Taylor & Francis
Taylor & Francis Group

International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: http://www.tandfonline.com/loi/tprs20

Data cleansing for energy-saving: a case of Cyber-Physical Machine Tools health monitoring system

Changyi Deng, Ruifeng Guo, Chao Liu, Ray Y. Zhong & Xun Xu

International Journal of Production Research, 2017
https://doi.org/10.1080/00207543.2017.1394596

Data cleansing for energy-saving: a case of Cyber-Physical Machine Tools health monitoring system

Changyi Deng[a,b,c], Ruifeng Guo[a,c], Chao Liu[b], Ray Y. Zhong[b] and Xun Xu[b]*

[a]*Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang, China;* [b]*Department of Mechanical Engineering, The University of Auckland, Auckland, New Zealand;* [c]*Shenyang Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing, China*

(*Received 26 June 2017; accepted 27 September 2017*)

Cyber-Physical Production Systems (CPPS) often use wireless sensor networks (WSNs) for monitoring purposes. However, data from WSNs may be inaccurate and unreliable due to power exhaustion, noise and other issues. In order to achieve a reliable and accurate data acquisition while ensuring low energy consumption and long lifetime of WSNs, data cleansing algorithms for energy-saving are proposed in this research. The cleansing algorithms are computationally lightweight in local sensors and energy-efficient due to low energy consumption in communications. Dynamic voltage scaling and dynamic power management are adopted for reducing energy consumption, without compromising the performance at system level. A low-power protocol for sink node communication is proposed at network level. A health monitoring system for a Cyber-Physical Machine Tool (a typical example of CPPS) is designed. Experiment results show that the proposed energy-saving data cleansing algorithm yields high-performance and effective monitoring.

Keywords: data cleansing; energy-saving; Cyber-Physical Production Systems (CPPS); Cyber-Physical Machine Tools (CPMT); Cyber-Physical Systems

1. Introduction

Cyber-Physical Systems (CPS) are enabling technologies which bring the virtual and physical worlds together to create a truly networked world where intelligent objects can communicate and interact with each other (Kumar et al. 2016; Lee, Bagheri, and Kao 2015; Liao et al. 2017; Nayak et al. 2016; Wang, Törngren, and Onori 2015). Cyber-Physical Production Systems (CPPS) relying on the advanced CPS make full use of information and communication technologies in a production environment (Monostori 2014; Xu 2012), and this new type of production system is also called Cyber-Physical Production Network which has the ability to control the networked production system easily (Mladineo, Veza, and Gjeldum 2017). As a typical example of CPPS, Cyber-Physical Machine Tools (CPMT) have been proposed as the next generation of machine tools which deeply integrate machine tool, machining processes, computation and networking (Liu and Xu 2017; Xu 2017). A CPMT owns a Cyber Twin (the digital abstraction of machine tool) that is equipped with computational and networking capabilities, allowing real-time feedback loops to be established in which machining processes can affect computations and vice versa (Liu et al. 2017). At component level, once the sensory data from critical components (e.g. spindle and slide ways) is gathered, the Cyber Twin of each component analyses the data in order to achieve autonomous functions such as self-awareness and self-prediction (Bagheri et al. 2015). In system integration and control, transfers of motion, energy and data among different modules of machine tools are carried out (Gadalla and Xue 2017). Furthermore, similarities between the current performance and historical information of a machine tool can be mined to predict its future behaviour. Thus, CPMT becomes intelligent, resilient and self-adaptable (Lee, Bagheri, and Kao 2015; Wang 2013).

Real-time status monitoring is a key function for CPMT and it requires monitoring of not only the operating status of the servomotor, but also other key components such as spindles, screws and rails. With the rapid development of wireless transmission technology, Ad Hoc network technology and low-power sensing technology, the new wireless sensor network (WSN) as the core of CPS (Gamba, Tramarin, and Willig 2010) can achieve an intelligent monitoring system for machine tools. The raw data collected by WSNs however may become inaccurate and unreliable as sensors' battery power drops (Huang, Thareja, and Shin 2006; More and Raisinghani 2017). In addition, an unstable network also introduces noise or outliers. Such noise can be categorised into four types: incomplete, inaccurate, duplicated, and

*Corresponding author. Email: xun.xu@auckland.ac.nz

missing data (Zhong et al. 2013; Zhong et al. 2014). Therefore, sensor data gathering requires suitable mechanisms such as data cleansing algorithms to ensure the reliability and accuracy. Another critical issue lies in energy consumption. In order to maximise the lifecycle of a sensor, the circuits, architecture, algorithms and network protocols need to be designed in energy-efficient manners. At system-level, dynamic voltage scaling (DVS) and dynamic power management (DPM) may be deployed to reduce energy consumption by setting the sensors into a different energy-saving mode or even shutting down depending on the level of urgency of the event (Pughat and Sharma 2015). Consequently, it is necessary to develop an algorithm that provides reliable data, tracks the changes of a machine status, infers additional knowledge from historical information, and passes the outcomes to the next level. An approach within the framework of Design Science Research Methodology and prototyping has been proposed to address the challenge of modelling, simulation and data analytics in manufacturing (Jain, Shao, and Shin 2017). Zhong, Wang, and Xu (2017) proposed a multi-objective hybrid artificial bee colony (MOHABC) algorithm for service composition and optimal selection in cloud manufacturing. This paper considers both the Quality of Service and the energy consumption from the perspectives of economy and environment. In fact, Small and Medium Enterprises do not exploit all the resources for implementing Industry 4.0 and there is still a lack of real applications in the field of production planning (Moeuf et al. 2017).

In this paper, we developed a low-power and reliable Health Monitoring System for Cyber-Physical Machine Tools (HMS-CPMT), to address two research questions: (1) How to ensure a long lifetime of a sensor while maintaining a low energy consumption? and (2) How to provide reliable real-time manufacturing data for local analysis as well as for the cloud server? Firstly, a data cleansing algorithm for energy-saving (DCAES) was proposed to guarantee lightweight computing and energy efficiency in local nodes. Secondly, an energy-saving scheduling algorithm for sensors (ESSAS) was developed based on DVS and DPM for reducing power consumption without significant performance degradation at the system level. Furthermore, a low-power protocol for sink node communication was proposed at network level.

The rest of this paper is organised as follows: Section 2 reviews the related work. The model overview and design are in Section 3, which describes the framework of the HMS–CPMT as well as the task model, energy model and data cleansing model. Section 4 presents the realisation of the proposed energy-saving data cleansing algorithm in the CPMT. Performance evaluations of the proposed algorithm are conducted in Section 5. Section 6 concludes this paper with key findings and future work.

2. Related work

2.1 *Data cleansing*

A key characteristic of CPMT is the ability to collect accurate and reliable data from machine tools and their critical components. The data might be directly measured by sensors or obtained from a controller. On one hand, there are various types of data, a seamless and tether-free method to manage data acquisition procedure and transferring data to local and cloud server is necessary with specific protocols such as MTconnect (Vijayaraghavan et al. 2008) and OPC-UA (Hannelius, Salmenpera, and Kuikka 2008). On the other hand, an effective data cleansing algorithm for processing raw data is required (Lee, Bagheri, and Kao 2015).

Owing to the natural properties of the sensors and the uncertainties of the environmental conditions, data loss usually happens during the data acquisition processes; the ratio of the acquired data compared to the total generated data is around 60–70%. (Wang et al. 2014). In order to obtain sufficient data, sensors need to be sufficiently deployed to sample data periodically. However, duplicated samplings are usually generated when the sensors are intensively distributed (Lu et al. 2017). Sending duplicated data to a cloud server results in time delays as well as a significant waste of network resources such as energy (Bashir et al. 2011). It is, therefore, both wise and necessary to eliminate the redundant and unreliable data at the sensor end by a cleansing approach. The objective of cleansing data method is to reduce energy consumption and time delays, i.e. less redundant data, less power consumption and less time delay in the network.

There are two types of time and space-related sensor data cleansing technologies: centralised cleansing and in-network cleansing. For centralised data cleansing, Jeffery proposed a method with temporal and spatial correlation of the perceived data to recover the missing data and remove the isolated points (Jeffery et al. 2006). Zhuang et al. (2007) proposed a weighted moving average algorithm to reduce the energy consumption of the local node samples and to increase the response of the perceived data by using the combination of the local node and the neighbour node test. However, centralised cleansing methods cannot meet the real-time requirements because of the large amount of perceptual data being sent to the sink nodes for centralised processing (Lei et al. 2016). In addition, the energy consumption due to the transmission of large amounts of data also makes the centralised cleaning methods difficult to implement. Xu et al. (2013) proposed a hierarchical framework for abnormal detection in WSNs, where data is pre-processed in sensors before being sent to the cloud server. However, the correlations among different sensor nodes are not discussed to recognise the event outliers. In order to address this issue, neighbourhood voting method was used for fault or event outlier detection (Yim and Choi 2010). Unfortunately, the accuracy of these methods was constrained by the number of neighbour nodes. Branch et al. (2013) provided an in-network outlier detection algorithm which was derived from neighbourhood voting algorithm without considering the power consumption.

2.2 *Energy saving*

Energy efficiency has been a crucial problem when applying sensors in CPS (Zhang and Wang 2006). Wireless devices have a stringent requirement of power consumption. Sensors in most of the existing networks have limited battery. It is crucial to manage the power consumption at the design and planning stage (López et al. 2009). Majority of studies on energy-efficient routing only focuses on a portion of, rather than the overall, system (Heo, Hong, and Cho 2009).

Energy optimisation has been widely studied in CPS. High-performance applications usually consume more energy that can lead to reduction of the battery lifetime and cripple the whole network (Rountree et al. 2009). In fact, the peak computing is not always required as the processor's operating frequency can be dynamically scaled based on the workload. The goal of DVS is to manage the power supply and operating frequency. And DPM is another effective technology for reducing system power consumption without significant performance degradation. The basic idea is to turn off devices or components when they are not needed (Deng et al. 2017). Then, energy-saving algorithms put the system into a sleep mode when slack time is long enough. In fact, the sleep mode transition and wake-up also have the switching energy overhead and also take a certain period of time.

Literature has studied the main techniques of saving-energy (Jiang et al. 2007; Khan, Qureshi, and Iqbal 2015). DVS has been discussed in Sausen et al. (2008) and interplay of both DPM and DVS has been considered. Also, some scheduling algorithms have been discussed (Dgharkut and Lohiya 2014; Kulau, Büsching, and Wolf 2013). An algorithm has been proposed to reduce the power consumption in the processing unit through 'undervaluing', where the powering electric circuits were lower than the specified voltage levels (Wu and Yang 2005). However, this technique depends on the circuit state and has operational difficulties. This threshold is unpredictable because it is not versatile for other platforms. Besides, the risk of instantaneous errors grows with lower clock rates. Unlike the aforementioned methods, this paper proposes a novel lightweight in-network approach for sensor data cleansing with the correlations of time and space.

3. A health monitoring system for Cyber-Physical Machine Tools

3.1 *System framework*

The framework of the HMS–CPMT is presented in Figure 1 which comprises three parts. The bottom part is the key components of a machine tool. The middle part is the key body where sensor network, local monitoring system, acquisition nodes and sink nodes are deployed. It sets up a wireless network for data collection. In the cloud services part, a remote health monitoring system is developed to supervise the system by making full use of the captured big data from machine tools.

HMS–CPMT is based on wireless sensors and includes four layers (Figure 2): physical layer, cyber layer, computing platforms layer and application layer. In the WSN layer, the acquisition nodes are attached to the spindle and the ball screw to obtain vibration and temperature data (Zhong, Wang, and Xu 2017). The acquisition nodes cleanse the received data and remove the noisy information before sending it to the sink node (Zhong et al. 2015). MTConnect is an international standard based on an open protocol for data integration, which strives to strengthen the data acquisition capabilities of devices and towards developing a plug-and-play component to reduce the cost (Vijayaraghavan et al. 2008). Figure 2 shows a schematic diagram of HMS–CPMT with MTConnect integration. The data are encapsulated into MTConnect format and sent to local CNC system by the sink nodes. HMS–CPMT is equipped with a wired connected router for the link between sinks and local CNC system in order to ensure the security, reliability and real-time performance of data communications in the manufacturing environment. The sink nodes can also send vibration and temperature information to remote a health diagnosis system via the GPRS module to facilitate remote maintenance. Finally, HMS–CPMT is designed to use data mining algorithms in the cloud services layer.

- Physical layer. The basic physical structure system consists of 'physical elements'. The key mechanical parts and hydraulic systems that affect the operation of the machine are monitored, such as the wear or damage of the bearings, gears, screws and the automatic tool changer.
- Cyber layer. The cyber level acts as a key information hub where data is passed from each connected machine to the machine's network. With massive data gathered, specific analytics are used to extract additional information to

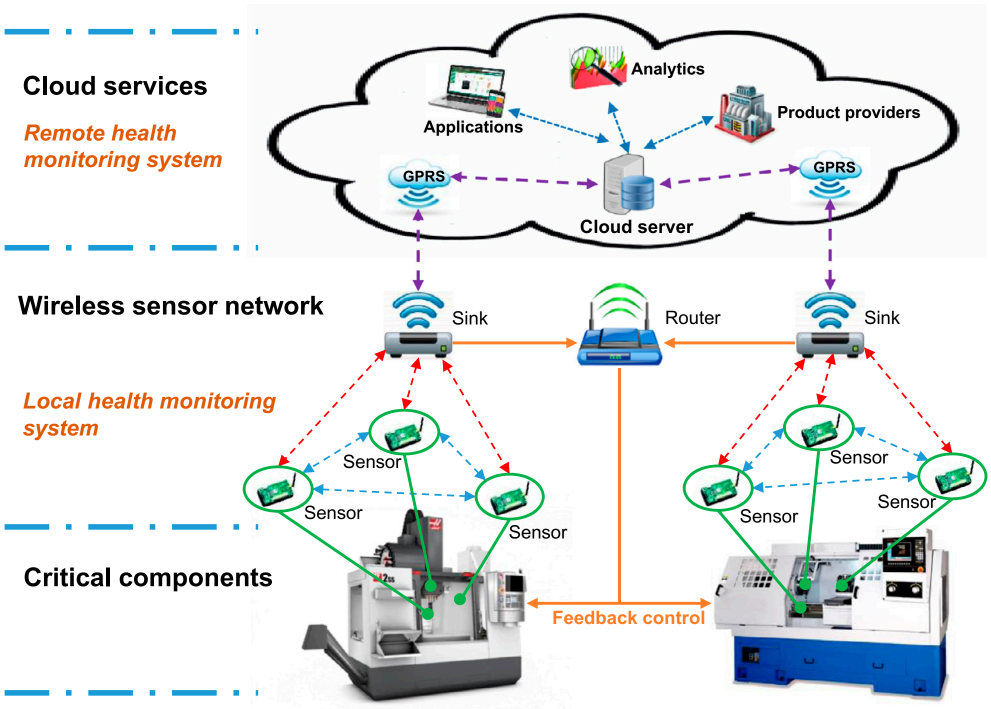


Figure 1. Framework of health monitoring system for CPMT.

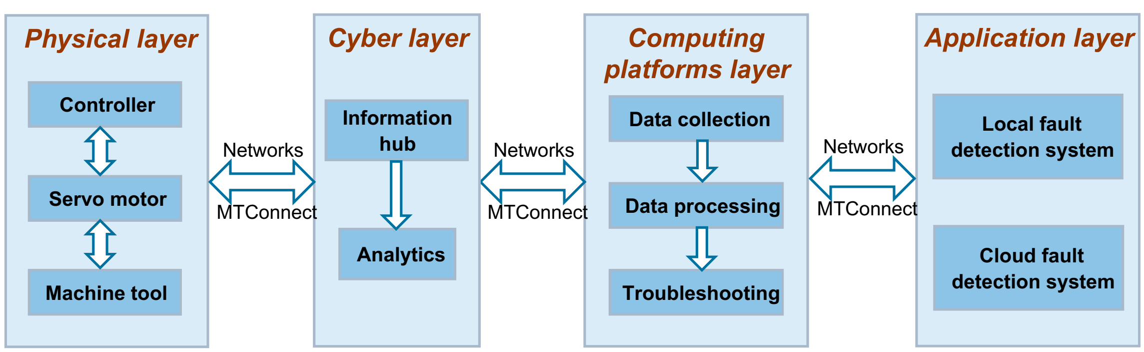


Figure 2. Model of health monitoring system for CPMT.

observe over the status of individual machines. The network can compose the 'cyber' part of CPS and the sensors are connected via a communication network, e.g. Bluetooth or ZigBee.

- Computing platforms layer. Aiming to examine a typical fault and friction of the machine tool system, the health diagnosis of the key components such as spindle, gear box and cutting tool is carried out by the combination of performance parameter, trend analysis and unit detection. The in-network DCAES is given for data collection, data processing and troubleshooting at this level.
- Application layer. The performance parameters, fault types and safety prediction of components are used for machine tool monitoring. The parameter database, dynamic database, historical database and knowledge base are used to realise an effective management of health information. Each machine tool has its own health records, which are updated using a data-driven algorithm. The health conditions of the machine tool are feedback to the local CNC system for real-time adaptive control and remote health diagnosis (Lee, Kao, and Yang 2014).

3.2 *Task model*

Based on the characteristics of data processing, each sensor node is assigned with three categories of tasks, i.e. sensing tasks, processing tasks and transmission tasks. These tasks are carried out periodically. Considering a period task set, $J = \{J_1, J_2, \ldots, J_n\}$, the worst-case total utilisation of task set is U_p. J_i can be represented by a three-tuple (AC_i, C_i, P_i). AC_i is the actual execution time of J_i. C_i is the worst-case execution time at the maximum processor speed. P_i is the minimum period between the two consecutive executions of instances. D_i is the relative deadline within which the system must guarantee the task response. We assume that D_i is equal to P_i for all tasks. The execution speed is assumed to have a linear relationship with the task execution time. If a task J_i scheduled on a different node, a communication between these nodes is required. In such a case, J_i cannot be executed until the communication is finished and the result of J_i is received. However, if all tasks are assigned on a same node, the result of communication delay is considered to be zero and J_i can be executed after J_m is completed.

3.3 *Energy model*

The energy consumption of a sensor for transmitting and receiving l-bit data over a distance d that is less than a threshold d_o, which are defined as $E_{tr}(l, d)$ and $E_{re}(l)$, respectively (Tian, Ekici, and Ozguner 2005):

$$E_{tr}(l, d) = E_{el} \cdot l + \varepsilon_{amp} \cdot l \cdot d^2 \quad (1)$$

$$E_{re}(l) = E_{el} \cdot l \quad (2)$$

where d means the transmission distance between sensors, E_{el} and ε_{amp} are both hardware parameters (Basu, Ke, and Little 2003; Giannecchini, Caccamo, and Shih 2004). When a sensor executing N clock cycles at CPU frequency f and supply voltage V_d, the energy consumption E_{com} is given as:

$$E_{com}(V_d, f) = N \cdot C \cdot V_d^2 + V_d \cdot (I_o \cdot e^{\frac{V_d}{n \cdot V_T}}) \cdot \left(\frac{N}{f}\right) \quad (3)$$

$$f \cong K \cdot (V_d - c) \quad (4)$$

where V_T is the thermal voltage and C, I_o, K and c are corresponding dependent parameters (Basu, Ke, and Little 2003).

The mode basically describes the power consumption at different levels. Each node can be in an active, idle or sleep mode as shown in Figure 3, which includes a specific combination of different component powers (Sinha and Chandrakasan 2001). Each node has two active modes: receive and transmit. These modes are activated directly using Sensor

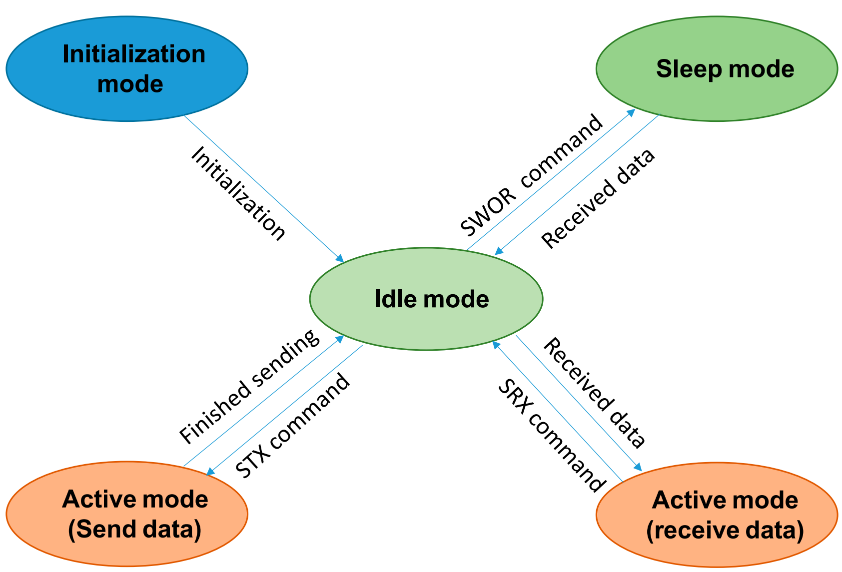


Figure 3. The sensors are switched in different mode.

Receive Mode (SRX) and Sensor Transmit Mode (STX) commands defined by the device provider, or automatically by Sensor Wake-On-Radio (WOR). When enabling STX, the modulator will start transmitting; when enabling SRX, the modulator will start receiving. SWOR functionality enables the radio to periodically wake up from sleep mode and listen for incoming packets with minimal CPU interaction. Table 1 lists the component power including three different modes, i.e. active mode, idle mode and sleep mode. Sleep mode has two options (normal and deep), which are selected based on the actual operating conditions of the sensors. A deeper sleep mode is characterised by increased latency and reduced power consumption. In general, the deeper sleep mode, the less power consumed and the longer wake-up time.

3.4 *Data cleansing model*

In a WSN, the current value of a perceived node compared with the previous period has a larger fluctuation, which introduces large uncertainties which require more information from the neighbouring nodes. Elastic space is the adjacent space that changes with the fluctuation of the perceived data under ensuring that each node has high spatial correlation (Fang, Dobson, and Hudges 2013). The elastic space model is defined as follows:

$$SP = R \cdot e^{\alpha \cdot \Delta^2} \tag{5}$$

where SP is the size of the neighbouring space, indicating the highest correlation of neighbouring nodes. R is the global correlation of the current neighbourhood space ($0 < R \leq 1$). e is a mathematical constant. Δ indicates the amount of change in the measured value, which is the difference between the current measured value and the previous period value. α indicates the fluctuation adjustment parameter ($\alpha > 0$). The value of α depends on the reliability of the data to be obtained. To ensure the reliability of the data, set γ as the lower bound of the elastic space. To prevent the elastic space from being too large, set δ as the upper bound of the elastic space. When the elastic space exceeds the upper bound δ, the data from the sensor is identified as unreliable data by the local node.

4. Realisation of data cleansing algorithm for energy-saving

There are two phases in the DCAES. The first is the in-network data cleansing algorithm for energy-saving (IDCAES), which can complete the data cleansing work, taking into account the proximity of the neighbour correlation. The second is the ESSAS, which is used to schedule periodic sampling task with lower power consumption.

4.1 *Data exchange standard: MTConnect*

Modern manufacturing systems usually comprise manufacturing devices from different providers who use their proprietary communication protocols (Koren, Wang, and Gu 2017). Hence, the data in a manufacturing system usually varies from machine to machine. That causes difficulties in data exchange, integration and management in the CPMT. MTConnect, which is a lightweight, open and extensible protocol designed for the exchange of data between shop-floor equipment and software applications, is a feasible solution for addressing the difficulties. MTConnect allows for disparate entities in a manufacturing system along with their associated devices to share data (Vijayaraghavan et al. 2008). It is built upon the most prevalent standards in the manufacturing and software industry including eXtensible Markup Language (XML) and Hypertext Transport Protocol (HTTP).

MTConnect provides a detailed data model for field-level manufacturing equipment. It defines the vocabulary and semantics of manufacturing data, enabling a unified definition such as name, units, values and context. MTConnect standard also provides a machine-readable XML schema which defines a hierarchical information model for a machine tool. The hierarchical structure enables the data related to the same component to be grouped together and bound to that component. For example, the temperature and vibration data of a spindle acquired from different sensors can be grouped

Table 1. The sensors are switched in different mode.

Serial number	Mode	Sensor, analogue–digital converter	Radio
S_0	Active	On	T_x, R_x
S_1	Idle	On	R_x
S_2	Sleep	On/Off	R_x

Note: T_x = transmit, R_x = receive.

together so that all relevant data items can be retrieved by a single command instead of inquiring each data item separately. Figure 4 shows an example of using MTConnect information model for a spindle motor of a machine tool.

4.2 *In-network data cleansing algorithm for energy-saving*

We assume sensor m can be denoted by (x, y, z), which represents the location of sensor in space. $N(m)$ is the neighbour set of sensor m. In the data cleansing algorithm, the closer the two nodes are, the greater correlation between them will be. The algorithm uses Gaussian Radial Basis Function (GRBF) to measure the correlation according to Euclidean distance between the two nodes (Wang, Wang, and Ma 2007). The following definitions are proposed.

Definition 1: $dis(m, \ n)$ is the Euclidean distance between the node m and n. The value of $dis(m, n)$ is used to find out the outlier.

$$dis(m,n) = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2} \tag{6}$$

Definition 2: GRBF is used to measure the spatial correlation between nodes, $r(m, n, \beta)$ is denoted the spatial correlation.

$$r(m,n,\beta) = e^{\frac{-dis(m,n)^2}{2\beta^2}} \tag{7}$$

where e is a constant and β is the width parameter of the function. By adjusting the parameter β, the far distance and low correlation of data can be eliminated in the data cleansing process. In order to balance the spatial correlation of the entire neighbourhood space, the method can calculate the correlation average between each node and the local master m as follows:

$$R(m) = \left(\sum_{n \in N(m)} r(m,n,\beta) \right) / |N(m)| \tag{8}$$

where $R(m)$ is the global correlation of the neighbourhood space. $N(m)$ is the neighbour sensor set of m, $|N(m)|$ represents the number of neighbours; $n \in N(m)$, which is the neighbouring sensor of m.

Set the lower bound γ and the upper bound δ of the elastic space, the acquisition sensor data removal noise and outliers recognition method is as follows:

(1) Store the acquisition data with MTConnect format, from which to obtain the time of the sensor acquisition data and value. When the current data are a large fluctuation compared with the previous ones, neighbouring sensors eliminate the uncertainty of the measured values.

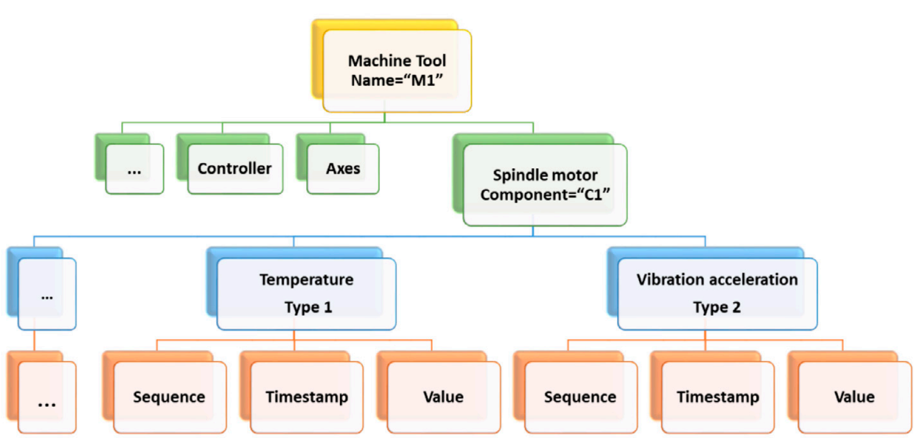


Figure 4. A paradigm hierarchy for HMS–CPMT with MTConnect.

(2) If the neighbour space *SP* is larger than the sensor's measured value, the weighted average of the neighbourhood is used to remove the noise:

$$\bar{x}(m,t) = x(m,t) + \left(\sum_{n \in N(m)} x(n,t) \cdot r(n,m,\beta) \right) / |N(m)| \tag{9}$$

$x(n, t)$ is the original measured value of sensor n at time t. $\bar{x}(m,t)$ is the correction value of sensor m at time t. $r(n, m, \beta)$ can be calculated by Equation (7), which is the weight of the measured value of the node n.

(3) If the neighbourhood space *SP* is less than or equal to δ, the measured value and the nearest neighbour measurement of the weighted average to remove the noise following:

$$\bar{x}(m,t) = \frac{x(m,t) + \sum_{n \in N(m)} x(n,t) \cdot r(n,m,\beta)}{|N(m)| + 1} \tag{10}$$

Based on the definitions, the proposed DCAES–IDCAES is as follows:

1. *Select a master sensor m_r from acquisition sensors*
2.   *Calculate* dis (m, n) and $r(m, n, \beta)$; *// find the optimal neighbour space.*
3. ***If** sensors are not selected, **then***
4.   *Sensors are going into sleep mode until the wake-up*;
5. ***If** m_r obtained a significant deviation from the previous data, **then***
6.   *The data is obtained from the neighbour sensor and the data is manipulated*;
7. *Input the elastic space lower γ and the upper δ, current measured value and previous period correction value*;
8. ***For** (node i = 0; k < N(m); k++)*
9. {
10. *Calculate m_r of neighbourhood spatial correlation R_i, Δ_I, and SP_i*;
11.   ***If** $SP_i > \delta$*
12. $N_i = \delta$;
13.   ***Else If** $SP_i < \gamma$*
14. $N_i = \gamma$;
15.   ***Else***
16. $N_i = SP_i$;
17. }
18. *Obtain the data of n neighbouring sensors with the highest spatial correlation $SP_{\max}$*;
19.   ***IF** $SP_{\max} > \delta$*
20.   *Calculate the weighted average of neighbour sensors*;
21.   ***Else***
22.   *Calculate local nodes and neighbour sensors to measure the weighted average*;
23. *Update the energy consumption of m_r by Equation (3)*;
24. *Put other sensors into sleep mode waiting to wake-up*;
25. *Send data to the sink sensor*;

In the above algorithm, lines 1–6 calculate the size of the neighbouring space. Lines 7–17 determine whether the adjacent space beyond the elastic space. Lines 18–22 find data in outliers. Line 25 sends the cleansed data to the sink node. Since the master sensor needs to communicate with other neighbouring sensors, the time performance of the algorithm mainly depends on the communication delay.

Due to the discrete feature of task mapping and scheduling, a schedule may yield some slack time before the deadline. The unbalanced workload of sensors and the communication scheduling also yields CPU slack time. In the DVS phase, the CPU slack time can be exploited by scaling CPU frequency to reduce energy consumption. When the slack is long enough, DPM can be used to decide CPU into different sleep mode.

DCAES–ESSAS

1. $S = S_{min}$, *t is the current scheduling point.*
2. *For all event job i = 1 to all do*
3. *Sort task set with deadline task in DQ*
4. *End for*
5. ***While** true do {*
6. ***If** J_k is activated, **then***
7. *J_k is scheduled and update $rem_i(t)$;*
8. ***If** J_k is finished, **then***
9. *Remove J_k from RQ;*
10. *Calculates the available slack time ST for the task J_k;*
11. ***If** $Rq \neq \varnothing$, **then***
12. $S_i = w(t)_i/(ST + U_i)$
13. ***If** no task is activated and $\mathrm{Rq} = \Phi$, **then***
14. {
15. *Calculates the slack_time;*
16. *System into idle mode;*
17. ***If** A task is active(external), **then***
18. *System comes back to active mode;*
19. ***If** slack_time > t_0, **then***
20. *System into sleep-state;*
21. ***If** no task is active in the sleep-state, **then***
22. *System into deep-sleep-state;*
23. ***If** task is coming(external), **then***
24. *System comes back to sleep-mode;*
25. ***If** the task is valid event, **then***
26. *System comes back to active mode;*
27. }
28. ***End While**}*
29. *Update the energy consumption of all active node by Equation (3);*

Firstly, the processor initialises the minimum speed (line 1). All tasks are sorted by descending order of the deadline and added to the deadline queue DQ (lines 2–4). Secondly, when a task is completed early, the slack time is recycled and the task is removed from the Run Queue (RQ). The processor speed is scaled down, which is used for the rest task (lines 6–12). Finally, DPM technology can be used to determine sensors into the general sleep mode or deeper sleep mode depending the length of the slack time (lines 13–26), which can achieve the collaborative optimisation scheduling with DVS and DPM. Time complexity of DCAES–ESSAS mainly comes from sorting tasks and recycling slack time, which is $O(n)$.

4.3 *Low-power protocol for sink node communication*

The sink sends the collected data in two ways. The first is the local CNC system with real-time feedback control machine tools by a wire router. The second is the cloud services for remote health diagnosis by GPRS. Machine tools network achieves ad hoc protocol, which is a three-way handshake communication between the sink and the acquisition node (Wheeler 2007). We design an array which can collect network nodes to manage the network. As shown in Figure 5, the sink receives the command confirmation packet from the destination. When a node cannot work properly due to circuit failure or battery depletion, the sink needs to remove these 'dead' nodes from the ad hoc network.

In the implementation of network protocol, the Tick_Check function is used to find an array element whose member variable Max_Time is equal to Ticks and Active is 1 (networked state) in the node structure array. As these 'dead' nodes are removed from the ad hoc network by the sink, the Max_Time member variable of the array elements will be overwritten. Every time the processors enter into a hardware timer interrupt, Tick_Check function will be executed once. Since the trigger cycle of hardware timer interrupt is set to 10 ms, that means the sink nodes check whether there is a network failure node every 10 ms.

4.4 *An example*

As an example, Table 2 shows some sample data. There are one master node 0 and three nodes to demonstrate DCAES–IDCAES. $\bar{X}(t_1)$ is the previous period value at time t_1 and $X(t_2)$ is the current measured value at t_2. Δ is the

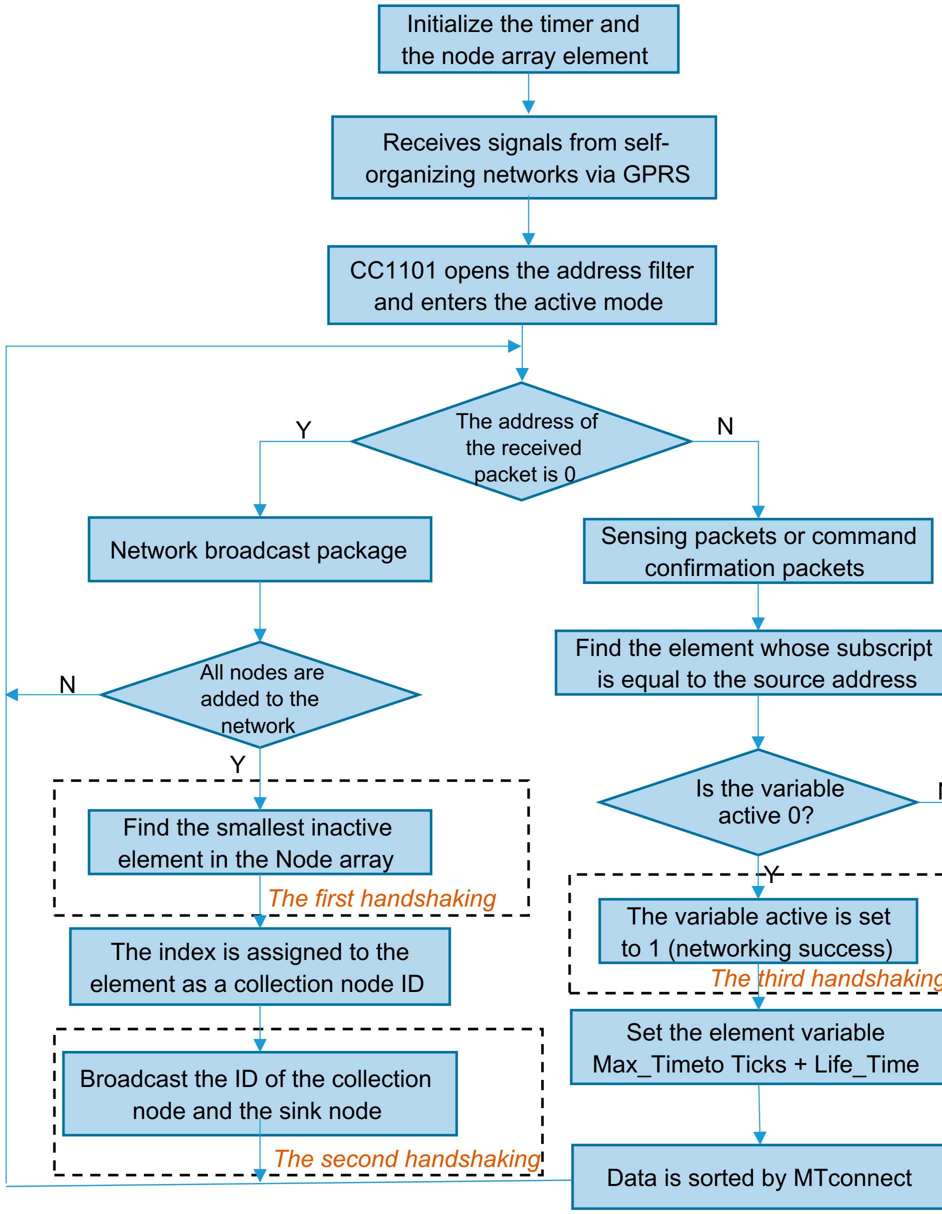


Figure 5. Implement of the low-power protocol of the sink node.

Table 2. A sample data for data cleansing algorithm.

Node	$\bar{X}(t_1)$	$X(t_2)$	Δ	R	SP	$\bar{X}(t_2)$
0	35.56355	32.86434	−2.69921	0.94	35.909142	34.044352
1		36.26367	0.70012	0.95		
2		35.58566	0.02211	0.98		
3		36.86362	1.30007	0.89		
0	32.56366	33.86368	1.30002	0.935	2.1767159	31.06079
1		31.54269	−1.02097	0.95		
2		33.16363	0.59997	0.98		
3		31.36356	−1.2001	0.89		

difference between the current measured value and the previous period value. Based on the experimental results and statistical analysis, we set $R = 1$ ($0 < R \leq 1$), $\alpha = 1$ and $\Delta^2 = 2.6$, so the upper bound of the elastic space δ is 14 according to Equation (5). Similarly, the lower bound of the elastic space γ is calculated as 2 when we set $R = 0.5$, $\alpha = 1$ and $\Delta^2 = 1$. When the current measured value $X(t_2)$ has a large fluctuation compared with the previous value $\bar{X}(t_1)$, neighbouring nodes are used to eliminate the uncertainty of the measured values. Since the first neighbour space $SP = 35.909142$ is larger than the upper bounds of the elastic space 14, the weighted average of the neighbourhood is used to remove the noise by Equation (9). $r(n, m, \beta)$ can be calculated by Equation (7), which is the weight of the measured value of the node n, here $R(1) = 0.95$, $R(2) = 0.98$, $R(0.89)$. $R(0)$ is average of other nodes R (1, 2, 3) and $\bar{X}(t_2)$ is amended as 34.044352 by DCAES-IDCAES.

When the neighbourhood space $SP = 2.1767159$ is less than the lower bounds of the elastic space $\delta = 2$, the measured value of node 0 and the other neighbour measurement of the weighted average are used to remove the noise by Equation (10), and $\bar{X}(t_2)$ is amended as 31.06079.

5. Experiments and discussions

This section uses a vibration fault to test the proposed algorithm. The radial drilling machine is a common type of machine tool, which can be used for drilling, reaming and modifying the scraping face. DCAES–IDCAES is used to cleanse the vibration data that can effectively predict the product's pass rate. Z3050X16 radial drilling machine is used in this experiment. A/B two-arm radial drilling machine spindle boxes are selected as the test samples. Group A spindle box was tested by special testing equipment, which meets the precision machining requirements. Group B spindle box did not pass the testing, which was identified as an improper spindle assembly.

In the experiment, low-power embedded processor MSP430 and wireless RF Transceiver CC1101 are adopted for HMS-CPMT. The sink node integrates a heterogeneous gateway which can communicate with each node. The data format follows MTConnect, which stores data in the MTConnect agent repository. Table 3 shows a sample request returns in the form of a number of vibration information readings from Z3050X16.

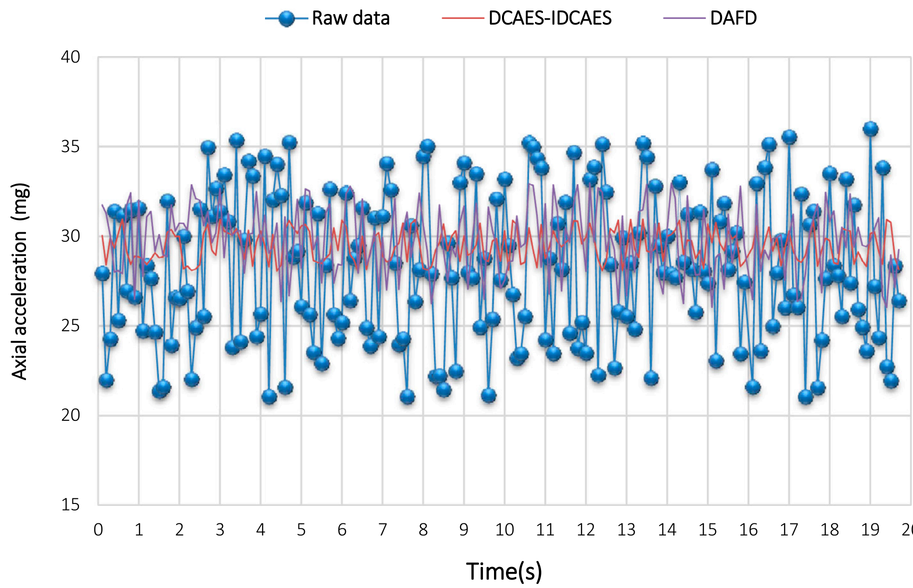
5.1 *Data cleansing experiment*

This section compares the data cleansing algorithm for energy-saving (DCAES–IDCAES) with the demand-based adaptive fault detectors (DAFD) (Wang et al. 2014), which can cleanse redundant data while maintain the integrity of original data.

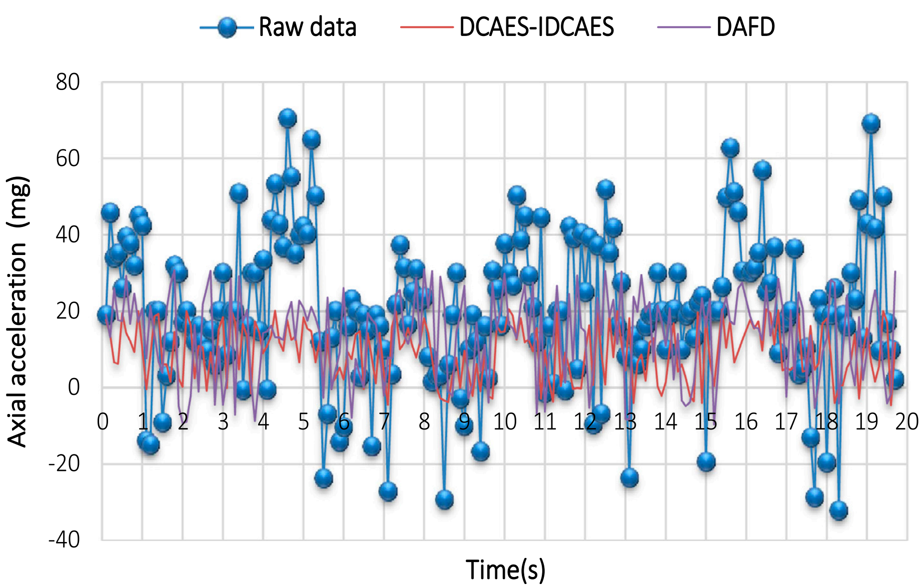
In the experiment, the sampling frequency of the vibration acquisition node is among 1–8 Hz, and sampling period is 0.1 s. The number of total acceleration samples of each spindle box is 200 times. Samples work in 2000 rpm/s under group A and B for X-axis acceleration data sampling. We use the upper and lower bounds of the elastic space from Section 4.4, and set the upper to 14 and the lower to 2, respectively, and the spatial correlation of 9 nodes closest to the local node is {0.98, 0.86, 0.83, 0.76, 0.72, 0.68, 0.62, 0.44, 0.38}. Finally, two groups of spindle data are obtained, and the raw sampling data and cleansing data are shown in Figure 6.

Table 3. A sample from Z3050X16 with MTConnect.

Machine	Component	Type	Sequence	Timestamp	Value
M1	C1	Vibration acceleration	11	2017-04-26T02:02:36.483034	35.56367
			12	2017-04-26T02:02:37.594045	34.14571
			13	2017-04-26T02:02:37.693075	28.65562
			14	2017-04-26T02:02:37.804056	28.07524
			15	2017-04-26T02:02:37.915051	34.77705
			16	2017-04-26T02:02:38.034055	33.96236
			17	2017-04-26T02:02:38.151053	27.49765
			18	2017-04-26T02:02:38.294055	32.25249
			19	2017-04-26T02:02:38.402056	22.29572
			20	2017-04-26T02:02:38.514058	34.59113
			21	2017-04-26T02:02:38.637054	21.29226
			22	2017-04-26T02:02:38.754035	26.30168
			23	2017-04-26T02:02:38.898050	21.79475
			24	2017-04-26T02:02:39.004045	35.56367



(a) Waveforms of axial vibration acceleration in group A

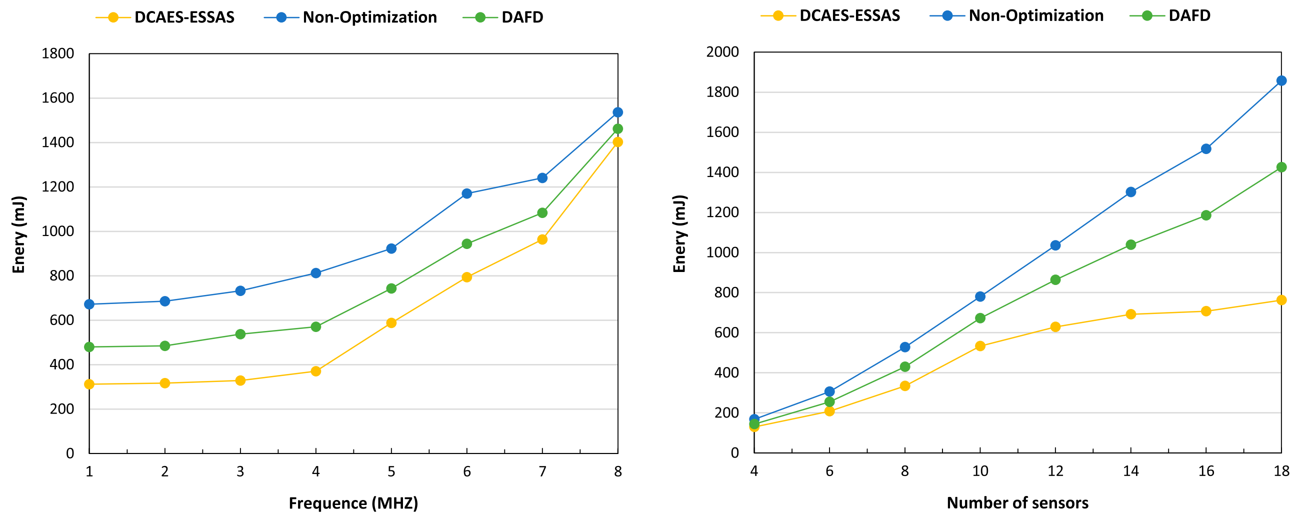


(b) Waveforms of axial vibration acceleration in group B

Figure 6. The axial vibration acceleration waveform in the normal state and improper state.

It can be seen from Figure 6(a) that the axial vibration acceleration waveform in the normal state is a mild fluctuating curve. The average value of the 200 raw samples is 28.6 mg and the standard deviation is about 4.32 mg. While the axial vibration acceleration waveform is a violent curve of volatility in the group B, the average value of 200 raw samples is 22.57 mg and the standard deviation of about 18.69 mg.

From the data map, it shows that DCAES–IDCAES can improve the data compressibility and accuracy more effectively compared with DAFD. The vibration acceleration of group A is fluctuating and after applying DCAES–IDCAES to eliminate the noise, the waveform is more stable than B group. By comparing the vibration acceleration data and waveforms in two different groups, it can be seen that the axial acceleration of B group with improper spindle assembly is bigger (the mean value) than the normal state during its motion. And the fluctuation is larger (variance big) as shown in Figure 6(b). The reason is that DCAES–IDCAES cleanses data by exploiting two-phase data cleansing model. In the first phase, sensor nodes preliminarily filter out potential errors in the raw data collected by themselves. In the second



(a) Energy consumption of 12 sensors with frequency

(b) Energy consumption with the number of sensors

Figure 7. The average energy cost consumed by sensors in each round of monitoring.

phase, the master nodes eliminate the potential outliers of the data from the first phase by applying the elastic space on the sensor nodes. The local CNC system or remote health diagnosis system can determine whether the failure occurred by analysing the above vibration information. Thus, the lathe operator can replace the spindle box in a more reasonable way.

5.2 *Energy consumption of HMS-CPMT*

On the basis of the above experiment, Energy Trace was adopted for getting the energy usage of MSP430 processors (Ra et al. 2012). There were 10 acquisition nodes and 2 sink nodes in the HMS-CPMT. Each acquisition frequency was tested by 60 s.

The energy consumption of 12 sensors is shown in Figure 7(a). With the increasing of acquisition frequency, energy consumption of each algorithm is increased. DCAES–ESSAS algorithm performs better at the beginning of the acquisition (1–4 MHz), and DCAES–ESSAS algorithm yields better performance because the lower frequency, more slack time it generates. Thus, more processors can be scaled, which consumes less energy. However, as the acquisition frequency increases by 5–8 MHz, the difference among the algorithms gets smaller. This is because the higher frequency, the less available slack time. Energy consumption of DCAES–ESSAS is always lower than DAFD, indicating that the proposed algorithm is better in the energy consumption. The reason is that DCAES–ESSAS adopts DVS and DPM in the system level to manage energy consumption. Experimental results show that DCAES can save the energy consumption on average by 19.48% over DAFD.

The energy consumption of DCAES-ESSAS algorithm is lower than that of DAFD and non-optimisation with the increase of the number of sensors in Figure 7(b). There are several reasons. Firstly, DCAES–IDCAES algorithm can cleanse data in the local nodes, which can reduce the energy consumption for communication. Therefore, when there are 10 sensors in the system, the energy consumption of DCAES is significantly lower than that of DAFD and no-optimisation algorithm. This is because it is not necessary for all sensors to keep running mode. DCAES–ESSAS communicates with 8.6 sensors on average. Secondly, DCAES–ESSAS algorithm can dynamically adjust the processor frequency for reducing energy consumption meanwhile ensuring their deadlines. In addition, at the system level, the non-working sensors can enter a different sleep mode depending on the workload, which can further reduce energy consumption. Finally, the optimisation of the sink node network protocol can also reduce the energy consumption. Additionally, DCAES–ESSAS adopts DVS and DPM in the system level to manage energy consumption, which is a good performance compared with DAFD. Although DAFD can reduce redundant data to save energy and reduce time delay, DAFD is too conservative without consideration of saving energy at the task and system level, or even the network level. The results show that the average energy consumption of DCAES-ESSAS algorithm is 33.58% less than DAFD, and 58.16% less than that of the non-optimisation.

6. Conclusions

In this paper, a HMS–CPMT with WSNs was proposed by considering data cleansing, task scheduling and network protocol. The aim is to achieve low-power and reliable data acquisition. Two algorithms have been proposed for energy-saving and data cleansing in a Cyber-physical machine tool. Contributions are summarised as follows.

(1) The developed HMS–CPMT offers high reliability and low power consumption. Sensors are embedded with computational and networking capabilities, allowing real-time feedback loops so that machining processes can affect computations and vice versa.
(2) The data cleansing algorithm for energy-saving (DCAES–IDCAES) can achieve data cleansing with the proximity of the neighbour correlation using lower energy consumption in local sensors.
(3) The energy-saving scheduling algorithm for sensor (DCAES–ESSAS) is used to schedule the period task from sensors with lower power consumption. DVS and DPM are effective technologies for reducing system power consumption without significant performance degradation at the system level.
(4) Finally, a low-power protocol for sink node communication is proposed. It can send the cleansed data to the local CNC system and the cloud service for remote health diagnosis.

The proposed DCAES–IDCAES can be utilised in different types of WSN, allowing more accurate and reliable data for further Big Data Analytics. Based on these reliable data, local and cloud-based machine health monitoring and analysis can be realised. Furthermore, various machine learning algorithms can be applied based on the proposed HMS-CPMT, thus the machining performance can be improved, while the production cost can be reduced.

The system is data-hungry when diagnosis is performed. The relevant eigenvalue extraction and data mining algorithms need to be further studied. Industrial Big Data Analytics and related mining algorithms are also necessary in HMS–CPMT so as to make the system more intelligent. The future work includes the further development of the cloud application based on this platform and the servitisation of CPMT.

Acknowledgments

The authors are grateful to the researchers at the Industry 4.0 Laboratory for Smart Manufacturing Systems, the University of Auckland.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China [grant number 2014ZX04014-021]; China Scholarship Council; University of Auckland Joint Scholarship.

References

Bagheri, Behrad, Shanhu Yang, Hung-An Kao, and Jay Lee. 2015. "Cyber-physical Systems Architecture for Self-aware Machines in Industry 4.0 Environment." *IFAC-PapersOnLine* 48 (3): 1622–1627.

Bashir, Ali Kashif, Se-Jung Lim, Chauhdary Sajjad Hussain, and Myong-Soon Park. 2011. "Energy Efficient In-network RFID Data Filtering Scheme in Wireless Sensor Networks." *Sensors* 11 (12): 7004–7021.

Basu, Prithwish, Wang Ke, and Thomas D. C. Little. 2003. "Dynamic Task-based Any Casting in Mobile Ad Hoc Networks." *Mobile Networks and Applications* 8 (5): 593–612.

Branch, J. W., C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta. 2013. "In-network Outlier Detection in Wireless Sensor Networks." *Knowledge and Information Systems* 34: 23–54.

Deng, C. Y., R. F. Guo, X. Xu, R. Y. Zhong, and Z. Y. Yin. 2017. "A New High-performance Open CNC System and Its Energy-aware Scheduling Algorithm." *International Journal of Advanced Manufacturing Technology* 93: 1513–1525.

Dgharkut, M. Ali, and P. Lohiya. 2014. "Scheduling Task of Wireless Sensor Network Using Earliest Deadline First Algorithm." *International Journal of Scientific Research in Computer Science and Engineering* 2 (2): 1–6.

Fang, Lei, Simon Dobson, and Danny Hudges. 2013. " An Error-free Data Collection Method Exploiting Hierarchical Physical Models of Wireless Sensor Networks." In *Proceedings of the 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, Barcelona, 81–88.

Gadalla, Moustafa, and Deyi Xue. 2017. "Recent Advances in Research on Reconfigurable Machine Tools: A Literature Review." *International Journal of Production Research* 55 (5): 1440–1454.

Gamba, Giovanni, Federico Tramarin, and Andreas Willig. 2010. "Retransmission Strategies for Cyclic Polling over Wireless Channels in the Presence of Interference." *IEEE Transactions on Industrial Informatics* 6 (3): 405–415.

Giannecchini, Simone, Marco Caccamo, and Chi-Sheng Shih. 2004. "Collaborative Resource Allocation in Wireless Sensor Networks." In *Proceedings of Euromicro Conference on Real-time Systems (ECRTS'04)*, 35–44. Sicily.

Hannelius, Tom, Mikko Salmenpera, and Seppo Kuikka. 2008. "Roadmap to Adopting OPC UA." In *Proceedings of the 6th IEEE International Conference on Industrial Informatics*, 756–761. Daejeon.

Heo, Junyoung, Jiman Hong, and Yookun Cho. 2009. "EARQ: Energy Aware Routing for Real-time and Reliable Communication in Wireless Industrial Sensor Networks." *IEEE Transactions on Industrial Informatics* 5 (1): 3–11.

Huang, C. T., S. Thareja, and Y. J. Shin. 2006. "Wavelet-based Real Time Detection of Network Traffic Anomalies." In *Proceedings of the 2006 IEEE Securecomm and Workshops*, 1–7. Baltimore, MD.

Jain, Sanjay, Guodong Shao, and Seung-Jun Shin. 2017. "Manufacturing Data Analytics Using a Virtual Factory Representation." *International Journal of Production Research* 55 (18): 5450–5464.

Jeffery, Shawn R., Gustavo Alonso, Michael J. Franklin, Wei Hong, and Jennifer Widom. 2006. "Declarative Support for Sensor Data Cleaning." In *Proceedings of 4th International Conference on Pervasive Computing*, 83–100. Dublin.

Jiang, Xiaofan, Jay Taneja, Jorge Ortiz, Arsalan Tavakoli, Prabal Dutta, Jaein Jeong, David Culler, Philip Levis, and Scott Shenker. 2007. "An Architecture for Energy Management in Wireless Sensor Networks." *ACM SIGBED Review* 4 (3): 31–36.

Khan, Junaid Ahmed, Hassaan Khaliq Qureshi, and Adnan Iqbal. 2015. "Energy Management in Wireless Sensor Networks: A Survey." *Computers & Electrical Engineering* 41: 159–176.

Koren, Y., W. Wang, and X. Gu. 2017. "Value Creation through Design for Scalability of Reconfigurable Manufacturing Systems." *International Journal of Production Research* 55 (5): 1227–1242.

Kulau, Ulf, Felix Büsching, and Lars Wolf. 2013. "A Node's Life: Increasing WSN Lifetime by Dynamic Voltage Scaling." In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems*, 472–473. Cambridge, MA.

Kumar, Ajay, Ravi Shankar, Alok Choudhary, and Lakshman S. Thakur. 2016. "A Big Data MapReduce Framework for Fault Diagnosis in Cloud-based Manufacturing." *International Journal of Production Research* 54 (23): 7060–7073.

Lee, Jay, Behrad Bagheri, and Hung-An Kao. 2015. "A Cyber-physical Systems Architecture for Industry 4.0-based Manufacturing Systems." *Manufacturing Letters* 3: 18–23.

Lee, J., H. A. Kao, and S. Yang. 2014. "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment." *Procedia CIRP* 16: 3–8.

Lei, Jianjun, Haiyang Bi, Ying Xia, Jun Huang, and Haeyoung Bae. 2016. "An In-network Data Cleaning Approach for Wireless Sensor Networks." *Intelligent Automation & Soft Computing* 22 (4): 599–604.

Liao, Yongxin, Fernando Deschamps, Eduardo de Freitas Rocha Loures, and Luiz Felipe Pierin Ramos. 2017. "Past, Present and Future of Industry 4.0 – A Systematic Literature Review and Research Agenda Proposal." *International Journal of Production Research* 55 (12): 3609–3629.

Liu, Chao, Sheng Cao, Wayne Tse, and Xu Xun. 2017. "Augmented Reality-assisted Intelligent Window for Cyber-physical Machine Tools." *Journal of Manufacturing Systems*. Available online publication 44 (2): 280–286.

Liu, Yongkui, and Xun Xu. 2017. "Industry 4.0 and Cloud Manufacturing: A Comparative Analysis." *Journal of Manufacturing Science and Engineering* 139 (3): 034701.

López, Tomás Sánchez, Daeyoung Kim, Gonzalo Huerta Canepa, and Koudjo Koumadi. 2009. "Integrating Wireless Sensors and RFID Tags into Energy-efficient and Dynamic Context Networks." *The Computer Journal* 52 (2): 240–267.

Lu, S. P., C. Xu, R. Y. Zhong, and L. H. Wang. 2017. "A RFID-enabled Positioning System in Automated Guided Vehicle for Smart Factories." *Journal of Manufacturing Systems* 44: 179–190.

Mladineo, Marko, Ivica Veza, and Nikola Gjeldum. 2017. "Solving Partner Selection Problem in Cyber-physical Production Networks Using the HUMANT Algorithm." *International Journal of Production Research* 55 (9): 2506–2521.

Moeuf, Alexandre Pellerin, Robert Lamouri, Samir Tamayo-Giraldo, Simon Barbaray, and Rodolphe Barbaray. 2017. "The Industrial Management of SMEs in the Era of Industry 4.0." *International Journal of Production Research.*: 1–19. doi:10.1080/00207543.2017.1372647.

Monostori, L. 2014. "Cyber-physical Production Systems: Roots, Expectations and R&D Challenges." *Procedia CIRP* 17: 9–13.

More, A., and V. Raisinghani. 2017. "A Node Failure and Battery-aware Coverage Protocol for Wireless Sensor Networks." *Computers & Electrical Engineering*. Available online publication. ISSN 0045-7906. doi:10.1016/j.co-mpeleceng.2016.12.024

Nayak, Ashutosh, Rodrigo Reyes Levalle, Seokcheon Lee, and Shimon Y. Nof. 2016. "Resource Sharing in Cyber-physical Systems: Modelling Framework and Case Studies." *International Journal of Production Research* 54 (23): 6969–6983.

Pughat, Anuradha, and Vidushi Sharma. 2015. "A Review on Stochastic Approach for Dynamic Power Management in Wireless Sensor Networks." *Human-Centric Computing and Information Sciences* 5 (1): 4.

Ra, Moo-Ryong, Bodhi Priyantha, Aman Kansal, and Jie Liu. 2012. "Improving Energy Efficiency of Personal Sensing Applications with Heterogeneous Multi-processors." In *Proceedings of the 14th International Conference on Ubiquitous Computing*, 1–10. Pittsburgh, PA.

Rountree, B., D. K. Lownenthal, B. R. de Supinski, M. Schulz, V. W. Freeh, and T. Bletsch. 2009. "Adagio: Making Dvs Practical for Complex Hpc Applications." In *Proceedings of the 23rd International Conference on Supercomputing*, 460–469. New York.

Sausen, Paulo Sérgio, José Renato de Brito Sousa, Marco Aurélio Spohn, Angelo Perkusich, and Antônio Marcus Nogueira Lima. 2008. "Dynamic Power Management with Scheduled Switching Modes." *Computer Communications* 31 (15): 3625–3637.

Sinha, Amit, and Anantha Chandrakasan. 2001. "Dynamic Power Management in Wireless Sensor Networks." *IEEE Design & Test of Computers* 18 (2): 62–74.

Tian, Yuan, Eylem Ekici, and Fusun Ozguner. 2005. "Energy-constrained Task Mapping and Scheduling in Wireless Sensor Networks." In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 8. Washington, DC.

Vijayaraghavan, A., W. Sobel, A. Fox, D. Dornfeld, P. Warndorf. 2008. "Improving Machine Tool Interoperability Using Standardized Interface Protocols: MT Connect." In *Proceedings of the 2008 International Symposium on Flexible Automation (ISFA), 2008*, Atlanta, GA, USA.

Wang, Lihui. 2013. "Machine Availability Monitoring and Machining Process Planning towards Cloud Manufacturing." *CIRP Journal of Manufacturing Science and Technology* 6 (4): 263–273.

Wang, Lihui, Martin Törngren, and Mauro Onori. 2015. "Current Status and Advancement of Cyber-physical Systems in Manufacturing". *Journal of Manufacturing Systems* 37 (Part 2): 517–527.

Wang, Xue, Sheng Wang, and Jun-Jie Ma. 2007. "An Improved Particle Filter for Target Tracking in Sensor Systems." *Sensors* 7 (1): 144–156.

Wang, Li, Li Da Xu, Zhuming Bi, and Yingcheng Xu. 2014. "Data Cleaning for RFID and WSN Integration." *IEEE Transactions on Industrial Informatics* 10 (1): 408–418.

Wheeler, Andrew. 2007. "Commercial Applications of Wireless Sensor Networks Using ZigBee." *IEEE Communications Magazine* 45 (4).

Wu, Jie, and Shuhui Yang. 2005. "Energy-efficient Node Scheduling Models in Sensor Networks with Adjustable Ranges." *International Journal of Foundations of Computer Science* 16 (01): 3–17.

Xu, Xun. 2012. "From Cloud Computing to Cloud Manufacturing." *Robotics and Computer-integrated Manufacturing* 28 (1): 75–86.

Xu, Xun. 2017. "Machine Tool 4.0 for the New Era of Manufacturing." *The International Journal of Advanced Manufacturing Technology.* 92 (5–8): 1893–1900.

Xu, L., Y. R. Yeh, Y. J. Lee, and J. Li. 2013. "A Hierarchical Framework Using Approximated Local Outlier Factor for Efficient Anomaly Detection." *Procedia Computer Science* 19: 1174–1181.

Yim, S. J., and Y. H. Choi. 2010. "An Adaptive Fault-tolerant Event Detection Scheme for Wireless Sensor Networks." *Sensors* 10: 2332–2347.

Zhang, Lei, and Zhi Wang. 2006. "Integration of RFID into Wireless Sensor Networks: Architectures, Opportunities and Challenging Problems." In *Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops*, 463–469. Changsha.

Zhong, R. Y., Q. Dai, T. Qu, G. Hu, and G. Q. Huang. 2013. "RFID-enabled Real-time Manufacturing Execution System for Mass-customization Production." *Robotics and Computer-Integrated Manufacturing* 29 (2): 283–292.

Zhong, R. Y., G. Q. Huang, Q. Y. Dai, and T. Zhang. 2014. "Mining SOTs and Dispatching Rules from RFID-enabled Real-time Shopfloor Production Data." *Journal of Intelligent Manufacturing* 25 (4): 825–843.

Zhong, R. Y., G. Q. Huang, S. L. Lan, Q. Y. Dai, X. Chen, and T. Zhang. 2015. "A Big Data Approach for Logistics Trajectory Discovery from RFID-enabled Production Data." *International Journal of Production Economics* 165: 260–272.

Zhong, R. Y., L. H. Wang, and X. Xu. 2017. "An IoT-enabled Real-time Machine Status Monitoring Approach for Cloud Manufacturing." *Procedia CIRP* 63: 709–714. doi:10.1016/j.procir.2017.03.349.

Zhuang, Yongzhen, Lei Chen, X. Sean Wang, and Jie Lian. 2007. "A Weighted Moving Average-based Approach for Cleaning Sensor Data." In *Proceedings of 27th International Conference on Distributed Computing Systems (ICDCS '07)*, 38–38). Toronto.