

## Wireless Fault Detection System for an Industrial Robot Based on Statistical Control Chart

Alaa Abdulhady Jaber<sup>1</sup>, Robert Bicker<sup>2</sup>

<sup>1</sup>Mechanical Engineering Department, University of Technology, Iraq

<sup>2</sup>School of Mechanical and Systems Engineering, Newcastle University, UK

---

### Article Info

#### Article history:

Received Dec 7, 2016

Revised May 31, 2017

Accepted Jun 25, 2017

---

#### Keyword:

Condition monitoring

Fault detection

Industrial robot

Statistical control chart

Wireless system

---

### ABSTRACT

Industrial robots are now commonly used in production systems to improve productivity, quality and safety in manufacturing processes. Recent developments involve using robots cooperatively with production line operatives. Regardless of application, there are significant implications for operator safety in the event of a robot malfunction or failure, and the consequent downtime has a significant impact on productivity in manufacturing. Machine healthy monitoring is a type of maintenance inspection technique by which an operational asset is monitored and the data obtained is analysed to detect signs of degradation and thus reducing the maintenance costs. Developments in electronics and computing have opened new horizons in the area of condition monitoring. The aim of using wireless electronic systems is to allow data analysis to be carried out locally at field level and transmitting the results wirelessly to the base station, which as a result will help to overcome the need for wiring and provides an easy and cost-effective sensing technique to detect faults in machines. So, the main focuses of this research is to develop an online and wireless fault detection system for an industrial robot based on statistical control chart approach. An experimental investigation was accomplished using the PUMA 560 robot and vibration signal capturing was adopted, as it responds immediately to manifest itself if any change is appeared in the monitored machine, to extract features related to the robot health conditions. The results indicate the successful detection of faults at the early stages using the key extracted parameters.

Copyright © 2017 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Alaa Abdulhady Jaber,  
Mechanical Engineering Department,  
University of Technology,  
Baghdad/Iraq.  
Email: 20039@uotechnology.edu.iq

---

## 1. INTRODUCTION

In machines condition monitoring (CM), fault detection/diagnostic approaches can be classified into two types, dependent on whether the diagnosis evaluation is based on deterministic or stochastic information (e.g., historical, statistical parameters). The first of which has been termed a model-based or “white box” approach, while the second is known as data-driven or “black box” approach [1]-[3]. The model-based fault detection approach employs a mathematical model of the system under observation, by assuming that a fault in the system will lead to deterministic changes in the model parameters. It relies on comparing the model outputs with the actual system outputs to generate a residual signal, and based upon the properties of the generated residual signal, potential fault conditions are identified and useful information is extracted [4]. Data-driven approaches depend on the features extracted from the measured process data for building a model that represents the process, and these are categorised into two types; statistical based methods and

those based on artificial intelligence (AI) techniques [5],[6]. However, accurate analytical models of industrial robots are often not practicable and difficult to be constructed due to dynamic complexity of the robot and unavoidable uncertainties [7]. Accordingly, model-free approach based on statistical control chart (SCC) is considered in this research for establishing a fault detection system for an industrial robot.

The concept of model-free based fault detection and diagnosis has stimulated the interest of using novelty detection for condition monitoring, which focuses on identifying any deviations between the features extracted from the recent measured data and the data measured under normal (healthy) operating conditions. The features obtained from a machine in its undamaged state will have a distribution with an associated mean and variance. However, a variation in the mean and/or variance will appear if the machine is damaged. Statistical control charts (SCCs) provide a framework for monitoring the distribution of the features and detection if they are inconsistent with the past healthy state, and any change in the distribution characteristics of the features will indicate damage, termed outlier analysis. SCCs are one of the earliest statistical fault detection techniques dating back to 1931 [6]. Starting with the advent of Shewhart control charts for averaging, usually called X-bar ( $\bar{X}$ ) chart, which is normally used in combination with a range chart (R chart) or standard deviation chart (S chart) [8]. Further modification to Shewhart charts have resulted in cumulative sum (CUSUM) and the exponentially weighted moving average (EWMA) charts in the early 1950s [6]. Because these charts are easy to construct, implement, and interpret, they received a large acceptance in the field of machines and processes monitoring.

Techniques based on SCCs can be classified to two approaches: the univariate and multivariate approaches. In the former method each characteristic of interest is monitored independently whereas in the latter the concurrent monitoring of characteristics is accomplished, considering the correlation that may exist among the various characteristics [6],[9]. Baydar, et al. [10] presented a multivariate statistical methodology for helical gears monitoring. The gathered time-domain vibration signals were employed to form a reference condition model using principle component analysis (PCA). The T-square control chart, type of multivariate SCCs, was adopted as health condition indicator. Researchers concluded that when tooth failures occur, the probability density function (PDF) of the measured signal will change which gives good indication about the health condition.

Another paper applied two statistical techniques for wind turbine gearbox CM [11]. The first technique was based on data-mining algorithms used to build a statistical model for predicting the jerk indicated by the vibration excitement of the gearbox. This model was utilized in conjunction with experimentally captured vibration signal to produce residual signals. Two control charts, X-bar and EWMA charts, were constructed to evaluate the residual and fault prediction. Another application, amongst others, of statistical control charts is for roller bearings condition monitoring. Niknam, et al. [12] at University of Tennessee investigate the use of CUSUM chart for detecting bearing failures, such as unbalance, based on acoustic emission signal analysis. Similarly, Zhou, et al. [13] presented an approach for in situ induction motor bearing fault detection by combining noise cancellation and X-bar chart. In this work the motor current signature was analysed to extract features related to bearings deterioration. Two control charts were developed based on Shewhart average chart to identify the initial start point of the bearing defect [14]. These charts are named adaptive moving average chart, and adaptive Shewhart average level charts. Based on these charts the researchers were able to produce warning and action limits. The findings of this study suggest that the adaptive Shewhart average level chart overcomes the drawback of adaptive moving charts by working out the limits using all the bearings' data.

In the majority of CM systems, with a special focus on industrial robots, it has been noted that for signal acquisition, processing, storage and decision making, data acquisition (DAQ) cards connected to PCs, as their main processing core, were extensively used, which adds considerably to the cost of CM system. Recent developments in electronics and computing have opened new horizons in the area of condition monitoring, and embedded devices and wireless systems other promising solutions, and have shown their practicality in fault detection and diagnosis processes in many of areas. For instance, a research study has implemented the envelope analysis algorithm for wireless bearing health monitoring based on vibration signals measurement, however, to overcome the limitations of memory size and restricted computational capabilities in the commercially available wireless nodes, the authors have used a 32-bit microcontroller type TM4C1233H6PM from Texas Instruments along with Zigbee wireless module.

In this research, the concept and design steps of the proposed wireless system are provided first. The components selection, embedded system construction and installation on the robot with preliminary system testing are then discussed. After that, the principle of SCC and how it can be implemented in fault detection is deliberated. Based on the extracted feature from the robot vibration an appropriate control chart is designed and implemented on the designed wireless system. Lastly, the performance of the designed system is then investigated.

## 2. DESIGN OF THE SENSOR NODE

The sensor node, as depicted in Figure 1, is an embedded system usually outfitted with a microcontroller unit (MCU) (or digital signal processor (DSP)), radio frequency (RF) transceiver, power supply, and memory along with various sensors and actuator depending on the application. The microcontroller with memory to store the data acts as the central computing and controlling device of the sensor node. The radio transceiver module, which allows two-way radio communication between several nodes in order to distribute the information, represents the communication subsystem of the node. The power supply subsystem is in charge of powering the whole sensor node, and is normally composed of batteries, which offer an easily available, low cost, and high capacity source of power, and have become companion with sensor nodes. In this work, one node will be fixed on the robot arm and will be in charge of performing the vibration signals capturing, conduction the SCC algorithm and then sending the information to the base station, which is connected to a PC for data visualization and robot health assessment. So, the node has to be of light weight and small size, easily attached and does not add extra loading on the robot arm. The node has to be of a reasonable price and easy to produce, for future expansion of the wireless systems. In this section the selection of the accelerometer, microcontroller, transceiver and the power supply source of the node is discussed. However, the proposed design methodology is not limited for industrial robots and could be utilized for health monitoring of different machines.

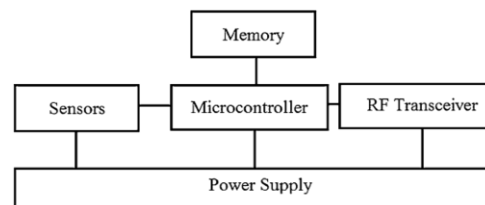


Figure 1. Wireless sensor node hardware architecture [18]

### 2.1. Accelerometer Selection

Accelerometers are electromechanical devices that convert the mechanical signals, such as vibration and force, to electrical signals, and are an extensively used for fault detection in many machines because of their accuracy, robustness and sensitivity. Selecting an appropriate accelerometer and the way it is mounted on a machine are significant factors in determining the success of any condition monitoring program. Misleading data can be produced if an unsuitable accelerometer is selected for the machine under study or an appropriate one is mounted in an incorrect location on the machine. The major parameters which need to be considered for accelerometer selection are [19]: sensitivity, range, bandwidth, frequency resolution, reliability, accuracy, operation environment and cost. MEMS (Micro-electro-mechanical-sensors) accelerometers compared with the conventional piezoelectric accelerometers, consume less power, are small in size, light weight, low cost and can achieve good performance [20]. Consequently, they are more suitable for wireless systems design, and will be used here. However, a variety of MEMS accelerometers are commercially available. After taking the all-important technical requirements in the consideration, the ADXL001 MEMS accelerometer has been chosen.

The ADXL001 is a MEMS based, single-axis accelerometer developed by Analog Devices. It provides a high performance, wide bandwidth of 22 kHz, and is small in size. The ADXL001 operates on a 3.3V or 5V supply, and outputs an analogue voltage, which allows the direct connection of the accelerometer output to the analogue input pins on a data acquisition device (DAQ) or a microcontroller. The accelerometer is available in 3 full-scale dynamic ranges of  $\pm 70$ ,  $\pm 250$ , and  $\pm 500$  g. For this work the  $\pm 70$  g range was deemed appropriate. Furthermore, using just one accelerometer for robot health monitoring was not considered sufficient, since when the robot is performing a general task its joints are rotating around different axes, therefore, the accelerometer may be more sensitive to specific joints than others yielding the monitoring system unreliable. For this reason, it was decided to fix three ADXL001 accelerometers in an orthogonal configuration to measure the vibration in X, Y, and Z directions using a purpose designed aluminum adapter, as shown in Figure 2. To fix this adapter correctly on the robot for accurate pick-up of vibration signals, adhesive mounting using superglue was adopted here.

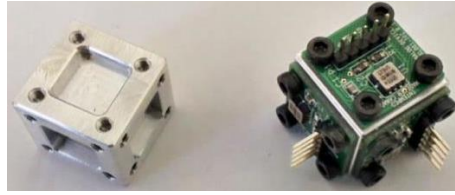


Figure 2. Designed 3 axis accelerometer adapter

## 2.2. Microcontroller Selection

A microcontroller unit (MCU) is a programmable device that combines electronics components, include central processing unit (CPU), memory and peripheral devices, integrated into a single chip [21]. Arduino offer several microcontroller models with different characteristics. The main differences between these models are the type of processor, the number of inputs and outputs ports, and the capacity of memory. Arduino controllers are relatively efficient, consume less power, cheap, and suitable for use in a harsh environment [22]. To reduce the number of available Arduino options it was decided to look at Arduino boards that have high specification and meet the desired requirements, such as DUE and Intel Galileo, which is a new board designed to be compatible with Arduino hardware and software and is based on Intel architecture. The Intel Galileo board was thoroughly tested, but despite the high functionality it was established that the signal capturing process takes very long time in order to capture 4096 samples, as it is the default number of samples, from one analogue input channel; as a result, was rejected. The Arduino DUE board was tested and found to be more than capable for this work in terms of signal capturing speed, processing speed, memory size, power consumption and, of course, cost. The Arduino DUE (Figure 3) is an open-source, single-board microcontroller based on a 32-bit, RISC, Atmel SAM3X8E ARM Cortex-M3 processor. It offers a relatively small size form, measuring 101.6 mm x 53.3 mm, and compatibility with most of the standard Arduino shields. The Arduino DUE board has an 84 MHz clock frequency, USB connection, four UARTs (Universal asynchronous receiver/transmitter) serial ports, a power jack, a reset button, and an erase button. There is a 12-bit resolution analogue to digital convertor (ADC) built in inside the processor with 1 MSPS (mega samples per second) sampling frequency and 12 input channels. The board comes with 512 KB flash memory, and 96 KB of SRAM. Whilst other Arduino boards accept up to 5V on I/O pins, the Arduino Due board is based on 3.3V on the I/O pins, fortunately, the selected accelerometer can work using 3.3V or 5V which makes the connection of the accelerometer to the board does not require any conditioning circuit.

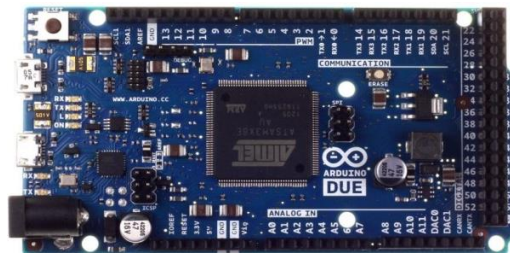


Figure 3. Arduino DUE microcontroller board [https://www.arduino.cc]

## 2.3. Wireless Technology Selection

Communication devices are used to exchange data between the nodes of wireless networks. The advances in wireless applications have led companies to develop different types of wireless standard. These standards are usually classified by their capabilities and properties, and designed to suit different applications, such as fault detection or human health monitoring. In this section, three widely used wireless network technologies are discussed, in order to investigate their pros and cons and also to select the appropriate technology for the work here. These technologies are [23],[24].

The IEEE802.11x family of standards is meant for wireless local area network (WLAN), which is also known as Wireless Fidelity (Wi-Fi). There are four generations of Wi-Fi products available which are IEEE 802.11a/b/g/n, operate in high frequency, unlicensed Industrial, Scientific, and Medical (ISM) radio

bands ranging from 2.4 GHz to 5 GHz. Typically, it is adapted for relatively high bandwidth and high data transfer rate, ranges from as low as 1 MBPS (megabyte per second) to over 50 MBPS, and commonly used for mobile computing devices, such as laptops. With the use of a standard antenna the transmission range can be up to 300 feet, and it can be significantly improved by utilizing a directional high gain antenna. Although the data transmission and rate ranges are enough for wireless sensor network application, the power requirement generally limited its usage in wireless sensor application. Bluetooth (IEEE802.15.1) is another standard that has lower power than IEEE802.11, an operating frequency 2.4 GHz within the ISM band, and represents a personal area network (PAN) standard. It is specifically aimed to serve applications that require short range communication, such as data transfer between computers and other peripheral devices like keyboards or cell phone to replace wire connectivity. Bluetooth supports star network topology, and can enable up to seven remote nodes to communicate with a single base station. However, in addition to its short range application and scalability problem (low number of nodes per network), the other disadvantages of Bluetooth are high power consumption and nodes that need a long time to be synchronized with the network when returning from sleep mode, and which increases the average system power.

The IEEE802.15.4 standard was designed for low data rate transmissions, low cost, and low power consumption wireless personal area networks (WPAN). In terms of communication range, this standard can be considered as a middle ground solution between IEEE802.11 (Wi-Fi) and IEEE802.15.1 (Bluetooth), and supports multiple transmission frequencies, multiple data rates, and two topologies, star and point-to-point (pair topology), which makes it a flexible standard. It operates in the unlicensed ISM bands at 868 MHz in Europe, 915 MHz in the USA and 2.4 GHz worldwide, with data rates 20 Kbps (kilo bit per second), 40 Kbps and 250 Kbps, respectively. ZigBee is a standard designed by the ZigBee Alliance, which is an association of companies working together to enable reliable, cost-effective, and low-power wireless network [25]. It is based on IEEE802.15.4 standard which means that ZigBee can take full advantage of this standard. In addition, it can accommodate multiple networks topologies like star, point-to-point, and mesh networks. Also, a ZigBee network can have at most 65000 nodes, making it a very scalable standard. Because of the aforementioned features, the ZigBee standard has been adopted in many wireless sensor network applications, and will be the best candidate for the work described in this thesis. Additionally, ZigBee modules are nowadays available in the market in small size with very affordable prices.

#### 2.4. ZigBee Module Selection

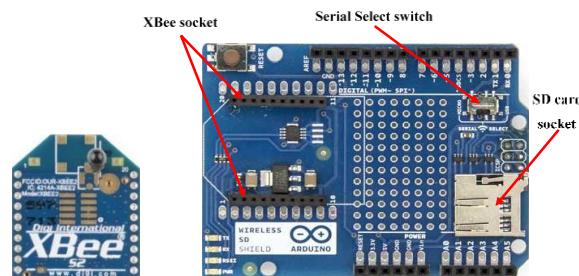
There are many parameters need to be considered when selecting a ZigBee module. From these parameters are power consumption, operating frequency, flexibility, coverage range, the module and testing costs, and the compatibility with the microcontroller unit. However, many certified semiconductor companies, such as Texas Instruments, Freescale, Digi International, are providing successful design for ZigBee products. After revising the available products, the XBee module from Digi International has been selected, as it meets the above mentioned requirements, is compatible with the selected Arduino board, and includes other factor, i.e. previous successful experience, popularity, and available development resources. XBee is a trade name from Digi international, and it is a wireless module designed for applications that require reduced data communication while having long range capabilities with less power consumption. XBee modules come in different formats for different kinds of application. There are two types of XBee modules, series 1 (s1) and series 2 (s2). As shown in Table I, XBee s2 consumes slightly less power and has a better range than the s1. The XBee s1 and s2 are pin-for-pin compatible, but based on different chipsets and running different protocols. The s1 module uses the IEEE 802.15.4 standard protocol, while the s2 module relies on ZigBee protocol. Furthermore, both series come in two different transmission powers, regular and Pro. The regular version is simply called XBee, and is less expensive than Pro version; the Pro version uses more power and is slightly larger than the regular version, but, on the other hand, has a longer communication range. The XBee s2 (Figure 4) has been selected for this research, as it provides good in-door range, good data rate, low-energy consumption, better receiver sensitivity and it supports mesh and tree networks. This makes the system scalable and reliable for future development. XBee modules can be attached to DUE using a wireless shield (Figure 5) designed for easy connection to the XBee.

XBee modules must have an antenna to send and receive signals, of which there are four different types offered on XBee s2, namely whip, chip, U.FL, and RPSMA [17]. The wire (or whip) and chip antennas come pre-connected to the XBee modules, while the U.FL and RPSMA are derivatives of the connector types, offering chip with connectors on the board. However, based on preliminary communication range tests [26] was establish that the RPSMA (external) antenna is the best option in this research. Also, in WSN must be there is a node configured to work as a coordinator (or base station) [17] and the others as sensor nodes. The coordinator represents the network controller and is responsible for setting up and maintaining the network, which should be connected to a reliable, uninterrupted power source, and requires to be interfaced to a PC for further data processing and visualization. In order to allow the transceiver modules on the sensor

node and base station to talk to each other, they have to be correctly configured individually, by utilizing specific software, before using them in a network. The configuration includes various aspects such as classification of the node (coordinator, router or end device), network ID, destination address and so on. X-CTU (XBee configuration and testing utility) is windows-based application software developed by Digi and represents the official configuration software for XBee modules [17]. Many versions of firmware can be selected and written into a ZigBee module via RS232 or USB port, depending on the used interface board. For this purpose, the XBIB-U-DEV development kit from Digi International was employed [26].

Table 1. Comparison between XBee s2 and s2

Specifications	XBee s1	XBee s2
Indoor range (m)	30	40
Outdoor range (m)	100	120
Frequency band (GHz)	2.4	2.4
Transmit power (mW)	1	2
Supply voltage (V)	2.8-3.4	2.8-3.6
Data rate (Kbps) (kilo bit per second)	250	250
Transmit current (mA)	45	40
Receive current (mA)	50	40
Receiver sensitivity (dbm)	-92	-96



Figur 4. XBee module and the wireless shield for Arduino [<https://www.sparkfun.com>; <https://www.arduino.cc>]

## 2.5. Power Source Selection

Rechargeable batteries are commonly employed as a power source for the wireless sensor nodes. Several characteristics should be considered when choosing a battery for a wireless sensor node. The most important of these attributes are energy density, charge/discharge cycle, size, self-discharge rate and cost. There is no any battery technology that meets all these criteria, so a compromise must be made. Each battery type has its advantages and disadvantages with a wireless sensor node. The most common rechargeable batteries types are: lead acid, nickel-cadmium (Ni-Cd), nickel-metal hydride (Ni-Mh), lithium-ion (Li-Ion), and lithium-ion polymer (Li-Po). The comparison of the different types of batteries can be seen in Table II [27]. Lithium-based batteries are technically more advanced, most widely used and fastest growing energy sources. Comparing to the other three battery types, Li-Ion and Li-Po have higher specific energy densities and offer a lower self-discharge rate than both Ni-Cd and Ni-Mh, with only lead acid having a lower rate. Lithium-Po batteries are similar to Lithium-Ion however with a different type of electrolyte used, and furthermore offer the advantage of being very thin and light weight, thus allowing them to be easily included in sensor nodes while occupying very little space. In this regard, a Turnigy Li-Po battery (Figure 5) has been chosen as a power source for the wireless node. The battery has an energy capacity at 7.4 V/1000 mAh (1 hour continuously working at 7.4 V with 1 A discharge current) with weight 62 g and size 74 x 35 x 13 mm, which can fit easily in the node enclosure as shown later.

Table 2. Comparison Among Rechargeable Batteries

Specifications	Lead Acid	Nickel-Cadmium	Nickel-Metal hydride	Lithium-Ion	Lithium-Ion polymer
Energy density (Wh/kg)	30-50	45-80	60-120	110-160	100-130
Charge/Discharge cycle	200-300	1500	300-500	500-1000	300-500
Self-discharge/Month	5%	20%	30%	10%	~10%



Figure 5. Lithium-Ion polymer (Li-Po) battery

### 3. SYSTEM SET-UP

To protect the wireless node and base station from dust and other contaminants, enclosures have to be used. Available enclosure for the Arduino DUE is designed to enclose just the boards without the other connected peripherals such as the sensors, battery, XBee shield and the antenna; thus, it was necessary to look for alternatives. Metal or plastic boxes are commonly used for creating custom-built enclosures for embedded equipment. For this purpose, two plastic boxes of size 179 x 86 x 51 mm, for the sensor node and base station were purchased and modified. Bolts were used to fix the Arduino microcontrollers to the boxes. Small rectangle and circular cuts were made on the base and right side of the sensor node box, to allow attaching the sensors adapter to the robot using super glue, and the antenna, via the wireless shield, and power supply port, from the battery, to the microcontroller. A double sided adhesive tape at the bottom of the battery was utilized to affix it on the box. The base station should be located next to a host PC, for reporting the analysis results and also to permit control of the entire monitoring system through the user interface. The sensor node was attached to the robot using two cable ties. The sensor node along with the final experimental set-up is depicted in Figure 6. Prior to deploying the designed system for robot condition monitoring, system accuracy was tested and validated. However, in this research the experimental work has been performed using the PUMA 560, which is a PC-controlled serial manipulator with six revolute joints/degrees of freedom (DOF).

The testing was performed by setting up the system for real-time robot vibration analysis and the captured signals were compared with their peer that were acquired using NI 6009 DAQ device (from National Instruments). Significant differences between signals are not preferred, as it could lead to increase rate of false fault detection. The result of the healthy signals comparisons for each of the three axes of measurements is presented in the Figure 7. Signals from both NI DAQ and the sensor node are compared with the robot executing a pick and place task, using the same sampling frequency (383 Hz); the time-domain signals are associated with their probability distributions. It can be seen that the wireless system signals are identical with NI DAQ signals, and normally distributed. It can be observed that the signals from the designed system are having slightly higher amplitudes, also the other statistical features, i.e. the standard deviation (STD), root mean square (RMS) and maximum and minimum amplitudes, as shown in Table III. This could be attributed to attaching the sensor node to the robot structure, which cannot be avoided, making the system capturing extra vibration from the robot, while this is not the case when the NI DAQ. Nevertheless, it was concluded that the signals captured using the Arduino DUE board have very reasonable accuracy, and thus it is expected that by employing the DUE board the desired credibility level of the designed robot health monitoring system can be achieved.

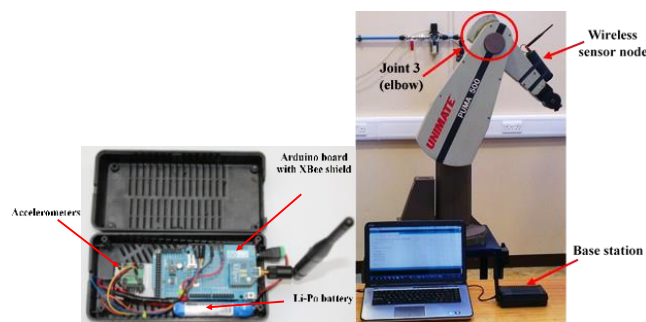


Figure 6. Wireless sensor node hardware and the experimental set-up

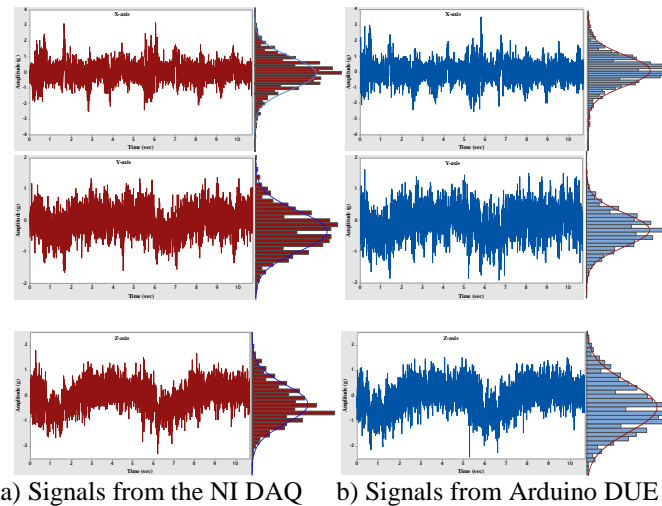


Figure 7. Time-domain signals from the X, Y and Z accelerometers captured using NI 6009 DAQ and Arduino DUE board

Table 3. Statistical Features Extracted from Time-Domain Signals Captured Using NI 6009 DAQ and Arduino DUE Board

Axis of measurement	Used signal acquisition system	STD (g)	RMS (g)	Maximum (g)	Minimum (g)
X	NI DAQ	0.62613	0.626	3.18387	-2.50729
	Arduino DUE	0.6636	0.664	3.5530	-2.5378
Y	NI DAQ	0.45126	0.448	1.39296	-1.67155
	Arduino DUE	0.50545	0.506	1.63550	-1.91748
Z	NI DAQ	0.52054	0.521	1.79090	-2.30827
	Arduino DUE	0.57752	0.578	1.52271	-2.42505

#### 4. PRINCIPLE OF CONTROL CHARTS

Control charts are a statistical tool for graphically displaying a *quality* characteristic plotted against the sample number (or time) with a centre line and two (upper and lower) control limits [28], as illustrated in Figure 8. Control charts represent one of the most important and widely applied methods for detection of abnormal process operations based on process variables such as dimensions, temperature, vibration and forces etc. They provide a clear differentiation between changes that are a result of unpredictable disturbances in the system and changes that occur as a result of a system fault. When control charts are being used for health monitoring of any machine, despite how well it is designed and maintained, some amount of inherent or natural variability will always be presented [9]. This natural variability, which is usually referenced to as background noise, is the cumulative effect of many small and unavoidable causes. However, from the statistical quality control point of view, a system that has this natural variability is often called a stable system and in-control [9].

On the other hand, if the variability in the monitored parameter is large enough when compared to the background noise, this is said due to assignable causes and represents an unacceptable level of system performance. In this case, the sources of variability are not part of the natural causes and commonly entitled *special causes*; a system that is operating in the presence of special causes is said to be an out-of-control. The conventional control chart is composed of three horizontal lines; a centre line (*CL*) that represents the average value of the quality characteristic corresponding to the in-control state; and two other lines named the upper control limit (*UCL*) and the lower control limit (*LCL*) (Figure 1). These control limits are chosen so that if the process is in-control, nearly all of the sample points will fall between them. It is common practise to connect the sample points on the control chart with straight-line segments since it is easier to visualize how the sequence of points has evolved over time. However, even if all the points fall inside the control limits, but behave in a systematic or non-random manner, then this could be an indication that the process is out of control. If the process is in control, all the plotted points should have an essentially random pattern.



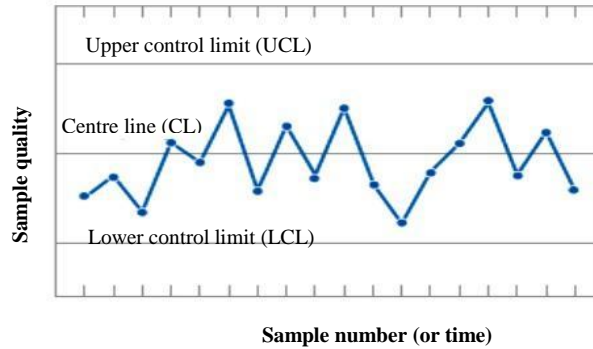


Figure 8. A conventional control chart [28]

#### 4.1. Control Chart Limits

The choice of control limits is one of the critical decisions when designing a control chart; and is connected with the risk of faulty prediction of in or out of control. Fundamentally, there is a close similarity between the principles of control chart and hypothesis testing. By applying the control chart technique, a continuous hypothesis testing is carried out through the progress of the process. For a normally distributed process the null hypothesis ( $H_0$ ) is that the treated sample mean is equal to the reference (healthy) process mean, while the alternative hypothesis ( $H_1$ ) assumes that the treated sample mean does not equal to the reference mean. The  $H_0$  hypothesis refers to that the monitored parameter is in a good condition and the process should continue, whereas  $H_1$  indicates that there is a problem and actions should be taken. However, two types of errors can occur during the fault prediction process using control charts (or hypothesis testing), these are called *type I* and *type II* errors [9, 28]. *Type I* error is accrued when a point falls outside the control limits, indicating an out-of-control condition when there is no assignable cause presented (reject the null hypothesis  $H_0$  when it is true). Whereas the *type II* error results if a point falls within the control limits when the process is actually out of control (fail to reject the null hypothesis  $H_0$  when it is false). Therefore, the risk of *type I* error is decreased by moving the control limits further from the centre line. Widening the control limits, however, will also increase the risk of *type II* error. However, the opposite effect occurs if the control limits are shifted closer to the centre line. The general mathematical formulation for a control chart is

$$CL = \mu_W \quad (1)$$

$$UCL = \mu_W + k\sigma_W \quad (2)$$

$$LCL = \mu_W - k\sigma_W \quad (3)$$

where  $W$  is a sample statistic that measures some quality characteristic of interest, and  $\mu_W$  and  $\sigma_W$  are its mean and standard deviation, respectively, while  $k$  represents the distance of the control limits from the  $CL$ , expressed in standard deviation units. The common practice is to make the  $UCL$  and  $LCL$  equal three standard deviations (three-sigma) from the  $CL$  of the monitored data, i.e.  $k = 3$ . By using three-sigma control limits it is assumed that the distribution of the quality characteristics is approximately normally distributed. Then, by doing so it is presumed that while the system is in statistical control, nearly 99.7% of the points will fall within the control limits [28],[29]. In this way, a good balance is made between *type I* and *type II* errors. In addition to the three-sigma limits, additional warning limits can be utilized. These are named the inner limits, usually placed at two-sigma [30].

## 5. ROBOT VIBRATION ANALYSIS AND FEATURE EXTRACTION

This paper concentrates on fault detection in the elbow joint (joint 3) of the robot. The mechanical construction of this joint is illustrated in Figure 9, which consists of a two-stage gear train system. The gear train is housed in the end of the upper arm and connected to a DC motor, which is housed in the beginning of the upper arm as well, by a drive shaft. There is a bevel pinion on the input shaft drives meshed with a bevel wheel on one end of an idler shaft. Also, a spur pinion at the other end of the idler shaft engages with a spur wheel fixed to the forearm, and therefore rotates the forearm around the elbow axis. A number of deep groove ball bearings are used to carry the input and idler shafts. The main fault mechanisms that may appear in the joint gearbox are basically the same as those arising in other types of gearboxes, such as tooth faults.

The most common types of gear tooth failure are scuffing, cracking, macro- and micro-pitting, wear, bending fatigue, and fracture due to overload as well as backlash between mating teeth. Moreover, since gears are normally supported on rolling element bearings, faults in these bearings such as wear in the inner or outer races represent another typical type of fault in gear transmissions. For the purpose of fault robot detection in this paper, backlash has been introduced in the bevel gear set of joint 3. To introduce backlash in joint 3, a number of screws should be rotated to adjust the pitch between the gear pairs. After adjusting the backlash, the robot was programmed to mimic a standard robot task by undertaking a pick and place sequence.

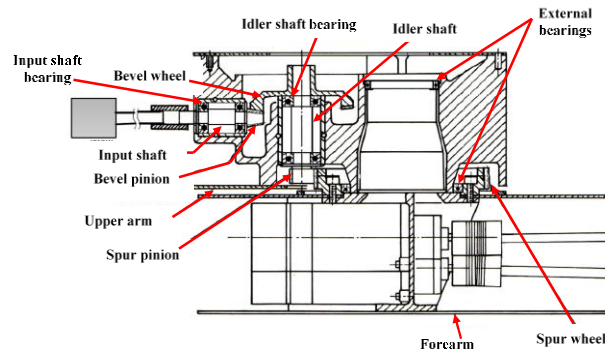


Figure 9. Schematic diagram of joint 3 (elbow joint)

Several features from the time-domain signals, such as root mean square (RMS), standard deviation (STD), and kurtosis, are extracted when the robot is healthy and when different backlash levels are introduced. A comparison of these features, to investigate which is the most faults sensitive one, was accomplished. The sensitive feature must be normally distributed, as this is a condition that has to be met in order to calculate the threshold values using SCC [28],[30]. The STD feature, which is normally used as a measure of extent of variation of the processed data and has the same units as the data, was found to be the most faults sensitive, and normally distributed one; and hence it was selected for comparison among fault severities.

To compute sets of STD values from the time-domain data, in order to be used for developing and testing of a SCC, the robot was programmed to execute the pick and place task for 100-time and the STD values for each repetition were calculated. The STD results when the backlash fault was simulated and even with the other fault types have shown that the X-axis vibration is the best to be utilized for monitoring the effect of fault development in the robot, as a clear designation can be recognized amongst the different faults. However, Y- and Z-axis could experience higher vibration level if the robot executes different tasks that involve a different sequence of movements or if a fault is progressed in the other joints. Because of this and in order to achieve reliable fault detection, the vibration of the three axes has to be considered. Thus, the resultant of the X-Y-Z standard deviations will be computed, by taking the square root of the sum of the squares of the three STDs, and used as the fault indicator. Figure 10 shows the resultant STD values for the robot when it is healthy and with three backlash levels (high, small and interference); a clear difference can be realized amongst the four trends related to robot health conditions. The high backlash conditions show lower resultant STDs whereas they are the highest in the interference condition.

The normality of the healthy features was studied through graphical techniques used to assess whether or not the data are consistent with the normal distribution. These are the histogram, which is a bar graph plot of data and represents an approximation to a probability density function, and the normal probability plot. The vertical axis of a histogram represents the data type frequency, while it signifies the cumulative probabilities scale in the normal probability plot; the horizontal axes in both plots are the values of the variable. At the center of the normal probability plot, a line of normal probability is drawn passing through the mean of the variable and the 50% cumulative probability. If most of the data points are placed on this line, the data is considered to be normally distributed. If the points appear in a curvature shape, then the indication is that the data are not normally distributed. To achieve this and to calculate the upper and lower control limits, as will be explained later, the Minitab 17 statistical package has been used and the result is shown in Figure 11; the histogram (left) is indicating a bell-shape distribution and the data in the normal probability plot (right) looks reasonably straight, meaning that the resultant STD is normally distributed, thus it will be used for designing the control chart.

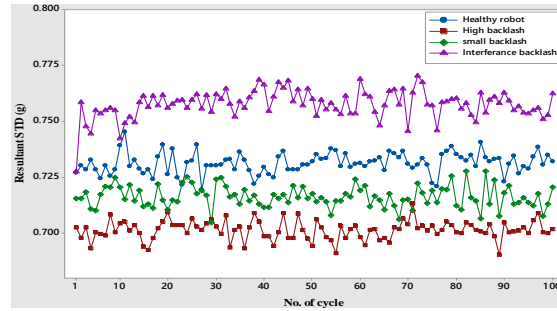


Figure 10. Resultant STD for healthy and with different backlash levels robot

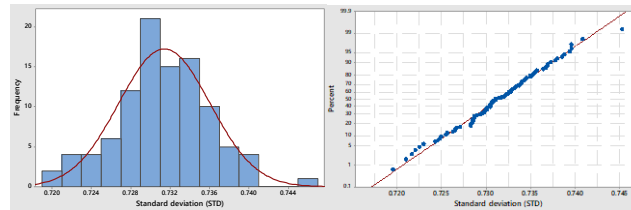


Figure 11. Normal distribution histogram and normal probability plot for the resultant standard deviation feature

## 6. CONTROL CHART DESIGN

The most commonly used control charts are: Shewhart, cumulative sum (CUSUM), exponentially weighted moving average (EWMA), which are called univariate charts, and T-square control chart, which is one of multivariate control charts. Shewhart control charts are the most popular charts used for process monitoring, and can be easily implemented since they do not require a lot of computational effort; and for this reason has been used here, based on the following discussion.

It is a standard practice when a variable is monitored using Shewhart charts that both the mean and the variability of the variable are considered. The mean of a variable is monitored with the X-bar ( $\bar{X}$ ) chart and the variability of the variable is monitored using a range chart (R-chart) or a standard deviation chart (S-chart). The X-bar chart informs whether the process is stable with respect to its healthy level, whereas the R- and S-chart provide information regarding the variability of the process and if it is stable over time or not. Significant shifting of the mean and the unusual large variability are indications of special causes or a fault starting to develop in the robot. The R-chart is relatively insensitive to small or moderate shifts for small sample size [28], thus, in a situation that demands tight control of process variability, moderately large sample sizes will be required, and the S-chart should be used. The control limits of the mean and standard deviation charts are derived from the healthy features after dividing the data set into subgroups (samples) of ten observations each. The population mean ( $\bar{x}$ ) and standard deviation ( $\bar{s}$ ) can then be estimated based on these data, having  $m$  preliminary samples, each of size  $n$ , with  $i$ th sample mean ( $\bar{x}_i$ ) and standard deviation ( $s_i$ ), as shown:

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i \quad (4)$$

$$\bar{s} = \frac{1}{m} \sum_{i=1}^m s_i \quad (5)$$

The upper and lower control limits and centre line of  $\bar{X}$  chart are given by:

$$UCL_x = \bar{\bar{x}} + 3 \frac{\bar{s}}{c_4 \sqrt{n}} \quad (6)$$

$$CL_x = \bar{\bar{x}} \quad (7)$$

$$LCL_x = \bar{\bar{x}} - 3 \frac{\bar{s}}{c_4 \sqrt{n}} \quad (8)$$

and the upper and lower control limits and centre line of S-chart are:

$$UCL_s = \bar{s} + 3 \frac{\bar{s}}{c_4} \sqrt{1 - c_4^2} \quad (9)$$

$$CL_s = \bar{s} \quad (10)$$

$$LCL_s = \bar{s} - 3 \frac{\bar{s}}{c_4} \sqrt{1 - c_4^2} \quad (11)$$

where  $c_4$  is a tabulated constant depends on the sample size and can be found in [28]. Minitab results for the  $\bar{X}$ -S charts, based on the resultant STD data from the healthy robot are presented in Figure 12. It can be seen from the charts that all the points are randomly distributed and within the control limits which means the robot is healthy.

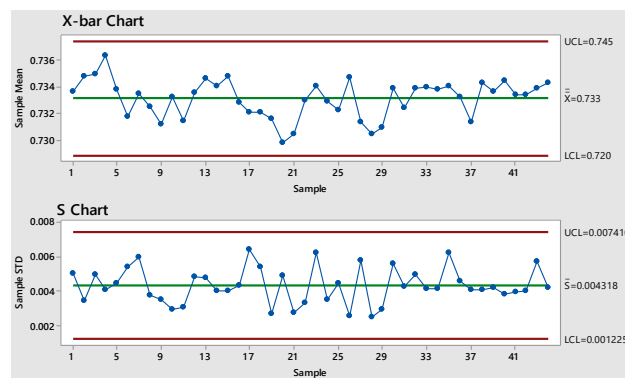


Figure 12. Designed Xbar-S charts

## 7. PERFORMANCE TESTING OF THE DESIGNED CONTROL CHART

Firstly, the established resultant standard deviation control chart was tested offline using unseen data; here it is tested under real-time conditions, whilst running on the wireless sensor node. The testing was done when the robot is in the healthy state and with different backlash levels, while it executes its repetitive task. The robot was stopped periodically in order to introduce different backlash levels. The result of the test is shown in Figure 13. In this test extra limits are added, these shall be called the upper and lower warning limits (*UWL* and *LWL*) respectively, equal to  $CL \pm 2\sigma$  [28], where  $\sigma$  is the standard deviation of the samples used for constructing the control chart, and can be used to give an early indication about the robot health deterioration. From the figure it can be seen that the extracted features for the low and interference backlash cases are outside the control limit; this is a very obvious indication of a fault in the robot. In the small backlash level, it is observed that all the STD features are inside the warning limits, on the same side below the center line. This is also an indication of a fault, based on the fourth rule of control charts [28], which states that a run of eight (or more) consecutive points on one side of the center line is an indication of a fault. Samples of the healthy case are randomly distributed inside the warning limits, indicating a healthy condition. Occasional some non-consecutive points fall outside the control limits or one of successive faulty points located within the control limits, these are marked in black ellipses in the figure and is due to the intrinsic variation within the calculated features [28]. In addition to the rule of eight consecutive points (mentioned above), if there are three or more successive points exceeding the warning limits or six points in a row steadily increasing or decreasing, a fault development in the robot is considered. These three rules are applied in the embedded system for the early robot fault detection.

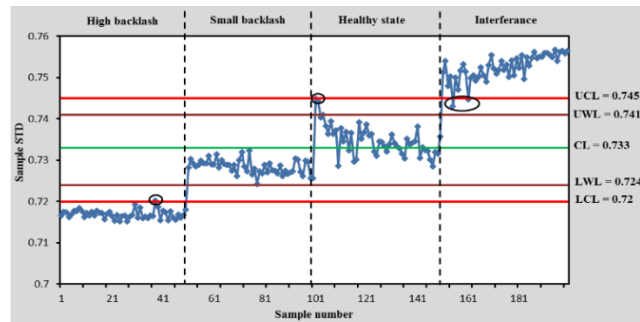


Figure 13. Fault detection result from the designed embedded system

To simulate faults similar to the natural fault development standard, by which a machine deteriorates gradually from its healthy condition, it would have been better if the backlash fault could be developed progressively while the robot is executing the pick and place task, which currently cannot be achieved. One possible idea is to fix a controlled device to the backlash adjustment grub-screws on the robot and programme it to increase or decrease the backlash level concurrently during the robot task execution. However, this is out of the scope of this research, but could be applied in future work. Thus, in this work to present something similar to gradual deterioration fault, first from the robot healthy condition the backlash was changed (increased or decreased) and ten measurement points were taken; then the power was disconnected from the robot arm and also the backlash was changed and another ten points were captured. Each time the grub-screw was turned by a quarter turn, where each turn is equivalent to change the backlash by 0.072 mm. Figure 14 shows the captured result when the robot was healthy (backlash = 0.68 mm) and the backlash progressively increased (each time by 0.018 mm) until the excessive backlash is reached (backlash = 0.92 mm), while Figure 15 presents the result of decreasing the backlash (also each time by 0.018 mm) starting from the robot healthy condition until gears interference was introduced (backlash = 0.56 mm). A total of 140 points were captured in the case of backlash increasing, whereas 80 points were captured in the case of backlash decreasing, as a higher number of steps are required to obtain excessive backlash than obtaining the gear interference.

It can be clearly seen from these figures that the resultant STD values are gradually decreased, as the backlash is increased, and increased, as the backlash is decreased. Obviously, the resultant STD values indicated by the samples 1 to 30 in Figure 14 and to 20 in Figure 15 are randomly distributed around the center line (CL), signaling a healthy robot condition. This is, however, at the healthy backlash range, which is a little higher or lower 0.68 mm. In both cases when a fault was detected, which was at sample 38 in Figure 15 (as eight consecutive points under the CL) and sample 23 in Figure 16 (as three consecutive points above the UWL), the embedded system showed a message to the user flagging that a fault was detected, Figure 16. From the result was concluded that the designed control chart is sensitive enough to the small changes in the backlash level.

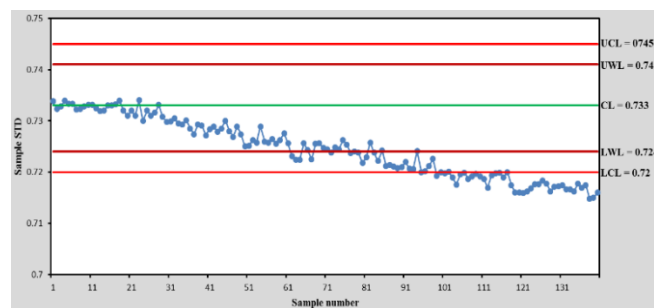


Figure 14. Fault detection result from the designed embedded system when the backlash was gradually increased

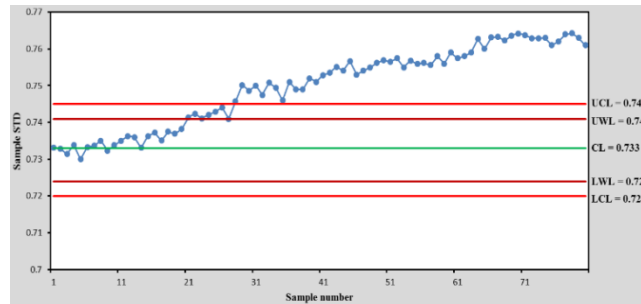


Figure 15. Fault detection result from the designed embedded system when the backlash was gradually decreased

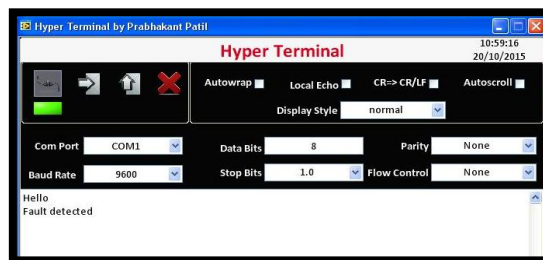


Figure 16. A screenshot from the serial communication software indicating detection of a fault in the robot

## 8. CONCLUSION

This paper has presented a methodology for design and implementation of a robot fault detection system based on statistical control charts using wireless embedded system. The design approach and requirements of hardware selection for developing a wireless vibration monitoring system that can be utilized for machines health monitoring was discussed. The Arduino DUE was selected since it was appropriate as a core of the wireless sensor unit. The wireless capability was put on the DUE board by using Arduino wireless shield. A comparison between the existing wireless technologies like Wi-Fi, Bluetooth and ZigBee was presented and it was established that the ZigBee wireless transmission consumes less power and is very reliable when it operates within the recommended range. The XBee wireless module, which is based on ZigBee protocol, was employed to establish the communication between the sensor node and the base station. Enclosures for the sensor node and base station hardware were developed relying on commercially available boxes, and then the final system installation and testing illustrated. In order to demonstrate the validation of the developed wireless embedded system, captured vibration data were compared to their peers that were captured from a healthy robot using NI 6009 DAQ. Then, the utilized robot was programmed to execute a handling task that mimics one of its real tasks and with different backlash faults seeded inside. Vibration signals from the three axes of measurements were captured from the robot while it is performing this task and the different statistical features were extracted and evaluated. The signals standard deviations (STDs) were used to distinguish between the healthy and faulty robot conditions, as it was found to be the best fault sensitive feature.

Joint 3 of the robot was selected in order to simulate different faults in the robot. A detailed description about the mechanical construction of the selected joint along with the different faults that may accompany the robot power transmission system has been discussed. The obtained STDs from the healthy signals have been utilized to establish the threshold limits in the control chart that can be applied in order to realise the abnormality development in the robot. Minitab software was used to establish the chart, and then C programming language was used to develop the designed chart on the Arduino board. The SCC showed very good capability in detecting changes in the robot health condition. However, the main advantage of control charts is that their design requires only data from the machine healthy conditions, which is applicable in situations where faulty condition data are lacking whereas their main disadvantage is that they are only able to detect the damage, rather than its quantification and location. Finally, this type of test could also be used by the robot's manufacturers for checking the backlash level before putting the robots in the service. However, the designed system needs more development that could make it more intelligent. Also, only one robot was monitored throughout this study, therefore, the hardware and software could be developed to

accommodate monitoring of more than a robot. This can be achievable by building and using more than one wireless sensor node and updating the designed embedded software to be able to communicate with more than one sensor node.

## REFERENCES

- [1] H. G. Park and M. Zak, "Gray-box approach for fault detection of dynamical systems," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 125, pp. 451-454, 2003.
- [2] H. E. Kim, "Machine Prognostics Based on Health State Probability Estimation," *Ph.D Thesis*, Queensland University of Technology, 2010.
- [3] S. Butler, "Prognostic Algorithms for Condition Monitoring and Remaining Useful Life Estimation," *Ph.D Thesis*, National University of Ireland, Maynooth, 2012.
- [4] S. X. Ding, "Model-Based Fault Diagnosis Techniques," Germany, Springer, 2008.
- [5] A. K. S. Jardine, *et al.*, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, pp. 1483-1510, 2006.
- [6] S. K. Yadav and P. K. Kalra, "Condition monitoring of internal combustion engine using EMD and HMM," vol. 275, pp. 167-185, 2010.
- [7] W. Verdonck, *et al.*, "Experimental robot identification: Advantages of combining internal and external measurements and of using periodic excitation," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 123, pp. 630-636, 2001.
- [8] J. Shiroishi, *et al.*, "Bearing condition diagnostics via vibration and acoustic emission measurements," *Mechanical Systems and Signal Processing*, vol. 11, pp. 693-705, 1997.
- [9] E. Kisić, *et al.*, "Fault Detection in Electric Power Systems Based on Control Charts," *Serbian Journal Of Electrical Engineering*, vol. 10, pp. 73-90, 2013.
- [10] N. Baydar, *et al.*, "Detection of incipient tooth defect in helical gears using multivariate statistics," *Mechanical Systems and Signal Processing*, vol. 15, pp. 303-321, 2001.
- [11] Z. Zhang, *et al.*, "Fault analysis and condition monitoring of the wind turbine gearbox," *IEEE Transactions on Energy Conversion*, vol. 27, pp. 526-535, 2012.
- [12] S. A. Niknam, *et al.*, "Analysis of acoustic emission data for bearings subject to unbalance," *International Journal of Prognostics and Health Management*, vol. 4, 2013.
- [13] W. Zhou, *et al.*, "Bearing fault detection via stator current noise cancellation and statistical control," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 4260-4269, 2008.
- [14] W. Wang and W. Zhang, "Early defect identification: Application of statistical process control methods," *Journal of Quality in Maintenance Engineering*, vol. 14, pp. 225-236, 2008.
- [15] G. J. Feng, *et al.*, "Implementation of envelope analysis on a wireless condition monitoring system for bearing fault diagnosis," *International Journal of Automation and Computing*, vol. 12, pp. 14-24, 2015.
- [16] N. K. Verma, *et al.*, "Android app for intelligent CBM," in *IEEE International Symposium on Industrial Electronics*, 2013.
- [17] R. Faludi, "Building Wireless Sensor Networks," O'Reilly Media, Inc, 2011.
- [18] S. H. Yang, "Wireless Sensor Networks," Principles, Design and Applications, 2014.
- [19] J. K. Sinha, "Vibration Analysis, Instruments, and Signal Processing," First Edition ed., CRC Press, 2014.
- [20] A. Albarbar, *et al.*, "Suitability of MEMS accelerometers for condition monitoring: An experimental study," *Sensors*, vol. 8, pp. 784-799, 2008.
- [21] M. D. P. Emilio, "Embedded Systems Design for High-Speed Data Acquisition and Control," Springer, 2015.
- [22] H. H. Yan and Y. Rahayu, "Design and Development of Gas Leakage Monitoring System Using Arduino and ZigBee," in *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2014)*, Yogyakarta, Indonesia, pp. 207-212, 2014.
- [23] C. Townsend, *et al.*, "Wireless Sensor Networks: Principles and Applications," in *Sensor Technology Handbook*, vol. 1, 2005.
- [24] S. Giannoulis, *et al.*, "Wireless sensor network technologies for condition monitoring of industrial assets," in *IFIP Advances in Information and Communication Technology* vol. 398, pp. 33-40, 2013.
- [25] V. Bhandari and P. Abrol, "Field Monitoring of Treated Industrial Waste Water," *International Journal of Electrical and Computer Engineering*, vol. 4, pp. 237-242, 2014.
- [26] A. A. Jaber and R. Bicker, "Design of a Wireless Sensor Node for Vibration Monitoring of Industrial Machinery," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, pp. 639-653, 2016.
- [27] I. Buchmann, "What's the Best Battery?" 2003.
- [28] D. C. Montgomery and G. C. Runger, "Applied Statistics and Probability for Engineers," Sixth Edition ed., Wiley, 2014.
- [29] R. M. Khan, "Problem Solving and Data Analysis using Minitab," John Wiley & Sons, Ltd, 2013.
- [30] M. A. S. El-Din, *et al.*, "Statistical Process Control Charts Applied to Steelmaking Quality Improvement," *International Journal of Quality Technology and Quantitative Management*, vol. 3, pp. 473-491, 2006.