Aalborg Universitet

Arrowhead Framework services for condition monitoring and maintenance based on the open source approach

Campos, Jaime; Sharma, Pankaj; Albano, Michele; Jantunen, Erkki; Baglee, David; Ferreira, Luis Lino

Arrowhead Framework services for condition monitoring and maintenance based on the open source approach

Jaime Campos, Pankaj Sharma, Michele Albano, Erkki Jantunen, David Baglee, Luis Lino Ferreira

Abstract— **The emergence of new Information and Communication Technologies, such as the Internet of Things and big data and data analytics provides opportunities as well as challenges for the domain of interest, and this paper discusses their importance in condition monitoring and maintenance. In addition, the Open system architecture for condition-based maintenance (OSA-CBM), and the Predictive Health Monitoring methods are gone through. Thereafter, the paper uses bearing fault data from a simulation model with the aim to produce vibration signals where different parameters of the model can be controlled. In connection to the former mentioned a prototype was developed and tested for purposes of simulated rolling element bearing fault systems signals with appropriate fault diagnostic and analytics. The prototype was developed taking into consideration recommended standards (e.g., the OSA-CBM). In addition, the authors discuss the possibilities to incorporate the developed prototype into the Arrowhead framework, which would bring possibilities to: analyze various equipment geographically dispersed, especially in this case its rolling element bearing; support servitization of Predictive Health Monitoring methods and large-scale interoperability; and, to facilitate the appearance of novel actors in the area and thus competition.**

I. Introduction

Different trends in both society and the market, competition puts requirements on the development of new technologies. The requirements involve, for instance on the Information and Communication Technologies (ICTs) to support product and service production in today's society [1]. Therefore, different stakeholders demand new approaches. Consequently, companies that are not able to keep up with the pace of the latest ICT developments may lose their competitive advantage and the ability to compete under the same conditions as their competitors, leading to shrinking market share and profitability [2]. It is, therefore, important to understand the potential use of the emergent ICTs, i.e. new open source tools, data mining as well as big data analytics solutions systems for the domain of interest. The work of standardisation for purposes of integration of data as well as software components for interoperability becomes crucial in this context.

For some years, attempts have been made to introduce standards in the software industry. Such attempts have the advantages of easier upgrading of system parts, more suppliers systems, providing more alternatives to technology, faster technology development and reduced prices [3]. Simultaneously, there have been attempts also to standardize the systems and subsystems for condition monitoring and maintenance. One such attempt is Open System Architecture (OSA) which allows the user to use standard systems to achieve flexibility, functionality and interoperability, Condition Based Maintenance (CBM) which allows maintenance to be carried out when a failure is about to occur, based upon a set of parameters. Part of the OSA-CBM is the condition monitoring layer. The condition monitoring of rolling element bearing (REBs) fault is vital to be able to keep an equipment running. The REBs are the most exposed mechanical parts in rotary machines [4]. Failures of REBs reduces the disposal of appropriate equipment, which in turn could cause economic losses. In addition, it is well known that REBs and analytics are significant enablers for the finding of patterns as well as hidden ones, which results in enhancing decision-making and supporting knowledge creation within an organisation. In addition, too much maintenance incurs expenditures, and it may not be economical to perform it too frequently as well as too little maintenance might result in costly consequences [5]. Moreover, it is well known that one of the few methods to reduce the cost of production is to diminish the cost of operation together with maintenance. In some industries, it is the second highest or even the highest element of the operating costs [6]. Consequently. it is, therefore, crucial to understand when maintenance should be performed and it is, therefore, the open source solutions and their usability becomes important in the domain of interest. The reason is that the development of such systems or Information Communication Technology (ICT) systems, in general, is costly and time-consuming, which in itself demonstrates that these are variables that might increase more than expected.

Thus, the current paper highlights aspects that are relevant for the domain of interest, i.e. the industrial maintenance engineering and asset management. The novelty of the approach lies in the usage of open source solutions, together with adherence to mature standards (e.g.: OSA-CBM) and servitization of the access to Predictive Health Monitoring methods. This approach can have a deep impact on interoperability of solutions, and in the long term democratization of the access to advanced monitoring and maintenance solutions, thus stimulating competition and

Jaime Campos is with Linnaeus University, Sweden (e-mail: jaime.campos@lnu.se).
Pankaj Sharma is with the National University of Singapore (e-mail: pankajsharma@nus.edu).
Michele Albano is with the Department of Computer Science, Aalborg University, Denmark (e-mail: mialb@cs.aau.dk).
Erkki Jantunen is with VTT Technical Research Centre of Finland Ltd (e-mail: erkki.jantunen@vtt.fi).
David Baglee is with the Faculty of Engineering and Advanced Manufacturing, University of Sunderland, UK (e-mail: david.baglee@sunderland.ac.uk).
Luis Lino Ferreira is with CISTER, ISEP, Polytechnic Institute of Porto, Portugal, (e-mail: llf@isep.ipp.pt).

empowering the industry with better condition monitoring and maintenance. The authors provide the aspects of the Open system architecture – condition monitoring (OSA-CBM) and Machinery Information Management Open Systems Alliance (MIMOSA) cris in Section 2. In Section 3, predictive methods are gone through and in Section 4 the Arrowhead framework is briefly presented, together with its potential for the servitization of the access to Predictive Health Monitoring methods. Next in Section 5 a data analysis solution is presented, which is possible to incorporate into the arrowhead framework. Finally, the paper is concluded in Section 6.

II. THE OPEN SYSTEM ARCHITECTURE - CONDITION BASED MAINTENANCE

Condition-based maintenance (CBM) is getting more and more acceptance within the industry and military. A CBM system contains several functional capabilities. The system to function requires the integration of various hardware and software components. These components may come from different sources but it should be possible to integrate and interchange them. OSA-CBM started as a solution to this problem. Its objective was precisely to facilitate the integration and interchangeability of the components available from a variety of sources. They are working to build de facto standards that cover all functionalities starting from data collection through the recommendation of specific maintenance actions. OSA-CBM was an industry initiative, partly funded by the US Navy through a DUST (Dual Use Science and Technology) programme with an objective to develop and demonstrate an Open System Architecture for Condition Based Maintenance. The participants come from industrial, commercial and military applications of CBM technology.

Other important contributors have been MIMOSA CRIS and IEEE standards. With Internet and LAN (local area network), distributed software architectures can be used for CBM. It is also cost-effective. It consists of the following 7 layers. Layer 1- Data Acquisition: It senses the signal from the machine and digitises it. The digitised sensor data is then passed on to the second layer. Layer 2- Data Manipulation: In this layer, different signal processing operations are performed on the data received from the sensor and/or other signal processing modules. Layer 3- Condition monitor: It compares the received values from the previous layer with the expected values. It may also generate alerts or give the operational state of the machine. Layer 4- Health assessment: It receives data from condition monitor or other health assessment modules. The main function of the module is to determine if a deterioration has occurred in the machine and if it has, it is expected to provide the different relevant fault conditions. Layer 5- Prognostics: The main function of this layer is to give a projection on the future health of the machine or its remaining useful life (RUL). While doing so, it should consider different estimates of future usage profiles of the machine. Layer 6- Decision support: The primary function of this layer is to recommend actions and in case of multiple alternatives, the implications of each alternative. One of the alternatives, for example, can be a maintenance action schedule. Layer 7- Human Interface (Presentation layer): This layer provides displays health of the machine (layer-4), prognosis of its condition (layer-5) and recommendations for action, if any, (layer-6) to the user. When an abnormal condition is reported,

the user can dig into multiple layers to trace the sources of data and the process of the decision-making.

In 1994, a non-profit trade organisation founded MIMOSA. The objective was to encourage exchange of information among the plants on machine maintenance, to come up with publications and develop open international agreements (standards) on machine maintenance information systems, [7], (www.mimosa.org) MIMOSA developed a Common Relational Information Schema (CRIS). It is a relational database model for different data types that need to be processed in CBM application. The system interfaces have been defined according to the database schema based on CRIS. The interfaces' definitions developed by MIMOSA are an open data exchange convention to use for data sharing in today's CBM systems.

III. PREDICTIVE HEALTH MONITORING METHODS

Condition-based maintenance (CBM), as a maintenance strategy, requires data for diagnosis and prognosis. The collection and analysis of data is a complex process that is to be customized according to the asset being monitored [8]. Data collected in a CBM program can be categorized into two main types: the so-called event data and condition monitoring data. Event data include the information on what happened (e.g., installation, breakdown, overhaul, etc., and what the causes were) and/or what was done (e.g., minor repair, preventive maintenance, oil change, etc.) to the targeted physical asset. Condition monitoring data are the measurements related to the health condition/state of the physical asset [9]. Data processing is a step that follows data acquisition. It is a two-stage process where the data is cleaned and analysed. Manual inspection, sensor fault isolation, graphical tools are some of the methods that are used for improving the data quality. Once the quality of the data is adequately improved, it is subjected to data analysis. These steps are presented in Figure 1.
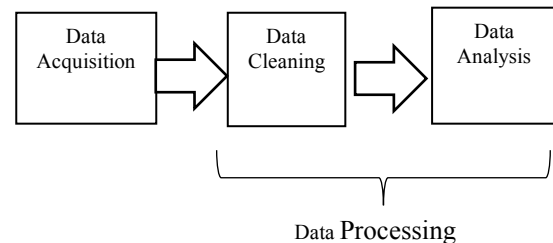
Figure 1 - Asset health monitoring process.

The acquired data can be in one of the three forms; value type, waveform or multi-dimension type. The processing step for waveform and multi-dimensional data is also known as signal processing. Waveform data in condition monitoring are vibration signals, acoustic emissions, ultrasonic signals, partial.

Various techniques of data processing are shown in Figure 2. The time-domain analysis is directly based on the time waveform itself. The traditional time-domain analysis calculates characteristic features from time waveform signals such as mean, peak, peak to peak interval, standard deviation, crest factor, skewness, kurtosis, etc. [10, 11]. These time-domain features are used for feature extraction. The frequency-domain

analysis is based on the transformed signal in the frequency domain. The advantage of frequency-domain analysis over time-domain analysis is its ability to easily identify and isolate certain frequency components of interest [12, 13]. The most widely used conventional analysis is the spectrum analysis using a Fast Fourier transform (FFT). The main idea of spectrum analysis is to either look at the whole spectrum or look closely at specific frequency components of interest and thus extract features from the signal [9].

Frequency-domain analysis cannot handle non-stationary waveform signals, which are very common when machinery faults occur. Time-frequency analysis analyses non-stationary waveform signals. The method investigates waveform signals in both time and frequency domain. The traditional time-frequency analysis uses time-frequency distributions, which represent the energy or power of waveform signals in two-dimensional functions of both time and frequency to better reveal fault patterns for more accurate diagnostics [9].
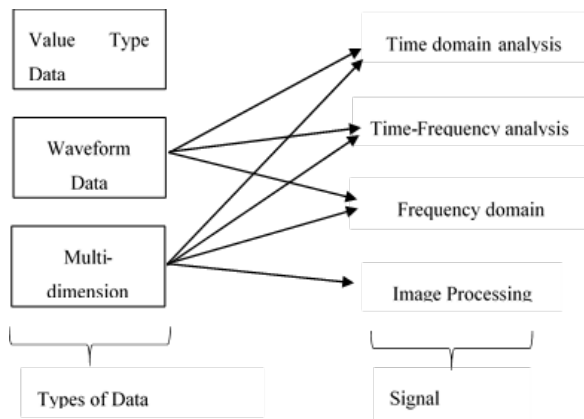

Figure 2 - Data Processing.

Health monitoring of assets can be of two broad types: Diagnostic and Prognostic. Diagnostic health monitoring is concerned with the detection of faults on occurrence. Prognostics, on the other hand, attempts at preventing the failures from occurring by predicting them. Ideally, prognostics is a better solution than diagnostics because it can help in ensuring that unexpected breakdowns do not jeopardize the machine operations. Even if the prognostics are not able to prevent the failure, it still gives a forewarning to the maintenance personnel to keep essential resources ready in case of the eventual failure. This cuts down the logistical delays of repair considerably. However, diagnostics cannot be entirely ruled out because no prediction technique can be correct all the time. When prognostics fails, diagnostics will help in isolating and identifying the fault. The prognostics can predict two things, remaining useful life (RUL) of the component or the probability that the component will work without failing for a specified duration of time. While the first type of prognostics (RUL estimation) is more commonly used, critical assets like military weapon systems, nuclear plants, etc. prefer the second type of prognostics, i.e., the chances of surviving a specific mission. Both the prognostic methods use the past failure data and the current asset health for predicting future faults. Rotating machinery failure prediction is done by three most common

methods: Traditional reliability approaches—event data based prediction, Prognostics approaches—condition data based prediction and Integrated approaches— a prediction based on both event and condition data. Traditional approaches to reliability estimation are based on the distribution of event records of a population of identical units. Many parametric failure models, such as Poisson, Exponential, Weibull, and Log-Normal distributions have been used to model machine reliability. The most popular among them is the Weibull distribution due to its ability to accommodate various types of behavior including infant mortality in the ''bath-tub'' curve [14]. There are two types of methods of prognosis: physics 'model'-based and data-driven i.e. statistical and artificial intelligence (AI). The physics-based prognostic models describe the physics of the system and failure modes based on mathematical models such as Paris' law, Forman law, fatigue spall model, contact analysis and 'stiffness based damage rule' model [15].

These approaches utilize physics specific, an explicit mathematical model of the monitored machine. Based on this explicit model, residual generation methods such as Kalman filter, parameter estimation (or system identification) and parity relations are used to obtain signals, called residuals, which is indicative of fault presence in the machine [16-20]. Finally, the residuals are evaluated to arrive at fault detection, isolation and identification. The data-driven prognostic models attempt to be driven by routinely and historically collected data (CBM, SCADA measurements, etc.) [21,22]. The data-driven prognostic models covers a high number of different techniques and artificial intelligence algorithms such as simple trend projection model, time series prediction model, exponential projection using ANN, data interpolation using ANN, particle filtering, regression analysis and fuzzy logic, recursive Bayesian technique, HMM, hidden semi-Markov model, system identification model, etc. [23,24]. The data-driven methods utilise data from past operations and current machine conditions, to forecast the remaining useful life. However, with the emergence of big data analytics, it is possible to find works where researches are using for instance clustering as well as classification.

The factors of the 3Vs (volume, velocity and variety of data), which are more relevant to the area of industrial maintenance, are "velocity" and "volume" of the data. However, if we talk about asset management, then the "variety" aspects of the data could also come into the picture. The use of, for instance, unsupervised and supervised approaches, that is, clustering and classification, are essential to comprehend, i.e. to be able to understand if the open source tools are adequate to use in the domain of interest. For instance, the wavelet characteristic, such as its multi-resolution property, permits the use of cluster algorithms, especially the WaveCluster algorithm, since it could efficiently detect arbitrary shape clusters on various scales with different degrees of correctness [25]. The Wavelets may be highly suitable for classification as well. For instance, classification algorithms can be applied to the wavelet domain data. The authors discuss several other approaches, such as linear regression and neural network [25]. All of these methods and approaches can be found in open source tools. In addition, for all flow of data, such as sensor data, time data, signal analysis, diagnosis, it is possible to investigate further the different output with data mining/big data analytics like clustering and/or classification to find possible hidden patterns. Other works that

combine maintenance approaches and data mining can be found in [26-29].

IV. THE ARROWHEAD FRAMEWORK

The Arrowhead Framework is the result from a set of European projects (Arrowhead, SOCRADES, IMC-AESOP, ARUM, INTER-IoT, etc.) in which Service Oriented Architectures (SOA) principles have been applied to IoT and industrial applications. The framework was the main result of the Arrowhead project, but continued its development independently and is now being used in multiple industrial installations and further developed in other projects. This Framework consists of what is needed for anyone to design, implement and deploy an Arrowhead compliant system. It aims at enabling all of its users to work in a common and unified approach – leading towards high levels of interoperability. The Arrowhead Framework includes principles on how to design SOA-based systems, guidelines for its documentation [33] and a software framework capable of supporting its implementations [1].

To discuss and define a local cloud architecture, it is necessary to define a few keywords, which may have other definitions in different domains and contexts. When it comes to the Arrowhead Framework, the main keywords are as follows [1]. A *service* is what used to exchange information from a providing system to a consuming system. Services can use different SOA protocols such as REST, COAP and XMPP. A service is provided by a software system. A *system* in the Arrowhead Framework provides and/or consumes services. Therefore, a system can play the role of both provider of one or more services as well as at the same time consume one or several services. A specific system is executed on a *device*.

The Arrowhead Framework builds upon the local cloud concept, where local automation tasks should be encapsulated and protected from outside interference. Application services are not able to communicate with services outside of the local cloud, except with other Arrowhead compliant local clouds through the Gatekeeper Service. Each local cloud must contain at least the three mandatory core systems: Discovery, Authorization and Orchestration. Thus allowing the establishment of connections between Arrowhead application services. Examples of these services are services capable of providing various sensor readings, controlling AC devices, getting energy consumption, etc.

The Discovery Service, through its ServiceRegistry System stores information about all active producing services within a local cloud. It allows all producer and consumer services to find each other, allowing endpoints to be dynamically changed during run-time. All application services within the local cloud must publish its producing services to the ServiceRegistry by using the Service Discovery service.

The Authorization system guarantees that a service can only be accessed by an authorized consumer. It consists of two service producers and one service consumer and it maintains a list of access rules to system resources (i.e. services). The AuthorizationManagement service provides the possibility to manage the access rules for specific resources. The AuthorizationControl service provides the possibility of managing the access for an external service to a specific resource. The Authorise service enables an application (or the Orchestrator system) to request access to another producer service, which is checked against the rules previously configured.

The Orchestration system is the workhorse of the Arrowhead Framework, being used to control how systems are deployed and in what way should they be interconnected. Orchestration can be viewed as the system that supports the establishment of all other systems by providing coordination, control and deployment services. The Orchestration system offers two services, one, the OrchestrationStore, that stores the connection requirements by analyzing systems models and another, the OrchestratioManagement which executes look-up and service provisioning. These systems are also capable of contacting other non-mandatory services, like the QoS Manager in order to configure the QoS requirements of services. Figure 3 highlights the Arrowhead core services.
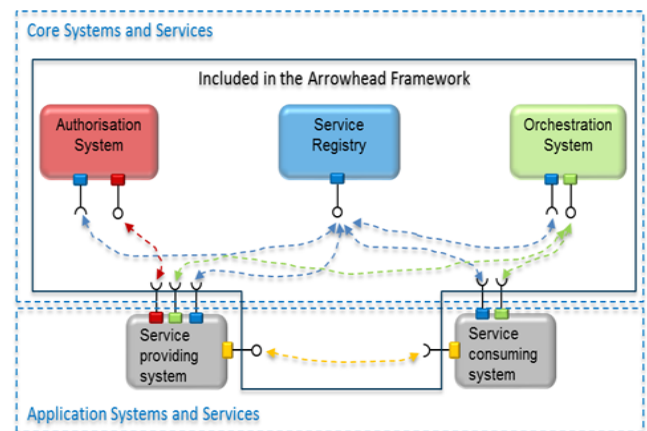

Figure 3 – Arrowhead core services

V. THE PROTOTYPE

The prototype was developed with the intention to be part of the Arrowhead Framework, namely, as a service which in turn would be provided by the prototype, i.e. software system. This choice can provide the capability of a secure remote access to the prototype, and increase its compatibility since Arrowhead services have an unambiguous and well documented definition of their interfaces.

The prototype was designed and implemented taking into consideration the OSA-CBM and MIMOSA CRIS. More specifically the layer of the OSA-CBM is the one part of the condition monitoring and the database model of MIMOSA CRIS that highlight the condition monitoring as well. The choice of a specific programming language is normally guided by some real-world constraints like cost, availability, training, and prior investment [31]. All things equal, other aspects become relevant. For instance, Python programs are generally expected to run slower than Java programs. However, they also take less time to develop, since two factors concur in assisting the development phase.

First of all, the structure and syntax of the language were studied to maximise the correctness of an executable program. Moreover, concerning languages such as Java and C++, there is a dramatic difference when it comes to the length of the

developed code, both when considering the Line of Codes (LoC) metric and the total number of characters. In fact, current heuristics consider that Python code is typically 3-5 times shorter compared to Java and 5-10 times shorter than equivalent C++ code [32]. Moreover, languages can be combined to be able to harvest the positive aspects of each one. The large user base of Python is another advantage of this language, together with the fast-growing code base of available software. Being quite different from the traditional languages (C, C++, Java, PERL, etc), the developers tend to be quite polarized when Python is involved, and they either love it or hate it, leading to stronger commitment to the Python code base, with benefits spanning from more robust code, to better-documented libraries. When hybridization of programming languages is taken into account, different modules can be developed in, for instance, Java and C++, and be combined as part of a software application in Python. In addition, Python can also be used to prototype components until their design can be "hardened" in a Java implementation [31]. Bottom line, usage of Python is on the rise in a large number of application areas, and both the industry and the academia are investing a lot of efforts with this language, especially to develop open source solutions. In this case, the prototype was developed in Python. This becomes handy when incorporating the prototype into the Arrowhead Framework, which is a language that is, supported for interacting with the services.

The work aimed at closing the gap between what has been done within other application areas in Python (big data applications, machine learning, pattern recognition) and monitoring of industrial machinery, with a particular focus on REBs. In this section, are some of the relevant and most widely accepted REBs fault diagnostic and analytics presented. The REBs fault signal was simulated. In the simulation, the influence of unbalance, and misalignment together with noise were also included. In the Figures 4 – 6. Figure 4 is the time series of the signal presented and in Figure 5 the amplitude spectrum. Next in Figure 6 the Wavelet "Haar" with de-noising "soft". Even though the solution that was developed is still at embryonal phase, the work leveraged on existing strategies to make a more manutenable and extendable code, and mature existing open source libraries were used as a computational platform to design, for example, a high-performance big data analytical module. The developed parts can with less effort be added into the Arrowhead Framework, and can also be part of a chain of services that are called together to provide more complex capabilities, such as the automatic generation of supply requests, or of request for urgent repairs, when the current prototype identifies a critical condition of REBs. The resulting code was able to replicate the data analysis strategies presented in the first part of this section, and to scale them to much larger data. In fact, the code is able to apply the techniques, such as the Waveler de-noising algorithm", Time series and Fourier Transform. Currently, the code is ready to be merged within existing tools, as mentioned above, to add data analysis capabilities for REBs to small industry actors that decide to apply open source philosophy, python, and other aligned strategies to their maintenance business.
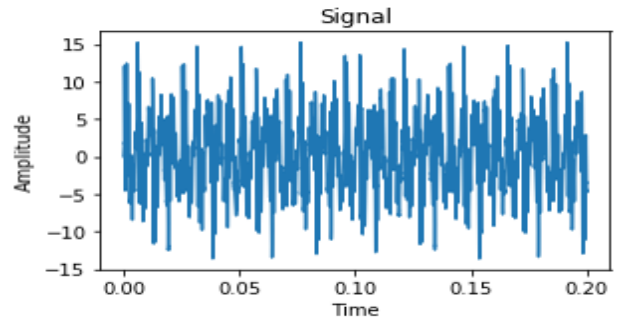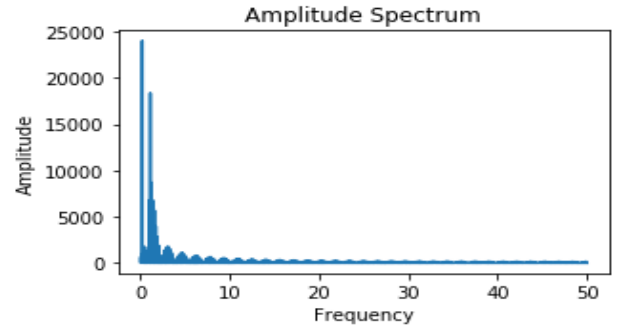

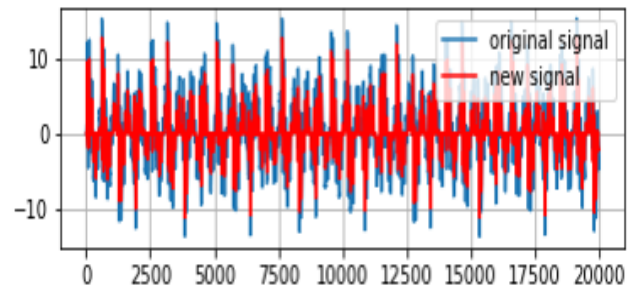*Figure 4 - Time series.*

Figure 5 - Amplitude Spectrum.

Figure 6 - Wavelet "Haar" with de-noising "soft".

VI. CONCLUSION

In this paper, an approach for fault diagnosis was presented which used Python as the underlying language. The open source solutions come with a large number of inbuilt libraries that can reduce the model usability and complexity. The approach ensured that the time for development is much shorter, thereby reducing the investment costs. This enables smaller companies with limited budgets to plan and execute such fault diagnosis systems in their facilities. Lower costs of implementation can be the required stimulant for small-sized companies to acquire the appropriate tools for their Condition Monitoring and asset management processes. In addition, the use of the Arrowhead Framework to be used in conjunction with the condition monitoring techniques as shown in the current paper provides an ecosystem that brings benefits into the monitoring and maintenance of geographically dispersed equipment, which are part of a local cloud. Thus, the benefits of the standardized Arrowhead Framework is that it provides interoperability

between the various systems part of the local cloud as well as incorporate them successfully for automation purposes.

REFERENCES

[1] Delsing, J., Varga, P., Ferreira, L. L., Albano, M., Puñal Pereira, P., Eliasson, J., ... & Derhamy, H. (2017). The Arrowhead Framework architecture. IoT Automation: Arrowhead Framework, 45-91.

[2] Turban, E., Sharda, R., & Denlen, D. (2011). Decision support and business intelligence systems (9th ed.). Upper Saddle River, NJ: Pearson Prentice Hall. 2011.

[3] Thurston, M.G. 2001. "An open standard for Web-based condition-based maintenance systems", 2001 IEEE Autotestcon Proceedings. IEEE Systems Readiness Technology Conference, 2001, p 401-15

[4] Jantunen, E., Hooghoudt, J. O., Yang, Y., & McKay, M. (2018). Predicting the Remaining Useful Life of Rolling Element Bearings. In IEEE International Conference on Industrial Technology, ICIT 2018 (pp. 2035-2040). Institute of Electrical and Electronic Engineers IEEE.

[5] Endrenyi, J., Anders, G. J., & Da Silva, A. L. (1998). Probabilistic evaluation of the effect of maintenance on reliability. An application [to power systems]. IEEE Transactions on power systems, 13(2), 576-583.

[6] Moubray, J (2005). Reliability Centred Maintenance, 2nd ed., Butterworth-Heinemann Ltd., Great Britain, 2005, ISBN: 0750633581.

[7] Thurston, M.G. 2001. "An open standard for Web-based condition-based maintenance systems", 2001 IEEE Autotestcon Proceedings. IEEE Systems Readiness Technology Conference, 2001, p 401-15

[8] Albano, M, Jantunen, E., Papa, G., Zurutuza, U. (2018). The MANTIS book. Cyber-Physical Sytem Based Proactive Collaborative Maintenance. River Publishers ed.

[9] Jardine, A.K.S., Lin, D. and Banjevic, D., (2006), "A review on machinery diagnostics and prognostics implementing condition-based maintenance", Mechanical Systems and Signal Processing, Vol. 20, pp. 1483–1510.

[10] Schwach, D. W., & Guo, Y. B. (2006). A fundamental study on the impact of surface integrity by hard turning on rolling contact fatigue. International journal of fatigue, 28(12), 1838-1844.

[11] Yoshioka, T. and Shimizu, S., (2009), "Monitoring of ball bearing operation under grease lubrication using a new compound diagnostic system detecting vibration and acoustic emission", Tribology Transactions, Vol. 52 (6), pp. 725–730.

[12] Mori, K. , Kasashima, N., Yoshioka, T. and Ueno, Y., (1996), "Prediction of spalling on a ball bearing by applying the discrete wavelet transform to vibration signals", Wear, Vol. 195, pp. 162–168.

[13] Shibata, K., Takahashi, A. and Shirai, T., (2000), "Fault diagnosis of rotating machinery through visualization of sound signals", Mechanical Systems and Signal Processing, Vol. 14(2), pp. 229–241.

[14] Heng, A., Zhang, S., Tan, A.C.C and Mathew, J., (2009), "Rotating machinery prognostics: State of the art, challenges and opportunities", Mechanical Systems and Signal Processing, Vol. 23, pp. 724–739.

[15] El-Thalji, I. and Jantunen, E., (2015), "A summary of fault modelling and predictive health monitoring of rolling element bearings", Mechanical Systems and Signal Processing, Vol. 60-61, pp. 252–272.

[16] Howard, I., Jia, S. and Wang, J., (2001), "The dynamic modelling of a spur gear in mesh including friction and a crack", Mechanical Systems and Signal Processing, Vol. 15, pp. 831–838.

[17] Wang, W.Y., (2002), "Towards dynamic model-based prognostics for transmission gears", Component and Systems Diagnostics, Prognostics, and Health Management II, Vol. 4733, Bellingham, pp. 157–167.

[18] Baillie, D.C. and Mathew, J., (1994), "Nonlinear model-based fault diagnosis of bearings", In Proceedings of International Conference on Condition Monitoring, Swansea, UK, pp. 241–252.

[19] Loparo, K.A., Adams, M.L., Lin, W., Abdel-Magied, M.F. and Afshari, N., (2000), "Fault detection and diagnosis of rotating machinery", IEEE Transactions on Industrial Electronics, Vol. 47, pp. 1005–1014.

[20] Loparo, K.A., Falah, A.H. and Adams, M.L., (2003), "Model-based fault detection and diagnosis in rotating machinery", In Proceedings of the Tenth International Congress on Sound and Vibration, Stockholm, Sweden, pp. 1299–1306.

[21] Yan, J., Koç, M. and Lee, J., (2004), "A prognostic algorithm for machine performance assessment and its application", Production Planning and Control, Vol. 15 (8), pp. 796–801.

[22] Vlok, P.J., Wnek, M. and Zygmunt, M., (2004), "Utilising statistical residual life estimates of bearings to quantify the influence of preventive maintenance actions", Mechanical Systems and Signal Processing, Vol. 18, pp. 833–847.

[23] Wang, W.Q., Golnaraghi, M.F. and Ismail, F., (2004), "Prognosis of machine health condition using neuro-fuzzy systems", Mechanical Systems and Signal Processing, Vol. 18, pp. 813–831.

[24] Dong, Y., Gu, W., Yang, K. and Zhang, W., (2004), "A combining condition prediction model and its application in power plant", In Proceedings of the Third International Conference on Machine Learning and Cybernetics, pp.26–29.

[25] Li, T., & Ogihara, M. (2005). Semisupervised learning from different information sources. Knowledge and Information Systems, 7(3), 289-309.

[26] Peña, M., Alvarez, X., Jadán, D., Lucero, P., Barragán, M., Guamán, R., Sanchez, V and Cerrada, M. (2017, August). ANOVA and cluster distance based contributions for feature empirical analysis to fault diagnosis in rotating machinery. In Sensing, Diagnostics, Prognostics, and Control (SDPC), 2017 International Conference on (pp. 69-74). IEEE.

[27] Pacheco, F., de Oliveira, J.V., Sánchez, R.V., Cerrada, M., Cabrera, D., Li, C., Zurita, G. and Artés, M., 2016. A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions. Neurocomputing, 194, pp.192-206.

[28] Zhang, J., Ma, W., Ma, L., & He, Z. (2013). Fault diagnosis model based on fuzzy support vector machine combined with weighted fuzzy clustering. Transactions of Tianjin University, 19(3), 174-181.

[29] Cerrada, M., Zurita, G., Cabrera, D., Sánchez, R. V., Artés, M., & Li, C. (2016). Fault diagnosis in spur gears based on genetic algorithm and random forest. Mechanical Systems and Signal Processing, 70, 87-103.

[30] Delsing, J. (Ed.). (2017). Iot automation: Arrowhead framework. CRC Press.

[31] Python: www.python.org/ (14-09-2018)

[32] Comparing Python to Other Languages: https://www.python.org/doc/essays/comparisons/(14-09-2018)

[33] F. Blomstedt, L. Ferreira, M. Klisics, C. Chrysoulas, I. de Soria, A. Zabaˇsta, B. Moris, E. Eliasson, M. Johansson, and Varga P.The Arrowhead Approach for SOA Application Development and Documentation. In Proceeding of the 40th Annual Conference of the IEEE Industrial Electronics Society, IECON 2014. IEEE, 2014