Designing Algorithms for Condition Monitoring and Predictive Maintenance

Predictive maintenance allows equipment users and manufacturers to assess the working condition of machinery, diagnose faults, or estimate when the next equipment failure is likely to occur. When you can diagnose or predict failures, you can plan maintenance in advance, better manage inventory, reduce downtime, and increase operational efficiency.

Developing a predictive maintenance program requires a well-designed strategy to assess the working condition of the machinery and detect incipient faults in a timely manner. Doing so requires effective use of both available sensor measurements and your knowledge of the system. You must take many factors into consideration, including:

- The observed sources of faults and their relative frequency. Such sources can be the core components of the machine (such as impeller blades and flow valves in a pump), its actuators (such as an electric motor), or its various sensors (such as accelerometers and flow meters).
- Availability of process measurements through sensors. The number, type and location of sensors, and their reliability and redundancies all affect both algorithm development and cost.
- How various sources of faults translate to observed symptoms. Such cause-effect analysis can require extensive processing of data from the available sensors.
- Physical knowledge about the system dynamics. This knowledge might come from mathematical modeling of the system and its faults and from the insights of domain experts. Understanding system dynamics involves detailed knowledge of relationships among various signals from the machinery (such as input-output relationships among the actuators and sensors), the machine operating range, and the nature of the measurements (for example, periodic, constant or stochastic).
- The ultimate maintenance goal, such as fault recovery or development of a maintenance schedule.

Algorithms for Condition Monitoring and Prognostics

A predictive maintenance program uses condition monitoring and prognostics algorithms to analyze data measured from the system in operation.

Condition monitoring uses data from a machine to assess its current condition and to detect and diagnose faults in the machine. Machine data is data such as temperature, pressure, voltage, noise, or vibration measurements, collected using dedicated sensors. A condition monitoring algorithm derives metrics from the data called condition indicators. A *condition indicator* is any feature of system data whose behavior changes in a predictable way as the system degrades. A condition indicator can be any quantity derived from the data that clusters similar system status together, and sets different status apart. Thus a condition-monitoring algorithm can perform fault detection or diagnosis by comparing new data against the established markers of faulty conditions.

Prognostics is forecasting when a failure will happen based on the current and past state of the machine. A prognostics algorithm typically estimates the machine's *remaining useful life* (RUL) or time-to-failure by analyzing the current state of the machine. Prognostics can use modeling, machine learning, or a combination of both to predict future values of condition indicators. These future values are then used to compute RUL metrics, which determine if and when maintenance should be performed. For the gearbox example, a prognostics algorithm might fit the varying peak vibration frequency and magnitude to a time series to

predict their future values. The algorithm can then compare the predicted values to a threshold defining healthy operation of the gearbox, predicting if and when a fault will occur.

A predictive maintenance system implements prognostics and condition monitoring algorithms with other IT infrastructure that makes the end results of the algorithm accessible and actionable to end users who perform the actual maintenance tasks. Predictive Maintenance Toolbox™ provides tools to help you design such algorithms.

Workflows for Algorithm Development

The following illustration shows a workflow for developing a predictive maintenance algorithm.

Beginning with data that describes your system in a range of healthy and faulty conditions, you develop a detection model (for condition monitoring) or a prediction model (for prognostics). Developing such a model requires identifying appropriate condition indicators and training a model to interpret them. That process is very likely to be iterative, as you try different condition indicators and different models until you find the best model for your application. Finally, you deploy the algorithm and integrate it into your systems for machine monitoring and maintenance.

Acquire Data

Designing predictive maintenance algorithms begins with a body of data. Often you must manage and process large sets of data, including data from multiple sensors and multiple machines running at different times and under different operating conditions. You might have access to one or more of the following types of data:

- Real data from normal system operation
- Real data from system operating in a faulty condition
- Real data from system failures (*run-to-failure* data)

For instance, you might have sensor data from system operation such as temperature, pressure, and vibration. Such data is typically stored as signal or time series data. You might also have text data, such as data from maintenance records, or data in other forms. This data is stored in files, databases, or distributed file systems such as Hadoop®.

In many cases, failure data from machines is not available, or only a limited number of failure datasets exist because of regular maintenance being performed and the relative rarity of such incidents. In this case, failure data can be generated from a Simulink® model representing the system operation under different fault conditions.

Predictive Maintenance Toolbox provides functionality for organizing, labeling, and accessing such data stored on disk. It also provides tools to facilitate the generation of data from Simulink models for predictive maintenance algorithm development. For more information, see Data Ensembles for Condition Monitoring and Predictive Maintenance.

Preprocess Data

Data preprocessing is often necessary to convert the data into a form from which condition indicators are easily extracted. Data preprocessing includes simple techniques such as outlier and missing value removal, and advanced signal processing techniques such as short-time Fourier transforms and transformations to the order domain.

Understanding your machine and the kind of data you have can help determine what preprocessing methods to use. For example, if you are filtering noisy vibration data, knowing what frequency range is most likely to display useful features can help you choose preprocessing techniques. Similarly, it might be useful to transform gearbox vibration data to the order domain, which is used for rotating machines when the rotational speed changes over time. However, that same preprocessing would not be useful for vibration data from a car chassis, which is a rigid body.

For more information on preprocessing data for predictive maintenance algorithms, see Data Preprocessing for Condition Monitoring and Predictive Maintenance.

Identify Condition Indicators

A key step in predictive maintenance algorithm development is identifying condition indicators, features in your system data whose behavior changes in a predictable way as the system degrades. 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. Examples of condition indicators include quantities derived from:

- Simple analysis, such as the mean value of the data over time
- More complex signal analysis, such as the frequency of the peak magnitude in a signal spectrum, or a statistical moment describing changes in the spectrum over time
- Model-based analysis of the data, such as the maximum eigenvalue of a state space model which has been estimated using the data
- Combination of multiple features into a single effective condition indicator (fusion)

For example, you can monitor the condition of a gearbox using vibration data. Damage to the gearbox results in changes to the frequency and magnitude of the vibrations. The peak frequency and peak magnitude are thus useful condition indicators, providing information about the kind of vibrations present in the gearbox. To monitor the health of the gearbox, you can continuously analyze the vibration data in the frequency domain to extract these condition indicators.

Even when you have real or simulated data representing a range of fault conditions, you might not know how to analyze that data to identify useful condition indicators. The right condition indicators for your application depend on what type of system, system data, and system knowledge you have. Therefore, identifying condition indicators can require some trial and error, and is often iterative with the training step of the algorithm development workflow. Among the techniques commonly used for extracting condition indicators are:

- Order analysis

- Modal analysis
- Spectrum analysis
- Envelope spectrum
- Fatigue analysis
- Nonlinear time-series analysis
- Model-based analysis such as residual computation, state estimation, and parameter estimation

Predictive Maintenance Toolbox supplements functionality in other toolboxes such as Signal Processing Toolbox™ with functions for extracting signal-based or model-based condition indicators from measured or generated data. For more information, see Design Condition Indicators.

Train Detection or Prediction Model

At the heart of the predictive maintenance algorithm is the detection or prediction model. This model analyzes extracted condition indicators to determine the current condition of the system (fault detection and diagnosis) or predict its future condition (remaining useful life prediction).

Fault Detection and Diagnosis

Fault detection and diagnosis relies on using one or more condition indicator values to distinguish between healthy and faulty operation, and between different types of faults. A simple fault-detection model is a threshold value for the condition indicator that is indicative of a fault condition when exceeded. Another model might compare the condition indicator to a statistical distribution of indicator values to determine the likelihood of a particular fault state. A more complex fault-diagnosis approach is to train a classifier that compares the current value of one or more condition indicators to values associated with fault states, and returns the likelihood that one or another fault state is present.

When designing your predictive maintenance algorithm, you might test different fault detection and diagnosis models using different condition indicators. Thus, this step in the design process is likely iterative with the step of extraction condition indicators, as you try different indicators, different combinations of indicators, and different decision models. Statistics and Machine Learning Toolbox™ and other toolboxes include functionality that you can use to train decision models such as classifiers and regression models. For more information, see Decision Models for Fault Detection and Diagnosis.

Remaining Useful Life Prediction

Examples of prediction models include:

- A model that fits the time evolution of a condition indicator and predicts how long it will be before the condition indicator crosses some threshold value indicative of a fault condition.
- A model that compares the time evolution of a condition indicator to measured or simulated time series from systems that ran to failure. Such a model can compute the

most likely time-to-failure of the current system.

You can predict remaining useful life by forecasting with dynamic system models or state estimators. Additionally, Predictive Maintenance Toolbox includes specialized functionality for RUL prediction based on such techniques as similarity, threshold, and survival analysis. For more information, see Models for Predicting Remaining Useful Life.

Deploy and Integrate

When you have identified a working algorithm for handling your new system data, processing it appropriately, and generating a prediction, deploy the algorithm and integrate it into your system. Based the specifics of your system, you can deploy your algorithm on the cloud or on embedded devices.

A cloud implementation can be useful when you are gathering and storing large amounts of data on the cloud. Removing the need to transfer data between the cloud and local machines that are running the prognostics and health monitoring algorithm makes the maintenance process more effective. Results calculated on the cloud can be made available through tweets, email notifications, web apps, and dashboards.

Alternatively, the algorithm can run on embedded devices that are closer to the actual equipment. The main benefits of doing this are that the amount of information sent is reduced as data is transmitted only when needed, and updates and notifications about equipment health are immediately available without any delay. .

A third option is to use a combination of the two. The preprocessing and feature extraction parts of the algorithm can be run on embedded devices, while the predictive model can run on the cloud and generate notifications as needed. In systems such as oil drills and aircraft engines that are run continuously and generate huge amounts of data, storing all the data on board or transmitting it is not always viable because of cellular bandwidth and cost limitations. Using an algorithm that operates on streaming data or on batches of data lets you store and send data only when needed.