Cyber-Physical System Implementation for Manufacturing With Analytics in the Cloud Layer

Mahmoud Parto - George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology
Pedro Daniel Urbina Coronado – School of Engineering and Sciences, Tecnologico de Monterrey
Christopher Saldana - George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology
Thomas Kurfess - Oak Ridge National Laboratory

Abstract

Effective and efficient modern manufacturing operations require the acceptance and incorporation of the fourth industrial revolution, also known as Industry 4.0. Traditional shop floors are evolving their production into smart factories. To continue this trend, a specific architecture for the cyber-physical system is required, as well as a systematic approach to automate the application of algorithms and transform the acquired data into useful information. This work makes use of an approach that distinguishes three layers that are part of the existing Industry 4.0 paradigm: edge, fog, and cloud. Each of the layers performs computational operations, transforming the data produced in the smart factory into useful information. Trained or untrained methods for data analytics can be incorporated into the architecture. A case study is presented in which a real-time statistical control process algorithm based on control charts was implemented. The algorithm automatically detects changes in the material being processed in a computerized numerical control (CNC) machine. The algorithm implemented in the proposed architecture yielded short response times. The performance was effective since it automatically adapted to the machining of aluminum and then detected when the material was switched to steel. The data were backed up in a database that would allow traceability to the line of g-code that performed the machining. [DOI: 10.1115/1.4051663]

Keywords

Internet of things, industry 4.0, fog computing, edge computing, data analytics, smart factory, computer aided manufacturing, cyber-physical system design and operation, cybermanufacturing, industrial internet of things, manufacturing automation

1. Introduction

In the current manufacturing landscape, machines can produce and share data about their operations. Sensor and data transmission technology is ubiquitous and low cost with ever-improving sample quality and rates, resulting in low cost and high-quality manufacturing data. However, most of the generated data are not fully leveraged due to two major issues. First, the data do not follow a standard structure, even though numerous standards and protocols exist such as MTConnect (for machine tools) [1,2], and open platform communications united architecture (OPC UA) [3], respectively. Nevertheless, the industry still operates using old CNC machines, legacy equipment, and other sensors that do not adhere to any standard data structure. Second, once obtained, the data need to be transformed into useful information, a process that is not automated and sometimes without a clear framework.

The literature on the subject can be divided into two categories. In the first category, several case studies are addressing the incorporation of artificial intelligence to the data generated during machining processes [4]. Techniques such as neural networks (NNs), support vector machine [5], and random forests have been success- fully implemented to predict tool wear using data from milling operations [6]. The cutting parameters for high-speed turning are predicted efficiently using machine learning methods [5]. Works in this category present an implementation of machine learning techniques to specific problems related to machining operations; however, they do not present a clear unified model to implement systematically a path from data collection to information generation. In the second category, cyber-physical systems (CPSs) are explained through models [7]. The 5C model comprises the levels of connection, data conversion, cyber, cognition, and configuration [8,9]. The work of Monostori et al. [10] presents a comprehensive CPS framework with some case studies related to the use of OPC/UA and other smart connection capabilities. The work in this category, while providing the architecture, communication protocols, and other structure towards a model for CPS, do not provide the link between these protocols and an automated data-driven information generation system. Table 1 showcases works related to manufacturing and the use of fog computing (FC), which is one of the most recent elements that is receiving attention as key in the transformative process of data to information. The table synthesizes the role of the edge and cloud layer considered in their respective frameworks, if automatic analytics is included, and the general purpose of the application. Related to the architecture for CPS, there are strong arguments on the advantages of using edge and FC versus cloud computing [14]. Some others propose even more categories (other than edge, fog, and cloud) as is the case of the fluid manufacturing architecture [16] (which included mist and dew). Most of the works clearly distinguish the elements in the edge layer such as sensors and machine tools (those that produce data). Most of the works also recognize the functions that the cloud usually performs which are storage (database) and big data analytics. Study cases have been presented with the layers edge, fog, and cloud not completely distinguishable from each other [19]. The works that better provide a clear definition of the CPS architecture [12,15] considering the layers of edge, fog, and cloud did not explicitly manifest how to implement automated analytics for the generated data.

This work proposes an approach with three main objectives. The first objective is to allocate the computational resources to perform a data-driven generation of information on the layers

of edge, fog, and cloud. This is an original implementation that, while similar to models already available in the literature [12,15], adds an message queue telemetry transport (MQTT) broker and a database in the fog layer that buffers data in case of interruptions of communication with the cloud. The second objective is to present a model for a CPS incorporating the standards of communication available in the industry and an automated unsupervised classifier implementation based on the Nelson rules [20] for control charts (CCs), a convenient and widely used method from the industry. The third objective is to display the implementation of the model in a case study that implements the recognition of the material being processed by a CNC machine. The general aim of this work is to build the CPS architecture with the layers of fog, edge, and cloud, implement in this architecture efficient automated analytics, and put all of this into practice with data produced by machine tools. The main contribution of this work is the presentation of a framework enabling manufacturing operations to utilize current data streams and effectively integrate new sources of digital data to generate information.

The organization of the manuscript is as follows: Sec. 2 describes the structural CPS model developed in this work, Sec. 3 presents the real-time statistical process control (RT-SPC), and the case study is described in Sec. 4. Results are presented in Sec. 5 and the discussion of the results can be found in Sec. 6. Section 7 contains the conclusion of this research and Sec. 8 provides the identified future work.

2. Model for Cyber-Physical Systems

CPSs are the intersection and interaction between the digital and physical environment and associates them with analytics to improve the efficiency of industrial systems [10,21,22]. CPS architecture contains a variety of layers. The selection and configuration of these layers have an impact on the performance of the system, for example, in terms of latency and processing rates [23]. One of the layers of CPS applied to manufacturing purposes is the edge. In edge computing (EC), data processing occurs as close as possible to the data source (e.g., the sensor). The accepted convention is that EC is an umbrella term comprising FC, mobile edge computing (MEC), and cloudlet computing [24–26]. Nevertheless, considering the contrasts between EC (especially MEC) and FC [27], it is useful to distinguish a separate layer in the CPS for FC, a fog layer. The fog (where the clouds touch the ground/edge) layer takes data from the edge layer and centralizes it in the local area network and, when necessary, with additional processing and storage. It also determines which data and information are shared with or transferred to the cloud. The cloud represents another layer in the CPS.

In this work, a CPS for manufacturing systems was designed with the three layers presented in the block diagram in Fig. 1. The edge layer in Fig. 1(a) contains the machine tools, embedded systems (e.g., microcontroller units, PLCs, microprocessor units), and other data sources from automated systems. Modern machine tools are typically equipped to transmit the data produced in the machining process using protocols such as MTConnect. MTConnect relies on a structure composed of an adapter and agent. The latter communicates the data related to the machining process and the machine status to the outside via an XML file with a structure optimized to be read by an interpreter. Once interpreted, the data can be sent

through a gateway with a communication protocol. In this case, the application layer protocol of communication, MQTT, is ideal to transmit data because of its lightweight and low-latency characteristics. The components of the edge layer act as clients in MQTT communication. Certainly, besides MTConnect and MQTT, OPC/UA is another relevant protocol for machine-to-machine communication [28] in the edge layer; however, it was not implemented in this particular CPS.

The fog layer in Fig. 1(b) contains a local MQTT broker that handles communication between the components of the edge layer and the components of the cloud layer. The fog layer contains a database that buffers data while offline (interrupted communication with cloud) and transmits the data once online (communication with cloud reestablished). The fog layer also performs statistical analysis and applies machine learning models to the data.

The cloud layer shown in Fig. 1(c) has the functions of maintaining a database log of the raw data coming from the fog layer, processing data, and training/applying the algorithms (and maintains a database of the generated data) and produces analytics based on user-defined statistics. The results update the fog layer and web applications (client-server-based applications that run in a browser) that are used to communicate the information to users and other systems. While some of the functions can be performed either in the cloud or in the other layers (local), there are powerful reasons why the cloud can be of benefit. The cloud has the characteristic that can be accessed remotely, that is the reason why it can offer services [16] to several users at the same time, for example, accessibility to dashboards, monitoring, or the possibility of collaborating from different geographical spaces. Those services depending on the cloud platform can be “pay as you go” [12] which can include storage and computing capabilities without the need to invest in local resources. Among the limitations of the cloud layer are that the services are interrupted with internet access intermittencies, there are high latencies, an increase in network traffic load, and the risk of cyber-attacks or concerns on data privacy [14].

These three layers constitute the general architecture of the CPS. The components of each of the layers can be modified to adapt to any particular purpose. A demonstration of this architecture working with an unsupervised self-learning algorithm is presented in the ensuing sections. The algorithm is primarily based on the first Nelson rule (1NR) and CCs, which have applications in manufacturing shop floors [29]. The implementation of this heuristic makes up a self-adapting and RT-SPC. The implementation presented in this work was oriented to indicate when different materials (e.g., steel, aluminum, etc.) were processed in a CNC lathe machine.

3. Real-Time Statistical Process Control

Manufacturing systems are becoming increasingly complex. Modern systems produce a substantial amount of raw data that typically is not used effectively to optimize production. To satisfy the need of producing higher quality products with fewer resources and time, researchers have initiated utilizing machine learning algo algorithms in the manufacturing domain. Machine learning is usually categorized into four main sections of unsupervised learning, semisupervised learning, supervised learning, and reinforcement learning. This

study focuses on the implementation and automation of the RT-SPC on the proposed architecture. Implementation of RT-SPC is similar to unsupervised machine learning methods since the data are not labeled. In contrast to the NN and other complex machine learning methods that require a large amount of labeled data and substantial computational power, RT-SPC requires fewer data points and minimal processing power.

Traditional SPC charts have historically been essential tools for manufacturing process control. Significant areas of opportunity are in the manual setup and time-consuming procedure of implementing CCs especially considering the automated nature of modern manufacturing processes. RT-SPC uses the same methodology underlying conventional CCs, but with stream analysis of the data in real-time. Assuming that the data captured can be represented by a normal distribution with a mean μ and a standard deviation σ, the odds of having the data points fall into three times the standard deviation (3σ) away from the mean, μ, is 99.73%. Considering this statistic, the 3σ lower and upper control limits are defined as the 1NR.

The arithmetic mean of a set of data can be simply calculated as shown in Eq. (1), where μ_n is the mean of the dataset until the nth data point.

$$\mu_n = \frac{1}{n}\left[\sum_{i=1}^{i=n} x_i\right] \tag{1}$$

Alternatively, the mean can be calculated incrementally as shown in Eq. (2). This facilitates the analytics and computational power required for big data streaming intrinsic to CPS, as not all the raw data points need to be stored and used in subsequent calculations.

$$\mu_n = \frac{1}{n}(x_n + (n-1)\mu_{n-1}) \tag{2}$$

Similarly, rather than computing the standard deviation of a dataset in total as in Eq. (3), the standard deviation can be calculated incrementally as shown in Eqs. (4) and (5).

$$\sigma = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{i=n} (x_i - \mu)^2\right)} \tag{3}$$

$$S_n = S_{n-1} + (x_n - \mu_{n-1})(x_n - \mu_n) \tag{4}$$

$$\sigma = \sqrt{\frac{S_n}{n}} \tag{5}$$

Considering Eqs. (2), (4), and (5), along with the 1NR requirements, to incrementally analyze the data using the proposed RT-SPC method, only four parameters need to be allocated in the memory: count, mean, standard deviation, and a buffer that holds the last 15 data points

(n=15). These parameters are updated when a new data point is received. The control limits are also updated accordingly. When a new data point is generated, the oldest data point is discarded and a new mean and standard deviation are calculated using Eqs. (2), (4), and (5), now with the updated set of 15 data points.

The RT-SPC based on the 1NR was developed in the cloud layer of the architecture as shown in Fig. 2, where the algorithm analyzes the incoming messages received in real-time from the MQTT message broker. It is possible to implement the same development in the fog or edge layer depending on the needs of the application. The codes were developed in NODE-RED,2 a flow-based programing environment for Node.js. In this case, the cloud implementation had the advantage that programing and testing could be done remotely by several users in collaboration and no local computing resources were needed since the operation (heavy lifting) was performed in remote servers. In other words, this part of the process worked as a service.

As discussed above, rather than storing all the data points in a database and calculating their mean and standard deviation, incremental calculations only require a few parameters to be stored, making the algorithm extremely high performance with low storage needed. As the number of data points increases, calculating the mean and standard deviation of all data points for every incoming data point will be computationally expensive. Through incremental calculations (e.g., stream analytics), the computational requirements are constant, and the performance of the system does not decrease over time due to computational complexity, enhancing scalability. This set of nodes was integrated into subsequent elements of the architecture.

The Nelson rules and anomaly detection focus of this study were implemented on the spindle load data captured from a CNC lathe. The goal was to develop an unsupervised algorithm that detects the material type difference in the machining of a part in a CNC lathe. In other words, the objective of this section is to identify the repeating G-Code files and ensuring that a similar material was used for the similar G-Code blocks. This algorithm is, therefore, expected to generate an alert back if a part with different hardness and characteristics is a machine that does not match the previously cut materials with the same G-code. An entity relation diagram is shown in Fig. 3 for the database that was created to store and correlate the G-Code filename, G-Code lines, and spindle loads captured from the MTConnect data stream generated by the CNC controller. This database was implemented in the fog layer.

The MTConnect data for the case study of this work were captured and stored in comma separated values (CSV) format to be used in a simulation environment to ease the debugging and verification of the algorithm as shown in Fig. 4. This setup publishes the same messages to the MQTT message broker that was captured during the actual operation of the CNC machine.

In contrast with the traditional internet communication protocols such as HTTP that forces the user to receive all the messages and filter out the client-side, this architecture advocates the use of MQTT as it provides the flexibility of subscribing to the desired topics of interest, reducing the unnecessary traffic between the gate- ways. This was implemented in the

architecture to only subscribe to the topics of “Lpprogram,” “Lp1block,” and “LS1load” which represent G-Code filename, G-Code line, and spindle load, respectively. Since these messages are received with different timing, once each one is received, they are stored in the global variable of the NODE-RED flow as shown in Fig. 5.

The function for the spindle load has an additional task of incrementally adding the spindle loads together while recording their count number. When the spindle load messages related to a G-Code line are all received, this node calculates the mean of the captured spindle loads and passes that along with the G-Code file- name and G-Code line to the next stage. This process is shown in the flow chart in Fig. 6.

The “Incrementally Calculate” flow shown in Fig. 6 consists of multiple stages. The first stage in this process is to read the current values of the mean, standard deviation, and count of the spindle load as well as the buffer that holds the last 15 spindle load readings that are needed for computation of the Nelson rules and to update them with the incoming spindle load average. The critical challenge in this process is that reading and writing to the database is time- consuming and asynchronous. By simply transmitting the spindle load averages to these nodes, rather than incrementally updating the values one by one, all the data are read from the database at once and are updated simultaneously. This causes unnecessary computations as each data point must be read or written in a big database.

To solve this problem, a buffer is created to hold all the messages. One message is injected into the flow and when this message reaches the ending node, the buffer is commanded to release the next data point to the flow. Therefore, the messages are kept in the buffer until they are permitted through the computational flows, ensuring they are analyzed serially in the sequence. This entire process with the integration of the Nelson rules nodes is shown in Fig. 7.

4. Case Study: Machining Aluminum and Steel

The purpose of the case study is to evaluate the integrated algorithm RT-SPC on the architecture in detecting anomalies. More specifically, the case study determines whether the algorithm can detect the change in the material type by watching the spindle loads generated during the mass production of a part using a CNC lathe. An OKUMA-Genos L250 was used as the CNC machine. The data from the machine were accessed using a SmartBox3 to add security to the IoT configuration. The data were available in the standard MTConnect. A picture of this machine is shown in Fig. 8. A facing operation was per- formed with this lathe for different materials. Three samples of aluminum and one of steel were used in this experiment. The diameter of the materials, as well as the machining parameters for the facing operation, is shown in Table 2. All the data collection and generation for this case study were performed in the edge layer (Fig. 1(a)).

The data generate in the edge layer, then pass to the fog layer (Fig. 1(b)) in which the data were transmitted to an MQTT broker. Here the data were made accessible by the cloud. In the fog layer, the data were stored in a database for offline backup, as displayed in Fig. 3. Finally, the data were accessed by the cloud layer (Fig. 1(c)) where the analytics and further

data storage were produced.

5. Results

The developed Nelson rules section was first tested with a repeated set of the array {1.5,2.5,3.5,4.5,5.5,6.5,7.5,8.5,9.5}, which was injected in different time intervals. The result of the RT-SPC for the 1NR is shown in Fig. 9. Figure 10 shows the accumulation of the detected errors in Nelson rules 3 (more than six points in a row continually increasing or decreasing) and 7 (15 points in a row within the first standard deviation).

Note that there is no need to collect any sample data, compute the control limits, or implement any steps manually. Once the second data set is received by the algorithm, it automatically computes the control limits and updates them accordingly as the new data points are received. Also, note that as soon as any anomalies are detected, the algorithm responds back with the detected Nelson rule error.

The spindle loads for each G-Code line captured from MTConnect for a facing operation on an aluminum sample (AL-S) are shown in Fig. 11.

Running the AL-S1, AL-S2, AL-S3, AL-S4, AL-S5, and steel sample 1 (ST-S1), respectively, the captured spindle loads are shown in Fig. 12. Since there are multiple aluminum parts machined and the next sample was a steel part, it is expected that the algorithm detects this change in the material.

The mean for the spindle loads per G-Code blocks was calculated. The result of this calculation for one of the AL-Ss is shown in Fig. 13. As presented in Fig. 11, the G-Code line "G1 X0 F.03" represents the facing operation and is the most important G-Code considered in this study. Figure 14 presents the spindle loads for this G-Code block, the mean, and the control limits. That last data point in Fig. 14(a) represents the spindle loads related to the stainless-steel sample and as can be seen is statistically abnormal. Figure 14(b) provides a zoomed view into the zone of part (a) where the lower and higher control limits have not yet been calculated for the available data.

Figure 15 presents the actual data points calculated and fed to the architecture. This figure shows the computed statistical errors (abnormalities) in the average spindle loads per G-Code blocks for all six samples that are calculated using the above-mentioned flows in the architecture. All the values stored in the database to perform RT-SPC for this case study are shown in Fig. 16. Note that these values are updated as more data points are published in contrast with the traditional methods where all the data points are stored and the process gets more complex over time as presented in Fig. 17.

6. Discussion

During the implementation of the model of the CPS provided in Fig. 1, it was clear that some functions could be performed in either the cloud or the fog layer. For example, in this case, the application of the RT-SPC algorithm was implemented in the cloud layer using the

NODE-RED software as explained in Sec. 2; however, a similar implementation may be executed in the local fog layer. Other functions can only be applied in the layer in which they were designed, for example, creating a buffer in the fog layer to maintain the data even when disconnection from the internet occurs.

The implementation of the MQTT protocol (which started in the fog layer) for transporting the data provided the necessary performance and flexibility to be compatible with the rest of the architecture and to implement the process in the cloud as shown in Fig. 7.

From Figs. 14 and 15, the RT-SPC algorithm calculates the control limits immediately (from 1NR); however, it shows early errors until it becomes stable. Once stable, the algorithm now can identify (by a violation of the 1NR) the switch in working materials. The challenge now is to define when stability has been reached. One proposal is by accounting the rate of detection of abnormalities and when this rate approaches zero and then stability has been reached. Defects in the machining process or changes in the material being machined can be detected only after stability has been reached. After the switch of materials (from aluminum to steel), the algorithm now adapts to the new normal as shown in Fig. 15. Besides the points detected as abnormalities while the control limits are not stable, another limitation of this work is that it is focused only on the spindle loads of the CNC machines and only two materials. The effect of having different materials but with similar properties that affect its machining was not assessed in this work.

From Fig. 17, the advantage of having the RT-SPC is displayed since the traditional method of performing the computation of all data every time a new data point is sampled increases the computational cost each time new data arrive. The RT-SPC allows a constant, low computational time (3 ms) to calculate the control limits that are independent of the data sample size. The data record length independence also provides clear memory requirements (in this case 15 points) which are quite low.

While this work focused only on one machine tool, having multiple machine tools enabled in this model of CPS with automated analytics can bring great benefits towards the I4.0. First, the trace- ability of manufactured components can be greatly simplified since the framework allows to retrieve the data to the detail of line of G-Code in a G-Code program run in a particular machine. With this, defects or redesigns can be efficiently addressed. Second, since the process is automated, productivity can be monitored as described by key performance indicators such as runtime, through- put, change over time, cycle time, downtime, etc. Third, quality and failure prevention can be monitored effortlessly in real-time since the RT-SPC will detect abnormalities that can come from human errors or machine disfunction. And last, having the fog and cloud layers brings the balance between remote services accessible by multiple users in different geographic zones (cloud) and having offline backup and almost real-time response (fog).

This work presented the results for machine tools; however, the framework should work for other streams of data, for example, the data coming from sensors embedded on machines or not. When that limitation is removed, the CPS presented here can be employed in manufacturing facilities of different nature, which can be the topic of future research.

7. Conclusions

This work contributes to the implementation of Industry 4.0 in a way that includes not only the architecture of a CPS to allocate the use of the computational resources but also the systematic approach to implementing the automatic application of algorithms to trans- form data into insightful information. The proposed architecture consists of three layers: edge, fog, and cloud. The edge layer collects the data from different sources using MTConnect and other protocols to adapt to each case. The data are transmitted to the fog layer using the protocol MQTT. In the fog layer, the data are prepared to be sent to the cloud layer and in case of lost connectivity, the data can be buffered and sent once the connection is reestablished. The fog layer can also perform the application of algorithms. The cloud layer stores the data and applies analytics to turn data into information. The concept was deployed in an industrial environment in which a case study was performed. The case study had the objective of identifying variations in the material being worked by a CNC machine tool. A RT-SPC based on the Nelson rules was implemented in the CPS architecture proposed. The results present an algorithm that can identify different materials (in this case aluminum and steel). The algorithm implemented does not require training and provides a short and stable response time, compared with the construction of CCs from buffered data.

8. Future Work

Future work is necessary for the next topics: cyber-security, optimization, energy consumption, and digital twin integration. This work did not consider the aspect of cyber-attacks and online security. This aspect is one of the bottlenecks for I4.0 technologies and an important concern for manufacturing companies. Further work is necessary to assess the risk level of this proposal of CPS and to draft strategies to reduce the risk. The CPS proposed here was developed organically based on the needs and resources available in the lab; however, a systematic optimization can produce a lean model for CPS architecture which can assess, according to the application, the precise number of resources to allocate in each of the layers of the CPS. The aspect of energy consumption needs to be addressed since there are important environmental issues that will be impacted in the future when more companies use solutions that combine edge, fog, and cloud layers. The problem can be attacked from the point of view of energy efficiency, not only of the layer of the CPS but also of the source of energy used. Finally, the CPS proposed here and the analytics produced can be inserted into a Digital Twin for products, which can be helpful to increase traceability, quality, and to test different scenarios. The way to connect the CPS explored in this work and the Digital Twin is an aspect that requires further research.

Acknowledgment

This work was partially supported by National Science Foundation (Grant Nos. 1631803 and 1646013). This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05- 00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or repro- duce the published form of this manuscript, or allow others to do so, for US

government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.4
The authors would like to acknowledge the financial and technical support of Writing Lab, TecLabs, Tecnológico de Monterrey, CONACYT, and the research group Automotive Consortium for CPSs in the production of this work.

Table 1 Analysis of works related to manufacturing, CPS, and FC

Work	Year	Edge elements	Cloud functions	Automatic analytics	General purpose/application
[11]	2017	Cameras, sensors, and production line	Central server and storage	Convolutional NN	Inspection system
[12]	2019	CNC machine tools, smart robots, sensors, terminals, and edge connection devices	Storage and big data analytics	Not applicable	Smart manufacturing service system
[13]	2019	Terminal device layer	Datacenter, storage, and server with services like fault diagnosis, etc.	Not applicable	Task scheduling of FC for smart production line
[14]	2019	Supervisory control and data acquisition (SCADA) among others	Not applicable	Simple data analysis with the potential to implement machine learning algorithms	Optimization of the assembly process
[15]	2018	Radio frequency identification (RFID), programmable logic controller (PLC), sensors, and others	Server and data center	Semantic data model	Packing line as a case study
[16]	2018	Sensors, machines, and production lines	Database, monitoring, and dashboards	Time-sensitive analytics at the local level	Smart factories in general
[17]	2020	Infrastructure	Supervisory control	Integer linear programing	Partial assembly of a vehicle component using robots and humans
[18]	2021	PLC, microcontrollers, and sensor drivers	Database, virtual machine, an applications	Repeatability and reproducibility (R&R)	In-line measurement for aluminum parts

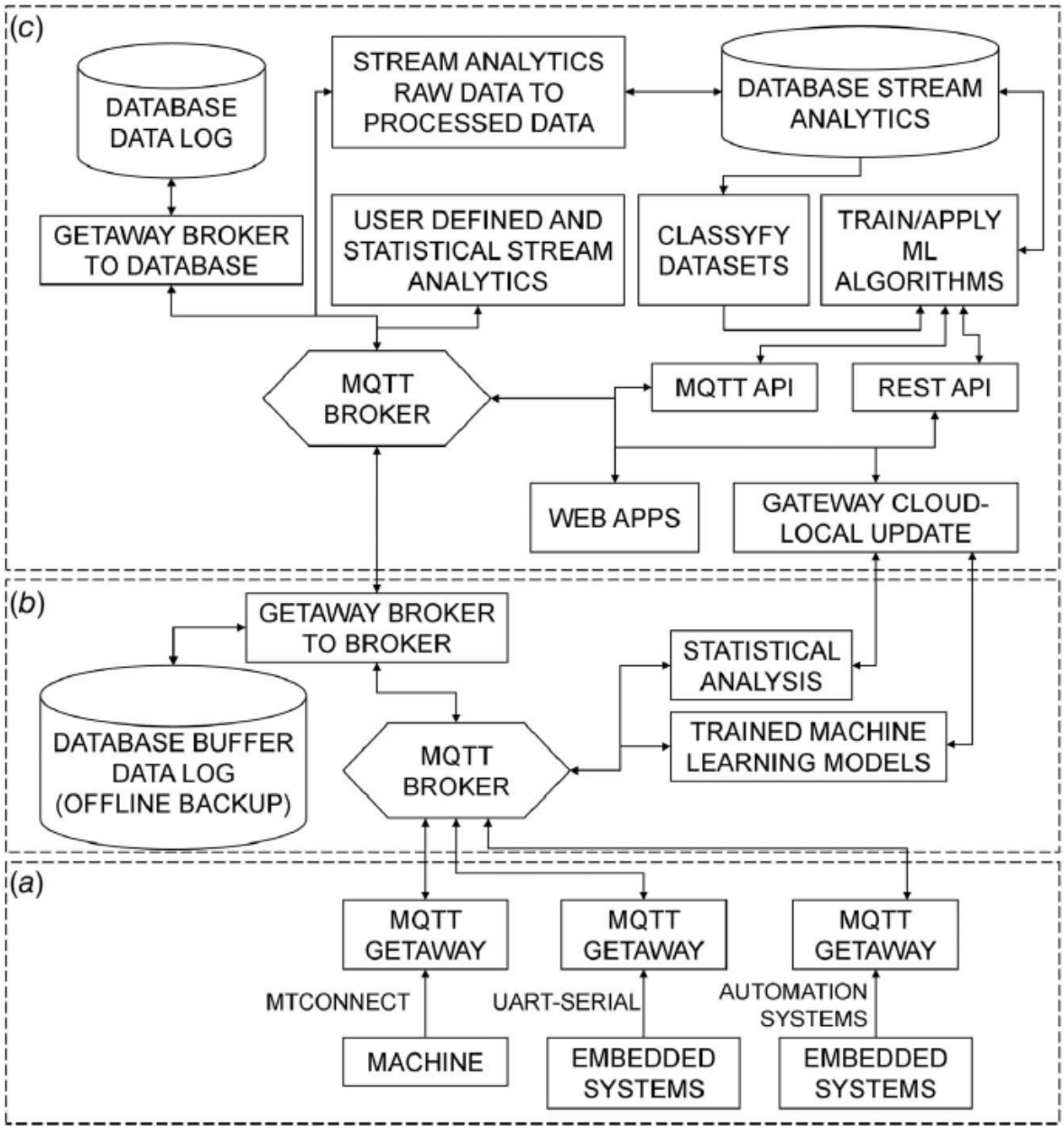

Fig. 1 Layers of the CPS for manufacturing: (*a*) edge, (*b*) fog, and (*c*) cloud

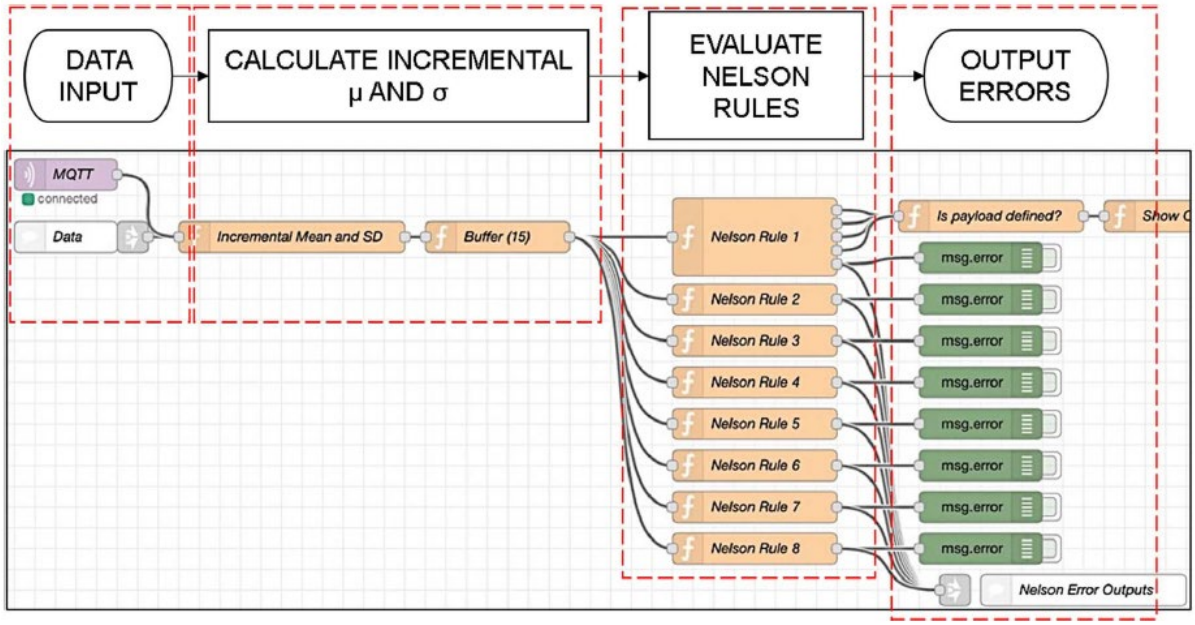

Fig. 2 Block diagram (top) and program elements in NODE-RED (bottom) to analyze the input data using the Nelson rules

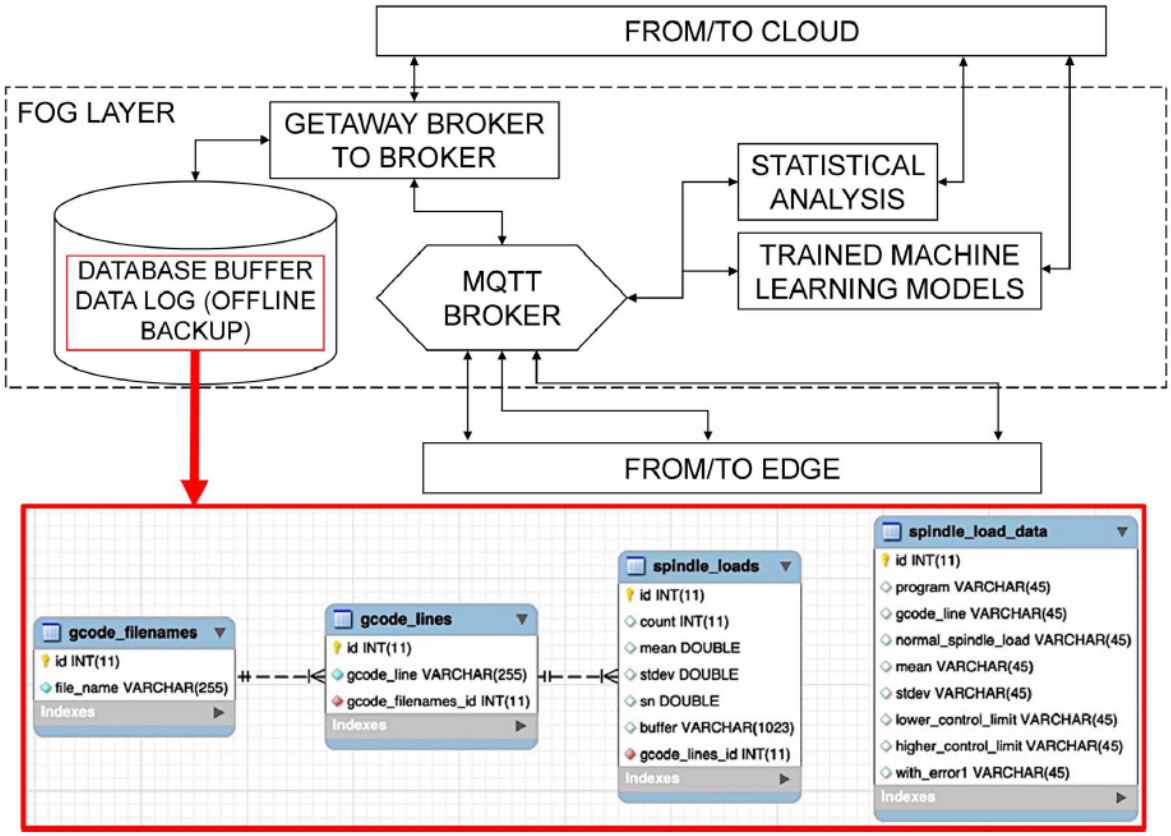

Fig. 3 Entity relation diagram (bottom) for the database tables created for spindle load analytics and implemented in the fog layer of the CPS architecture (top)

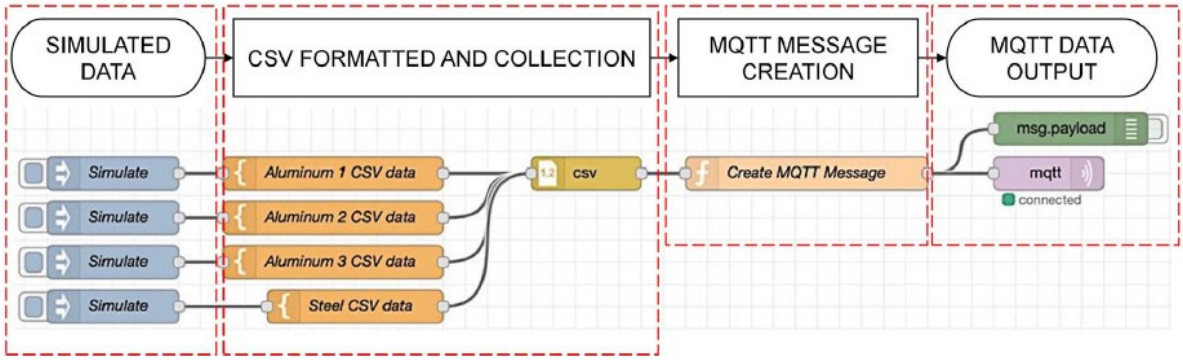

Fig. 4 Block diagram (top) and program elements in NODE-RED (bottom) for simulating MTConnect-MQTT data

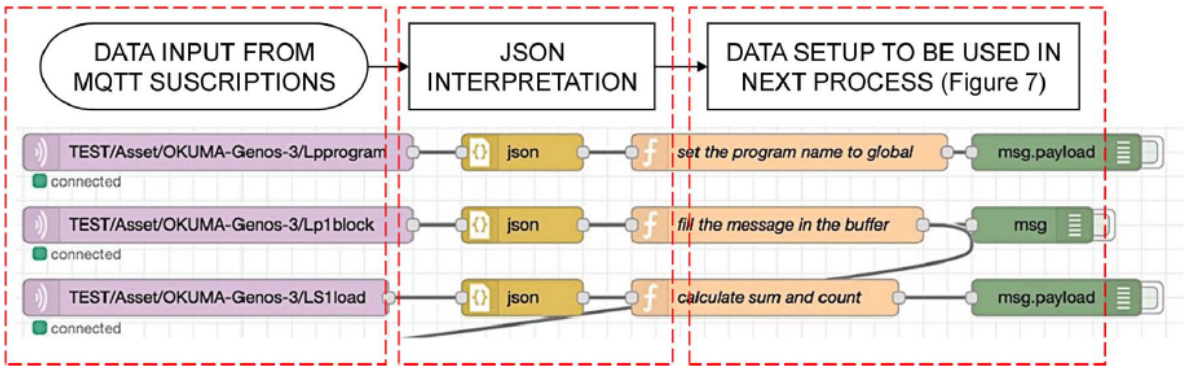

Fig. 5 Block diagram (top) and program elements in NODE-RED (bottom) to subscribe to G-Code names, lines, and spindle loads topics

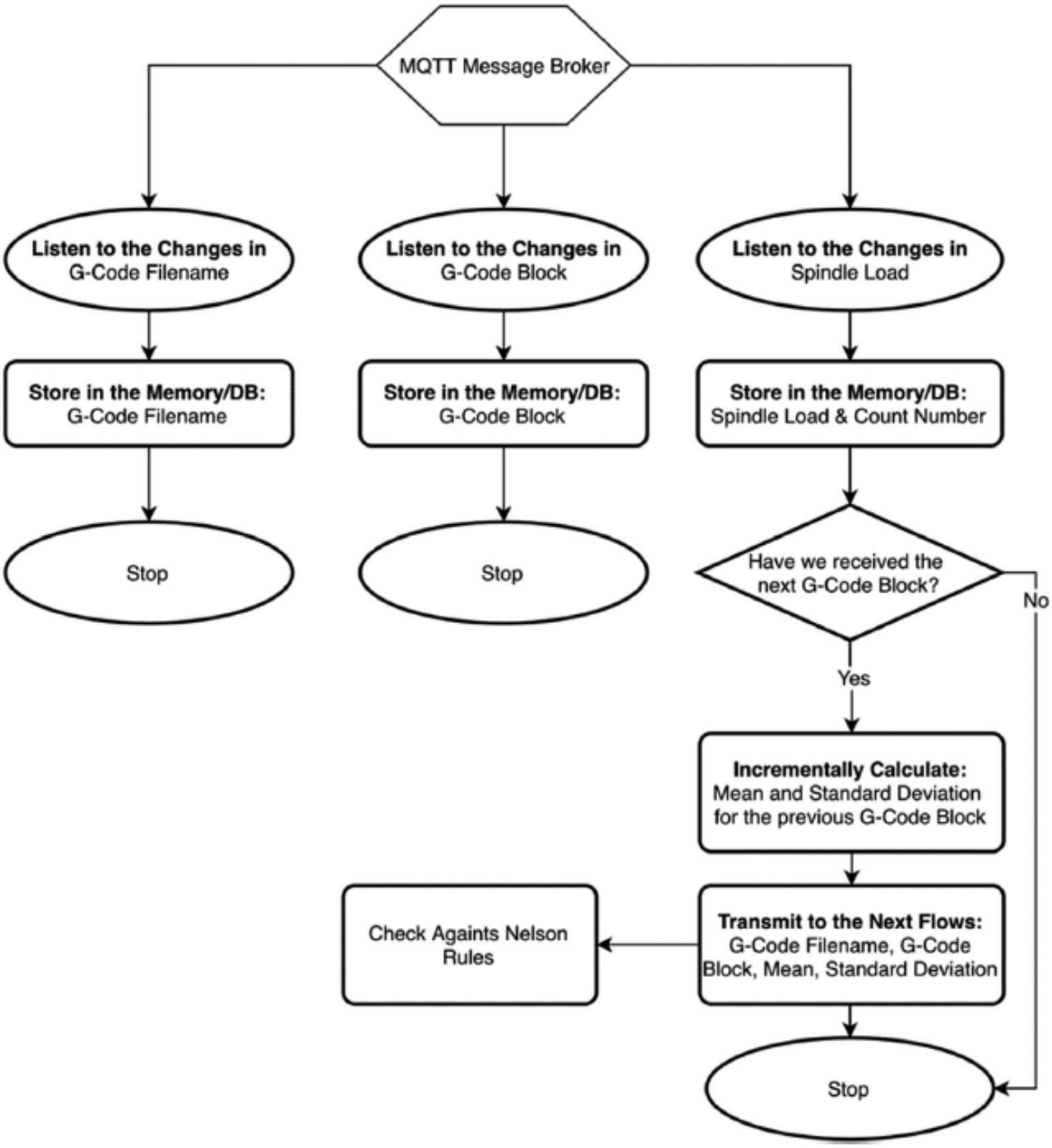

Fig. 6 Flowchart for RT statistical analytics for RT-SPC

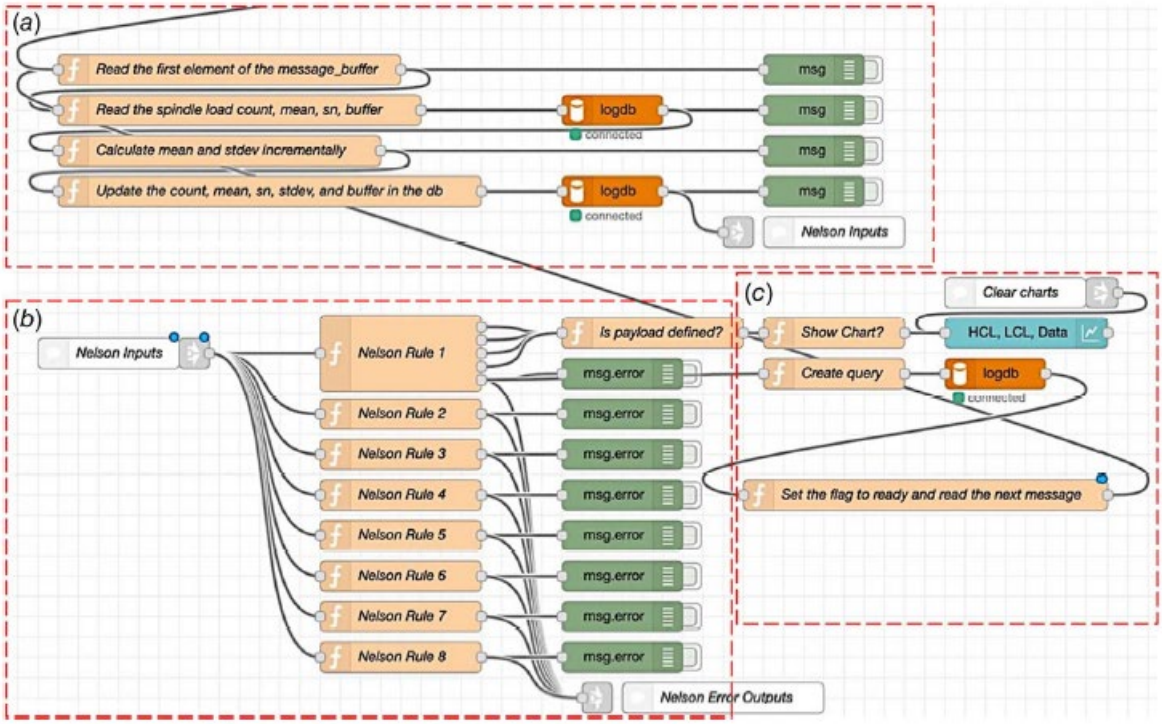

Fig. 7 (*a*) Management of the buffer of the data received, (*b*) evaluation according to Nelson rules, and (*c*) generation of CCs. The cycle *a*–*b*–*c* is repeated.

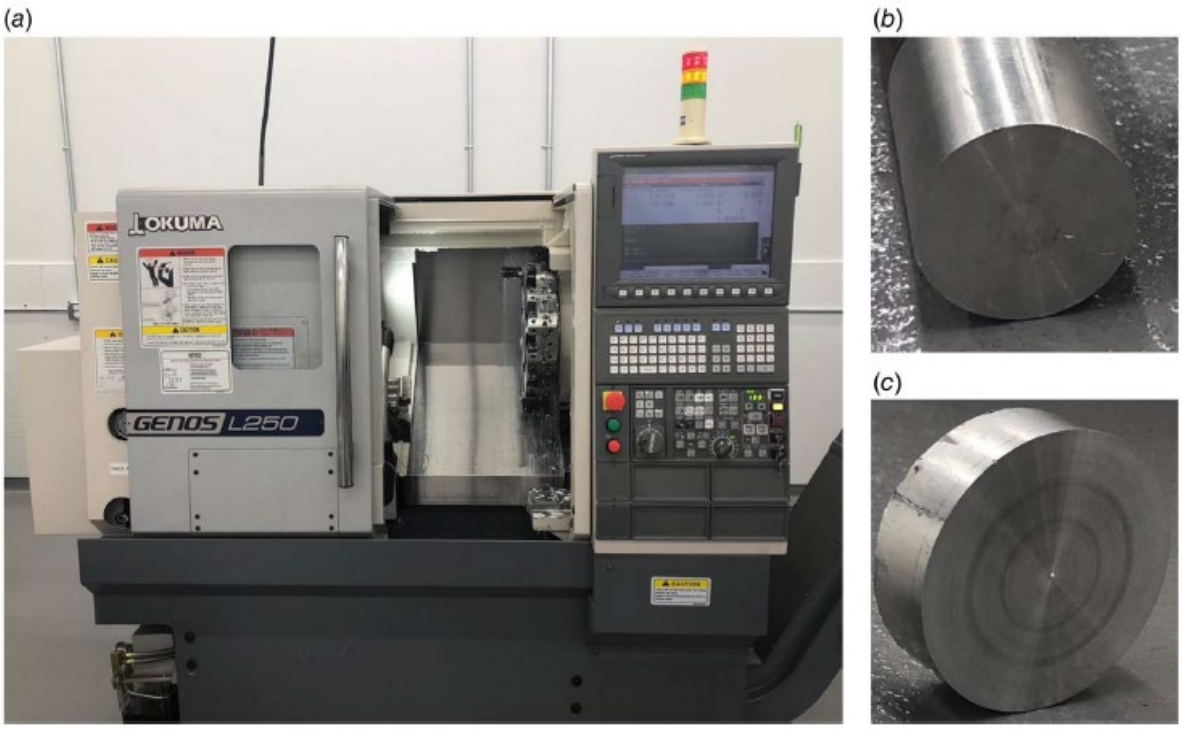

Fig. 8 (*a*) OKUMA-Genos CNC Lathe, (*b*) steel bar from which the steel sample was machined, and (*c*) aluminum workpiece from which the AL-Ss were machined

Table 2 Machining parameters and material diameters

Parameter	Value
Depth of cut	0.5 mm
Feed rate	0.03 mm/rev
Surface spindle speed	250 m/min
Aluminum diameter	2.25 in.
Steel diameter	1 in.

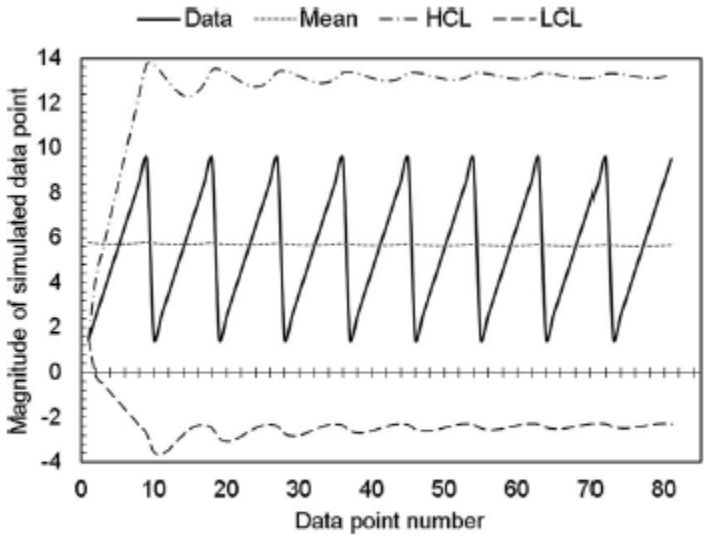

Fig. 9 Developed dynamic CCs with incremental calculation

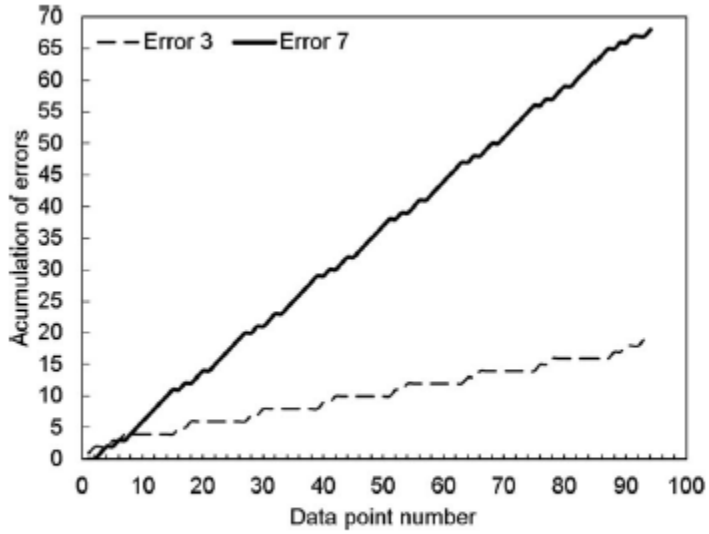

Fig. 10 Cumulative errors generated with Nelson rules engine for the raw spindle loads

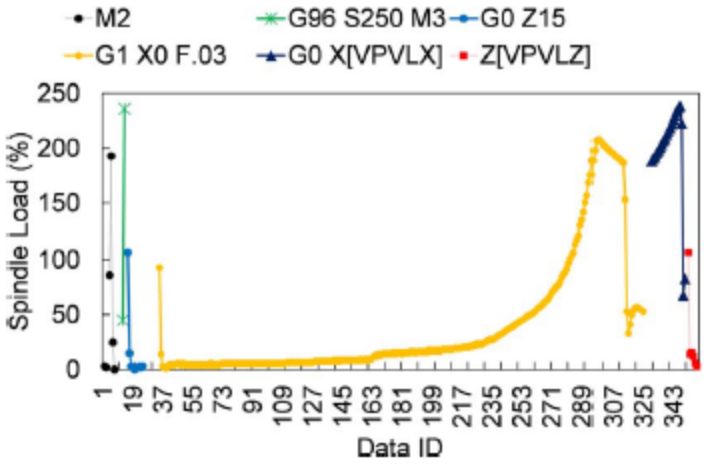

Fig. 11 Spindle loads for facing operation on an AL-S

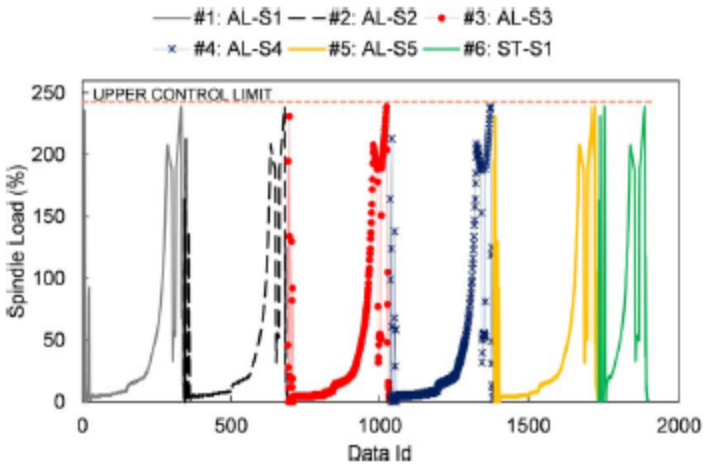

Fig. 12 Spindle loads for AL-S1, AL-S2, AL-S3, AL-S4, AL-S5, and ST-S1

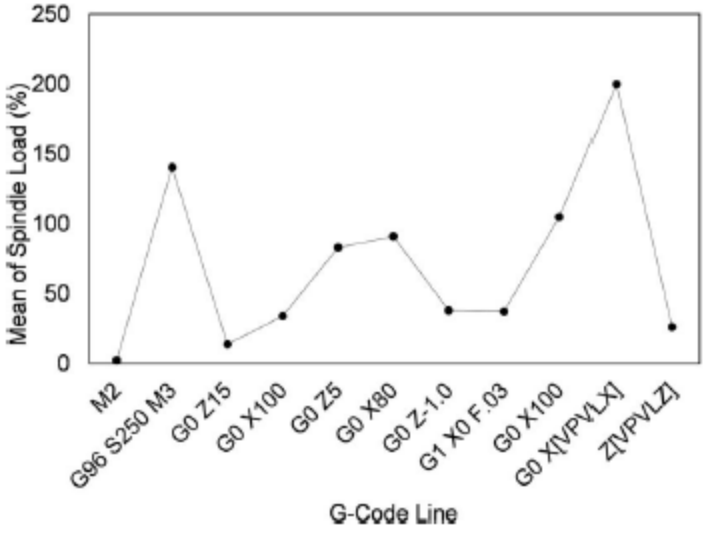

Fig. 13 Computed average spindle loads per G-Code for an AL-S

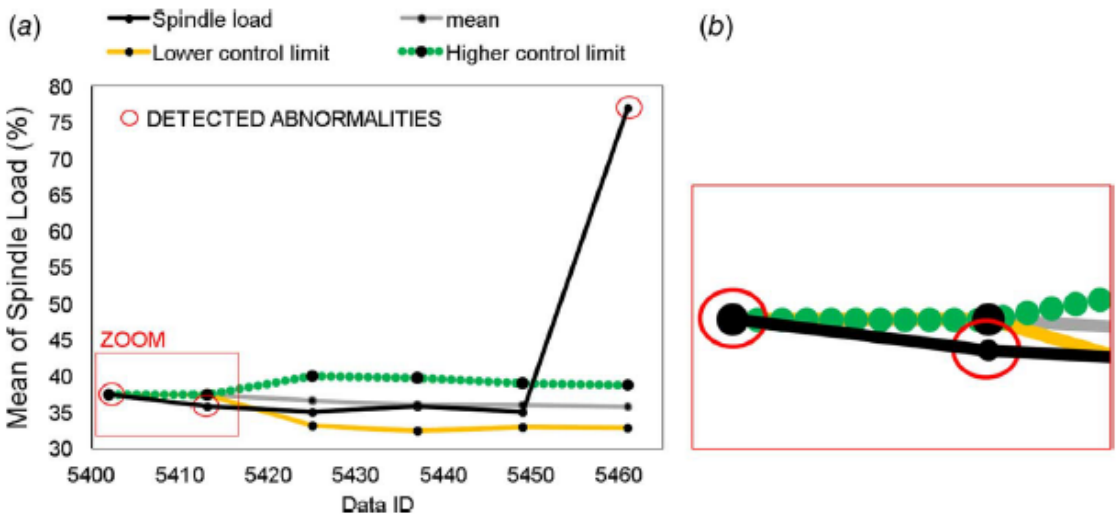

Fig. 14 (*a*) Mean of spindle loads for "G1 X0 F.03" G-Code block for all six samples and (*b*) zoom of the zone marked in the bottom left of (*a*)

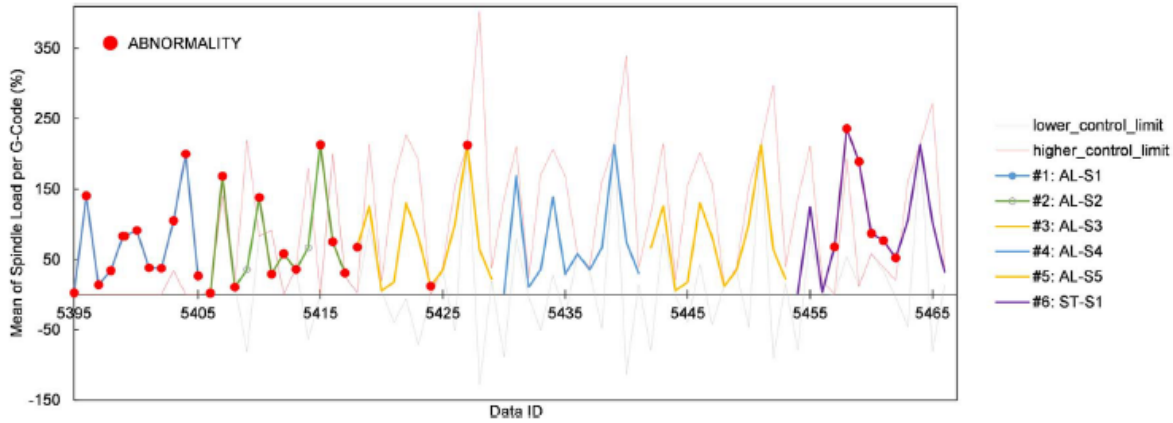

Fig. 15 Computed statistical errors in average spindle loads per G-Code for all six samples

id	count	mean	stdev	sn	buffer	gcode_line	file_name
16898	11	13.954545454545455	26.35166460155273	6944.102272727273	2,67.25,2,67.25,1.5,2.5,2,2,2.5,2	M2	N900.MIN
16899	11	149.09090909090907	19.490790365988477	3798.9090909090924	168.5,125.66666666666667,168.5,...	G96 S250 M3	N900.MIN
16900	12	13.390178571428573	12.62750802849637	1753.993549107143	10.571428571428571,5.4,10.57142...	G0 Z15	N900.MIN
16901	21	60.523809523809526	31.871412029621546	20315.7380952381	35.5,66.5,18,101,106,34,105,35.5,6...	G0 X100	N900.MIN
16902	11	130.45454545454547	42.63886404763531	18180.727272727272	138,130,138,130,236,83,138,138,8...	G0 Z5	N900.MIN
16903	10	58.2	30.954446817505524	8623.6	29,82,29,82,91,29,29,91,29	G0 X80	N900.MIN
16904	3	38	0	0	38,38	G0 Z-1.0	N900.MIN
16905	11	39.78984366552518	12.355443775727219	1526.5699089515647	35.74333333333333,34.976510067...	G1 X0 F.03	N900.MIN
16907	11	109.51948051948052	58.94451522472047	34744.55875077303	75,63.5,75,63.5,103,199.90476190...	G0 X[VPVLX]	N900.MIN
16908	11	28.05194805194805	3.5190092494495984	123.83426097711825	30.571428571428573,22.28571428...	Z[VPVLZ]	N900.MIN
16915	6	62.833333333333336	11.839200423452025	700.8333333333331	58,87,58,58,58	G0 Z-1.5	N900.MIN
16918	8	213.05592105263156	0.317176621303819	0.7042070637119436	212.57894736842104,213.1578947...	M9	N900.MIN
16927	3	71	102.19099764656376	20886	12,189	M8	N900.MIN
16960	1	67.75	0	0	NULL	G0 X45	N900.MIN

Fig. 16 Values stored in the database (using MYSQL software) to perform RT-SPC for this case study

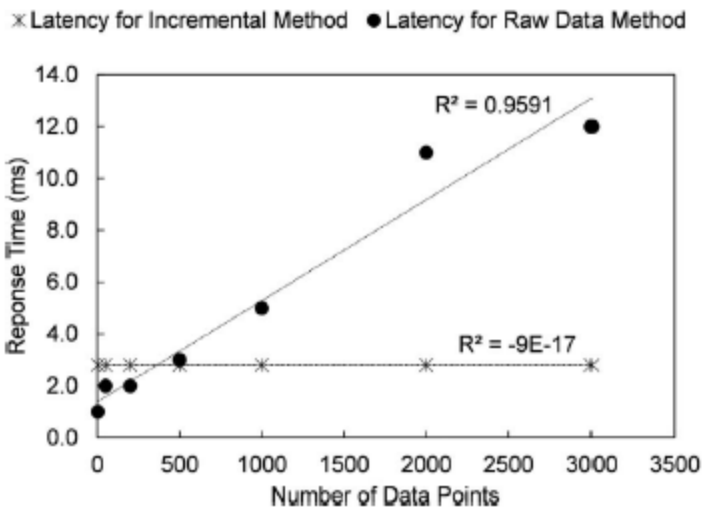

Fig. 17 Computation time needed to calculate mean and std. dev. versus number of data for both methods

References

[1] Lynn, R., Louhichi, W., Parto, M., Wescoat, E., and Kurfess, T., 2017, "Rapidly Deployable MTConnect-Based Machine Tool Monitoring Systems," 2017 ASME Manufacturing Science and Engineering Conference, Los Angeles, CA, June 4–8, pp. 1–10.

[2] Liu, X. F., Shahriar, M. R., Al Sunny, S. M. N., Leu, M. C., and Hu, L., 2017, "Cyber-Physical Manufacturing Cloud: Architecture, Virtualization, Communication, and Testbed," J. Manuf. Syst., 43, pp. 352–364.

[3] Mourtzis, D., Milas, N., and Athinaios, N., 2018, "Towards Machine Shop 4.0: A General Machine Model for CNC Machine-Tools Through OPC-UA," Procedia CIRP, 78, pp. 301–306.

[4] Kim, D.-H., Kim, T. J. Y., Wang, X., Kim, M., Quan, Y.-J., Oh, J. W., Min, S.-H., Kim, H., Bhandari, B., Yang, I., and Ahn, S.-H., 2018, "Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry," Int. J. Precis. Eng. Manuf. Green Technol., 5(4), pp. 555–568.

[5] Jurkovic, Z., Cukor, G., Brezocnik, M., and Brajkovic, T., 2018, "A Comparison of Machine Learning Methods for Cutting Parameters Prediction in High Speed Turning Process," J. Intell. Manuf., 29(8), pp. 1683–1693.

[6] Wu, D., Jennings, C., Terpenny, J., Gao, R. X., and Kumara, S., 2017, "A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests," J. Manuf. Sci. Eng., 139(7), p. 071018.

[7] Qin, J., Liu, Y., and Grosvenor, R., 2016, "A Categorical Framework of Manufacturing for Industry 4.0 and Beyond," Procedia CIRP, 52, pp. 173–178.

[8] Lee, J., Bagheri, B., and Jin, C., 2016, "Introduction to Cyber Manufacturing," Manuf. Lett., 8, pp. 11–15.

[9] Lee, J., Bagheri, B., and Kao, H. A., 2015, "A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems," Manuf. Lett., 3, pp. 18–23.

[10] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K., 2016, "Cyber-Physical Systems in Manufacturing," CIRP Ann., 65(2), pp. 621–641.

[11] Li, L., Ota, K., and Dong, M., 2018, "Deep Learning for Smart Industry: Efficient Manufacture Inspection System With Fog Computing," IEEE Trans. Ind. Inf., 14(10), pp. 4665–4673.

[12] Qi, Q., and Tao, F., 2019, "A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing," IEEE Access, 32(4–5), pp. 86769–86777.

[13] Wang, J., and Li, D., 2019, "Task Scheduling Based on a Hybrid Heuristic Algorithm for Smart Production Line With Fog Computing," Sensors, 19(5), pp. 1–13.

[14] Wang, J., Zheng, P., Lv, Y., Bao, J., and Zhang, J., 2019, "Fog-IBDIS: Industrial Big Data Integration and Sharing With Fog Computing for Manufacturing Systems," Engineering, 5(4), pp. 662–670.

[15] Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., and Yin, B., 2017, "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges," IEEE Access, 6, pp. 6505–6519.

[16] Beregi, R., Pedone, G., and Mezgár, I., 2019, "A Novel Fluid Architecture for Cyber-

Physical Production Systems," Int. J. Comput. Integr. Manuf., 32(4–5), pp. 340–351.
[17] Nikolakis, N., Senington, R., Sipsas, K., Syberfeldt, A., and Makris, S., 2020, "On a Containerized Approach for the Dynamic Planning and Control of a Cyber-Physical Production System," Rob. Comput. Integr. Manuf., 64, p. 101919.
[18] Castro-Martin, A. P., Ahuett-Garza, H., Guamán-Lozada, D., Márquez-Alderete, M. F., Urbina Coronado, P. D., Orta Castañon, P. A., Kurfess, T. R., and González de Castilla, E., 2021, "Connectivity as a Design Feature for Industry 4.0 Production Equipment: Application for the Development of an In-Line Metrology System," Appl. Sci., 11(3), p. 1312.
[19] Ahuett-Garza, H., and Urbina Coronado, P. D., 2019, "A Reference Model for Evolving Digital Twins and Its Application to Cases in the Manufacturing Floor," Smart Sustain. Manuf. Syst., 3(2), p. 20190049.
[20] Road, T. H., 2001, "A Fuzzy Reasoning Based Diagnosis System for X Control Charts," J. Intell. Manuf., 12(1), pp. 57–64.
[21] Ahuett-Garza, H., and Kurfess, T., 2018, "A Brief Discussion on the Trends of Habilitating Technologies for Industry 4.0 and Smart Manufacturing," Manuf. Lett., 15, pp. 60–63.
[22] Zhou, P., Zuo, D., Hou, K. M., Zhang, Z., Dong, J., Li, J., and Zhou, H., 2019, "A Comprehensive Technological Survey on the Dependable Self-Management CPS: From Self-Adaptive Architecture to Self-Management Strategies," Sensors, 19(5), pp. 1–53.
[23] Fernández-Caramés, T. M., Fraga-Lamas, P., Suárez-Albela, M., and Díaz-Bouza, M. A., 2018, "A Fog Computing Based Cyber-Physical System for the Automation of Pipe-Related Tasks in the Industry 4.0 Shipyard," Sensors, 18(6), pp. 1–26.
[24] Dolui, K., and Datta, S. K., 2017, "Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing," Proceedings of GIoTS 2017—Glob Internet Things Summit, Geneva, Switzerland, June 6–9.
[25] Baktir, A. C., Ozgovde, A., and Ersoy, C., 2017, "How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases; Future Directions," IEEE Commun. Surv. Tutorials., 19(4), pp. 2359–2391. [26] Roman, R., Lopez, J., and Mambo, M., 2018, "Mobile Edge Computing, Fog, et al.: A Survey and Analysis of Security Threats and Challenges," Future Gener. Comput. Syst., 78, pp. 680–698.
[27] Shirazi, S. N., Gouglidis, A., Farshad, A., and Hutchison, D., 2017, "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective," IEEE J. Sel. Areas Commun., 35(11), pp. 1–10.
[28] Angrish, A., Starly, B., Lee, Y.-S., and Cohen, P. H., 2017, "A Flexible Data Schema and System Architecture for the Virtualization of Manufacturing Machines (VMM)," J. Manuf. Syst., 45, pp. 236–247.
[29] Esmaeilian, B., Behdad, S., and Wang, B., 2016, "The Evolution and Future of Manufacturing: A Review," J. Manuf. Syst., 39, pp. 79–100.