Tolkku – a toolbox for decision support from condition monitoring data

Olli Saarela¹,³, Mikko Lehtonen¹, Jari Halme¹, Antti Aikala¹ and Kimmo Raivio²

¹ VTT Technical Research Centre of Finland, P.O.Box 1000, FI-02044 VTT, Finland
² Aalto University, School of Science and Technology, P.O.Box 15400, FI-00076 Aalto, Finland

E-mail: olli.saarela@vtt.fi

Abstract. This paper describes a software toolbox (a software library) designed for condition monitoring and diagnosis of machines. This toolbox implements both new methods and prior art and is aimed for practical down-to-earth data analysis work. The target is to improve knowledge of the operation and behaviour of machines and processes throughout their entire life-cycles. The toolbox supports different phases of condition based maintenance with tools that extract essential information and automate data processing. The paper discusses principles that have guided toolbox design and the implemented toolbox structure. Case examples are used to illustrate how condition monitoring applications can be built using the toolbox. In the first case study the toolbox is applied to fault detection of industrial centrifuges based on measured electrical current. The second case study outlines an application for centralized monitoring of a fleet of machines that supports organizational learning.

1. Introduction

The importance of measurement data in machine operation and maintenance (O&M) is continuously increasing. Measurement data provides a key basis for, e.g., condition-based maintenance (CBM) and operation of remote condition monitoring centres, both of which facilitate significant potentials in cost savings. Techniques for transferring data from remote locations to condition monitoring centres are available, and a multitude of useful data processing and analysis techniques are presented in the literature. At the current state of the art the analysis still requires considerable human expertise, especially in deciding what service and operational actions are needed to ensure reliable and efficient machine operation. Also, it is desirable to keep the number of sensors in an individual machine as low as possible, both to keep manufacturing expenses down and to reduce the number of failure prone components. Software tools are needed in extraction of relevant information from available measurement data and to support O&M personnel in their decision making. [1, 2, 3]

This paper describes the principles that have guided the development and the deployment of a software toolbox, named TOLKKU, which addresses data processing and decision support needs related to O&M. Scenarios for utilizing the TOLKKU toolbox include:

³ To whom any correspondence should be addressed.

Published under licence by IOP Publishing Ltd
- **For research and development**: Acquiring increased understanding of the system’s load and behaviour throughout its whole life cycle (e.g. information on the usage, operating situations, loadings etc.). Gaining quantitative evidence on the feasibility and reliability of different solutions.

- **For operation of the machines**: Providing feedback on the systems’ behaviour (e.g., a “traffic light” that informs the operator about her operation style). Optimizing operation (e.g. anticipation of service needs and avoiding emergency actions).

- **For service and maintenance**: Anticipating the wear and failure of the machine/process parts (e.g. information on the extensive load and possible breakdown of a part). Supporting condition based maintenance, remote condition monitoring, maintenance service etc.

Section 2 discusses the design philosophy behind toolbox development, section 3 the contents of the toolbox, and section 4 its documentation, a crucial element for toolbox deployment. Sections 5 and 6 show how the toolbox can be utilized in building O&M applications.

2. Toolbox design

In the toolbox development the aim was to implement a carefully selected set of methods proven useful in practice, not to implement a large number of computational techniques found in the literature. To achieve this, the development was decided to be based on use cases specified by machine manufacturers. Each case included measurement data from one or more machines and a specific objective such as, e.g., early fault detection, fault diagnosis, and comparison of load and use histories. The cases were solved, and especially the data analyses carried out were further studied on a higher abstraction level. The aim was to discover which functions would be useful in the toolbox to be developed, not only the particular cases solved, but also for similar cases expected to be encountered in the future. To support close cooperation with the companies, the “release early, release often” software development approach [4] was selected. This facilitated rapid and continuous feedback from participating companies, judged necessary for creating a toolbox for practical use.

Designing a toolbox for a particular application area requires consideration of aspects particular to that domain. Analysis of condition monitoring data is characterized by several factors imposing constraints to system design:

- **Large variability in the level of instrumentation**. Depending on machine type and age, they can significantly differ in the level of instrumentation installed. Consequently, the methods in the toolbox should impose as few restrictions on the data set available as possible, and should handle some missing data items.

- **Limitations in data transfer**. Especially when a fleet of machines is deployed in remote locations worldwide, the cost of data transfer can be considerable. Hence the toolbox must contain functionality that facilitates compaction of essential information in, e.g., load and use profiles for cost effective transmission.

- **Widely varying operational situations**. Work machines are used in multitudes of operational situations with differences in, e.g., environmental conditions, operator actions, and the task the machine is being used for. Measurement data is not in general comparable across different operational situations. Hence the toolbox must provide functionality for separating operational states from each other, analysing them separately.

- **Unreliability of measurement data**. Raw measurement data is typically ill suited for analysis due to, e.g., outlier values and missing values due to instrumentation failures. Hence, the toolbox must contain functionalities for screening and processing data, and handling missing data points in the analysis.

On the other hand, there are reasonably long manufacturing series of certain types of machines and subsystems that are used in several machine designs. This makes it cost-effective to enter a priori
knowledge, e.g., in the form of fault trees [5, 6]. It also facilitates accumulation of fault statistics and analysis of fault precursors.

The toolbox was implemented in the MATLAB language. This language was selected mainly because the potential application developers in companies are familiar with it and already use it in their work.

3. Toolbox functionalities

The actual computational methods implemented in the toolbox were selected based on the needs of participating companies, expressed mostly as use cases consisting of analysis goals, machine properties, and measurement data. Jointly the company needs covered a wide range of O&M topics, which also gave a basis for arranging the methods as modules. The module structure is shown in table 1.

| Module              | Description                                                                 | Examples of methods included                                      |
|---------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------|
| Data import         | Utility functions to assist parsing data files and importing data for use with the toolbox. For data storage the toolbox contains an SQL data base. | A SQL database for storing data and analysis results. |
| Data preparation    | Tools for ensuring the validity of the data used for analysis, and data processing. | Outlier detection, filtering, scaling of measurement data, selecting subsets of data. |
| Feature extraction  | Characterization of data by computed statistical figures and functions.     | Statistic figures, detection of trends and other waveforms.          |
| State recognition   | Identification of operational states from history data, classification of new data, and detection of deviations from previously encountered states. | Clustering, segmentation of time series.                            |
| Load profiles       | Compacting, logging, and analysing the operational history of machines.     | Time at level, rainfall cycle counting, cross tabulation. Comparison of profiles. |
| Anomaly detection   | Detection of values and patterns that deviate from those encountered in normal operation. | Clustering, detection of changes in correlations.                   |
| Analysis of causality | Identification of dependencies and fault causes.                        | Fault tree analysis, stepwise regression.                           |
| Specific analyses   | Analysis methods specific to particular component types: bearings and gearboxes. | Time-frequency analysis, computation of nominal frequencies.         |
| Decision support    | Cumulative organisational learning, support in determining future actions. | Optimization of maintenance plans, case-based reasoning.            |

The toolbox functions supplement each other, making the whole greater than the sum of the parts. In the area of machine operation there are some important synergies between toolbox functionalities. For example, data preparation, feature extraction, and state recognition must work together. On one hand feature extraction and state recognition perform best with validated data; on the other hand the operational state determines a valid range for data. In the toolbox this has been addressed by facilitating the use of two levels of operational states. On the first level the machine operation is classified robustly with states that can be easily detected mainly from control signals. This state...
classification is utilized in data preparation, and more detailed state classification can be done with more reliable data. The classified machines states can overlap, as shown in figure 1.

Another important area where the toolbox modules supplement each other consists of load profiling, fault trees, and decision support. Their joint use is discussed in section 6.

![Figure 1. The operational states of a machine can overlap.](image)

4. Toolbox documentation

The users of a toolbox are application developers who build applications related to machine operation and maintenance. Consequently, the technical documentation has to be aimed for this group. The most essential function of the documentation is to help developers to deploy the toolbox. Due to the popularity the programming language and environment and the vast amount of training material available, the users can be assumed to be familiar with it. Hence the documentation of the toolbox can be relatively focused.

Interviews with toolbox users in participating companies indicated a need for two levels of documentation: The lower-level documentation consists of descriptions of individual functions and their call interfaces, whereas the higher-level documentation focuses on concepts and how a number of toolbox functions can be used together to implement desired functionalities. Also the descriptions of data structures were included in the higher-level documentation, as some knowledge of them was judged essential in gaining an overall picture of the toolbox. The higher-level documentation was implemented as HTML pages, available also through the documentation browser of the programming environment.

![Figure 2. A snapshot of the Tolkku HTML documentation in a web browser. The top of the page “Rainflow cycle counting” is shown.](image)
platform. Snapshots of the higher-level documentation are shown in figures 2 and 3. Of a few visual styles examined, the one adapted from the documentation of the Python programming language [7] was deemed best.

The lower-level documentation was implemented in the header texts of the files where toolbox functions are defined. They are accessible via the hyperlinks in the higher-level documentation and with the standard “help” command of the development platform.

The toolbox documentation also includes a number of demonstrations, i.e., more comprehensive examples of toolbox use including data necessary for executing them. These demonstrations serve a dual role. From the toolbox users’ point of view they present sample code the users can try out and copy to any application they are writing. Invisibly to the end users, the demonstrations also form an important part of the test suite with which a considerable part of software testing can be automated [8]. During toolbox development these tests are run regularly in order to discover, e.g., any new programming errors introduced as a side effect of enhancing the toolbox functionality or correcting other programming errors.

Figure 3. A snapshot of the Tolkku HTML documentation in a web browser. The bottom of the page “Rainflow cycle counting” is shown.

5. Application example: centrifuge fault detection

In this application, the signal processing functionality of the Tolkku toolbox was deployed in centrifuge fault detection. In the industrial setup studied the accept fraction was regularly ejected from the centrifuges. Due to fouling the ejection mechanisms occasionally fail, causing all of the centrifuge input flow to be directed to the reject flow. The aim was to detect such faults, preferably in advance, without having to install additional instrumentation. The electric currents of centrifuges’ power supplies were selected as inputs to fault detection, as they were already measured and the operation of the ejection mechanisms was seen as current peaks (figure 4).
Figure 4. Electric current of centrifuge power supply. Absence of peaks indicates a fault.

A lower degree of fouling was known to cause the current peaks to decay slower than usual (figure 5). In some fault cases this has indicated approaching faults well in advance.

Figure 5. Electric current of centrifuge power supply. Slow decay of peaks indicates fouling, giving an advance warning of faults.

Changes in the input flow rates and raw materials cause the electric currents of the centrifuges vary considerably. Hence reliable fault detection required more detailed processing than, e.g., a straightforward level comparison. One of the waveform detection functions in the toolbox allows the application developer to define a series of “gates”, through which only the waveform of interest passes [9]. Figure 6 shows the gate system used to detect slow decay; a slightly simpler gate system was used to detect the occurrence of peaks.

Figure 6. Waveform detection with a series of gates (indicated as solid vertical lines). The waveform on the left doesn’t pass through the gates. Hence, the decay is classified as normal. The waveform on the right passes through the gates, and a slow decay is detected.
Internally each gate is defined as a tuple (time offset, lower limit, upper limit), indicated for gate $g_j$ by $(g_j^T, g_j^L, g_j^U)$. The waveform specified by a system of $N$ gates is detected from time series $x$ at time $t$ if

$$
\bigwedge_{j=1}^{N} g_j^L < x(t - g_j^T) < g_j^U
$$

where $\bigwedge$ is the “and” operator. The output from the waveform detection was passed through a trigger function. The hysteresis implemented in the Tolkku trigger function generated a steady enough indication to be shown to process operators (figure 7). The fault detection application is outlined in figure 8.

![Centrifuge 2 Current](image)

Figure 7. Early fault detection from centrifuge current was implemented using waveform detection and trigger with hysteresis.

![Slow decay detected with tolkku_find_waveform](image)

![Warning triggered with tolkku_hysteresis_trigger](image)

Figure 8. Signal processing implemented for each centrifuge. Slow peak decay raises a warning; an absence of peaks raises an alarm.

6. Centralized monitoring of a fleet of machines

In centralized condition monitoring data from a fleet of machines is collected to a centre for analysis. The most important output from the analysis is recommendation for further actions. Recommended actions can suggest, e.g., continuing machine operation as planned, limiting its use to make it last to the next scheduled service, or carrying out a set of service operations. At the current state of the art a number of techniques exist for reliable remote data collection and data processing, but generating recommendations for relevant actions requires considerable human expertise. This forms a bottleneck in the operation of the centres.

The data received from a fleet of machines, in addition to being used as input to analysis, also forms a rich basis for learning. The expertise of the personnel of the centre naturally increases as they
carry out their work, but individual learning cannot utilize the full potential of the information contained in the data. Furthermore, individual learning is difficult to transfer to new employees.

The Tolkku toolbox has been designed for implementing a more systematic approach to organisational learning. In the envisioned application (figure 9), data packets received from the fleet are used to update load and use profiles and fault trees. These condense the raw measurement data into descriptions of machine state, retaining information relevant to condition monitoring. In case of limited (expensive) data transfer this compaction can be done at each machine, and only the load and use profiles, fault trees, and possibly other signature-like information transferred to the monitoring centre.

The updated load and use profiles and fault trees are used in checking the loss of remaining theoretical lifetime of the machine components. In addition, Case-Based Reasoning [10] is applied to search for any earlier cases where a similar state of a similar machine or machine component has preceded a fault. The output from these analyses is used as input to optimization of maintenance plan.

![Figure 9. Envisioned Tolkku application in maintenance centres. Systematic collection of case data facilitates organisational learning.](image)

In the case of a machine fault the same toolbox functions are used slightly differently (figure 10). The fault tree is analysed to discover root causes that could have caused the observed fault, and to estimate their probabilities. Case-Based Reasoning is used to search for earlier cases where a similar machine (or subsystem) with a similar load and use history has the same fault symptoms. Most similar earlier cases are examined to discover what the actual root causes were, what service operations were carried out, and whether the fault had reoccurred. The discovered information is used to generate a check list and work plan for the service personnel.

![Figure 10. Operation in fault cases is supported by the same Toolbox functions.](image)

The envisioned application contains two important feedback loops. The more automated one (depicted in figures 9 and 10) involves storing new analysis cases in the case base, making them immediately available in the analysis of future cases. The other loop (not shown in the figures) involves more thorough analysis of the collected case data. Analysis of the fault cases, involving the load and use...
histories of each machine, can be used to tune estimation of remaining lifetime, even statistically if the
fleet being monitored includes a sufficiently large number of similar machines or subsystems. This
latter feedback loop also provides information for product development: the machine designers gain
understanding about load and use conditions that lead to faults.

7. Conclusions
With continuing need to improve the efficiency of machine operation and maintenance software tools
have become crucial in extracting relevant information from measurement data. Implementing data
processing methods as a toolbox facilitates building a wide range of O&M applications to address the
diverse and continuously changing needs of O&M companies. The developed tools facilitate increased
understanding of the system’s load and behaviour throughout its whole life cycle, automation of
diagnostics and prognostics, and more optimized operation and maintenance.

The architecture of the diagnostic system proposed for centralized condition monitoring
supplements earlier work [1] by introducing feedback loops facilitating organizational learning. This
allows information from single machines to be effectively utilized in monitoring, diagnosis, and
maintenance of fleets of machines.

Acknowledgements
This paper is based on work carried out within the TOLKKU project in FIMECC EFFIMA program.
The support and co-operation with Bronto Skylift, John Deere, Konecranes, Metso Minerals, and
FIMA (Forum for Intelligent Machines) are gratefully acknowledged.

References
[1] Muller A, Marquez A C and Iung B 2008 On the concept of e-maintenance: review and current
research Reliability Engineering & System Safety 93(8) pp 1165–87
[2] Jardine A K S, Lin D and Banjevic D 2005 A review on machinery diagnostics and prognostics
implementing condition-based maintenance Mechanical Systems and Signal Processing
20(7) pp 1483–1510
[3] Liao L and Lee J 2010 Design of a reconfigurable prognostics platform for machine tools
Expert Systems with Applications 37(1) pp 240–252
[4] Raymond E S 1999 The Cathedral and the Bazaar O’Reilly Media
[5] Ericson C A 1999 Fault tree analysis – a history Proc. of the 17th international system safety
conference (Orlando, FL, USA, 16-21 Aug 1999)
[6] Halme J and Aikala A 2012 Fault tree analysis for maintenance needs Submitted to COMADEM
2012 (Huddersfield, 18-20 June 2012)
[7] Documentation of the Python programming language 2012 http://docs.python.org/ (accessed
2.1.2012)
[8] Leung H K N and White L 1989 Insights into regression testing Proc. Conf. on Software
Maintenance (Miami, FL, USA, 16-19 Oct 1989), IEEE pp 60-69
[9] Söderholm K, Ihalainen H and Ritala R 1999 Fuzzy gate based feature detection Proc.
Information Decision and Control 99 (Adelaide, SA, 8-10 Feb 1999) IEEE pp 545–549
[10] Aamodt A and Plaza E 1994 Case-based reasoning: foundational issues, methodological
variations, and system approaches AI Communications 7(1) pp 39–59