Predictive Maintenance with MATLAB and Simulink

Alec Stothert
Development Manager
Design Optimization and Identification
Why Predictive Maintenance?

- Improved operating efficiency
- New revenue streams
- Competitive differentiator
What does a Predictive Maintenance algorithm do?
Helps make maintenance decisions based on large volumes of complex data

**Condition Monitoring**
Process of monitoring sensor data from machines (vibration, temperature etc.) in order to identify significant changes which can indicate developing faults

**Predictive Maintenance**
Technique that determines *time-to-failure/remaining useful life (RUL)* from sensor data & historical data in order to predict when maintenance should be performed
Predictive Maintenance Toolbox

- Develop and validate condition monitoring and predictive maintenance algorithms
- Apply signal processing and dynamic modeling techniques to extract features from your data to monitor machine health
- Train machine learning and time-series models to detect, classify, and predict machine failure
Reciprocating Pump Example

- Monitor pump condition and predict future condition
- Measure
  - Flow rate
  - Pressure
  - Engine current
- Faults
  - Leaks
  - Worn bearings
  - Blockages

<table>
<thead>
<tr>
<th>COMMON SPECIFICATIONS</th>
<th>U.S.</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bore</td>
<td>0.945&quot;</td>
<td>24 mm</td>
</tr>
<tr>
<td>Stroke</td>
<td>1.18&quot;</td>
<td>30 mm</td>
</tr>
<tr>
<td>Crankcase Capacity</td>
<td>42 oz.</td>
<td>1.26 L</td>
</tr>
<tr>
<td>Shaft Diameter</td>
<td>1.18&quot;</td>
<td>30 mm</td>
</tr>
</tbody>
</table>
Frequency Domain Indicators

- Use spectral peaks and harmonics to understand condition of pump

```matlab
% Remove the mean from the flow and compute the flow spectrum
fA = flow;
fA.Data = fA.Data - mean(fA.Data);
[flowSpectrum,flowFrequencies] = psspectrum(fA,'FrequencyLimits',[2 250]);
```
Classify Faults Based on Condition Indicators

- Train and test a support vector machine

```matlab
% Create and train the classifier
template = templateSVM(...
    'KernelFunction', 'polynomial', ...  
    'PolynomialOrder', 2, ...  
    'KernelScale', 'auto', ...  
    'BoxConstraint', 1, ...  
    'Standardize', true);
combinedClassifier = fitcecoc(...
predictors(cvp.training(1),:), ...
response(cvp.training(1),:), ...  
'Learners', template, ...  
'Coding', 'onevsone', ...  
'ClassNames', [0; 1; 2; 3; 4; 5; 6; 7]);
```
Other Condition Indicators

Extract features using signal-based and model-based methods to determine machine health

Signal-based Methods

Condition Indicator

Model-based Methods
Advanced Condition Indicators

- Capture time-varying dynamics (e.g. vibration data) by computing time-frequency moments

- Detect sudden changes in nonlinear systems using phase-space reconstruction methods (correlation dimension, approximate entropy, Lyapunov exponent)
Tools for Managing and Analyzing Data

- Generate data using Simulink
- Analyze data using datastores
Modeling Faults In Simulink

- Parameterize Blocks

FEMA analysis to choose parameters and failure modes
Datastore to Manage Data

- Ensemble (whole table)
- Member (one row)
- Independent variables
- Data variables
  - Source variables
  - Derived variables
- Condition variables

<table>
<thead>
<tr>
<th>Date</th>
<th>qOut_meas</th>
<th>pMid</th>
<th>qVar</th>
<th>qSkewness</th>
<th>qKurtosis</th>
<th>LeakFault</th>
<th>BlockingFault</th>
<th>BearingFault</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>103.56</td>
<td>9.5309</td>
<td>-0.55515</td>
<td>2.4113</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>17-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>87.039</td>
<td>9.2604</td>
<td>-0.57675</td>
<td>2.4018</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>18-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>117.22</td>
<td>9.3804</td>
<td>-0.54379</td>
<td>2.3226</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>19-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>97.658</td>
<td>9.5407</td>
<td>-0.5533</td>
<td>2.4124</td>
<td>4e-07</td>
<td>0.74</td>
<td>0.0002</td>
</tr>
<tr>
<td>20-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>106.05</td>
<td>9.232</td>
<td>-0.56334</td>
<td>2.403</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>21-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>109.96</td>
<td>9.732</td>
<td>-0.53987</td>
<td>2.3798</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>22-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>105.06</td>
<td>9.4902</td>
<td>-0.56641</td>
<td>2.3461</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>23-Jan-2015</td>
<td>[2001×1 timetable]</td>
<td>105.1</td>
<td>9.2956</td>
<td>-0.56135</td>
<td>2.3623</td>
<td>1e-09</td>
<td>0.8</td>
<td>0</td>
</tr>
</tbody>
</table>

```matlab
ten = simulationEnsembleDatastore('.\Data');
ten.SelectedVariables = [...
    "qOut_meas", "qVar", "qSkewness", "qKurtosis", ...
    "LeakFault"],

data = read(ens)
```

<table>
<thead>
<tr>
<th></th>
<th>qOut_meas</th>
<th>qVar</th>
<th>qSkewness</th>
<th>qKurtosis</th>
<th>LeakFault</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2001×1 timetable</td>
<td>9.5309</td>
<td>-0.5551</td>
<td>2.4113</td>
<td>1.0000e-09</td>
</tr>
</tbody>
</table>
Sensor And Data Access in MATLAB

- Data sources accessible through MATLAB
  - Files (.xls, .cvs, .txt, .mat, etc.)
  - Distributed file systems Azure Blob Storage
  - Amazon S3
  - Industrial Internet of Things
Predict Pump Failures

- Use condition indicators to predict future behavior – remaining useful life (RUL)
Using Fleet Data to Predict Remaining Useful Life

% Build an train the model
mdl = residualSimilarityModel(...
    'Method', 'poly2',...
    'Distance', 'absolute',...
    'NumNearestNeighbors', 50,...
    'Standardize', 1);
fit(mdl, trainData);

% Use the model to predict RUL
[estRUL,ciRUL, pdfRUL] = predictRUL(mdl, newData);
RUL Methods and when to use them

Requirement: Need to know what constitutes failure data

- System Data
  - Run-to-failure history
    - Similarity Models
      - Large data
        - Hash Similarity Model
      - Match signal shapes
      - Known degradation dynamics
        - Pairwise Similarity Model
        - Residual Similarity Model
  - Known failure threshold
  - Life time data with or without covariates

Weibull Distribution
MATLAB Coder and Compiler

- Run predictive model on embedded devices
MATLAB Production Server and Enterprise Integration

- Integrate predictive model with your enterprise system and cloud platform
Predictive Maintenance Development - Common pains

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

- How do I manage data and what if I don’t have any data?
  - Command line functions to manage and label data
  - Examples showing Simulink models generating failure data

- How do I choose condition indicators and estimate the RUL?
  - Functions provided for estimating RUL
  - Functions for computing condition indicators
Predictive Maintenance Toolbox

- Develop and validate condition monitoring and predictive maintenance algorithms
- Apply signal processing and dynamic modeling techniques to extract features from your data to monitor machine health
- Train machine learning and time-series models to detect, classify, and predict machine failure
- Use functionality from Signal Processing, Statistics and Machine Learning and System identification
Thank you!
Predictive Maintenance Toolbox

- Develop and validate condition monitoring and predictive maintenance algorithms
- Apply signal processing and dynamic modeling techniques to extract features from your data to monitor machine health
- Train machine learning and time-series models to detect, classify, and predict machine failure
- Use functionality from Signal Processing, Statistics and Machine Learning and System identification