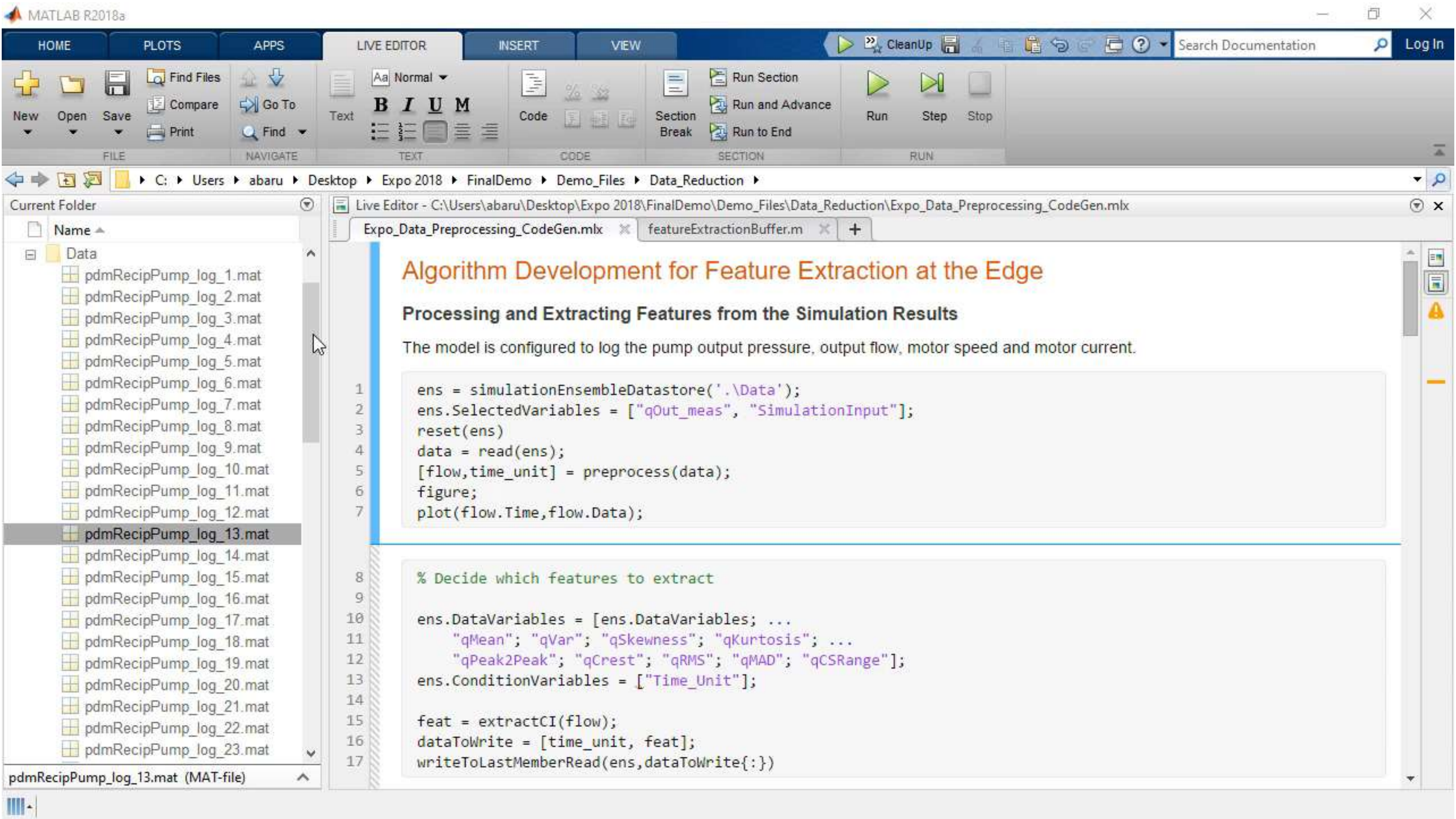
A *condition indicator* is a feature of system data whose behavior changes in a predictable way as the system degrades or operates in different operational modes. A condition indicator can be any feature that is useful for distinguishing normal from faulty operation or for predicting remaining useful life. A useful condition indicator clusters similar system status together, and sets different status apart.

You can derive condition indicators from signal analysis, by extracting time-domain or frequency-domain features of system data. You can also derive condition indicators by fitting static or dynamic models to your data, and examining model parameters or model behavior to distinguish fault states or predict system degradation. For more information, see Condition Indicators for Monitoring, Fault Detection, and Prediction.

Functions

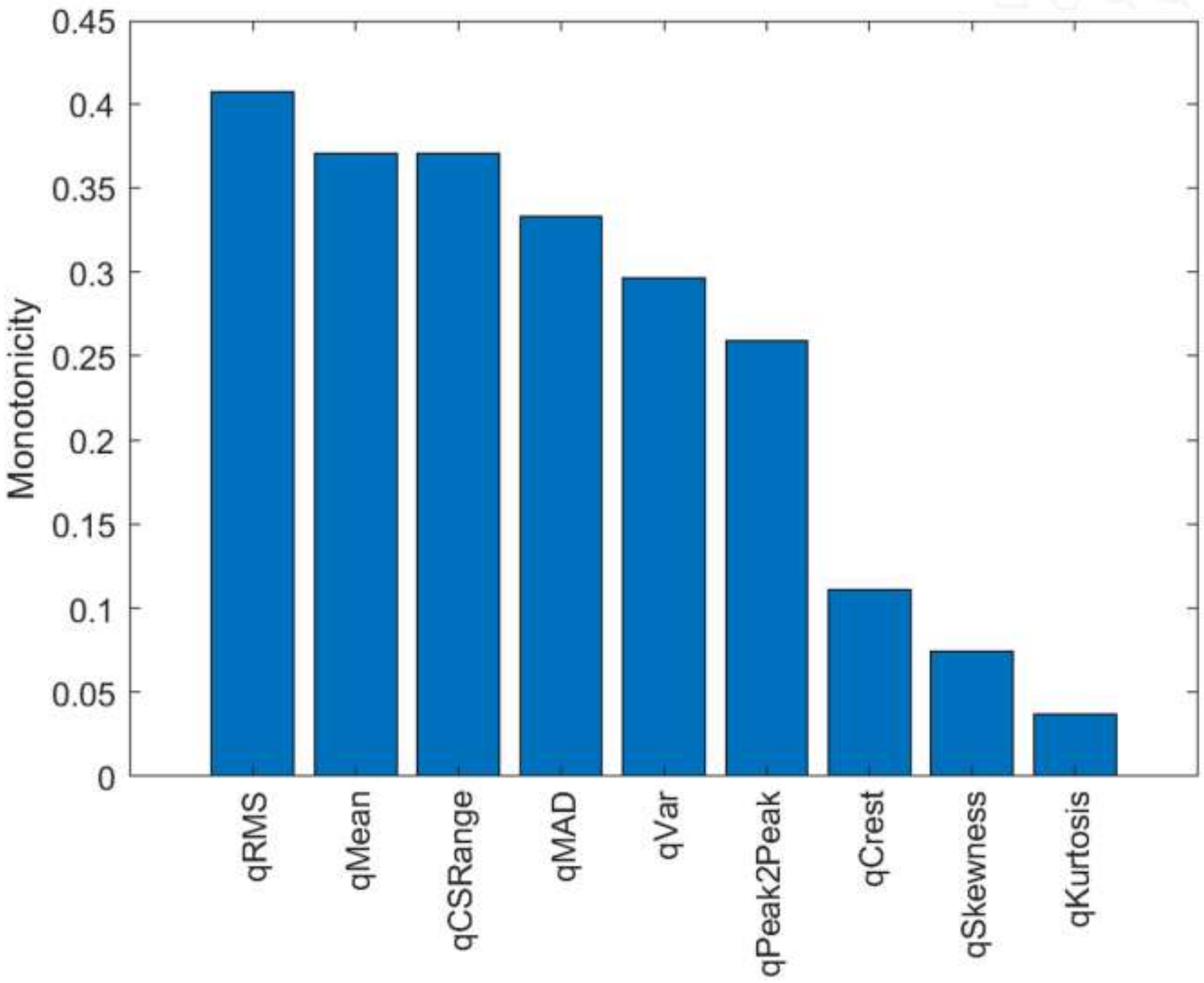
expand all

- **Signal-Based Features**
- **Model-Based Features and Residuals**
- **Feature Selection**

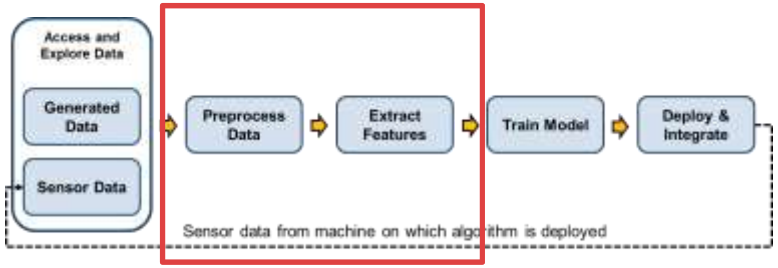

MATLAB R2018a
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
CleanUp
Search Documentation
Log In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Text
Normal
B I U M
TEXT
Code
CODE
Section Break
Run Section
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ abaru ▸ Desktop ▸ Expo 2018 ▸ FinalDemo ▸ Demo_Files ▸ Data_Reduction ▸
Current Folder
Name
Data
pdmRecipPump_log_1.mat
pdmRecipPump_log_2.mat
pdmRecipPump_log_3.mat
pdmRecipPump_log_4.mat
pdmRecipPump_log_5.mat
pdmRecipPump_log_6.mat
pdmRecipPump_log_7.mat
pdmRecipPump_log_8.mat
pdmRecipPump_log_9.mat
pdmRecipPump_log_10.mat
pdmRecipPump_log_11.mat
pdmRecipPump_log_12.mat
pdmRecipPump_log_13.mat
pdmRecipPump_log_14.mat
pdmRecipPump_log_15.mat
pdmRecipPump_log_16.mat
pdmRecipPump_log_17.mat
pdmRecipPump_log_18.mat
pdmRecipPump_log_19.mat
pdmRecipPump_log_20.mat
pdmRecipPump_log_21.mat
pdmRecipPump_log_22.mat
pdmRecipPump_log_23.mat
pdmRecipPump_log_13.mat (MAT-file)
Live Editor - C:\Users\abaru\Desktop\Expo 2018\FinalDemo\Demo_Files\Data_Reduction\Expo_Data_Preprocessing_CodeGen.mlx
Expo_Data_Preprocessing_CodeGen.mlx
featureExtractionBuffer.m
Algorithm Development for Feature Extraction at the Edge
Processing and Extracting Features from the Simulation Results
The model is configured to log the pump output pressure, output flow, motor speed and motor current.
ens = simulationEnsembleDatastore('.\Data');
ens.SelectedVariables = ["qOut_meas", "SimulationInput"];
reset(ens)
data = read(ens);
[flow,time_unit] = preprocess(data);
figure;
plot(flow.Time,flow.Data);
% Decide which features to extract
ens.DataVariables = [ens.DataVariables; ...
"qMean"; "qVar"; "qSkewness"; "qKurtosis"; ...
"qPeak2Peak"; "qCrest"; "qRMS"; "qMAD"; "qCSRange"];
ens.ConditionVariables = ["Time_Unit"];
feat = extractCI(flow);
dataToWrite = [time_unit, feat];
writeToLastMemberRead(ens,dataToWrite{:})

Feature Selection

```
function m = monotonicity(x)
% Compute monotonicity given a vector x
n = length(x);
dx = diff(x);
m = abs(sum(dx>0) - sum(dx<0))/(n-1);
end
```



Learn Further: Techniques for Preprocessing Data & Extracting Features



Time Domain

- Data smoothing, outlier removal, resampling
- Signal statistics (e.g. mean, moving average, etc.)
- Rain flow counting
- Time series models (linear & nonlinear)
- Non-linear time series features
- Recursive and batch based models
- Kalman filters (linear, unscented, & extended)

Frequency Domain

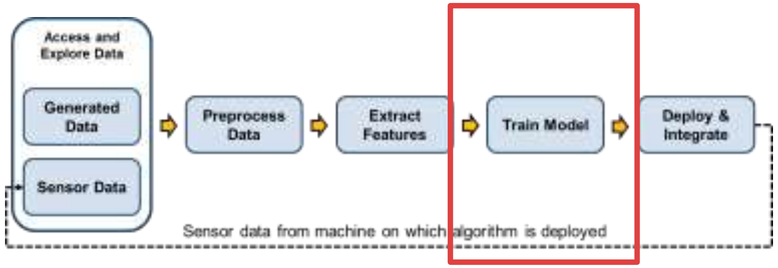
- Filtering
- Time synchronous averaging
- Spectral analysis and statistics (e.g. FFT, peak-to-peak values, bandwidth, etc.)
- Modal analysis using models/frequency data
- Envelope analysis
- Order analysis

Time – Frequency Domain

- Wavelet transforms
- Short-time Fourier transform
- Empirical mode decomposition, Hilbert-Huang transform
- Spectral Kurtosis
- Spectral Entropy
- Time-frequency moments

*Focus and functionality in the Predictive Maintenance Toolbox

Integrate Domain Knowledge with Machine Learning



Domain Knowledge → **Machine Learning**

Type of Learning

Categories of Algorithms

Supervised Learning

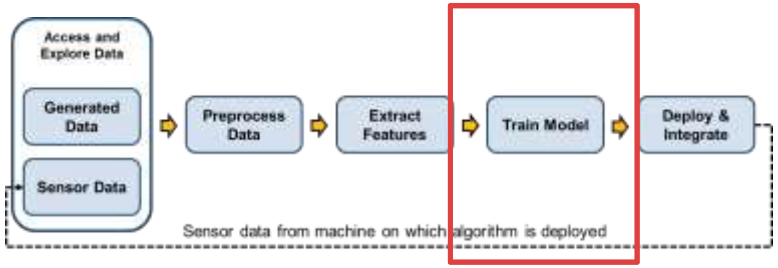
Develop **predictive model** based on both **input and output** data

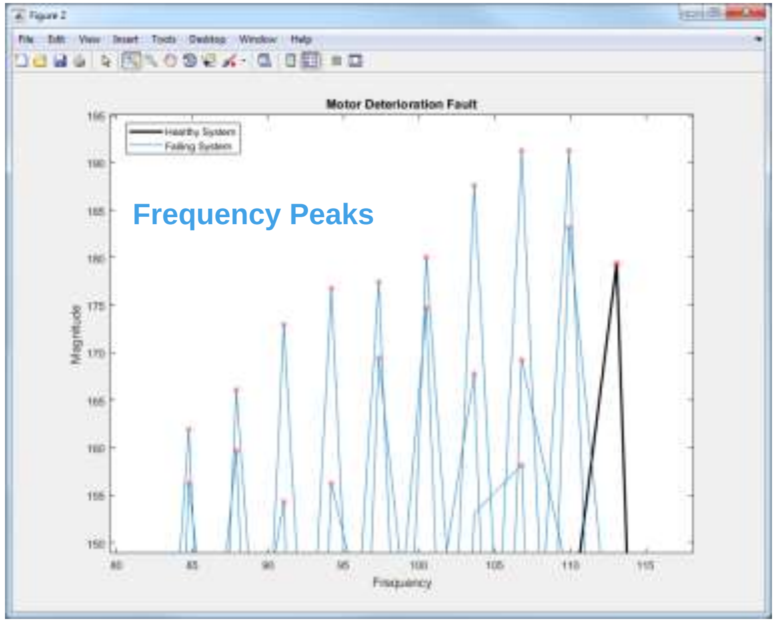
- **Regression**
- **Classification**

Unsupervised Learning

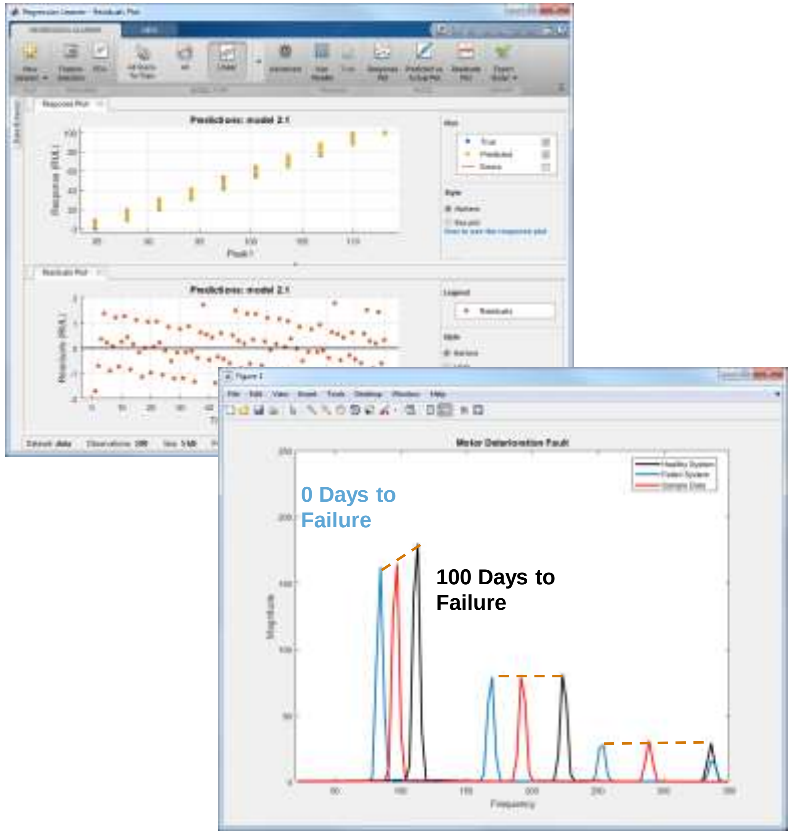
Group and **interpret** data based only on input data

Train Models to Predict Failures & Isolate Faults



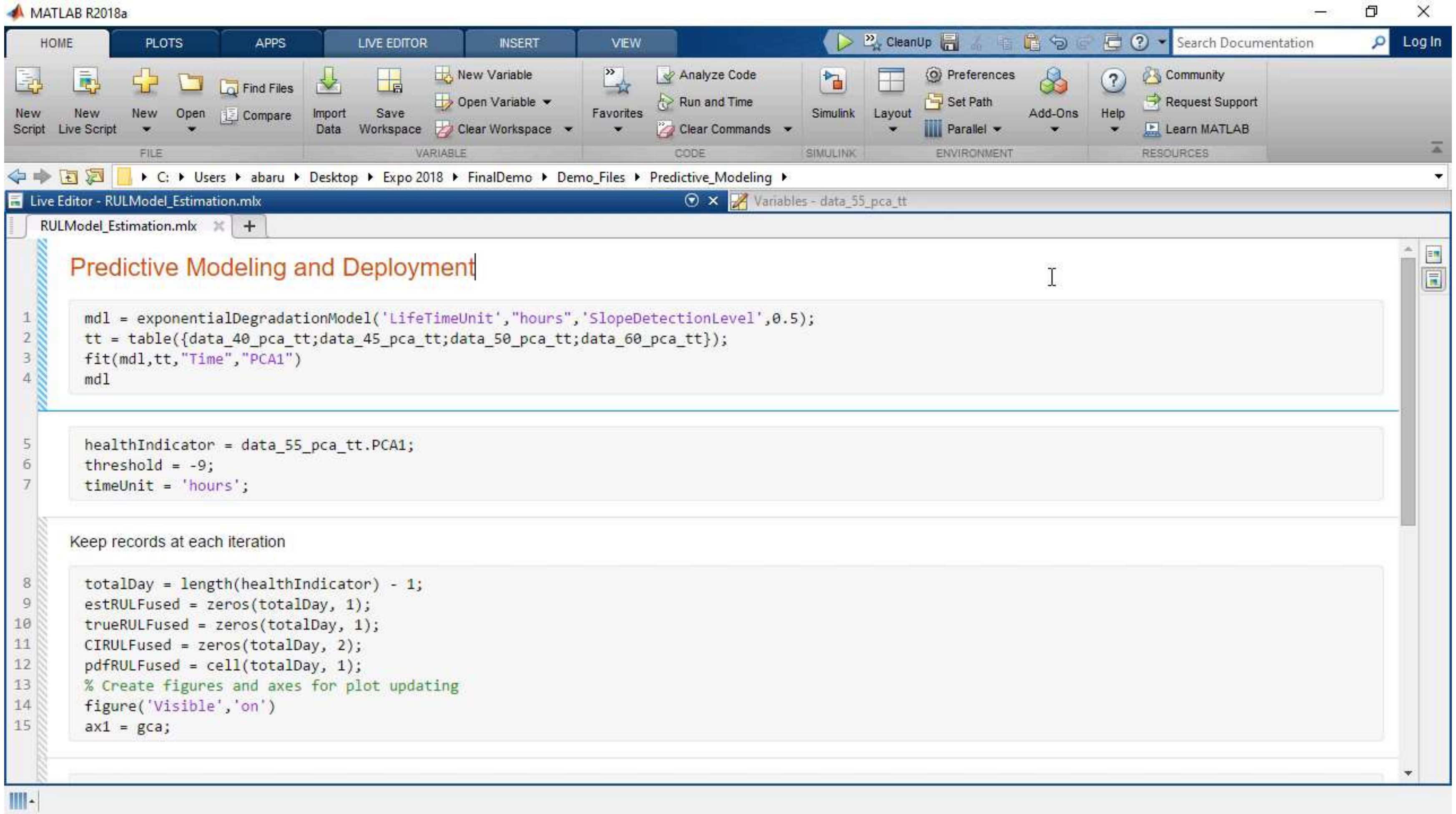


Health Indicators



Regression Model

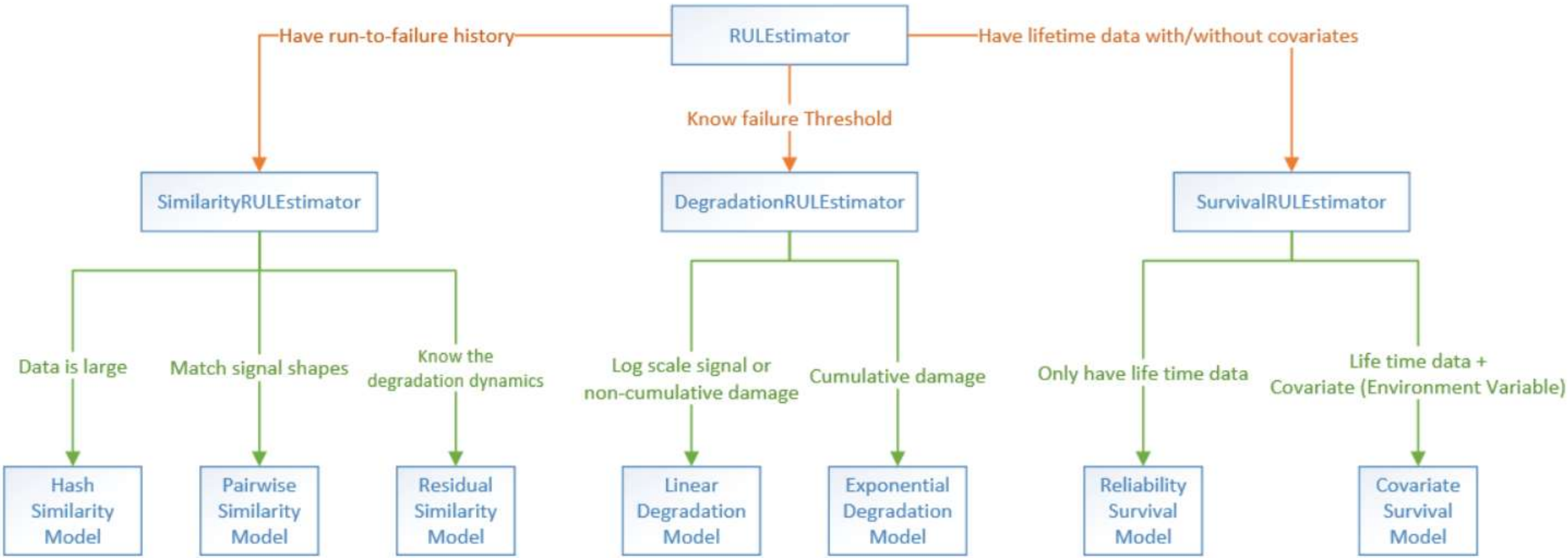
Demo video


MATLAB R2018a
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
CleanUp
Search Documentation
Log In
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Parallel
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ abaru ▸ Desktop ▸ Expo 2018 ▸ FinalDemo ▸ Demo_Files ▸ Predictive_Modeling ▸
Live Editor - RULModel_Estimation.mlx
Variables - data_55_pca_tt
RULModel_Estimation.mlx
Predictive Modeling and Deployment
mdl = exponentialDegradationModel('LifeTimeUnit',"hours",'SlopeDetectionLevel',0.5);
tt = table({data_40_pca_tt;data_45_pca_tt;data_50_pca_tt;data_60_pca_tt});
fit(mdl,tt,"Time","PCA1")
mdl
healthIndicator = data_55_pca_tt.PCA1;
threshold = -9;
timeUnit = 'hours';
Keep records at each iteration
totalDay = length(healthIndicator) - 1;
estRULFused = zeros(totalDay, 1);
trueRULFused = zeros(totalDay, 1);
CIRULFused = zeros(totalDay, 2);
pdfRULFused = cell(totalDay, 1);
% Create figures and axes for plot updating
figure('Visible','on')
ax1 = gca;

RUL Methods and when to use them

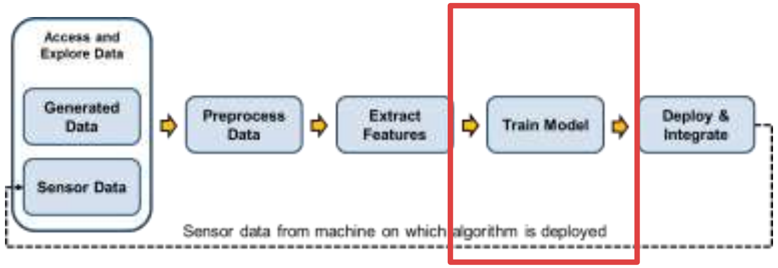
Requirement: Need to know what constitutes failure data

R2018a



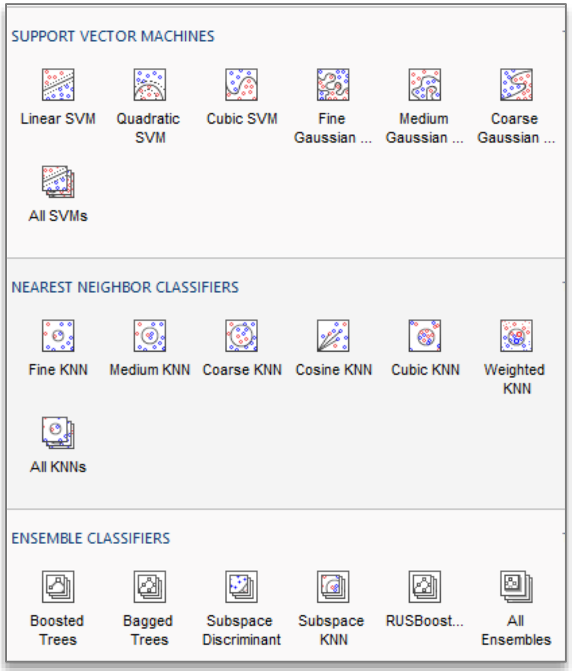
No equivalent RUL libraries in OSS

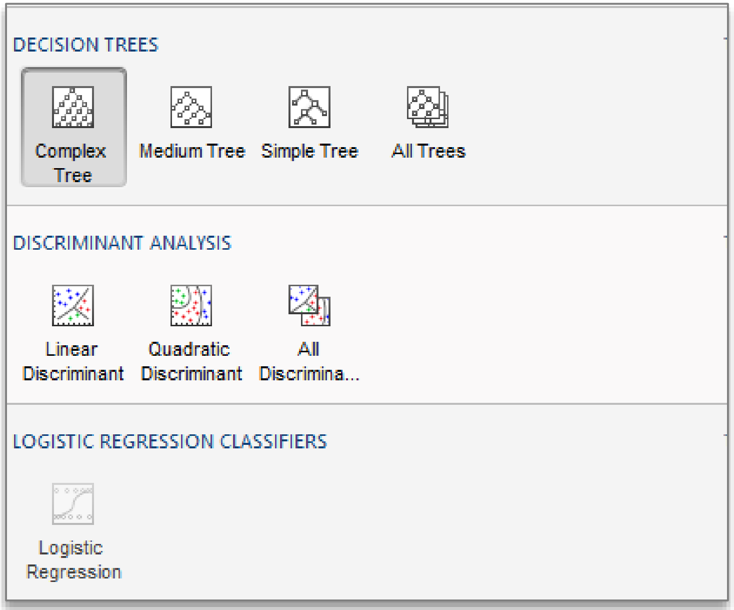
Learn Further: Techniques for Training Diagnostic & Prognostic Models



Diagnostic Models

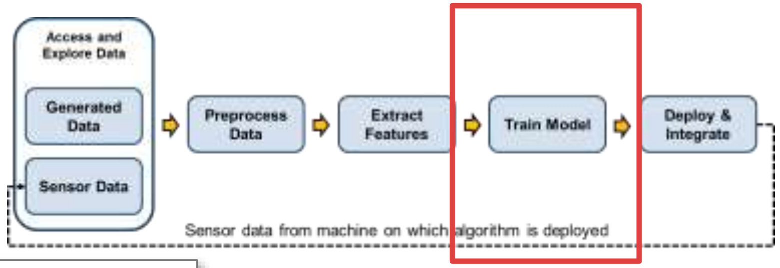
- Classification models
  - Support vector machines
  - Ensembles
  - Naïve Bayes, etc.
- Neural networks
- Change point detection
- Hypothesis testing
- Probability distributions





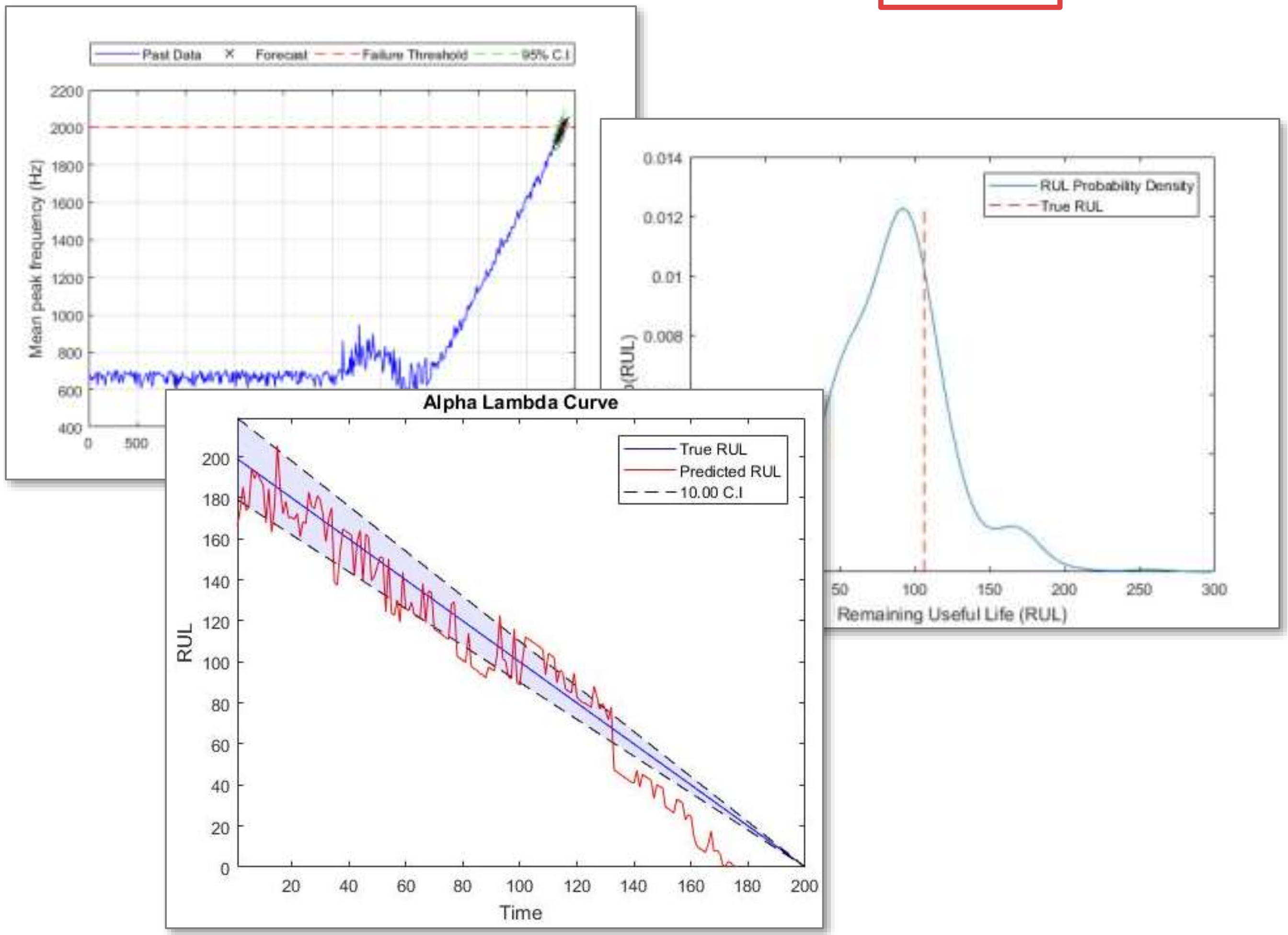
*Focus and functionality in the Predictive Maintenance Toolbox

Learn Further: Techniques for Training Diagnostic & Prognostic Models



Prognostic Models for RUL

- Static regression models (e.g. linear, logistic, nonlinear, etc.)
- Dynamic regression models (e.g. ARMAX, ARMA, etc.)
- Linear and nonlinear time series models
- Kalman filter prediction
- Similarity based methods
- Hidden Markov Models

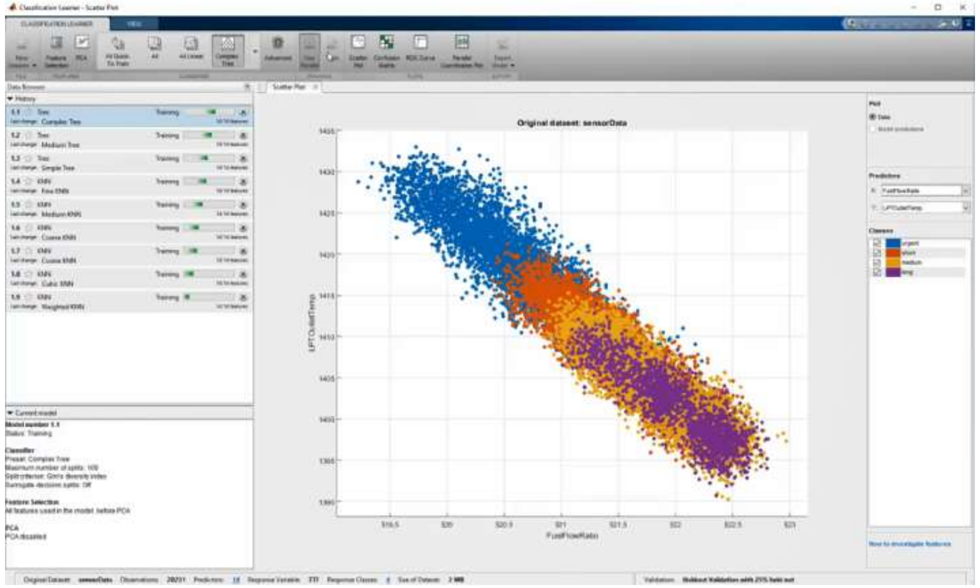


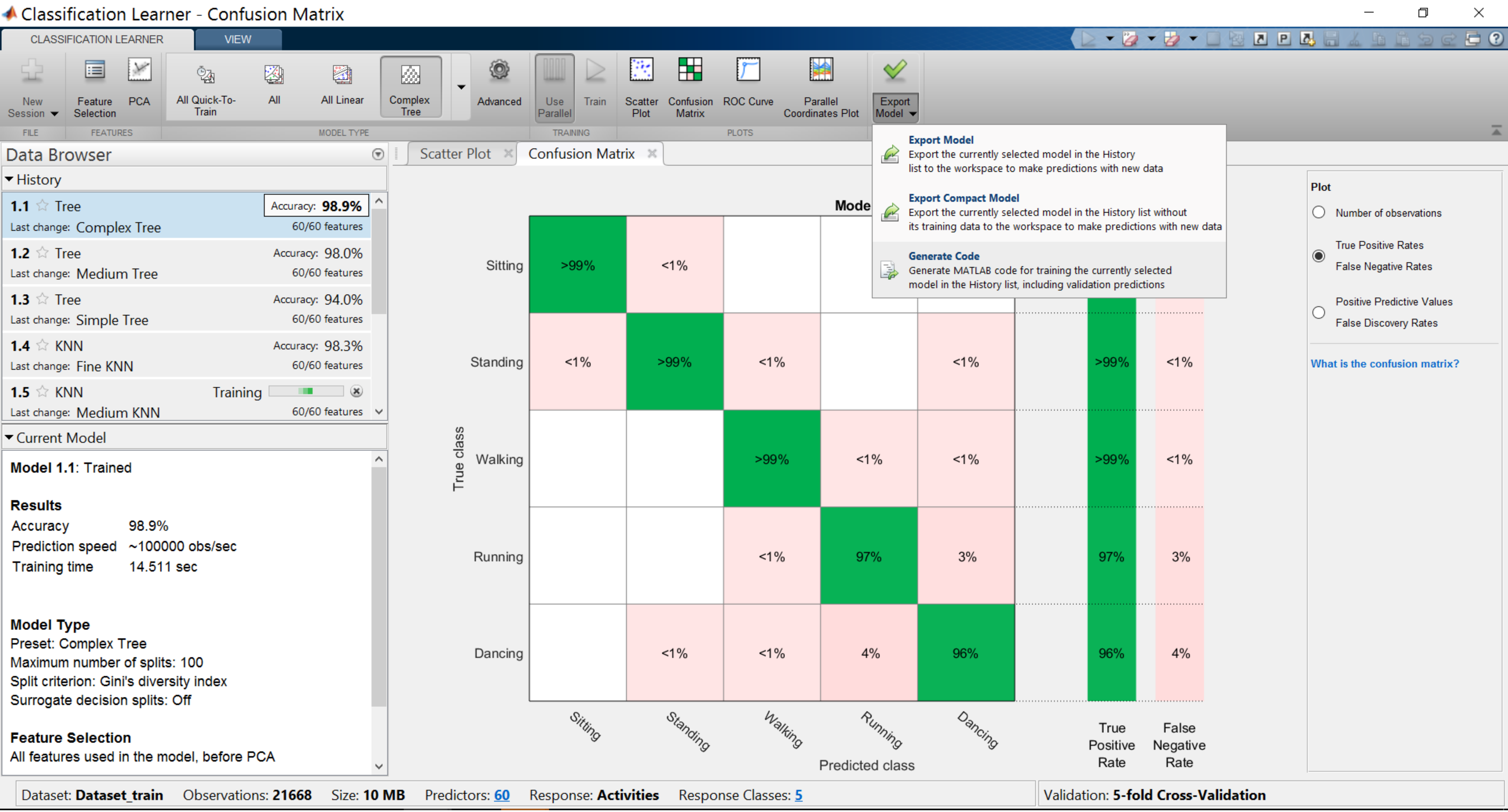
*Focus and functionality in the Predictive Maintenance Toolbox

Learn Further: Classification Learner App

App to apply advanced classification methods to your data

- Added to Statistics and Machine Learning Toolbox in R2015a
- Point and click interface – no coding required
- Quickly evaluate, compare and select classification models
- Export and share MATLAB code or trained models




Classification Learner - Confusion Matrix
CLASSIFICATION LEARNER
VIEW
New Session
Feature Selection
PCA
All Quick-To-Train
All
All Linear
Complex Tree
Advanced
Use Parallel
Train
Scatter Plot
Confusion Matrix
ROC Curve
Parallel Coordinates Plot
Export Model
FILE
FEATURES
MODEL TYPE
TRAINING
PLOTS
Export Model
Export the currently selected model in the History list to the workspace to make predictions with new data
Export Compact Model
Export the currently selected model in the History list without its training data to the workspace to make predictions with new data
Generate Code
Generate MATLAB code for training the currently selected model in the History list, including validation predictions
Data Browser
History
1.1 Tree Accuracy: 98.9%
Last change: Complex Tree 60/60 features
1.2 Tree Accuracy: 98.0%
Last change: Medium Tree 60/60 features
1.3 Tree Accuracy: 94.0%
Last change: Simple Tree 60/60 features
1.4 KNN Accuracy: 98.3%
Last change: Fine KNN 60/60 features
1.5 KNN Training
Last change: Medium KNN 60/60 features
Current Model
Model 1.1: Trained
Results
Accuracy 98.9%
Prediction speed ~100000 obs/sec
Training time 14.511 sec
Model Type
Preset: Complex Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off
Feature Selection
All features used in the model, before PCA
Scatter Plot
Confusion Matrix
Mode
Sitting
Standing
Walking
Running
Dancing
True class
Predicted class
>99%
<1%
97%
3%
4%
96%
True Positive Rate
False Negative Rate
Plot
Number of observations
True Positive Rates
False Negative Rates
Positive Predictive Values
False Discovery Rates
What is the confusion matrix?
Dataset: Dataset_train
Observations: 21668
Size: 10 MB
Predictors: 60
Response: Activities
Response Classes: 5
Validation: 5-fold Cross-Validation

and Many More MATLAB Apps for Data Analytics

Regression Learner

Distribution Fitting

System Identification

Signal Analysis

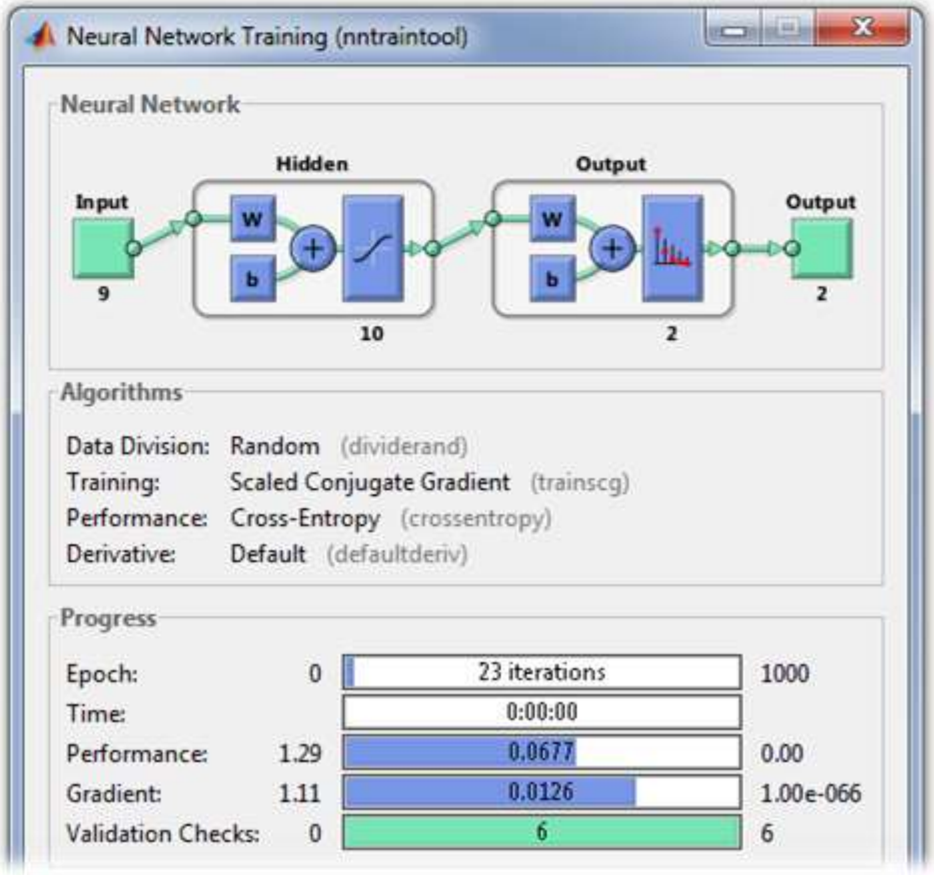
Wavelet Design and Analysis

Neural Net Fitting

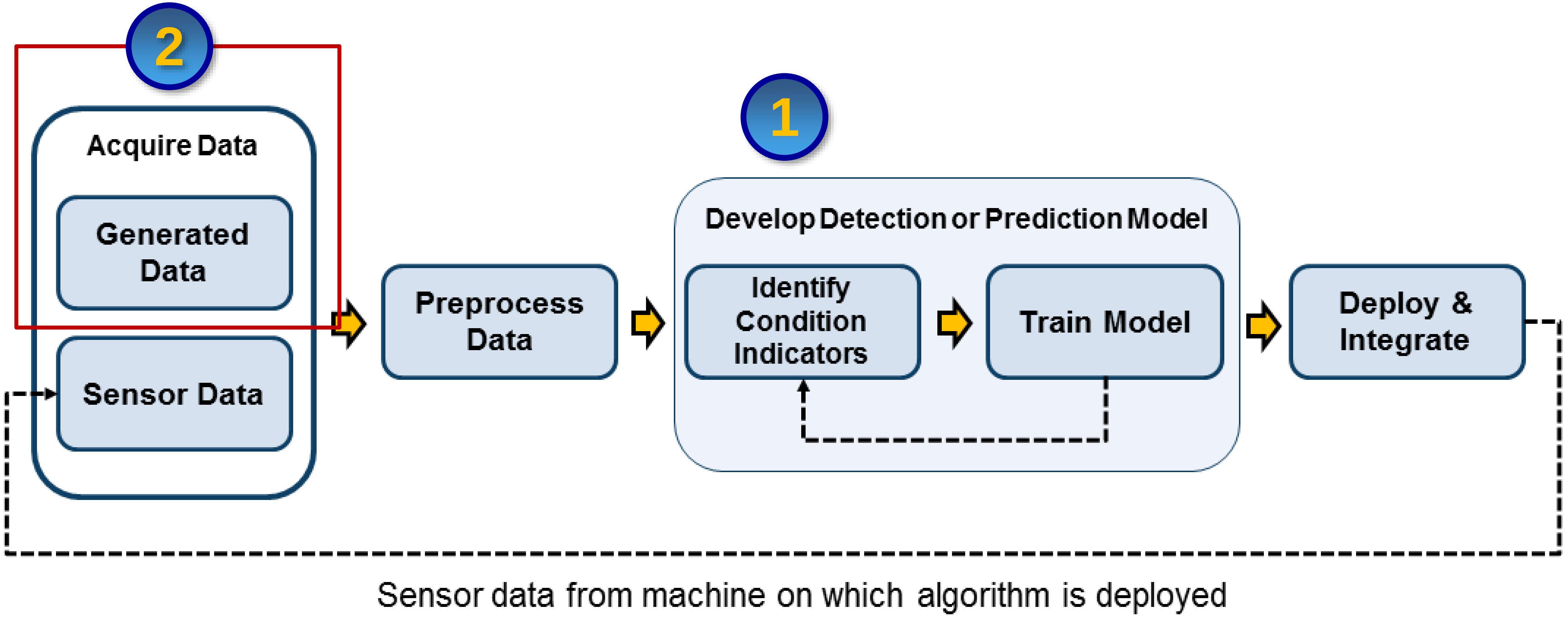
Neural Net Pattern Recognition

Training Image Labeler

and many more…



Workflow for developing a predictive maintenance algorithm



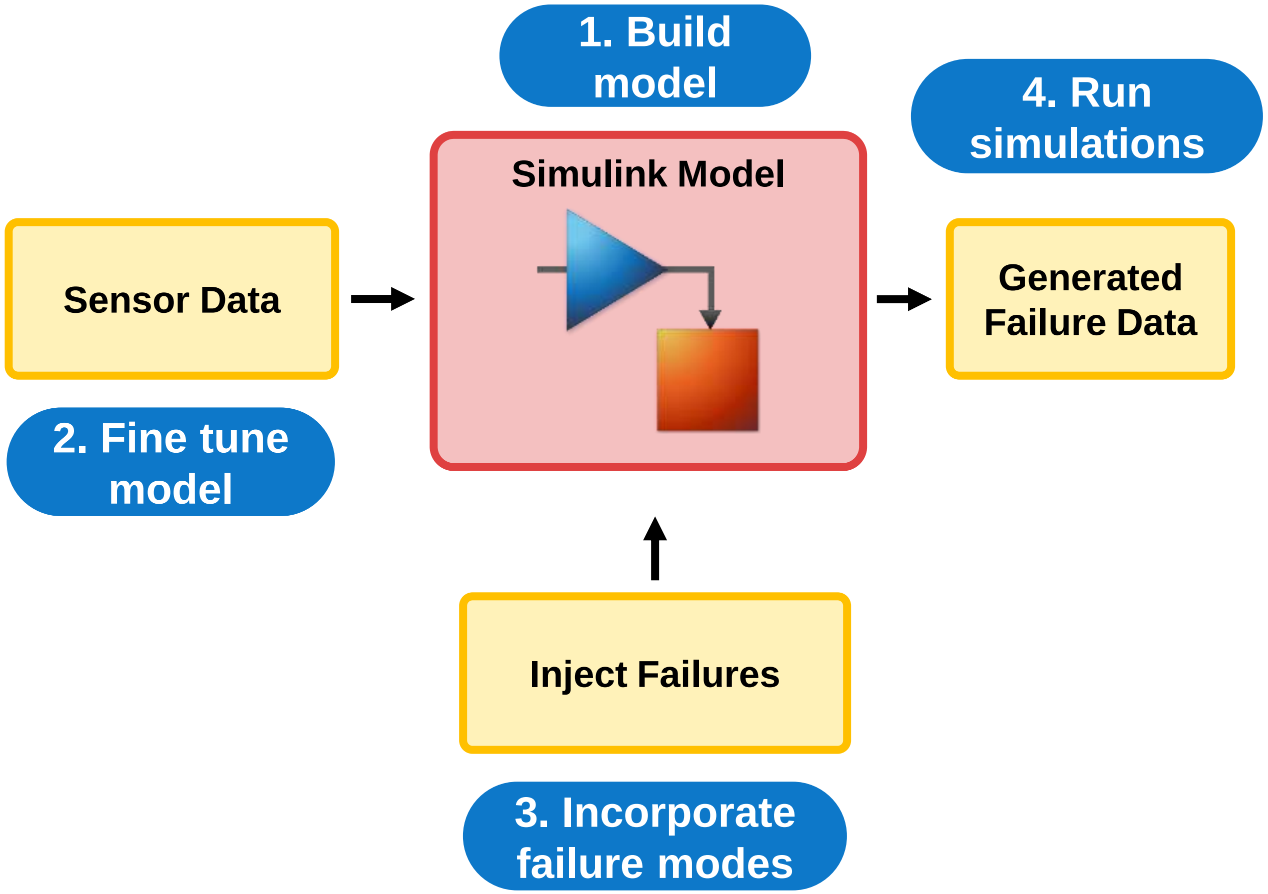
Why Generating failure data?

- Sensor data isn‘t always available
  - Failure conditions difficult to reproduce
  - Time consuming or costly to generate

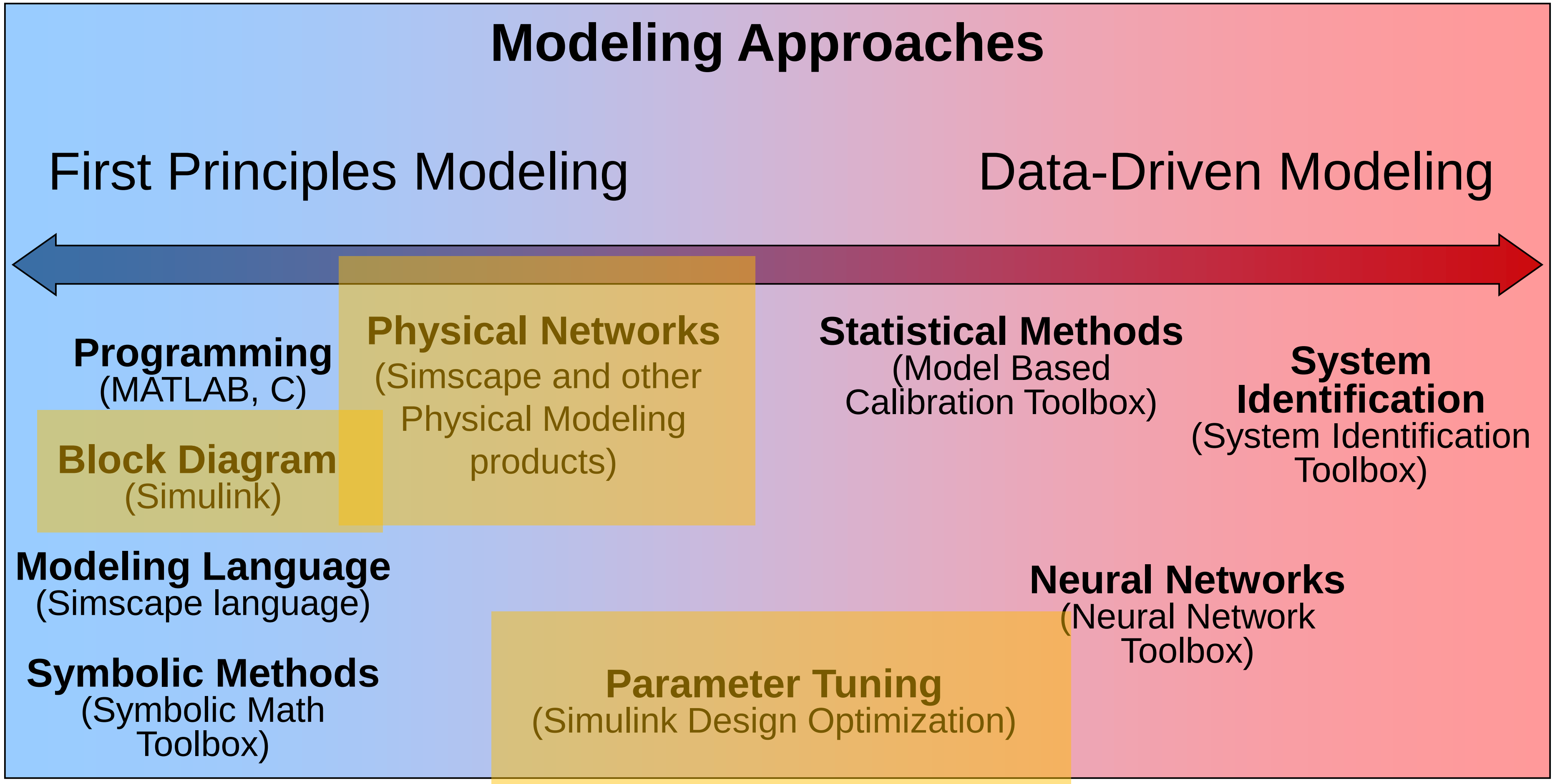
- Multiple failure modes and failure combinations possible

- Different machines can show different behavior for the same failure

Generating failure data from Simulink models



Different Approaches for Modeling Dynamic Systems

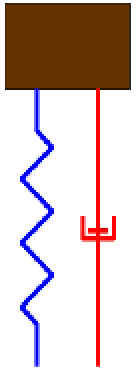


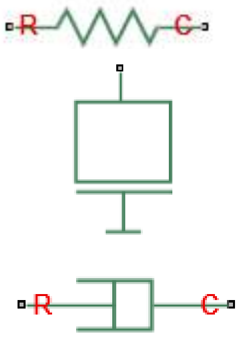
Simulink

Simscape

$$F = m\ddot{x} + b\dot{x} + kx$$

$$\ddot{x} = 1/m\,(F - b\dot{x} - kx)$$

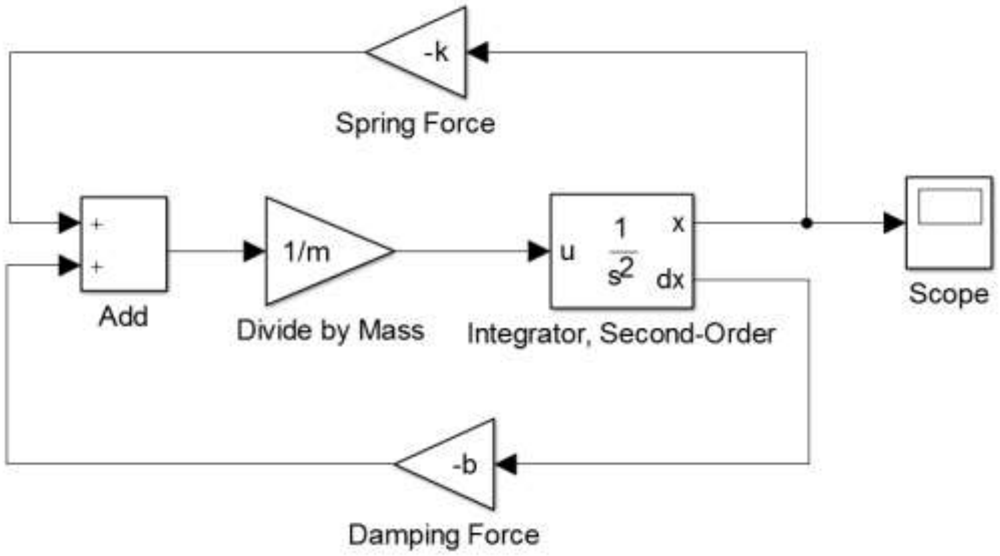


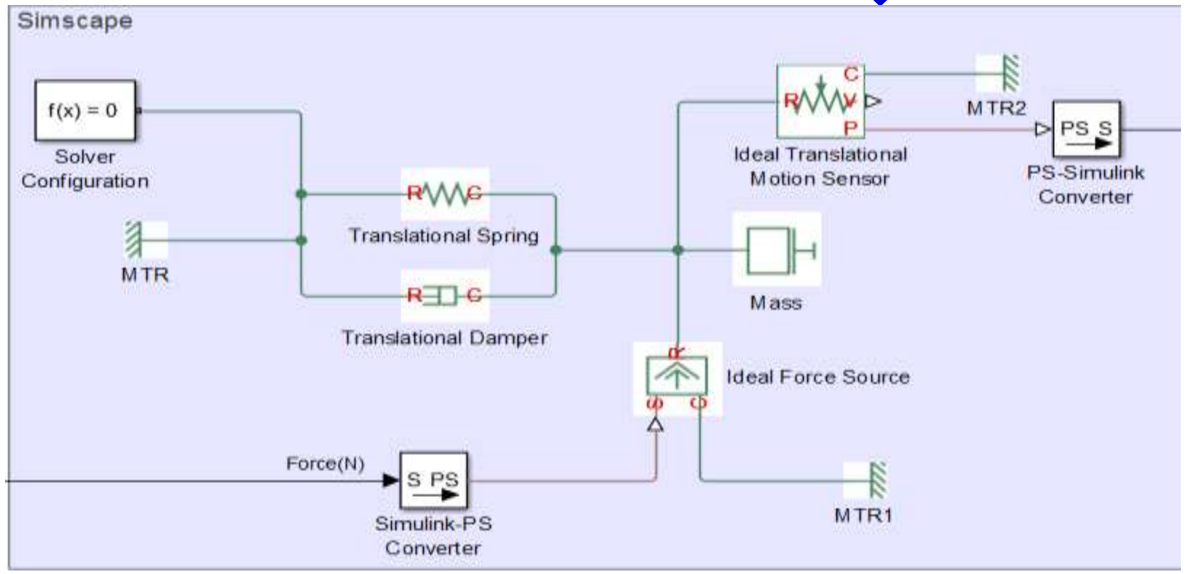


Spring

Mass

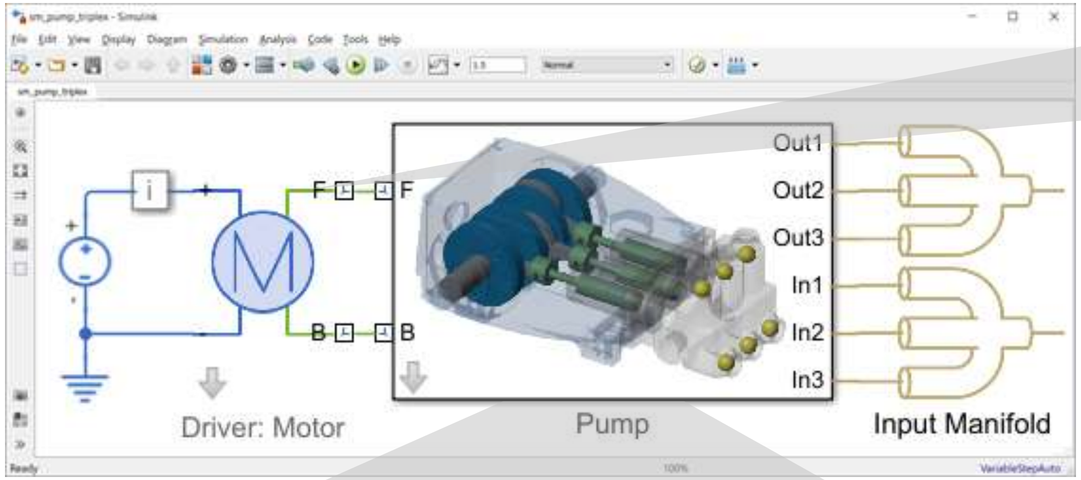
Damper

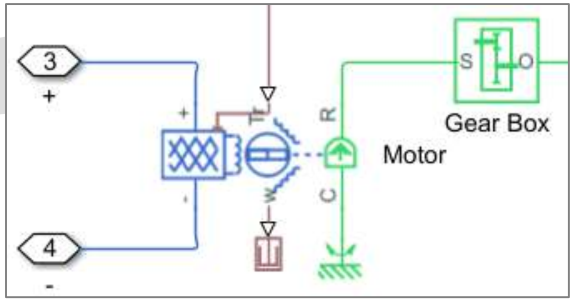


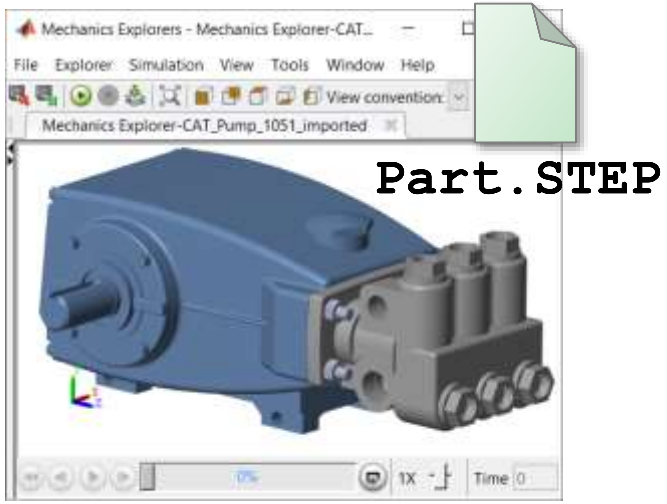


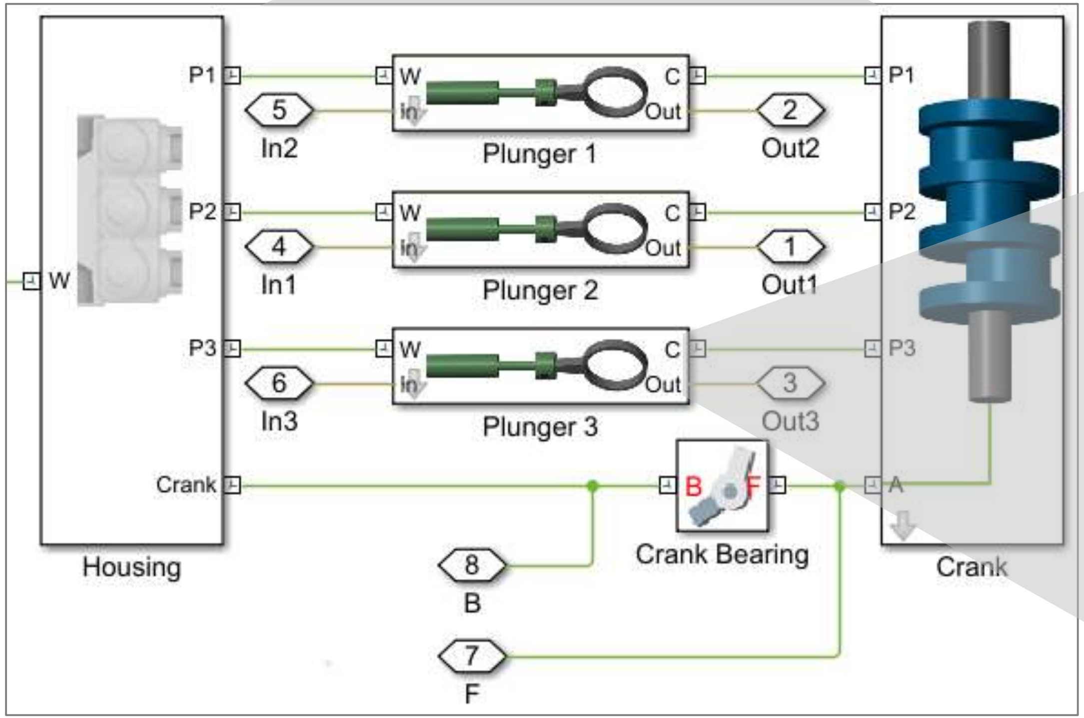
1. Build model

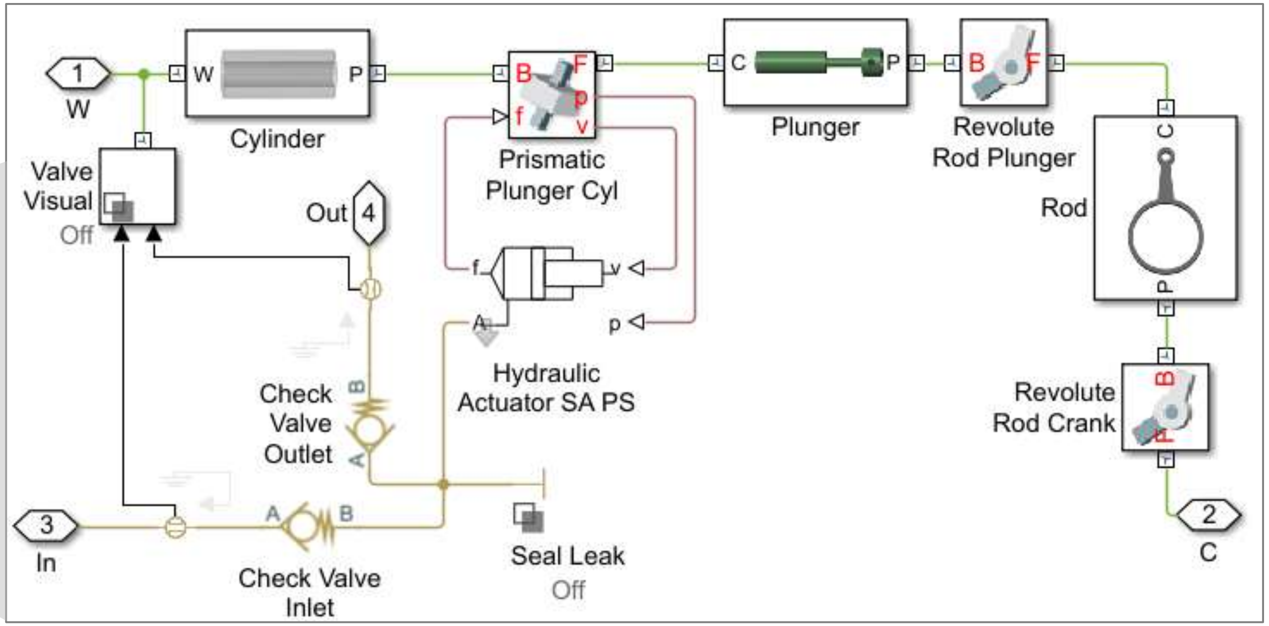
Simscape Model







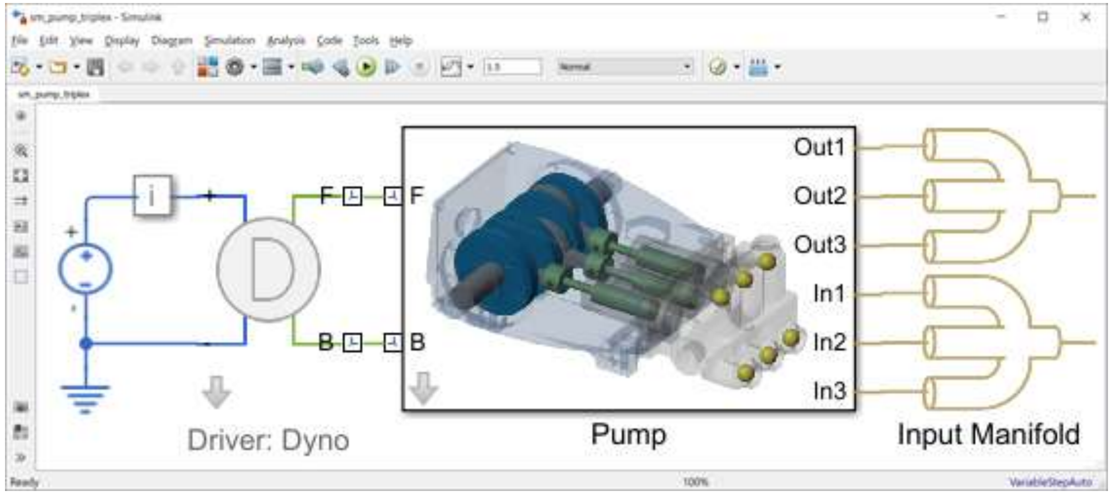




2. Fine tune model

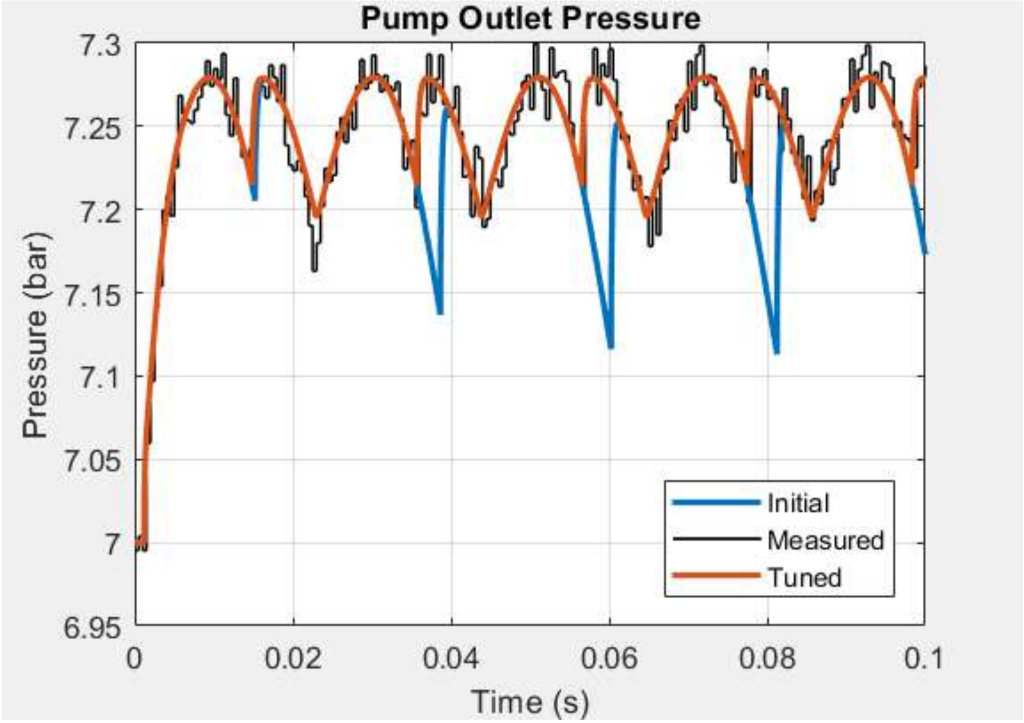
Estimate Parameters Using Measured Data

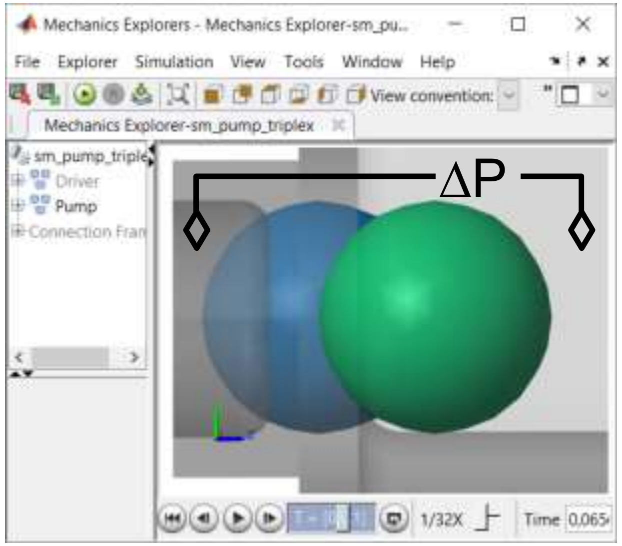
Model:

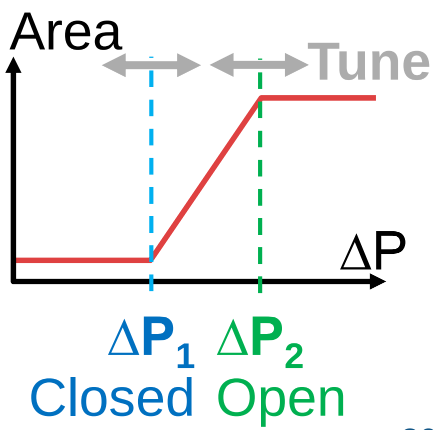


Challenge: Simulation results do not match behavior of real system

Solution: Use Simulink Design Optimization to automatically tune model parameters



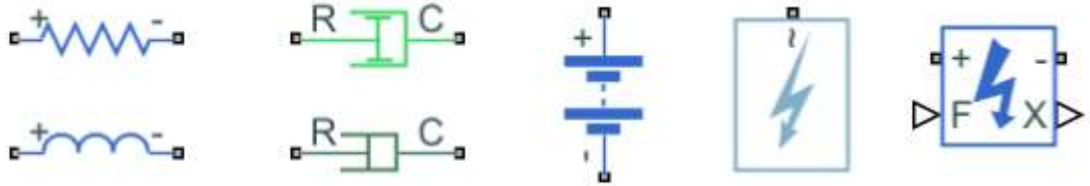


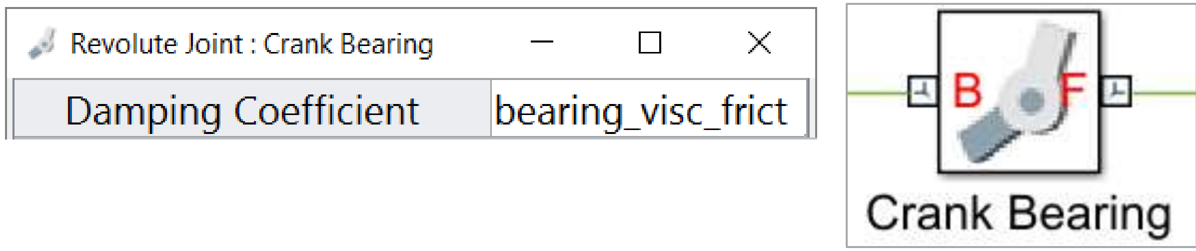


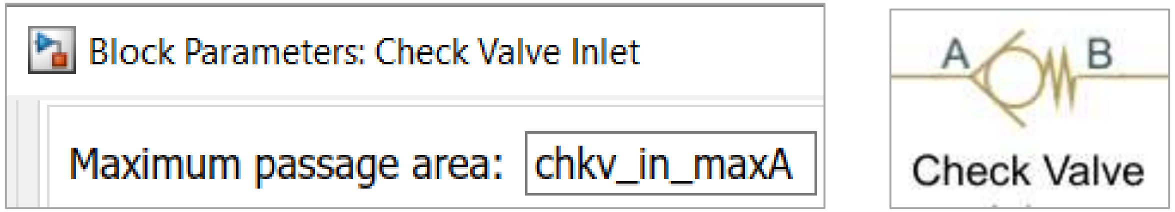
Model Component Failure

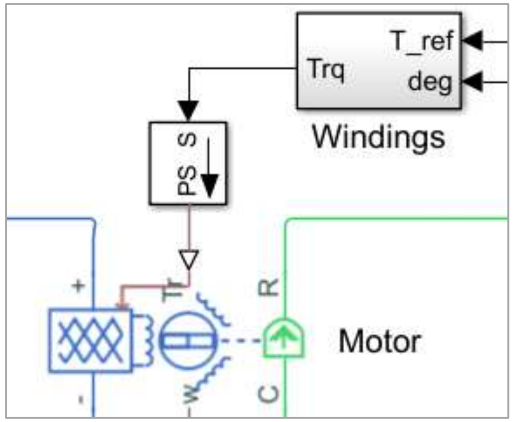
3. Incorporate failure modes

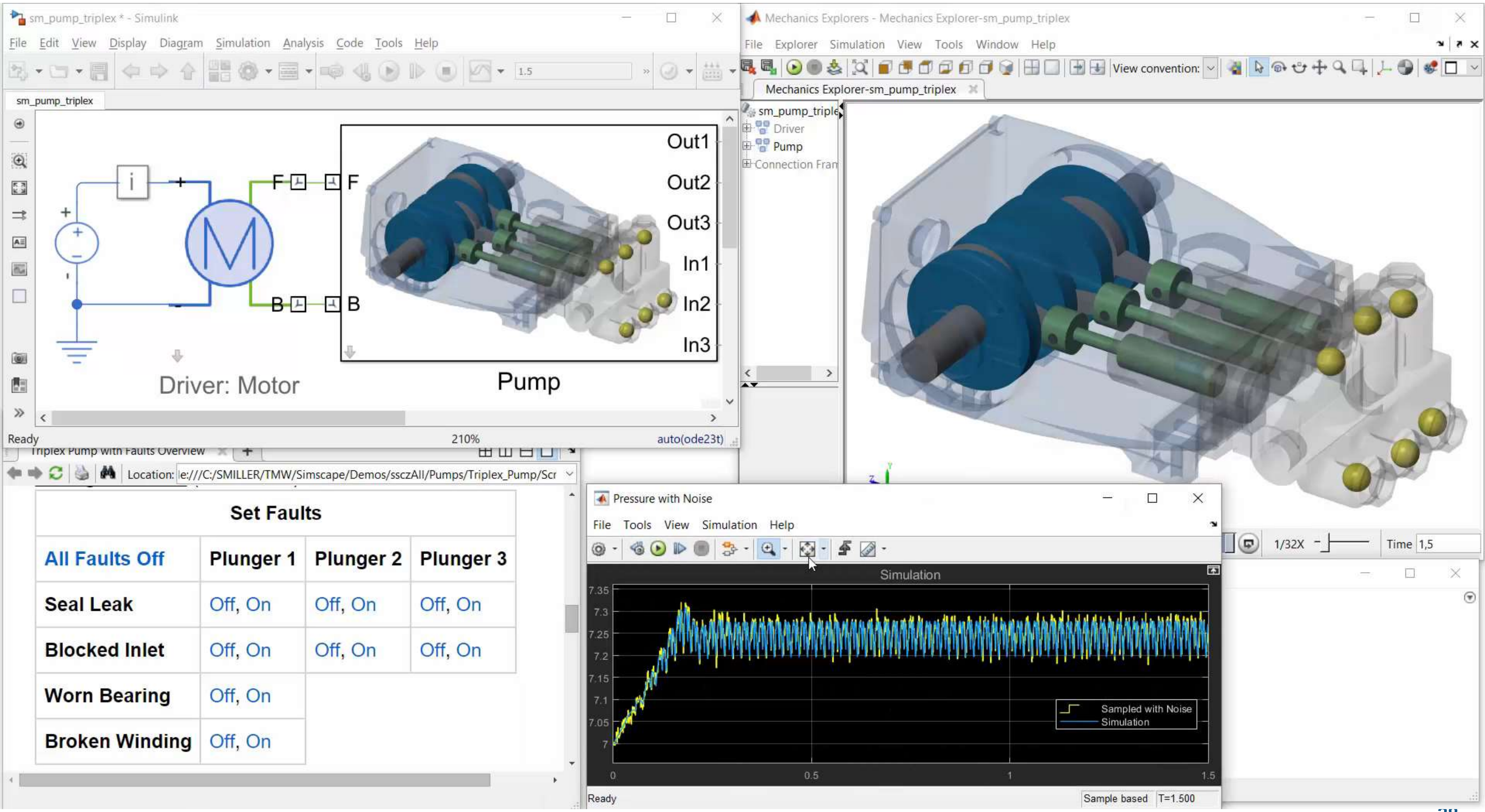
- Generic faults in many components
  - Short circuit, open circuit, friction, fade, etc.
  - Trigger based on time or conditions
- Adjust parameter values
  - Worn bearing adds friction
  - Blocked inlet has reduced passage area
- Adjust network
  - Seal leakage adds flow path
- Custom effects in Simulink
  - Broken winding applies no torque for 1/3 of every revolution










sm_pump_triplex * - Simulink
Driver: Motor
Pump
Out1
Out2
Out3
In1
In2
In3
Mechanics Explorers - Mechanics Explorer-sm_pump_triplex
Driver
Pump
Connection Frames
Triplex Pump with Faults Overview
Set Faults
All Faults Off
Plunger 1
Plunger 2
Plunger 3
Seal Leak
Blocked Inlet
Worn Bearing
Broken Winding
Off, On
Pressure with Noise
Simulation
Sampled with Noise
Simulation
Ready
Sample based
T=1.500

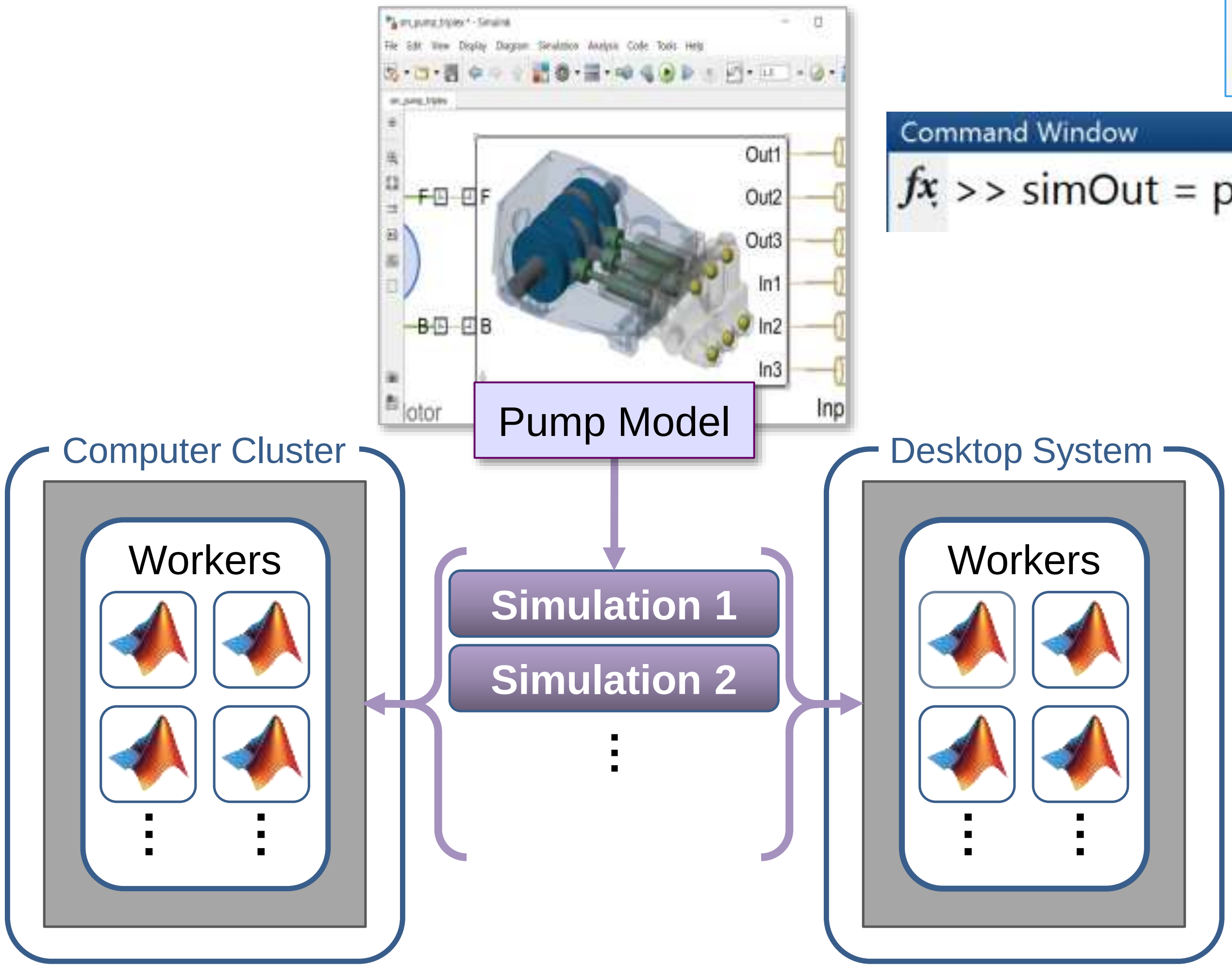
Generate Synthetic Data

```
for ct = 1:length(final_block)
    time_WKSP = ct;
    simInput(ct) = Simulink.SimulationInput(mdl);
    simInput(ct) = setVariable(simInput(ct),'leak_cyl_area_WKSP',final_leak(1));
    simInput(ct) = setVariable(simInput(ct),'block_in_factor_WKSP',final_block(ct));
    simInput(ct) = setVariable(simInput(ct),'bearing_fault_frict_WKSP',final_bearing(1));
    simInput(ct) = setVariable(simInput(ct),'noise_seed_offset_WKSP',ct-1);
    simInput(ct) = setVariable(simInput(ct),'time_WKSP',time_WKSP);
end
```

Running simulations in parallel speeds up your testing process.

Command Window

```
>> simOut = parsim(simInput)
```

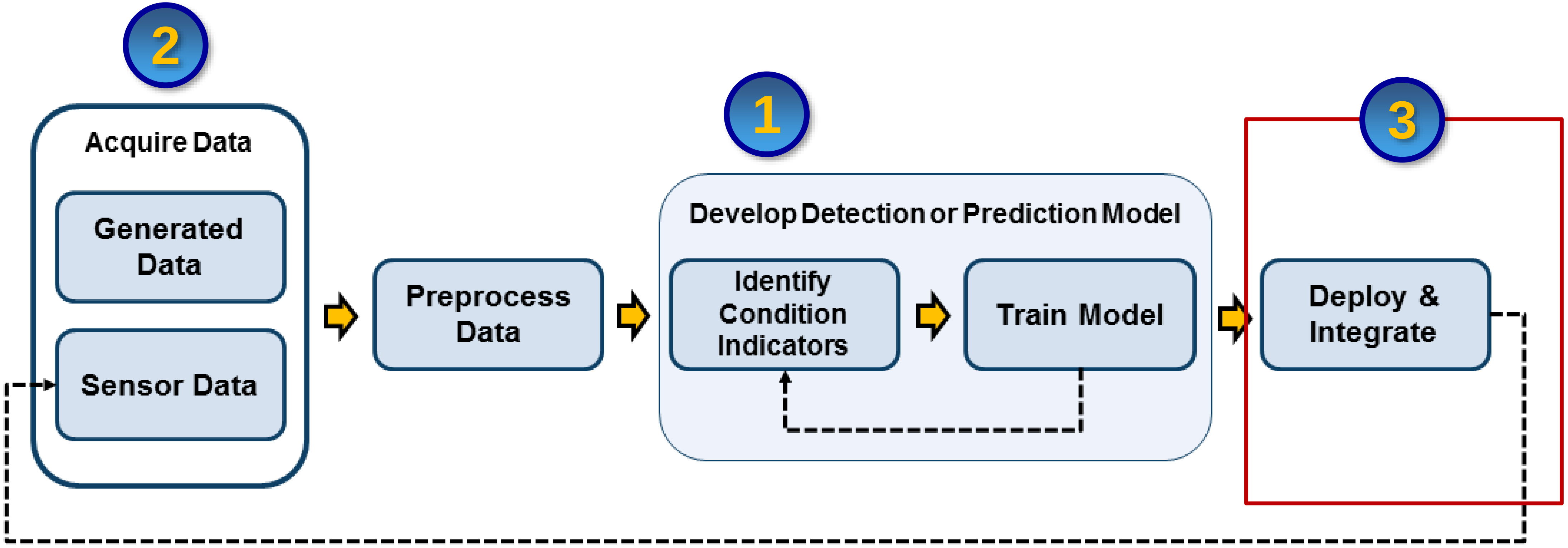


PumpData1.csv
PumpData2.csv
PumpData3.csv
PumpData4.csv
PumpData5.csv
PumpData6.csv
PumpD
PumpD
PumpD
PumpD
PumpData11.csv
PumpData12.csv
PumpData13.csv
PumpData14.csv
PumpData15.csv
PumpData16.csv
PumpData17.csv

simulationEnsembleDatastore

Manage ensemble data generated by generateSimulationEnsemble or by logging simulation data in Simulink

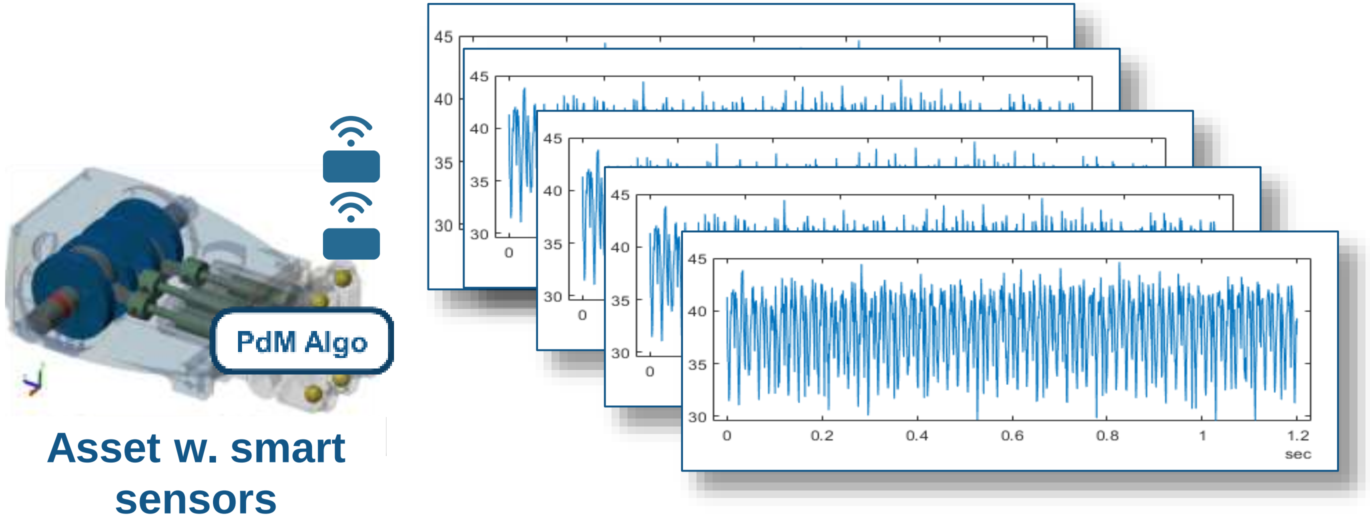
Workflow for developing a predictive maintenance algorithm



Sensor data from machine on which algorithm is deployed

Feature Extraction Algorithm at the Edge

Pump flow sensor *1 sec ~ 1000 samples ~16kB*

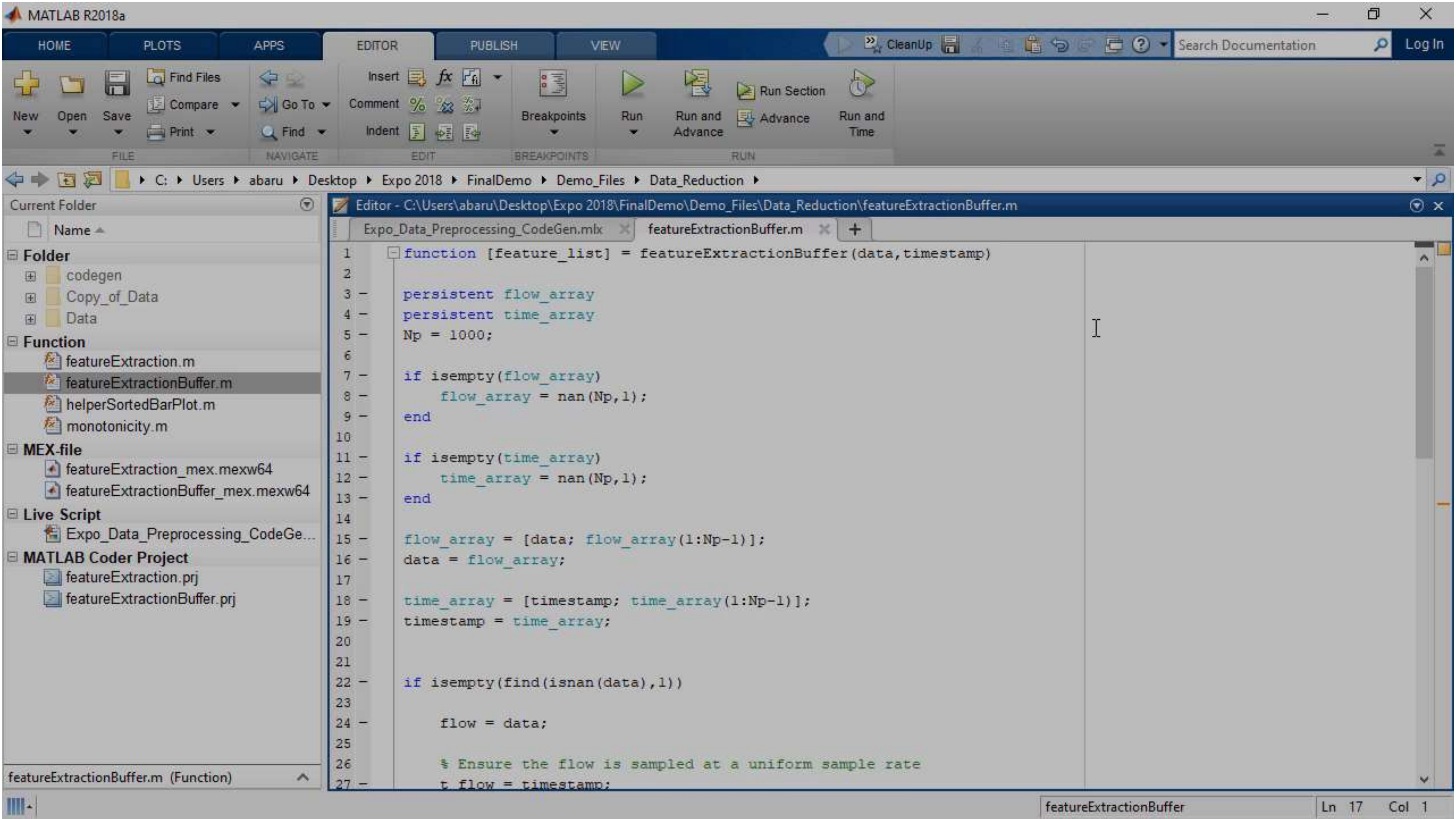


Asset w. smart sensors

- 1 day ~ 1.3 GB
- 20 sensors/pump ~26 GB/day
- 3 pumps ~ 78 GB/day

Challenge:
Data transmission cost is pretty high

Solution:
Extract only relevant information and send it to predictive model


MATLAB R2018a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
CleanUp
Search Documentation
Log In
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
C: ▸ Users ▸ abaru ▸ Desktop ▸ Expo 2018 ▸ FinalDemo ▸ Demo_Files ▸ Data_Reduction ▸
Current Folder
Name
Folder
codegen
Copy_of_Data
Data
Function
featureExtraction.m
featureExtractionBuffer.m
helperSortedBarPlot.m
monotonicity.m
MEX-file
featureExtraction_mex.mexw64
featureExtractionBuffer_mex.mexw64
Live Script
Expo_Data_Preprocessing_CodeGe...
MATLAB Coder Project
featureExtraction.prj
featureExtractionBuffer.prj
featureExtractionBuffer.m (Function)
Editor - C:\Users\abaru\Desktop\Expo 2018\FinalDemo\Demo_Files\Data_Reduction\featureExtractionBuffer.m
Expo_Data_Preprocessing_CodeGen.mlx
featureExtractionBuffer.m
function [feature_list] = featureExtractionBuffer(data,timestamp)
persistent flow_array
persistent time_array
Np = 1000;
if isempty(flow_array)
flow_array = nan(Np,1);
end
if isempty(time_array)
time_array = nan(Np,1);
end
flow_array = [data; flow_array(1:Np-1)];
data = flow_array;
time_array = [timestamp; time_array(1:Np-1)];
timestamp = time_array;
if isempty(find(isnan(data),1))
flow = data;
% Ensure the flow is sampled at a uniform sample rate
t_flow = timestamp;
featureExtractionBuffer
Ln 17
Col 1

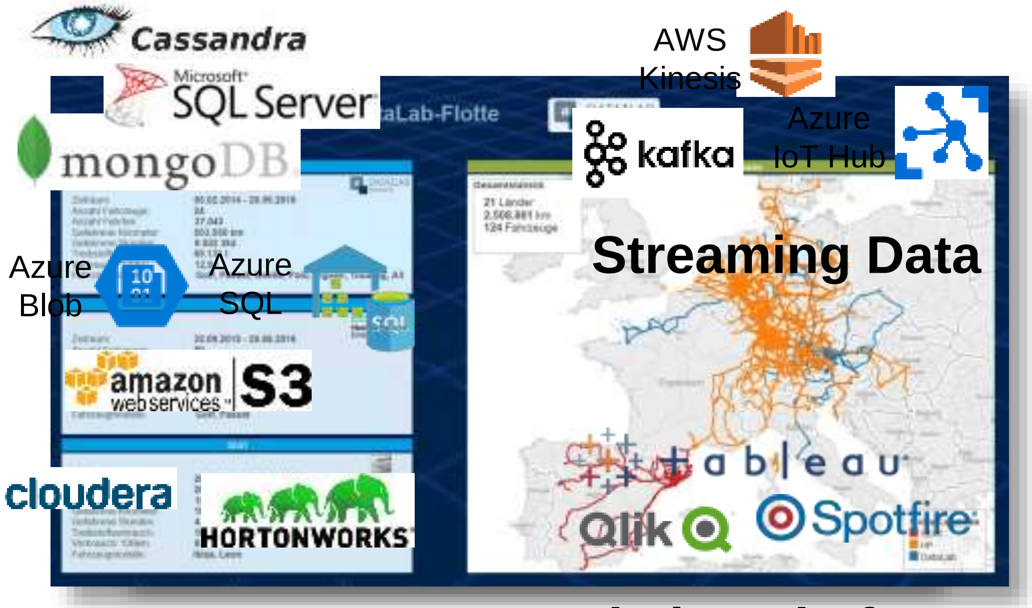
What do your end users want?

Flexible Deployment

- Maintenance needs simple, quick information
  - Hand held devices, Alarms

- Operations needs a birds-eye view
  - Integration with IT & OT systems

- Customers expect easy to digest information
  - Automated reports



Data Sources **Analytics Platforms**

Fleet & Inventory Analysis

Hand held Devices

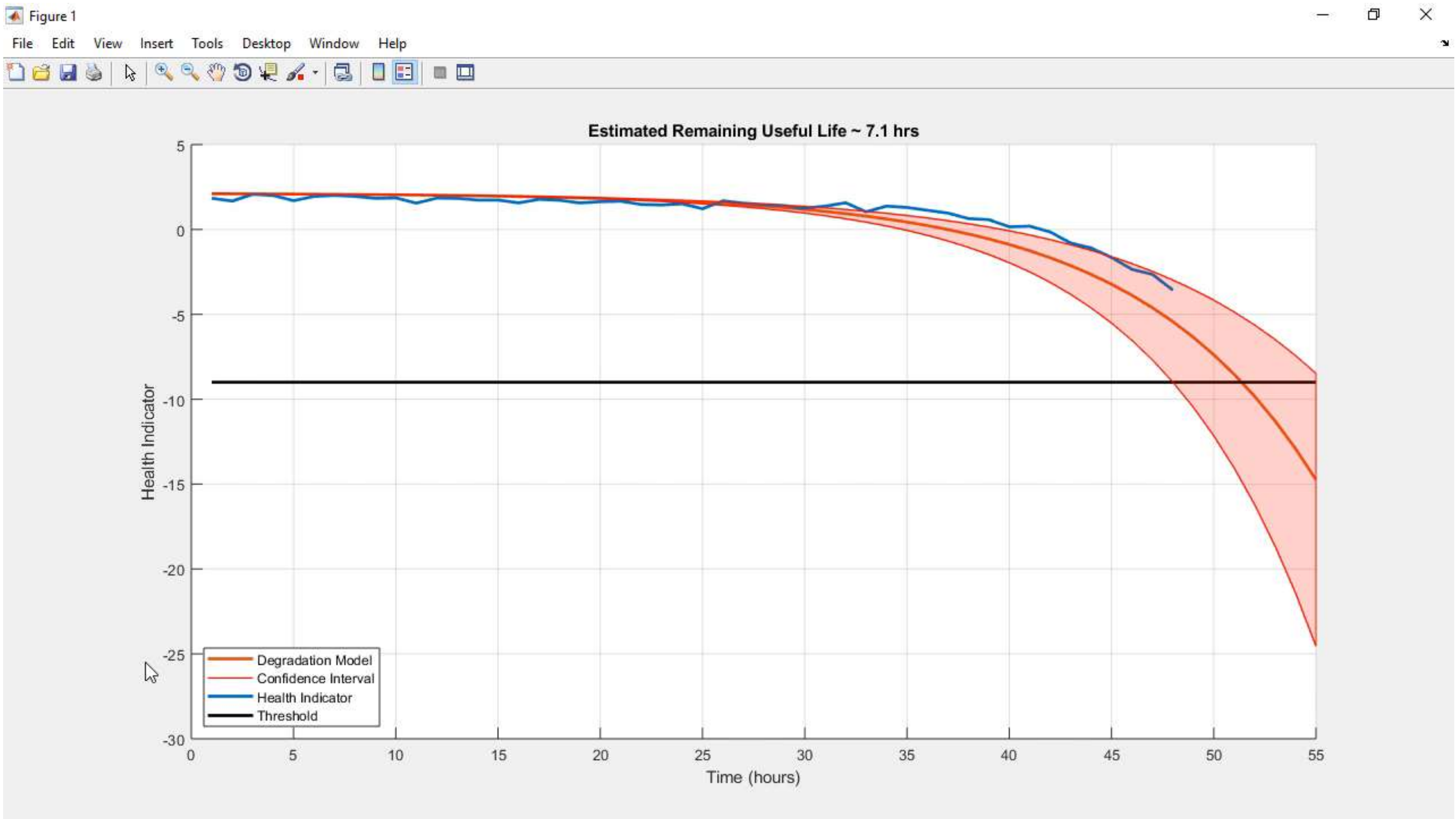
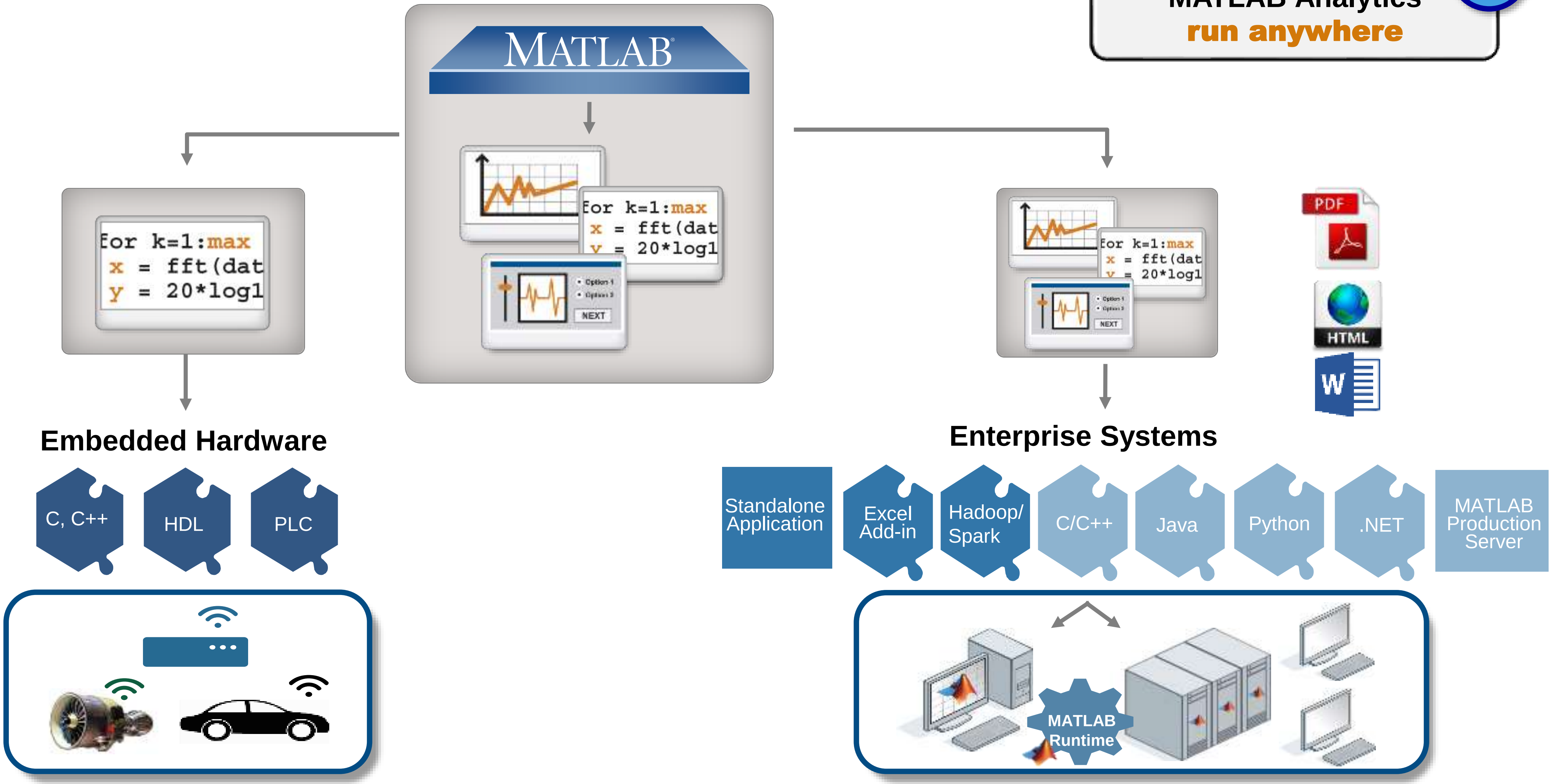


Figure 1
File Edit View Insert Tools Desktop Window Help
Estimated Remaining Useful Life ~ 7.1 hrs
Health Indicator
Time (hours)
5
0
-5
-10
-15
-20
-25
-30
0 5 10 15 20 25 30 35 40 45 50 55
Degradation Model
Confidence Interval
Health Indicator
Threshold

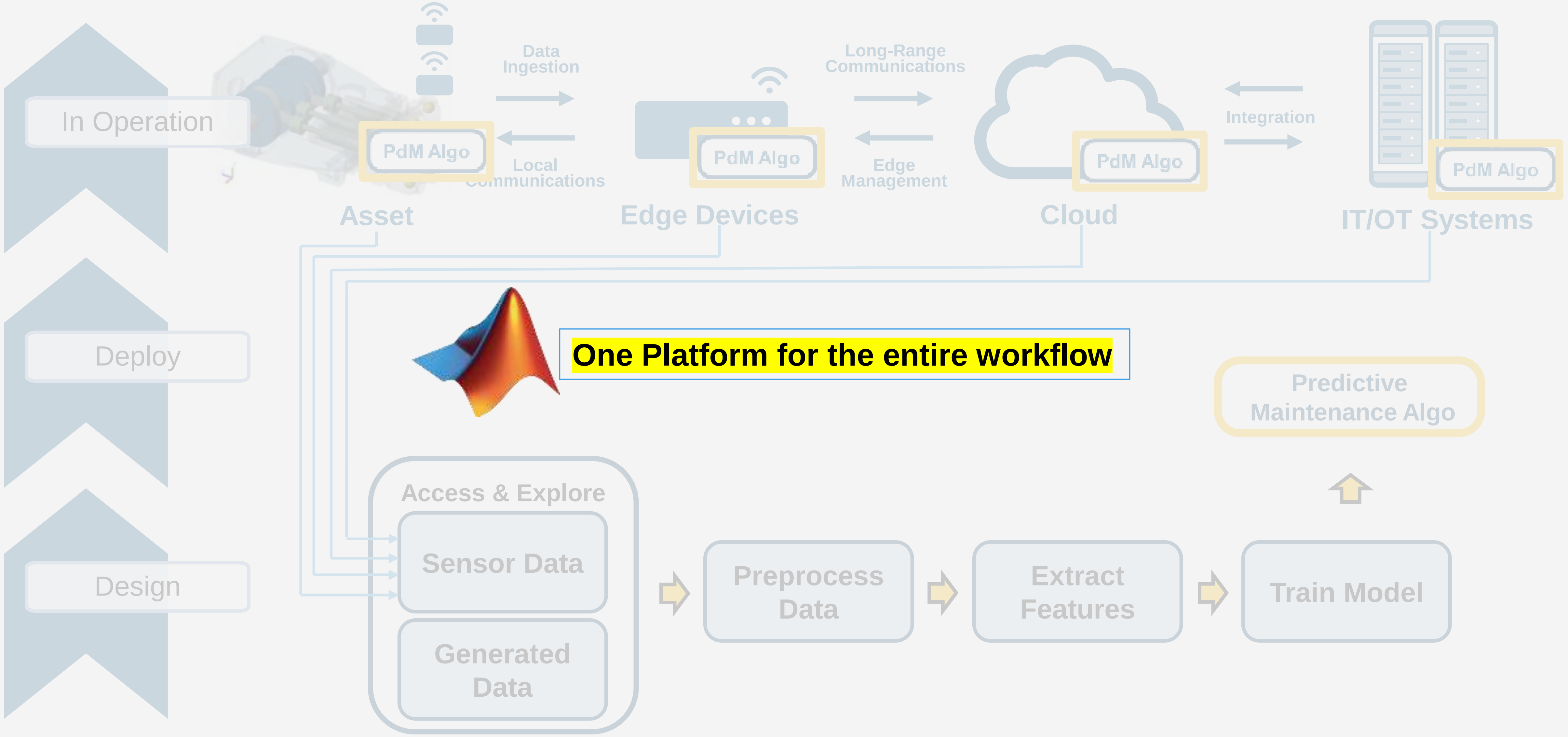
Integrate analytics with systems

3

MATLAB Analytics
run anywhere

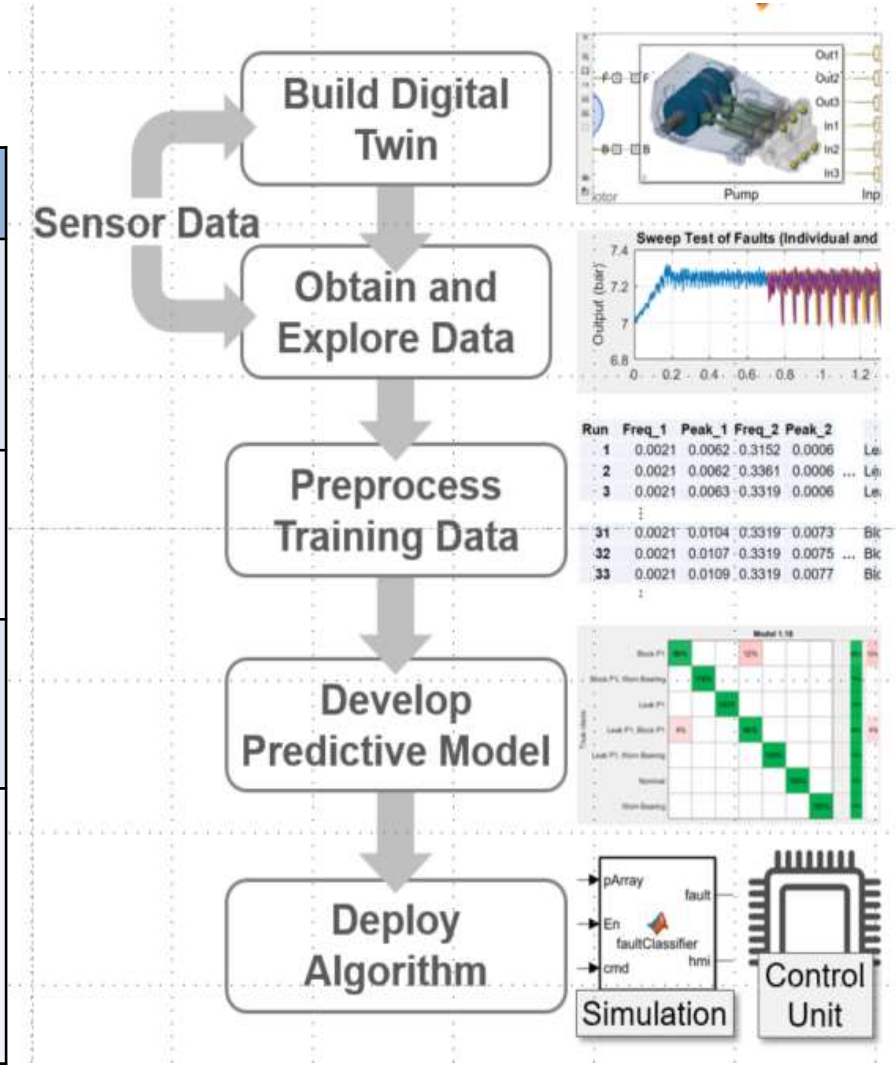


Summary: Workflow For Developing a Predictive Maintenance Algorithm



Key Takeaway: Predictive Maintenance using MATLAB and Simulink platform

Build Digital twin of the plant to generate sensor data and Simulate fault scenarios	***Use Simulink + Simscape***
Access data in **BIG DATA** - large text files, databases, or other file formats	***Use datastore + tall***
Apply **MACHINE LEARNING** for developing predictive models	***Use Apps + Documentation***
INTEGRATE machine learning models to work in **CLOUD/BUSINESS/Embedded** system	***Use Compiler / Coder***



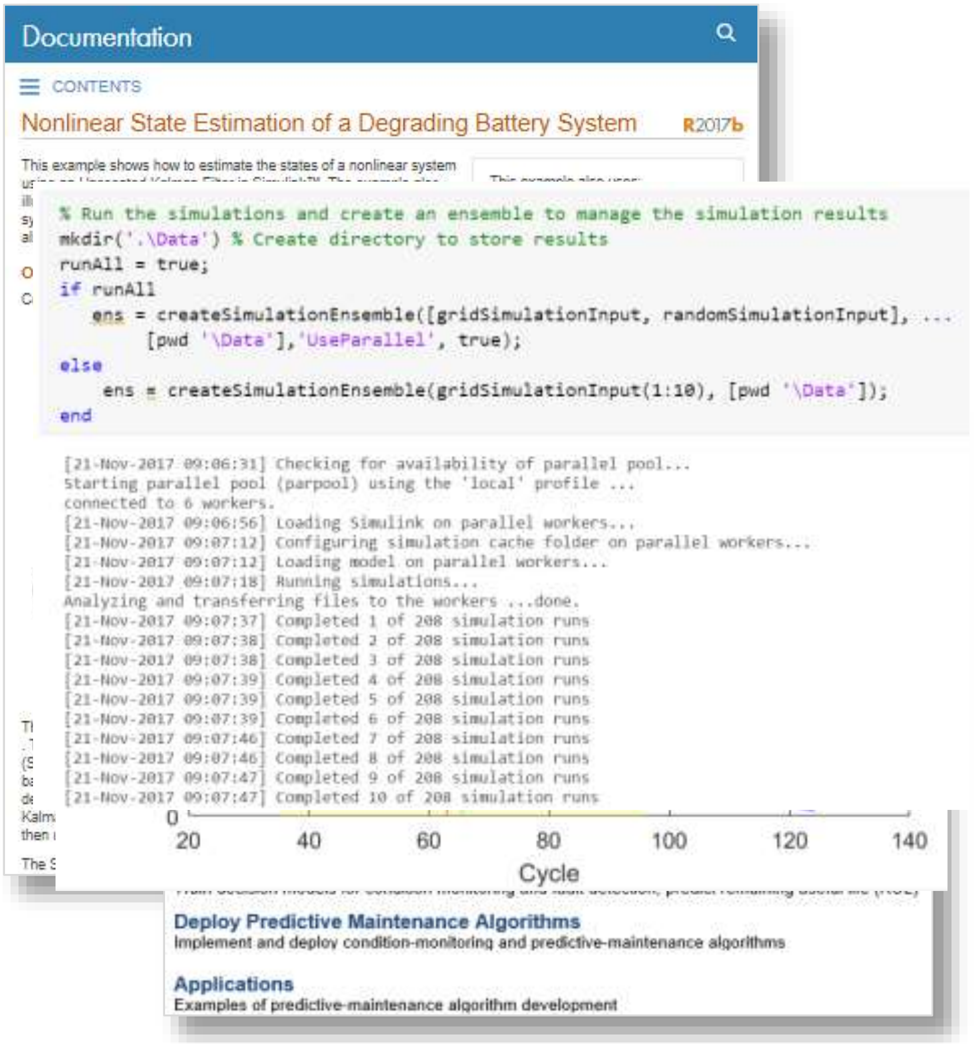
How can you get started?

R2018a

- **How do I get started with developing algorithms?**
  - Reference examples Predictive Maintenance Toolbox
  - Documentation based on the workflow

- **How do I manage my data?**
  - `ensembleDatastores` to manage and label data
  - Examples for Simulink models generating failure data

- **How do I choose which feature extraction and predictive modeling techniques to use?**
  - Functions provided for estimating RUL
  - Functions for computing condition indicators



Learn Further

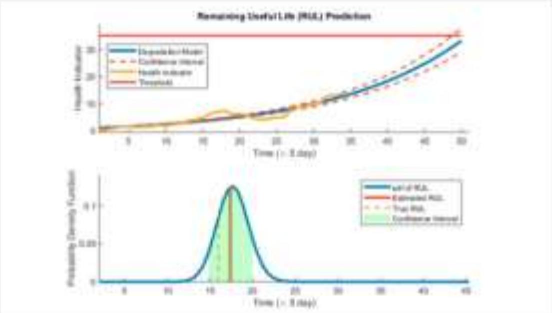
Predictive Maintenance Toolbox NEW PRODUCT

Search MathWorks.com

Overview | Features | Videos

Trial software | Contact sales

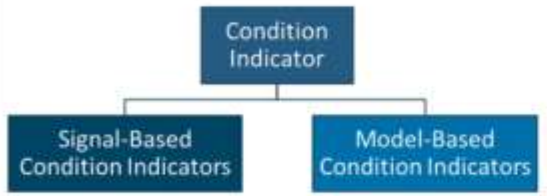
Capabilities



Remaining Useful Life (RUL) Estimation

Use time-series data and lifetime data to forecast RUL and compute confidence intervals.

» Learn more

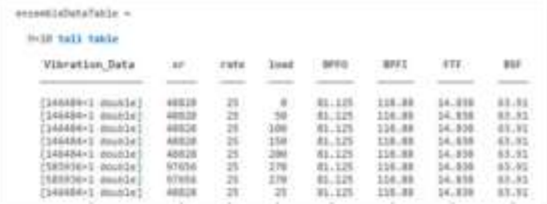


Condition Indicator Design

Extract features from sensor data that can be used as inputs to diagnostic and machine learning algorithms.

» Learn more

```
fileLocation = fullfile('.','RollingElementBearingFaultDiagnosis-Data','trai
fileExtension = '.mat';
ensembleData = fileEnsembleDatastore(fileLocation, fileExtension);
ensembleData = initializeEnsemble(ensembleData);
ensembleDataTable = tall(ensembleData)
```



Data Organization and Labeling

Access and manage data from files stored locally, on the cloud, or in HDFS.

» Learn more

```
[status,E] = generateSimulationEnsemble(simin,location
ensemble = simulationEnsembleDatastore(location);
ensembleData = tall(ensemble)

ensembleData =

  5×4 tall table

          SimulationInput                   Tacho
    ______________________________    __________________

    [1×1 Simulink.SimulationInput]    [20202×1 timetable]
    [1×1 Simulink.SimulationInput]    [20215×1 timetable]
    [1×1 Simulink.SimulationInput]    [20204×1 timetable]
    [1×1 Simulink.SimulationInput]    [20213×1 timetable]
```

Failure Data Generation from Simulink

Create simulation data that is representative of failures and store it automatically in MAT files.

» Learn more

Predictive Maintenance Toolbox uses Signal Processing Toolbox, System Identification Toolbox, Statistics and Machine Learning Toolbox

Learn Further: Predictive Maintenance Success Stories

Pump Health Monitoring System

- Spectral analysis and filtering on binary sensor data and neural network model prediction
- More than $10 million projected savings

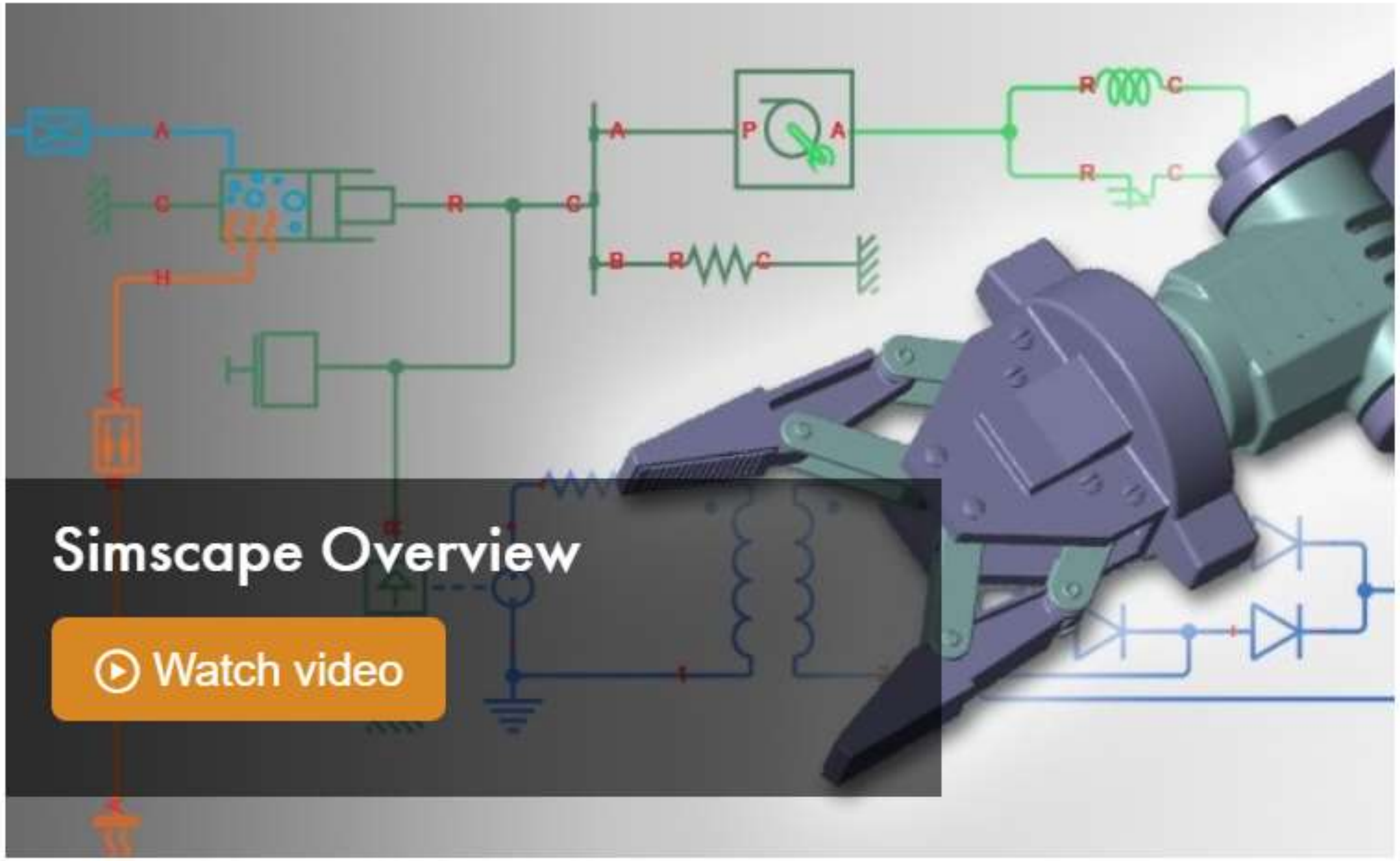
Online engine health monitoring

- Real-time analytics integrated with enterprise service systems
- Predict sub-system performance (oil, fuel, liftoff, mechanical health, controls

Production machinery failure warning

- Reduce waste and machine downtime
- MATLAB based HMI warns operators of potential failures
- > 200,000 € savings per year

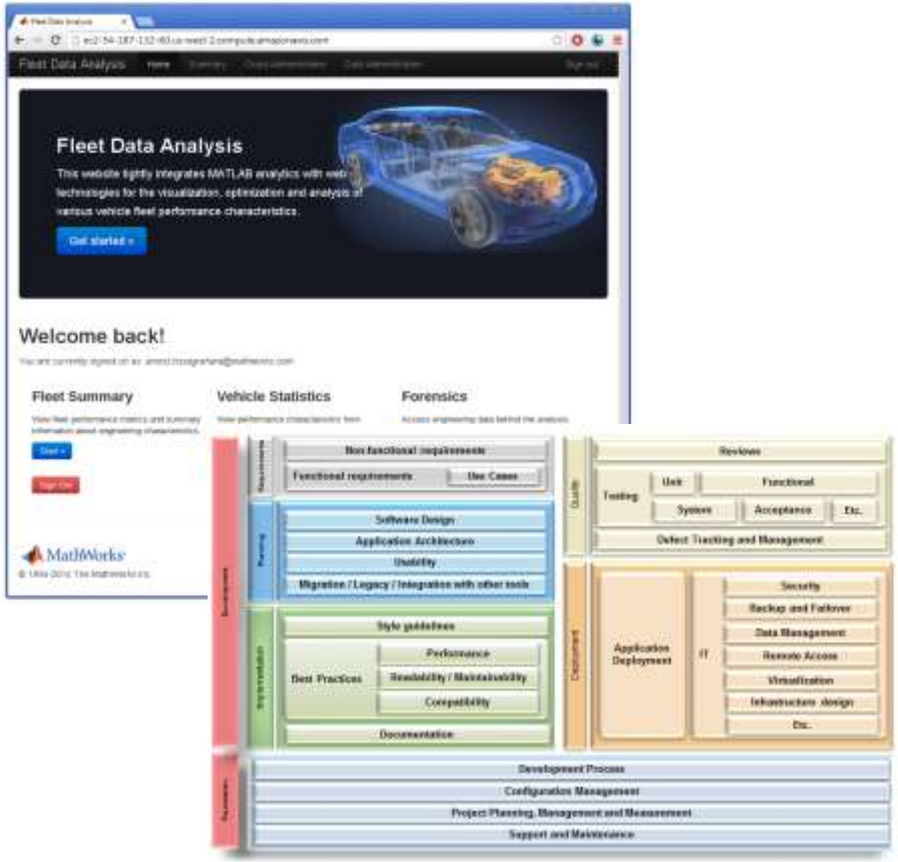
mathworks.com/big-data



mathworks.com/simscape

mathworks.com/machine-learning

MathWorks Services



- Consulting
  - Integration
  - Data analysis/visualization
  - Unify workflows, models, data

www.mathworks.com/services/consulting/

- Training
  - Classroom, online, on-site
  - Data Processing, Visualization, Deployment, Parallel Computing

www.mathworks.com/services/training/

Speaker Details

Email: Amit.Doshi@mathworks.in

LinkedIn: https://www.linkedin.com/in/amit-doshi/

Contact MathWorks India

Products/Training Enquiry Booth

Call: 080-6632-6000

Email: info@mathworks.in

- **Share your experience with MATLAB & Simulink on Social Media**
  - Use **#MATLABEXPO** on LinkedIn / Twitter

- **Share your session feedback:**

  Please fill in your feedback for this session in the feedback form