IoT based Vibration Analytics of Electrical Machines

D.Ganga and V. Ramachandran

Abstract— **The aim of this research is to propose an IoT based model for real-time condition monitoring of electrical machines, which addresses the challenges of data storage and scalability. The proposed model is evolved with an experimental setup having two sets of DC motor coupled to AC Generator and an IoT device to elucidate integrated monitoring and decision making. This IoT based vibration analytic model uses an IoT2040 Gateway with custom Linux OS image built for acquisition and streaming of vibration signals. The Python target application acquires DC motors' shaft vibration using vibration sensors and communicates the data as events to cloud through serial device driver interface. The IoT service running in cloud receives the data from multiple machines through lightweight RESTful HTTP and records the same which are retrievable for analysis and algorithm development in any platform. The retrieved data have been analysed using the proposed statistical classification based signal decomposition algorithm as well as time-frequency analysis to estimate the vibration thresholds of every machine connected to IoT cloud. Such estimated thresholds corresponding to different operating and environmental conditions maintained in cloud are used to build a repository of context specific solutions for machine conditions leading to improved maintenance decisions. The uniformity of threshold values obtained from IoT based model in comparison with that of analysis carried out on the machines locally using myRIO for data acquisition ensures the integrity of the proposed statistical classification algorithm and reliabilty of the IoT model for condition monitoring with assured scalability.**

Index Terms— **Electrical machines, Condition monitoring, IoT Gateway, Vibration Analytics, Signal Processing, Cloud.**

I. Introduction

Condition monitoring is the most predominant strategy used for predictive maintenance of machines. In any enterprise or industry, the objective of plant maintenance has always been to maximize the uptime and efficiency through better preventive or predictive maintenance and condition monitoring diagnostics so that the desired targets could be achieved with increase of revenue. At present, most of the condition monitoring systems are local systems, which collect vibration data from the machines and use various algorithms to check for defectiveness or unusual behaviour and compare the results with the knowledge base for effective decision making. This is the usual methodology adopted in many industries, which faces certain challenges such as inadequate storage space for data and especially scalability when multiple machines at different locations are to be monitored. The preciseness, volume, variety and analysis of the machine data are the major contributing factors for effectiveness in condition monitoring. High volume and variety of data to be collected from the machines at different locations during online monitoring for the interpretation of their behaviour at dynamic or abnormal operating conditions pose the challenges of data storage and scalability [1].

The practical challenges faced by maintenance engineers are the introduction of new technologies for the enhancement of plant productivity, methods of data acquisition and analysis, inconsistent outcomes and shortage of resources. The present condition monitoring systems possess advanced instrumentation that could acquire data at high throughput with less noise but lags in volume, variety and extent of data analysis. The preliminarily identification of the machine's abnormal behavior is carried out by comparing the measured value with the vibration severity limits prescribed in IS12075, Bureau of Indian Standards, 2008. The method is simple but lacks sufficient information to identify the behavioural patterns during dynamic conditions [2]. National Instruments while discussing the aforesaid challenges and benefits of fleetwide monitoring has cited that maintenance managers require innovative strategy for continuous and automated data collection from more industrial assets to make data comparison with baseline behaviour and analyse the performance using algorithms specific to application so that the maintenance and the real-time decisions are improved [3]. This kind of maintenance strategy could successfully be achieved by practising IoT based condition monitoring in cloud platform. Advantech in its white paper [4] has discussed on the importance of the implementation of cloud based SCADA system using Industrial IoT (IIoT) and points out that even though SCADA monitors the instantaneous conditions well within the enterprise, the adoption of cloud offers pervasive analytics and decisions additionally irrespective of the hardware used and thus making Industry 4.0 effective. It is observed that condition monitoring, a process which involves data acquisition, data processing and information extraction plays the lead role in bringing out successful diagnostics and prognostics. Mallikarjun Kande et al. [5] have extensively reviewed and discussed about existing machine condition monitoring techniques and industrial automation for plant-wide condition monitoring of rotating electrical machines, which includes machine diagnostics using artificial intelligence. They pointed out the importance for on-equipment, on-premise and on-cloud integration of condition monitoring and the Distributed Control System to provide continuous monitoring of the equipment with high update rates from the sensors, to collect and send sensor data to diagnostics running as part of plant operations and to offer the elasticity required for the data and computational resources. The effective implementation of real-time integration of various data acquisition devices demands lightweight and uniform communication standards. While discussing about on-equipment and on-premise integration methods, the need for on-cloud monitoring using IoT gateway has been substantiated to meet the requirements of advanced diagnostics and data platforms for enhanced computation. The integration of remote services gives the benefit of making intensive data analysis even with the application of basic data acquisition devices for condition monitoring.

Steve Lacey [6] has stated that condition monitoring carried out with the incorporation of cloud facilitates comparative analysis of the conditions of similar machines or related machines. The adoption of cloud allows data sharing and enables implementation of new analysis techniques when unknown signal patterns are observed at the user end. The cloud environment provides an added value of being able to share and compare the local machine condition data with other similar machines across the plant, or with other machines at multiple plants wherever they are located. The cloud based condition monitoring system can infer the data from the distributed databases for effective decision making in vibration analysis. The vibration data is further processed in the cloud with the combination of data of one machine and other similar machines' data with extensive analysis options. This increases the reliability of the diagnosis information for appropriate decision making. The perceptions of various industries on the adoption of cloud based condition monitoring have been portrayed by Sheila Kennedy [7]. The author has pointed out that Siemens has developed a cloud application for asset analytic services that receives high volume of physical and process data for analysis and generates alarms automatically under critical conditions. It is inferred that multiple access provided by the cloud environment to multiple condition monitoring experts improves the decisions for effective maintenance solutions. Fran Dougherty, CTO of the Worldwide Incubation Enterprise and Partner Group of Microsoft has outlined in the special report composed by Jim Montague [8] that industries look for innovation, scalability and business growth for which the use of private and public clouds has been appreciated. Hybrid cloud was considered to be the best option by him, as industries can choose the type of analysis dynamically as per the requirements.

Development of customized software layers based on the monitoring requirements and lightweight communication between cloud and the end user makes IoT devices to operate reliably with high speed and throughput so that performing data analytics meets the real-time requirements of operational decisions and seamless maintenance schedules for machines.

II. Experimental Setup and Data Acquisition

Out of various machine parameters namely vibration, humidity, temperature, pressure, sound, thermography, motor current, insulation resistance, electrical capacitance and electrical inductance, the choice of the parameter for condition monitoring depends on the type of industrial equipment and condition to be assessed (Hashemian et al. [9]). In order to improve the performance and uptime of electrical motors, the condition of each machine is monitored and assessed by observing the input electrical variable such as current as in Motor Current Signature Analysis (Mehala et al. [10]) or the mechanical parameters such as acceleration, velocity, displacement as in vibration analysis (Asoke Nandi et al. [11]). Measuring vibration is the widely used condition monitoring technique for detecting the faults and diagnosing the equipment behavior. It is proposed to use an IoT Gateway to acquire the vibration data from multiple machines. The IoT device can communicate using different protocols such as MQTT, XMPP, DDS, AMQP and HTTP, each of which follows specified format and mode of data communication. The high-level application, which is developed in the IoT enabled gateway collects the machine's physical data and automatically performs the task of transmitting the acquired data to the cloud more effectively with less programming overheads than conventional embedded systems. The present work illustrates the process of building a customized Linux OS image for embedding into IoT2040 Gateway, on which the required Python device drivers and application logic are run to acquire the data from the vibration sensors mounted on the shaft of the DC motors. The acquired data are sent to cloud through RESTful service developed in Python which uses lightweight RESTful HTTP protocol for communication. The HTTP protocol has the advantages of creating, updating, deleting and retrieving the resources from IoT Cloud service with the options of compressing headers and obtaining response as acknowledgement.

The IoT based framework proposed for machine vibration monitoring at enterprise level has been depicted in Figure 1 and implemented on the experimental set up shown in Figure 2. IoT based processing is adopted for condition monitoring of multiple machines operating at different locations as it evolves as a better choice due to the attributes of cloud storage, flexible application development, data aggregation, scalability and platform of multiple services. The proposed framework will enhance the machine condition monitoring functionality with methodologies of scalable and platform independent data aggregation and collaborative analysis that the real-time industrial applications demand extensively. The experimental set up consists of two similar sets of machines having DC shunt motor coupled to three phase AC Generator and a SIMATIC IoT2040 Gateway. The vibration signals have been acquired during the motor is started and ran to the rated speed of 1500 rpm at no load condition and then loaded by AC Generator at fixed load changes. To analyze the effects of industrial environment on shaft vibration, a 3-phase squirrel cage induction motor placed in the proximity of DC motor is made to run at constant speed of 1500 rpm and the shaft vibration data of the DC motor is acquired as carried out for standalone condition. The acquired vibration data under the operating conditions of starting to no load speed with and without external disturbance and loading are streamed to cloud through IoT2040 gateway. Similar experimentation has been carried out for acquiring the shaft vibration data using myRIO-1900 [12] as acquisition device and tri-axial accelerometer (ADXL345) as vibration sensor. The ADXL345 mounted on the rigid structure supporting the DC motor's rotor shaft senses the shaft vibration which is acquired by my-RIO in fast data transfer mode of I2C (400 kHz) with output data rate and bandwidth as 800 Hz and 400 Hz respectively. The ADXL345 is used in 13-bit resolution at measurement range of ±16g with sensitivity of 256 LSB/g. The data acquired in both cases have been analyzed using a statistical classification algorithm developed in LabVIEW DIAdem [19]. The algorithm extracts the major amplitude levels of non-stationary vibration oscillations and clusters the determined levels for precise enumeration of vibration thresholds at dynamic operating conditions.

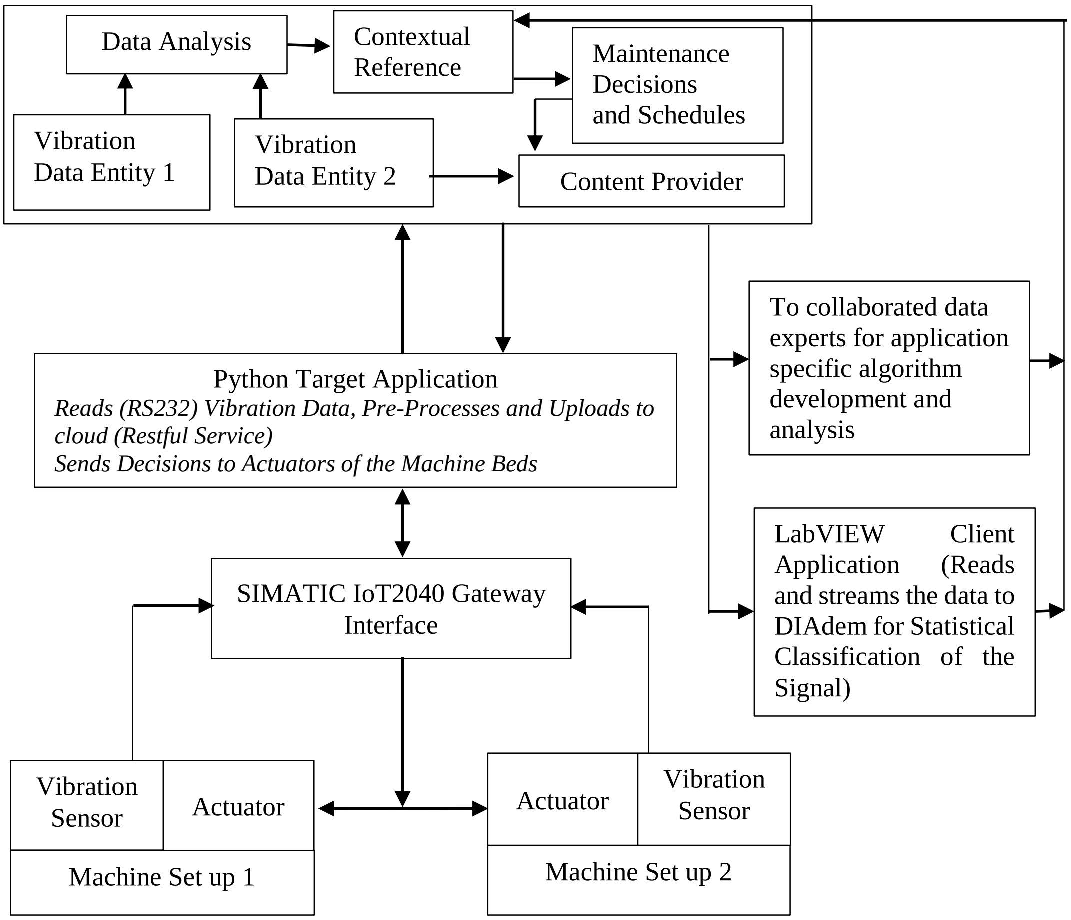

Figure 1. The IoT based condition monitoring model

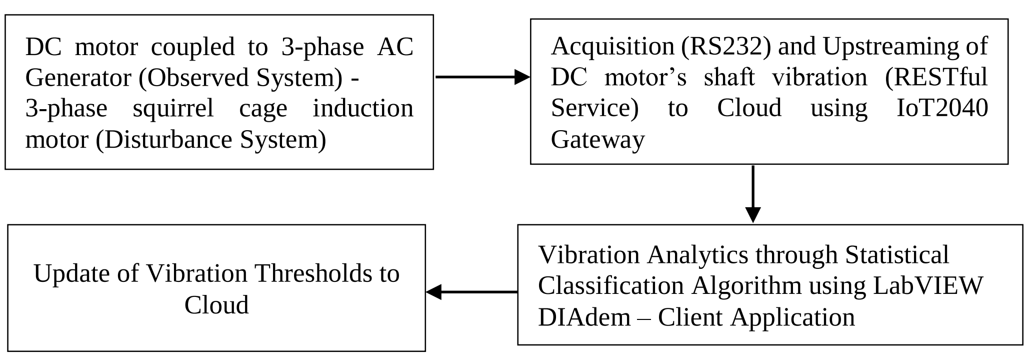

Figure 2. Experimental Set Up

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

III. Implementation of IoT based Condition Monitoring Model

Building a custom host OS Image for the development of Python interface to acquire data through the SIMATIC IoT2040 gateway is a challenging task and has been detailed as follows. The bottom up approach to build the custom Yocto Linux image that boots the IoT Gateway is depicted in Figure 3. SIMATIC IoT2040 is an Intel Quark X1020 based System on Chip Industrial IoT gateway which runs with Linux Open Embedded Core OS. It supports external RS232 / RS422 / RS485, Ethernet, USB and internal Arduino shield, Mini PCIe card hardware interfaces [13].

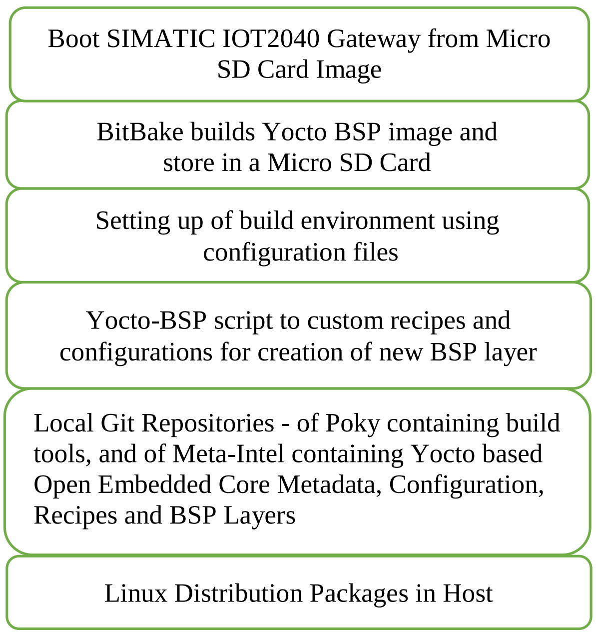

Figure 3. Development of Custom Host OS image for IoT2040 Gateway

The custom Open Embedded Linux core with specific hardware configuration as per application requirements facilitates flexible and faster operational features. The IoT2040 gateway utilized to monitor the vibration signals is operated on open source Linux platform (Yocto Linux image obtained from BitBake build process) with custom Board Support Package (BSP) optimized for Intel Galilieo development boards. The BSP contains directory of file structure that specifies about its hardware features, kernel configuration namely "standard, tiny or preempt-rt" and all the additional supporting hardware platforms and drivers. The BSP does not possess build system rather it contains information only about the hardware with a task executor and scheduler (BitBake) of an Embedded Linux build system. These are available in Git repository and cloned as local copy in the host project using "git clone git://git.yoctoproject.org/poky". The host build process parses the metadata of recipes, classes, and configuration files and builds hardware specific binary output that run on specific hardware or on Quick Emulator *(www.yoctoproject.org)*. The BitBake build process using either 'Native build' or 'Docker build' yields kernel configuration, tools and furnishes a bootable SD card image *(github.com)*. This layer built on the top of "meta-iot2040-bsp" provides services to exploit the features for application development in IoT2040. The application specific components such as drivers and cloud protocols available under the host OS are added in this layer. The image thus built boots the IoT2040 with the preconfigured IP address for an Ethernet interface.

The SDK installer script specific to the custom OS image has been run to install the toolchain, which is a collection of hardware specific cross-compilers, linkers and debuggers running on a target architecture that also supports development of software compatible with other target architectures. The environment setup script for the SDK with a configuration file, version file and root file system (sysroots) for the target system is also to be run to enable application development and deployment in IoT2040 platform. IoT2040 thus booted with custom embedded Linux OS image has been used in the proposed model to develop an application for machine condition monitoring.

The IoT2040 Gateway application development and software commissioning are carried out using remote desktop tool called MobaXterm *(mobaxterm.mobatek.net)* which connects the IoT device with a PC through Secure Shell (SSH) session. The IoT2040 Gateway and the PC network settings are configured to be in the same subnet using the command "nano/etc/network/interfaces" which opens a network configuration file with details as given below:

iface eth0 inet static
address xx.xx.xxx.xxx
netmask xxx.xxx.xxx.x
gateway xx.xx.xxx.x

After editing the values of the fields viz., address, netmask and gateway appropriate to the connected network, the MobaXterm identifies the IoT2040 Gateway through the newly configured IP and opens the Linux platform for application development through SSH. Any software package required for the application development can be installed using package manager, in addition to the Linux image built with Board Support Package. The package manager of Yocto, "opkg" is used to install the packages downloaded from the Intel or Git repositories.The application developed in Python not only acquires data through RS232 interface but also uploads the vibration signal data to the cloud using RESTful services. To read the vibration data from RS232 hardware interface of IoT2040 Gateway, it requires serial package compatible for Python. The installation of Python serial package requires virtual environment or Python Installer Program, "pip" which invokes the system to build the desired package i.e., the Python serial package called "pyserial" *(pip.pypa.io)*. The piezo electric vibration sensor fixed with magnetic mount on each of the rotor shaft's rig senses the shaft vibration and sends the raw data through RS232 interface to IoT2040 Gateway. The Python application reads the vibration data as 'x' and 'x1' from the shafts of two DC motors through serial communication interface of IoT2040 and processes the serial data of 16 bits length to convert the raw values into vibration in 'g'. The IoT device streams the vibration data to cloud through RESTful HTTP Request / Response communication. Any cloud that is running IoT service requires the client to send the unique API keys that are generated for read or write operations. The vibration data to be updated in the cloud are saved as 'parameters' along with the Write API key using the method available in urllib Python package,

parameters = urllib.urlencode ({'field1':x, 'field2':x1, 'key': Write API key})

The HTTP connection from the application running in IoT2040 Gateway to the resource in the IoT cloud service has been made by referring to the end point of the resource containing the address url and port number 80 as given below:

httplib.HTTPConnection ("cloud resource end point")

Consequently, while making HTTP Request, the POST method sends the vibration data, x and x1 saved as 'parameters' together with the headers that define the response type, data type and encoded features.

headers = {"Content-typZZe": "application/x-www-form-urlencoded", "Accept": "text/plain"}
conn.request("POST", "/update", parameters, headers)

Having established the network connection, the vibration data sent to the cloud is visualized as chart history and saved in data fields. The vibration data received over a period are stored as historical data in the cloud platform. The IoT service processes the vibration data online and enables other collaborated data experts in remote location to get the data shared through content provider as shown in Figure 1 to develop and execute application specific algorithms that provide more meaningful insight into data. In the proposed model, the vibration data present in the data fields of cloud are retrieved by LabVIEW client application developed using RESTful VIs of HTTP Client palette [14]. The LabVIEW application sends request to the cloud with authentication details of username and password through Open Handle VI of the palette. In addition, this VI opens a client handle which allows multiple requests and responses between the application and IoT service using the same credentials, thus provisioning for scalability. The GET VI uses GET HTTP method and combines the client handle, Read API Key, URL of the vibration data, number of data entries for retrieval while making the Web request to the cloud API end point. While running the application, this VI gets the Headers and Body from the cloud service of which the Headers contain the details such as protocol version, content length and meta data while the Body contains the vibration data in JSON format.

IV. Proposed Classification Technique for Non-Stationary Vibration Signal Analysis

Effective monitoring of vibration is the major criterion for precise identification of machine behavior specific to the type of physical component, environment and operating conditions. P.J.Tavner [15] has stated that the vibration signal analysis seems to provide comprehensive and reliable condition monitoring subject to availability of high data rate and advanced analytic techniques. It has been reviewed that the conventional spectral analysis remains suitable when the machine maintains a constant speed for substantial amount of time. In the cases where machine speed changes or when the machine is fed by electric drive with inbuilt harmonics, the complexity that is endured in capturing and interpreting the spectral content of signals having high bandwidth and low signal-to-noise ratio, demands the application of multi-parameter or soft-computing or effective non-stationary signal processing techniques. Rakesh et al. [16] performed online condition monitoring of induction motor through Motor Current Signature Analysis, which identifies frequencies corresponding to faults using the current spectrum and this method is said to lag during varying load torque conditions. As the signals are non-stationary, the characterization of the signal and the classification of the machine states are challenging tasks for condition assessment under dynamic load and speed variations. Due to the limitations observed in the Time Frequency Representation (TFR) and wavelet based TFR, Cardona Morales et al. [17] have proposed the application of Linear Frequency Cepstral Coefficients (LFCC) and Spectral Sub-Band Centroids (SSC) on time frequency response as a measure of reducing the feature loss in the estimation. The one class classifier applied on such extracted features has been said to give better classification of machine states under non-stationary operations. The results of the above stated work substantiate the occurrence of frequency interferences in time frequency response during the estimation of dynamic features.

From the Gaussian distribution of raw vibration data, the values of the mean and standard deviation are calculated. Jablonski et al. [18] have assessed the nature of vibration from the range of values falling in between the multiples of standard deviation determined from Gaussian distribution. Generally, the real time data do not take Gaussian distribution always and hence the calculations lead to false alarms. Thus different distributions such as Weibull probability distribution, generalized extreme value probability distribution, extreme value probability distribution and inverse Gaussian probability distribution are used instead of Gaussian distribution to characterize the vibration data for threshold fixation. Various such works portray different methodologies have been developed for condition monitoring, where focus is more towards the diagnosis of abnormalities from the available data nature than the threshold estimation adaptive to operating conditions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

The threshold calculations in condition monitoring are more important but are not given due consideration. Thousands of false alarms are generated due to adoption of default threshold levels. Unless thresholds are estimated precisely, the criticality of the abnormal conditions could not be realized to the fullest extent.

A statistical classification based signal decomposition algorithm is proposed for identification of denser vibrating regions dynamically under various machine operating conditions and thereby to enumerate adaptive thresholds for quick and accurate prediction of abnormalities. The vibration signal data received as JSON string is unflatten to actual values and has been segmented into classes of equal width over the range of maximum and minimum amplitudes. The data read from the cloud are streamed to DIAdem [19] as .tdms file for carrying out statistical classification of the vibration signal into 'n' number of classes and obtain the transition matrix which is fed as input to the signal decomposition algorithm developed in LabVIEW to identify the vibration thresholds. The proposed signal decomposition algorithm determines the vibration oscillations at multiple levels of the signal amplitude using the transition matrix obtained through statistical classification. The variations in the range of maximum number of vibration oscillations within the scope of segmented classes, which have been observed with respect to the operating conditions of starting to no load speed and loading along with environmental disturbances reveal the significance of computing the thresholds dynamically. The technique further traces the changes in signal transitions at every level of class accurately which helps to illustrate the machine behavior. Also, it attempts to identify the behavioral changes by determining the oscillations starting from a class and ending at the same class or at different class in any of the direction either upper or lower, which will aid to extract the useful information from the random signal that will be a significant indicator for condition monitoring using vibration analysis. Thus, the deceptive thresholds that hide the incipient changes in the behavioral pattern are clearly outlined, resulting to effective condition monitoring.

Using amplitude classification analysis based on transition matrix [19], the nature of the vibration of the machine is determined by extracting the oscillatory information at any of the amplitudes that the vibration has taken. The transition matrix shown in Table 1 contains the number of signal transitions from one class to every other class, where ST_{kn} and ST_{nk} represent Signal Transitions from class k to class n and class n to class k respectively. For 'n' number of classes, the n n transition matrix consisting of n^2 elements represent the transitions from the Start Classes (indexed in rows) to Target Classes (indexed in columns). Positive slope represents transition of the signal from a lower class to higher class and vice-versa for the negative slope. Thus, the upper diagonal matrix indicates the counts of positive slopes and lower diagonal matrix corresponds to the counts of negative slopes. In general, any row 'k' of the transition matrix gives the transitions of the signal from the class corresponding to the row '*class k'* (Start Class) to other [*n-1*] classes (Target Classes) and the same applies to any column 'k'. The sum of upper and lower off-diagonal elements along each column have been referred as level crossing counts of positive and negative slopes.

TABLE 1. TRANSITION MATRIX

		Target Class				
		Class 1	Class 2	Class 3	Class k	Class n
Start Class	Class 1	ST_{11}	ST_{12}	ST_{13}	ST_{1k}	ST_{1n}
	Class 2	ST_{21}	ST_{22}	ST_{23}	ST_{2k}	ST_{2n}
	Class 3	ST_{31}	ST_{32}	ST_{33}	ST_{3k}	ST_{3n}
	Class k	ST_{k1}	ST_{k2}	ST_{k3}	ST_{kk}	ST_{kn}
	Class n	ST_{n1}	ST_{n2}	ST_{n3}	ST_{nk}	ST_{nn}

Positive Slopes

Negative Slopes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

In the proposed vibration analysis technique, corresponding row-wise and column-wise class transitions have been considered to calculate the oscillations happening at every class i.e., to extract the actual vibration pattern of the shaft from the signal transition data described by the transition matrix by progressing along the columns and rows with diagonal elements as reference.

While analysing the transition matrix (Table 1) along its rows, the elements right to the diagonal element gives the transition of the physical signal from a class referred by the diagonal element to higher classes. Similarly, the elements left to the diagonal element gives the signal transitions to the lower classes. Hence the class ‘1’ of the transition matrix (represented by first diagonal element) will have only positive transitions to higher classes and the last diagonal element representing the class ‘n’ will have only negative transitions to lower classes. The classes in-between referred by the corresponding diagonal elements, possess signal transitions from one class to all the other higher classes as per the positive slopes and all the lower classes according to the negative slope values. The sum of positive slopes of each row gives the total number of transitions made by the signal from the respective class ‘k’ to the upper classes and the sum of negative slopes of each row gives the total number of transitions from class ‘k’ to the lower classes.

In the column-wise perception of the transition matrix, the elements above the diagonal element of every column reveal the signal transitions from lower classes to the class represented by positive slopes and the elements below gives the signal transitions from higher classes to the class as per the negative slopes. Thus Class ‘1’ (having only negative slopes), has transitions only from higher classes and class ‘n’ (having only positive slopes) has transitions only from lower classes. The in-between classes referred by the respective diagonal elements, have signal transitions from lower classes to class as per the positive slopes and from higher classes to the class as per the negative slopes. The sum of positive slopes of every column gives the total number of signal transitions from the lower classes to respective class ‘k’ and the sum of negative slopes gives the total number of transitions from upper classes to class ‘k’.

V. Algorithm

An efficient algorithm is proposed [20] to identify the shaft vibration patterns quickly and precisely that will lead to various condition monitoring decisions such as fixation of adaptive thresholds for various operating modes at normal conditions, tracing the abnormality patterns accurately from the shifts of oscillations to different class levels and measuring the intensity of abnormality from the range of shifts. The calculation framework for the determination of oscillations in the real time non-stationary vibration signal at multiple class levels is detailed below:

Step 1: Determine the maximum and minimum amplitudes of the shaft acceleration signal of the DC motor and choose them as the initial and end points of classification.

Step 2: Divide the amplitude of the shaft vibration signal in the range of selection into classes of considerable width and obtain the transition matrix and class mean of the classification.

Step 3: From the transition matrix extract the column-wise positive slopes (ST_{ij}) and row-wise negative slopes (ST_{ji}) pertaining to each class represented by the diagonal element and compare every pair of values that a class possess to identify the lowest value (OSC_{ji}) as shown in Equation 1. Every lowest value (OSC_{ji}) will represent the oscillations completed between corresponding class j and the lower class i.

$$OSC_{ji} = min(ST_{ij}, ST_{ji}) \qquad (1)$$

Step 4: Determine the difference between the positive and negative slopes pertaining to each class represented by the diagonal element of the transition matrix and record the resulting positive difference and negative difference values as separate matrices. The positive difference values DS_{ij} (Equation 2) represent the excess positive slopes to the class j, which refer to the existence of oscillations which have started from a lower class but have not ended at the same lower class. The negative difference values DS_{ji} give the excess negative slopes of the class j, which represent the existence of those oscillations ending at a lower class but have not started from the same lower class.

$$DS \;\; = \mathrm{Sub}(ST_{ij}, ST_{ji}) \dots \;\; DS_{ij} \; if \; DS > 0,$$

$$DS_{ji} \; if \; DS < 0 \qquad (2)$$

The oscillations between a class and the lower classes are calculated using the column-wise upper diagonal and row-wise lower diagonal elements of the transition matrix. Likewise, the oscillations between a class and upper classes are computed using column-wise lower diagonal and row-wise upper diagonal matrices. The proposed condition monitoring is based on the feature, Similar Amplitude Oscillations of non-stationary vibration signal that happen between every class and the lower classes. This methodology has been applied to classify the vibration pattern by determining the following aspects:

- Classes having higher number of signal transitions with lower and / or upper classes.
- Dominant classes having comparatively higher number of oscillations with lower, upper or both the classes.
- Class-wise oscillation distribution of the dominant classes.
- Identification of oscillation percentage of the dominant classes in respect of total oscillations and
- Clustering of dominant classes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

The classes which are identified to have comparatively higher number of oscillations with lower classes are considered as dominant upper classes. Such dominant upper classes which cumulatively account for 90 to 95 percent of the total oscillations are clustered as upper threshold classes. The distribution of the oscillations pertaining to all the dominant upper classes with respect to every lower class identifies those lower classes with which dominant upper classes make higher percentage of oscillations and are clustered as lower threshold classes. The shifts in the upper and lower threshold classes reveal the change of operating conditions or abnormalities within a specific operating condition. The range of deviations determines the intensity and criticality of abnormalities. This identification of cluster of classes as upper and lower thresholds adaptive to the operating nature of the machines enables accurate fixation of the vibration reference levels for condition monitoring. The erroneous conditions could be tracked precisely with the determined adaptive levels and hence the possibilities of incorrect failure diagnosis due to misleading thresholds could be overcome.

This non-stationary vibration analysis algorithm has been integrated with IoT service through LabVIEW client application to enable collaborated real-time condition monitoring of any machine whose data are streamed to cloud. The analysis results updated to the IoT service running in cloud lead to efficient decision making in machine condition monitoring and make the maintenance of other connected devices / machines automatic and effective. The updated results create contextual vibration references for assessing the condition of any other machine of same type that has been exposed to similar operating conditions.

VI. Results and Discussion

To comprehend the effectiveness of the vibration thresholds identified from the IoT based data analysis, a comparative analysis has been made with the results of vibration data acquired through myRIO. The above stated algorithm is implemented on the signals acquired in real-time to perform vibration analysis on the DC motor shaft when the motor is running under the following operating conditions:

- Starting to no load speed and Loading at standalone condition
- Starting to no load speed in the presence of the mechanical disturbance injected using a three-phase induction motor by running it under constant speed in the neighbourhood of DC machine

A. From Starting to No Load Speed – Standalone Condition

The shaft vibration signal of DC motor pertaining to the operating condition of starting to no load speed (standalone condition) has been acquired from the tri-axial accelerometer ADXL345 with sensitivity of 256 LSB/g by myRIO application developed using LabVIEW FPGA and RT programming. The acquired data are logged in an excel file and imported to NI DIAdem for statistical classification. The vibration signal pertaining to this mode holds the maximum amplitude of 368 LSB and minimum amplitude of -286 LSB, which ranges to the value of 654, has been divided into 12 classes with class width of 54.5. From the oscillations determined between every class and its lower classes by applying the above algorithm, the dominant classes which make comparatively higher number of oscillations with the respective lower classes have been identified. These dominant classes constituting to 91 percent of cumulative oscillations have been clustered to form the upper threshold class cluster for this operating condition. Similarly, for every class in the upper threshold class cluster that makes 65 and above percentage of oscillations cumulatively with its lower classes are extracted to constitute the lower threshold class cluster. The vibration data during the same operating condition are acquired by IoT2040 gateway (as shown in Figure 4) from a Piezo electric sensor through serial interface and transferred to cloud simultaneously. The data measured in ‘g’ are retrieved by LabVIEW client application and post multiplied by 256 LSB/g (sensitivity of the ADXL345) to map the IoT data scale in g equal to that of myRIO data scale in LSB. The scaled IoT data holds the maximum amplitude of 370 LSB and minimum amplitude of -285 LSB, which ranges to the value of 655. The signal of this range has been divided into 12 classes with class width of 54.58. The vibration threshold class clusters determined for the data acquired through IoT model as well as myRIO model using the statistical classification based signal decomposition algorithm are furnished in Table 2.

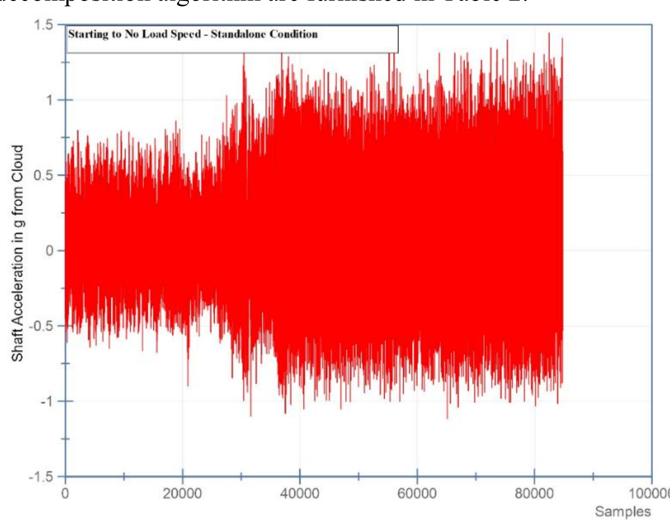

Figure 4. Shaft Acceleration acquired by IoT Gateway – Standalone Condition

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

TABLE 2. THRESHOLD CLASS CLUSTERS OF SHAFT VIBRATION SIGNAL - STARTING TO NO LOAD SPEED

Standalone Condition		**Disturbance Condition**	
IoT Based Data Analysis	myRIO Based Data Analysis	IoT Based Data Analysis	myRIO Based Data Analysis
Upper Threshold Class Cluster			
{15.20, 69.79, 124.37, 178.95, 233.54}	{13.88, 68.46, 123.05, 177.64, 232.23}	{0.03, 53.67, 107.32, 160.96, 214.60}	{-1.68, 51.88, 105.46, 159.03, 212.61}
Lower Threshold Class Cluster			
{-148.54, -93.95, -39.37}	{-149.88, -95.29, -40.70}	{-160.89, -107.25, -53.60}	{162.41, -108.84, -55.26}

Thus, the occurrence of faults or any abnormality at this operating condition can be diagnosed precisely with the shift in the oscillation percentages of the denser class regions and threshold class clusters from the predetermined values.

B. From Starting to No Load Speed with External Disturbance

The impact of the mechanical disturbance on the DC motor shaft vibration pattern has been examined by implementing the analysis on shaft vibration signals acquired during external disturbance condition using IoT based data acquisition as well as using LabVIEW with myRIO based system. In both cases the results have been compared with the shaft vibration pattern measured under starting to no load speed at standalone condition. The vibration acquired by myRIO during disturbance holds maximum amplitude of 453.6 LSB and minimum of -296.3 LSB while the signal from IoT2040 gateway holds the maximum and minimum amplitudes as 456 LSB and -295 LSB respectively. Both the data are segmented into 14 classes with class width of 53.5. Using the transition matrix which has resulted out of the classification and the algorithm proposed, the oscillations existing between every class and its lower classes are calculated and the dominant classes with more percentage of oscillations measured during the presence of external disturbance have been identified to form the upper threshold class cluster. To form the lower threshold class cluster, every class of the upper threshold class cluster that has made 65 percent or more number of oscillations cumulatively with its lower classes are considered and the results are tabulated (Table 2). This investigation brings out the changes that had happened in the vibration pattern due to the disturbance and discloses the fact of fixing adaptive condition monitoring threshold for a machine when exposed to external disturbances at a particular operating condition.

The criticality of the disturbances can be observed by measuring the range of shifts from the limits of the upper and lower threshold class clusters estimated during standalone condition.

C. Load Changes at Standalone Condition

The vibration signal acquired by myRIO during the load changes made at standalone running condition of the DC machine is shown in Figure 5, which possesses the maximum amplitude of 389 LSB and minimum amplitude of -292 LSB. Similarly, the vibration data corresponding to this operating condition acquired by IoT Gateway in 'g' are depicted in Figure 6, where the maximum and minimum amplitudes of the signal are 394 LSB and -291 LSB respectively. The multiple class level analysis is carried out by segmenting both categories of vibration data (acquired using myRIO and IoT based devices) into 13 classes between the maximum and minimum amplitude levels with class widths of 52.3 and 52.6 respectively. The results of upper and lower threshold class clusters obtained from the implementation of the signal decomposition algorithm are furnished in Table 3.

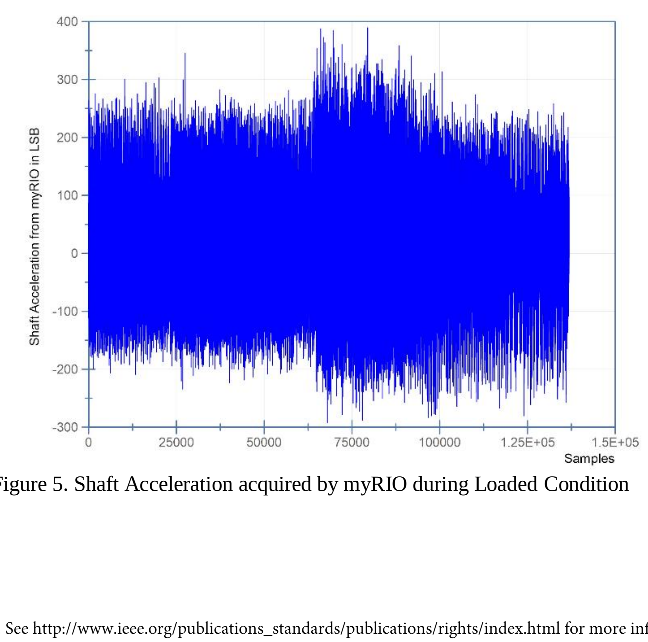

Figure 5. Shaft Acceleration acquired by myRIO during Loaded Condition

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

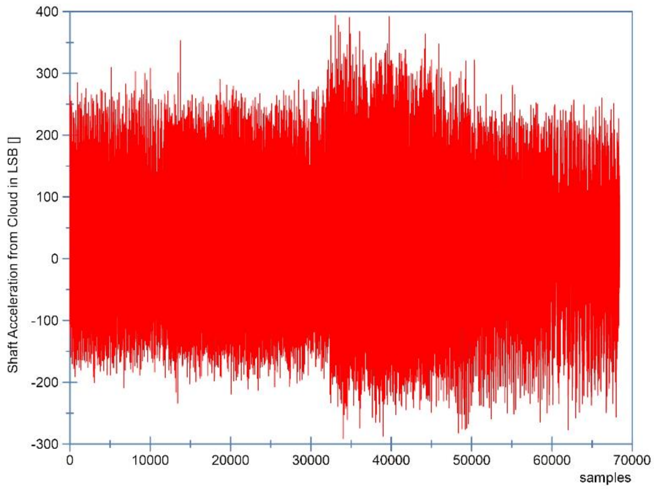

Figure 6. IoT based acquisition of Shaft Acceleration during Loaded Condition

The observed thresholds during loading at standalone condition imply that the external disturbance during starting to no load speed condition has created a vibration effect on the machine shaft equivalent to the loading at standalone condition. This analysis helps to prescribe about the setting of operational constraints for machines in real-time applications so that the machine performance and lifetime can be improved.

TABLE 3. THRESHOLD CLASS CLUSTERS DURING LOAD CHANGES MADE AT STANDALONE CONDITION

IoT Based Data Analysis	**myRIO Based Data Analysis**
Upper Threshold Class Cluster	
{ -0.76, 52, 104.76, 157.53, 210.30}	{-3.79, 48.67, 101.13, 153.60, 206.07}
Lower Threshold Class Cluster	
{-159.07, -106.30, -53.53}	{-161.18, -108.72, -56.25}

The upper and lower threshold class clusters of DC motor's shaft vibration determined from the analysis of the data acquired by IoT Gateway and myRIO are furnished in Tables 2 and 3, which define the scope of the amplitude levels between which the majority of shaft vibrations oscillate during the specified operating conditions. The incipient faults or abnormalities during any of the operating conditions can be diagnosed precisely by analysing the margin of deviations in the threshold class clusters. The considerable shift from the threshold values which are estimated during the operating mode of starting to no load speed under standalone condition with same spread pattern reveals the existence of continuous and constant disturbance. These distinct deviations in the upper and lower threshold clusters demarcate the standalone and disturbance conditions which are unseen in the measured values of DC armature current. The proposed analysis when implemented on the vibration signal corresponding to the loading condition has precisely brought out the intrinsic effect of mechanical disturbance which causes the motor shaft to vibrate equivalent to that of loading.

VII. COMPARATIVE STUDY OF THE PROPOSED VIBRATION ANALYSIS WITH JOINT TIME-FREQUENCY ANALYSIS TECHNIQUES

To illustrate the effectiveness of the statistical classification algorithm for the real-time condition monitoring of electrical machines, the same set of vibration signals acquired at different operating conditions are analysed using Joint Time-Frequency analysis [21] in LabVIEW and are compared with. The results of Short Time Fourier Transform (STFT) applied with different windows and window lengths are shown in Figure 7 with the details of window, its length and dominant frequencies along with time index. The rectangular window chosen with length of 6000 and time step of 1500 has captured the frequency of only 25 Hz in both normal and disturbance conditions, and has not distinguished the change at all. However, the increase in the window length to 24,000 with overlap of 6000, has shown additional frequencies of 125 Hz and 75 Hz in normal and disturbance condition respectively. Having set these levels as thresholds for condition determination, further analysis carried out with window length of 85000 and overlap of 56000, shows the existence of 25-160 Hz in normal condition and 25-150 Hz in disturbance condition. This overrides previous threshold of 125 Hz in normal condition (160 Hz) and 75 Hz of disturbance condition (150 Hz) which leads to the condition of false alarms. This study has been done on the vibration signals of normal and disturbance conditions based on same window and different window lengths. An alternate perception of using different windows for the same window length has been outlined below:

Out of the four windows chosen for STFT analysis, only Flat top and Gaussian identify closely similar pattern of frequencies for all window lengths in normal condition, whereas the same windows show different patterns during disturbance condition. On the other hand, considering the frequency of 75 Hz determined by both windows (for length of 6000) as threshold will cause a false alarm for 125 Hz present in the normal condition, which is identified by the window length of 24,000. The occurrence of 125 Hz is observed at normal condition for both windows of length 24,000, whereas, the Gaussian window detects 150 Hz at disturbance condition and the same is unidentified by Flat top. From the examination, it is observed that STFT using Gaussian window finds the presence of peak frequencies of 75, 125 and 140 (in Hz) under normal condition and 125, 150 and 25 (in Hz) under disturbance for the window lengths of 6,000, 24,000 and 85,000 respectively. Moreover, the performance of Gaussian window is not so convincing when compared to Flat top for the window length of 85000 in disturbance condition. The results of analysis made using Gabor Transform on the same signal finds the frequency content ranging between 0-150 Hz at different time instants in normal condition which is uncaptured by Gaussian window in normal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2835724, IEEE Internet of Things Journal

condition. Under disturbance, the frequencies from 0-270 Hz have been observed which remain unidentified by STFT analysis.

Figure 7. Time Frequency Details of Vibration Signal during Starting to No Load Speed (STFT)

Such inconsistency in the features extracted by STFT and Gabor Transform for the same vibration signal creates ambiguity in the aspect of threshold fixation. An intensive feature extraction is required to fix the thresholds precisely and thereby make comparisons for condition monitoring. Comprehensive information of the real-time vibration signal is required for early detection of abnormalities. It is observed that the features extracted using the proposed technique are more appropriate to precisely identify and validate the changes in the non-stationary vibration signal due to disturbance and other operating conditions than time-frequency techniques which depend on selection of window and length as major factors. The information obtained in terms of oscillations at classified amplitude levels and density based threshold class clusters using the proposed technique provide detailed reference for analysis of non-stationary vibration signal at dynamic conditions.

VIII. Conclusion

In either case of analysis based on myRIO or IoT device, the investigation uniformly brings out the changes that had happened in the vibration pattern and upholds the fact of fixing thresholds adaptive to the operating condition. The deviation between the threshold class clusters determined using the data acquired by IoT gateway and myRIO for starting to no load speed at standalone and disturbance conditions is around 0.2 percent of total amplitude range whereas the loaded condition, it is 0.6 percent. The deviations perceived are such that the threshold class clusters obtained from either IoT data or myRIO data do not lead to incorrect decisions and tends to recognize the change of operating conditions without ambiguity. Thus, the insight on the shaft vibration data remains reliable in spite of narrow variations in the threshold values. These attributes ascertain the reliability of the vibration data streamed by IoT gateway and available in cloud for performing condition monitoring analysis and decision making. The results also validate the efficiency of the statistical classification based signal decomposition algorithm in handling the non-stationary vibration signals at various operating conditions by providing consistent outcomes irrespective of difference in the data acquisition resources. The characteristics of IoT model to integrate the vibration sensors, actuators through Python and LabVIEW applications with cloud in real-time ascertain generic, interoperable and ubiquitous computational nature of the model for implementation of effective condition monitoring. Realization of machine maintenance and process automation platform with flexibility of data analysis in application specific platform has been substantiated with the real-time implementation of IoT based condition monitoring model. The stateless nature of REST architecture used for the deployment of condition monitoring is observed to enhance the scalability of the application. Thus, despite being remote, IoT based processing prevails as a better option for condition monitoring of multiple machines operating at different locations due to the attributes of cloud storage, flexible application development, data aggregation, platform of multiple services and scalability. The trait of the model developed to access the distributed databases of machine data and maintain a repository of analysis results as contextual references enhances the scope of precise decision making at the enterprise level.

References

[1] Divyakant Agrawal, Amr El Abbadi, Sudipto Das, Aaron J. Elmore, "Database Scalability, Elasticity, and Autonomy in the Cloud" *DASFAA'11 Proceedings of the 16th international conference on Database systems for advanced applications*, Hong Kong, Apr. 2011, Part I, pp.2-15.

[2] Bureau of Indian Standards IS 12075 (2008) Indian standard mechanical vibration of rotating electrical machines with shaft heights 56 mm and higher - measurement, evaluation and limits of vibration severity [ETD 15: rotating machinery].

[3] National Instruments. Addressing Challenges of Fleetwide Monitoring [Online]. Available: http://download.ni.com/evaluation/crio/24346_Fleetwide _Monitoring_WP_IA.pdf.

[4] Advantech Technical White Paper. (2015, October). Cloud-Based SCADA as an IIot Gateway. [Online]. Available: https://www.automation.com /pdf_articles/advantech/cloudbased.pdf

[5] Mallikarjun Kande, Alf J. Isaksson, Rajeev Thottappillil, Nathaniel Taylor (2017, October). Rotating Electrical Machine Condition Monitoring Automation—A Review. Machines, 5(4), 24. [Online]. Available: http://dx.doi.org/10.3390/machines5040024

[6] Dr Steve Lacey. (2017,01 March). From local to cloud-based condition monitoring. [Online]. Available: www.engineerlive.com/content/local-cloud-based-condition-monitoring

[7] Sheila Kennedy. (2015, 09 June). Condition monitoring in the cloud (Part 1) [Online]. Available: www.plantservices.com/articles/2015/technology -toolbox-condition-monitoring-cloud

[8] Jim Montague. (2012, Nov 08). Flexibility Driving the Future of Automation – Flexible Production, Cloud Computing Pave the Way to Future for Rockwell Automation and User, Automation Fair - A special report from the editors of CONTROL and control design. [Online]. Available:http://www.controlglobal.com/assets/wp_downloads/pdf/1212 11-ROK Auto Fair2012 SpecialReport.pdf

[9] Hashemian, H M and Wendell C Bean. (2011, October). State-of-the-art predictive maintenance techniques. IEEE Transactions on Instrumentation and Measurement. 60 (10), 3480-3492.

[10] Neelam Mehala and Ratna Dahiya. (2009). Condition monitoring methods, failure identification and analysis for induction machines. International Journal of Circuits, Systems and Signal Processing. 3 (1), 10-17.

[11] Asoke K Nandi, Chao Liu M L, Dennis Wong, "Intelligent Vibration Signal Processing for Condition Monitoring" *Proceedings of International Conference Surveillance 7*, Institute of Technology of Chartres, France, October 29-30, 2013, pp.1-15.

[12] *NI myRIO-1900 User Guide and Specifications*, National Instruments, 2013-2016, pp.1-31. [Online]. Available: http://www.ni.com/pdf/manuals / 376047c.pdf.

[13] *SIMATIC IOT SIMATIC IOT2020 SIMATIC IOT2040, Operating Instructions Manual - A5E37656492-AB*, Siemens, 2016. [Online]. Available: https://cache.industry.siemens.com/dl/files/658/109741658 /att_899623/v1/iot2000_operating_instructions_e_en-US.pdf

[14] *LabVIEW 2016 Help, Part Number: 371361N-01*, National Instruments, June 2016. [Online]. Available: http://zone.ni.com/reference/en-XX/help/ 371361N-01/lvcomm /http_client/

[15] Tavner, P J. (2008). Review of condition monitoring of rotating electrical machines. IET Electric Power Applications. 2 (4) 215-247.

[16] Chandra S Rakesh, Ayyappan, G S, Kota Srinivas, Ganesh, D (2016, April). Simulation and Testing of Induction Motor Faults in MATLAB for Online Condition Monitoring. IUP Journal of Electrical and Electronics Engineering, 9(2), 7-18.

[17] Cardona-Morales, O., Alvarez-Marin, D., Castellanos-Dominguez, G, "Condition Monitoring Under Non-Stationary Operating Conditions using Time–Frequency Representation-Based Dynamic Features", *CMMNO 2013 Proceedings of the Third International Conference on Condition Monitoring of Machinery in Non-Stationary Operations*, 2014, Lecture Notes in Mechanical Engineering LNME, 7, pp. 441–451. Springer, Heidelberg

[18] Jablonski, A, Barszcz, T, Bielecka, M, Breuhaus, P. (2013). Modeling of probability distribution functions for automatic threshold calculation in condition monitoring systems. Measurement [Online]. 46, 727–738. Available:http://www.sciencedirect.com/science/article/pii/S026322411 2003600

[19] *DIAdem 2017, Part Number 370858N-01*, National Instruments, 2017. Available: http://zone.ni.com/reference/en-XX/help/370858N-01/

[20] Ganga, D and Ramachandran, V. "Efficient Methodology for Estimation of Vibration Thresholds for Electrical Machines," presented at the International Conference on Machine Intelligence and Signal Processing (MISP 2017), Indian Institute of Technology Indore, Dec 22-24, 2017.

[21] *LabVIEW-Joint Time-Frequency Analysis Toolkit Reference Manual, Part Number 320544D-01*, National Instruments, 1998.

Ms. D. Ganga is from Tamilnadu, India and was born on 30th March 1980. She has completed M.E., in Electrical Drives and Embedded Control at College of Engineering, Guindy in 2007. She is currently pursuing Ph.D. at NIT Nagaland and serving as Assistant Professor at NIT Nagaland. Her research areas are Condition monitoring of machines, Internet of Things and industrial automation.

Dr. V. Ramachandran, born on 22 May 1958 at Virudhunagar in Tamilnadu, India received B.E., in Electrical and Electronics Engineering in Coimbatore Institute of Technology at the year 1980. After completion of M.E., in Power Systems Engineering at College of Engineering, Guindy, he obtained Ph.D. in 1990 from Anna University in Power Systems Planning and Reliability.

He is currently serving as Professor at College of Engineering Guindy with thirty-five years of teaching and research experience. He has acted as the Director of NIT Nagaland and Vice-Chancellor of Anna University Tiruchirappalli and has held many pivotal positions. His research areas are data analytics, smart grids, distributed services, Internet of Things and industrial automation.

Dr. Ramachandran has been an active member of various governance, executive, research and expert committees functioning for various educational motives.