Context-Sensitive Modeling and Analysis of Cyber-Physical Manufacturing Systems for Anomaly Detection and Diagnosis

Miguel A. Saez, Francisco P. Maturana, *Member, IEEE*, Kira Barton, and Dawn M. Tilbury, *Fellow, IEEE*

***Abstract*— Cyber-physical manufacturing systems (CPMS) can be defined by the integration of control, network communication, and computing with a physical manufacturing process. In this work, we present a hybrid model of CPMS combining sensor data, context information, and expert knowledge. We used the identification of global operational states and a multimodel framework to improve anomaly detection and diagnosis. The anomaly detection is based on context-sensitive adaptive threshold limits. Root cause diagnosis is based on classification models and expert knowledge. The proposed approach was implemented using the Internet of Things (IoT) to extract data from a computer numerical control machine. Results showed that using a context-sensitive modeling strategy allowed to combine physics-based and data-driven models for residual analysis to detect an anomaly in the part, machine, or process. The identification of root cause was improved by adding context information in classification models to identify worn or broken tools and wrong material.**

***Note to Practitioners*—Anomaly detection and diagnosis of manufacturing equipment is a complex problem. Some of the challenges are complex machine dynamics and nonstationary operating conditions. This paper describes a framework for modeling manufacturing equipment using a combination of sensor data, context information, and system knowledge. The proposed modeling framework is used to improve anomaly detection for diagnostics using a context-sensitive strategy. This work aims to support more effective maintenance actions by identifying problems in the machine, part, or process. The modeling and anomaly detection strategy was used to identify anomalies in computer numerical control machines and can be extended to other equipment on the plant floor.**

***Index Terms*— Cyber-physical systems (CPSs), fault detection, manufacturing systems.**

I. Introduction

EQUIPMENT and process monitoring play a key role in manufacturing. Anomaly detection has arisen as a critical first step in monitoring machine, part, and process to support health monitoring, scrap avoidance, and process optimization. An anomaly can be defined as an occurrence that is different from what is standard, normal, or expected, and it can be abrupt or gradual [1]. The root cause diagnosis focuses on finding the cause of abnormal behavior with as much detail as possible to determine the location and size of a fault. In manufacturing machines, proper anomaly detection and diagnosis represents a challenge partly due to machine interactions, multiple operational states, and similarities between symptoms of different failure modes.

Physics-based model fault detection and diagnosis (FDD) requires the knowledge of equations that govern the machine dynamics. Physics-based model FDD approach has been developed to detect machine tool faults. However, due to noise caused by component and part interactions during the manufacturing process, implementation has not been completely feasible [2]. Data-driven models have been used to detect anomalies in computer numerical controls (CNCs), gantries, and robots. Extensive research has been done to monitor machining operations to detect anomalies based on different signal processing and analysis strategies [3]. However, in order to improve detection and diagnosis, some knowledge of the system, whether from physics-based models or experts, is required [4]. A comparison between physics-based and data-driven models [5] shows that both have advantages and disadvantages based in part on factors such as the detail of data available, model development efforts, and implementation challenges. The goal of this work is to improve anomaly detection to answer the following questions: 1) How to detect anomalies considering the different machine–part interactions? and 2) How to improve the diagnosis of anomalies by considering the operational context?

Recent advances in machine communication, data extraction, and real-time analysis have enabled the development of cyber-physical systems (CPSs). A CPS is defined by the integration of cyber and physical components such as communication and control networks, sensors, and actuators in a multilayer architecture [6]. In this work, a novel approach to model manufacturing operations as a hybrid system is presented. This work is based on Hybrid Discrete Event System Specification (HDEVS), a general, scalable, and hierarchical formalism for modeling hybrid and discrete event system (DES) used for representing systems with a finite number of states in finite intervals of time. The model considers the

Manuscript received March 25, 2018; revised September 11, 2018 and January 21, 2019; accepted February 25, 2019. Date of publication November 8, 2019; date of current version January 9, 2020. This article was recommended for publication by Associate Editor E. Messina and Editor D. Popa upon evaluation of the reviewers' comments. This work was supported in part by the NSF under Grant 1462910 and in part by Rockwell Automation. *(Corresponding author: Miguel A. Saez.)*

M. A. Saez, K. Barton, and D. M. Tilbury are with the Mechanical Engineering Department, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: migsae@umich.edu; bartonkl@umich.edu; tilbury@umich.edu).

F. P. Maturana is with Rockwell Automation, Cleveland, OH 44124 USA (e-mail: fpmaturana@ra.rockwell.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2019.2918562

machine–part interaction to define discrete states based on the operational context of the machine. Moreover, the model leverages local computing, communication, and control for CPS in manufacturing to estimate discrete states and continuous variables. Initial work for anomaly detection was developed by Lee *et al.* [7]. This paper extends [7] with three main contributions:

The *first contribution* of this paper is a framework for modeling cyber-physical manufacturing systems (CPMS) merging both physics-based and data-driven models. The framework is based on a hybrid model combining discrete states and continuous dynamics (CD) developed using HDEVS formalism.

The *second contribution* is to develop a framework for anomaly detection based on context-sensitive adaptive threshold limits and diagnosis based on context-specific classification models. Context is defined based on machine, part, and process data and information.

The *third contribution* is an experimental demonstration of the proposed framework to detect and diagnose anomalies contained within the part, machine, and process of a machining operation. Data from the machine controller and electric drives were extracted using industrial communication protocols.

This paper introduces two new aspects to consider when modeling manufacturing equipment, first context information to provide insight into the machine operation and part interactions, and second noninstantaneous events as some events are defined by a pattern in a continuous signal. Moreover, other formalisms such as hybrid finite state machines (FSM) can be extended by defining global operational states (GOS) introduced in this work and be used to develop models of CPMS using other formal methods.

The remainder of this paper is organized as follows. Section II provides background on the research area. Section III defines the modeling framework providing details of discrete states and CD. Section IV describes the anomaly detection and diagnosis methods. Section V presents a case study to validate the approach for anomaly detection and diagnosis in a machining application. Finally, Section VI concludes the paper and discusses other applications and future work.

II. Background

In this paper, an abstract model of manufacturing operations studied as CPMS is presented for anomaly detection and diagnosis.

A. Cyber-Physical Manufacturing Systems

A CPMS is composed of cyber and physical components. The cyber component includes data, control algorithms, and communication networks. The physical component includes machines, robots, and actuators interacting with a product as part of the manufacturing process. The analysis of CPMS requires data extraction and model development.

1) Data Extraction: Communication networks in manufacturing have evolved over time from the transfer of a simple binary signal to a complex exchange of messages and variables in "bus" architectures. Recent developments in Ethernet Industrial Protocol (I/P) for the machine–machine communication have enabled data exchange between different machines on the manufacturing floor. Some of the most common protocols for data extraction are OPC-UA and MTConnect. Both protocols aim to standardize information exchange in a hierarchical fashion to enable machine controller data extraction. OPC-UA is more flexible when dealing with multiple machines in a system [8], while the MTConnect protocol has been developed specifically to extract controller data from CNC machines [9].

To model and study CPMS, information about the machine and physical process is needed to create an abstract representation. Extraction of the required sensor data and context information can be accomplished by setting up a message gateway from a local controller to a server. These messages contain data from sensors monitoring continuous variables, binary signals, machine states, and event occurrences. In [10], a CPS model of a CNC machine tool was developed by extracting energy consumption and instruction codes from the controller using OPC-UA. Electric current consumption data have also been used to improve manufacturing sustainability using MTConnect [11]. However, the capability of extracting sensor data and context information to provide insight into machine operations has not been fully developed for anomaly detection.

2) Modeling: CPSs are often modeled as hybrid systems based on both discrete and continuous variables. Different formalisms have been used to model hybrid systems such as hybrid automata or FSM and hybrid Petri-nets. The formalism can be seen as the "semantics" linking the cyber and physical domains. In [12], different formalisms and tools to model CPS are discussed and compared for different applications concluding that the selection of the proper formalism depends on the application (i.e., robot control design, software design, and simulation).

Formal methods such as hybrid FSM have also been explored for modeling manufacturing machines to evaluate the reachability and robustness of a control strategy at machine level [13]. However, the analysis of manufacturing systems with multiple machines and parts using finite states machines can present some scalability challenges, particularly when adding state that describes the machine–part interaction. Hybrid Petri-nets have been used to model manufacturing systems with multiple machines for verification of possible deadlock conditions in the control logic [14], [15]. No matter which formalism is used for modeling the discrete behavior of CPS, the increasing complexity of the manufacturing operation can represent a challenge due to possible "state explosion" as the number of states increases when studying the combination of machine, part, and process.

The modeling framework presented here is based on the HDEVS formalism developed for modeling and simulation [16]. This formalism can be used for representing discrete and continuous variables along with their transition and trajectories in a hierarchical fashion [17]. In [18], the supply chain of a semiconductor manufacturing facility was modeled and simulated using HDEVS to define inventory control policies. Results in [19] show the ability to simulate complex machine operation using HDEVS. The validation and verification of hybrid or DES developed using the HDEVS formalism has been developed based on Quantized

State Systems methods and translating a HDEVS model into a hybrid automaton for verification with tools such as UPPALL [20].

Physics-based models have been developed using the identification of model parameters to estimate state and output variables. For manufacturing machines such as milling machines, robots, and conveyors, the system identification and model development steps are presented in [2]. Model development without the need for prior knowledge is discussed in [21]. In [22] a hybrid timed automaton model was developed using energy consumption based on historical data for estimation of expected behavior. However, for many manufacturing applications, information about the control strategy can be combined with expert knowledge to improve both physics-based and data-driven models.

Analysis of CPMS in industry has had a wide range of applications such as process control, manufacturing planning and scheduling, condition monitoring, and network reconfiguration. In [23], the system-level control of CPMS for decision making shows how the implementation of communication networks and cloud computing can improve the flexibility of material handling systems. Anomaly detection models have also been improved by studying CPMS given that more data is made available for process monitoring. Different models have been suggested for modeling CPMS; however, many seem to converge on a hybrid model based on discrete and continuous variables. An algorithm to specify a hybrid automaton based on historical data is presented in [22]. However, applications are still limited, and expert knowledge is needed for diagnosis in cases where results require operational context considerations.

B. Anomaly Detection

In manufacturing, anomaly and fault detection on machine tools have been extensively studied using both physics-based and data-driven models. The former is based on a mathematical model representing physical parameters and machine dynamics. The latter is based on a statistical analysis of historical data. In [2], physics-based models for fault diagnosis were developed for different machines and actuators by monitoring the difference between real and expected values of state and output variables. However, case studies show implementation challenges due to changes in the machine dynamics and an increase in signal noise during the manufacturing operation caused by the machine–part interactions. In [24], fault diagnosis of linear drives subject to system noise was improved through the use of Kalman filters. However, model uncertainties and noise are not considered.

Data-driven models often implement machine learning to build a regression or classification model. In [25], a data-driven model for fault detection was developed using joint motor torque data. The study focused on changes in data distribution caused by a fault. The model used historical data from a repetitive task under the assumption of constant trajectory and working conditions. Faults have also been detected by evaluation of states of the plant and a DES model of fault-free behavior at any point in time [26]. Supervised machine learning, where knowledge of data class, source, or condition is used by the classification algorithm, has proven to be an effective tool for diagnosing anomalies. Nonetheless, the selection of the proper classification algorithms for studying time-series data should be based on the type of data and application [27].

Limit-based methods for anomaly detection often require consideration of the impact of FPs and false negatives (type I and type II errors, respectively). This consideration can be based on cost [28], [29] or risk [30], [31]. In manufacturing, the risks associated with part or process anomalies are evaluated using failure mode and effect analysis (FMEA) [32]. However, the ability to assign risk for specific threshold limits often requires the knowledge of the manufacturing task.

Efforts to model the dynamics and operations of CPMS have been constrained to physics-based or data-driven models. Moreover, anomaly detection and diagnosis methods often do not consider the different machine–part interactions. However, new data extraction techniques such as IoT have granted access to context information that can complement both modeling strategies and anomaly detection and diagnosis algorithms. This work aims to improve modeling and analysis of CPMS for anomaly detection by introducing a multimodel framework for detection and context-sensitive classification for diagnosis.

III. Modeling Cyber-Physical Manufacturing Systems

The interconnection of information management systems and plant floor data has set the groundwork for modeling and analysis of CPMS. Information from the cyber domain, the data from the physical domain, and expert knowledge can be combined to develop new abstractions of manufacturing machines and processes. In this section, we describe an approach to model CPMS as a hybrid system, merging contextual information about the part, machine, and process with sensor and controller data and knowledge-based models. The development of the hybrid system model requires three steps: identification of GOS, identification of CD models, and definition of the hybrid system by specifying the CD for each GOS of the manufacturing operations. The hybrid system here presented is developed using the HDEVS formalism for anomaly detection. Other formalisms such as hybrid FSM could be extended by defining each GOS including the appropriate CD within each GOS.

A. Discrete States

GOS represent the discrete set of states characterized by the operational context of the machine. In [7], GOS was defined as the combination of functional, dynamic, and interactive states identified using implicit process descriptors. In this work, we extend the GOS by adding explicit process descriptors.

1) Implicit Descriptors: Implicit descriptors require interpretation of machine data and control logic by an expert to provide context. In this work, implicit descriptors are defined as states in different domains: Functional (F), Dynamic (D), and Interactive (I) using Discrete Event System Specification (DEVS) [16]. Each domain is represented in an atomic model defined as a tuple H

$$
\begin{aligned}
H^i &= (E^i, S^i, \delta^i) \quad \text{for } i \in F, D, I \text{ where}\\
E^i &= \{e_1, e_2, \ldots\} \quad \text{Set of events}\\
S^i &= \{s_1, s_2, \ldots\} \quad \text{Set of States}\\
\delta &: S \times E \to \quad S \text{ Transition function.}
\end{aligned}
$$

a) Functional: The functional domain is defined by the working conditions of the machine based on states and events.

1) *Functional State:* A qualitative aspect that captures the working condition of the machine. The functional states can be defined from the control logic based on a discrete set of conditions in which the machine can be operating (e.g., idle, standby, positioning, processing, changing tool, setup, etc.).
2) *Functional Event:* An instantaneous occurrence that causes a transition from one state to another. Functional events can be determined by changes in digital signals or adjacent machine states (e.g., part arrival, e-stop pushed, etc.).

Identification of functional states requires some information about the control system. This information can be in the form of an FSM or control logic in the programmable logic controller (PLC). The study of the manufacturing operation may help identify the states, events, and transitions relevant for anomaly detection.

b) Dynamic: The dynamic domain is defined by the type of motion of the different actuators during the manufacturing process.

1) *Dynamic State:* It is defined as a quantitative aspect of the machine operation such as velocity. The behavior of continuous variables is bounded within specific ranges to define a discrete set (e.g., constant speed, accelerating, stopped, etc.).
2) *Dynamic Event:* An occurrence defined by rising or falling of a continuous state variable or its derivative beyond a specific limit. Dynamic events can be detected by monitoring changes in signal descriptors such as mean or slope, or root mean square (rms) (e.g., velocity or acceleration changes).

Dynamic states can be defined based on ranges of velocity, acceleration, or deceleration. Events or transitions can be detected using change-point detection [33].

c) Interactive: The interactive domain is defined by the type of contact between the machine and the part.

1) *Interactive State:* A description of the tasks or processes during a manufacturing operation based on the machine effects on the part (e.g., "cutting air," face milling, drilling).
2) *Interactive Event:* A change in the machine–part interaction characterized by a specific pattern in the time-series data. An interaction event e^I can be described by a matrix of machine output signals describing a specific pattern (Y_{pat}) (e.g., rise and fall of electric current when a machine starts cutting a part) $e^I = [Y_{\text{pat}}(1) \ldots Y_{\text{pat}}(n)]^T$.

In the manufacturing process, machines interact with a part in multiple ways. The nature of these interactions affects machine output signals differently. An understanding of the interactions can aid in anomaly detection and diagnostic processes. Identification of interactive states and events requires knowledge of the manufacturing operation to identify data patterns. Given a matrix of continuous output variables $G = [Y(1) \ldots Y(m)]^T$ collected during a manufacturing operation, the time instance when e^I has occurred can be obtained using the search algorithm in [7].

The functional, dynamic, and interactive states provide context information about the manufacturing process. The combination of all possible states from each domain can result in a state explosion. However, some combinations of states are unfeasible (e.g., idle, constant speed, face milling, etc.). A data- or knowledge-driven approach can help reduce the number of possible combinations to consider and avoid "state explosion" by identifying unreachable states. Unreachable states can be defined by those states of the machine–part interaction that cannot be reached as specific process steps are not defined as part of the manufacturing process. Moreover, when studying the machine–part interactions, some interactions are constrained due to the process requirements such as processing speeds or events for a specific manufacturing process. The knowledge of the control logic and the manufacturing operation can support limiting the number of combined states to a feasible set.

2) Explicit Descriptors: Explicit descriptors extracted from the machine- or system-level controller provide context information without the need for expert analysis. In this work, explicit descriptors are defined by the part (p), the tool (t), and the process step (s).

a) Part: A number identifying the type of part being processed is often available in the system level controller. Considering that modern machines have the ability to process different parts, extracting part type information allows one to differentiate between materials, geometries, or features when defining the operational context.

b) Tool: A number identifying the tool used in the manufacturing process is often available in the machine level controller. Considering that a machine could use different tools in a manufacturing process such as cutting tools on a CNC, or end-effectors on a robot, differentiation between tool size, geometry, or material can provide context information about the manufacturing operation.

c) Process step: A number identifying the specific step in a manufacturing process is often available in the machine level controller. Identifying the specific step in the process provides information about the task a machine is performing, which could be related to G-code instruction within a CNC machine or a moving instruction to a robot.

Machines are typically able to process various part types, operate with different tools, and perform a large number of process steps. However, the manufacturing operations for a specific part type are often limited to a finite number of tools and process steps. Expert knowledge can help identify the relationship between explicit descriptors.

3) Global Operational State: The abstraction of manufacturing equipment as a CPMS requires machine and system level controller data (e.g., continuous variables, discrete states of adjacent machines, internal and external events, part, tool, and process step) collected in discrete time given a fundamental timestep Δt. Variables are monitored every $k\Delta t$ where $k \in \mathbb{Z}^+$ represents the discrete-time unit. In this paper, we define the CPMS abstraction at a machine level as a coupled model describing a GOS defined based on implicit and explicit descriptors

$$GOS(k) = [S^F(k), S^D(k), S^I(k), p(k), t(k), s(k)].$$

For every timestep k, the GOS is defined by implicit descriptors given $S^F(k)$, $S^D(k)$, and $S^I(k)$ representing functional, dynamic, and interactive states and explicit descriptors as defined by $p(k)$, $t(k)$, $s(k)$ describing part, tool, and process step, respectively. The operational context of the machine then is studied based on a set of states represented in $\boldsymbol{GOS} = \{GOS_1, GOS_2, \ldots\}$. For example, if the machine is idle while waiting for a part to be loaded, one can define $GOS_1 = \{Idle, Stopped, NoInteraction, 0, 0, 0\}$. Once a part with ID number 1 has been loaded, tool number 5 is installed, and the manufacturing operation is initiated with process steps number 1, one can define $GOS_2 = \{Processing, Accelerating, NoInteraction, 1, 5, 1\}$.

B. Continuous

The CD model captures state and output variables in continuous time. In the most basic form, the machine dynamics can be captured in a differential equation of the form $\dot{x} = f(x, u, t)$ and $y = h(x, u, t)$, where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $u \in \mathbb{R}^q$ represent state, output, and input vectors, respectively. The functions $f(\cdot)$ and $h(\cdot)$ describe the evolution of continuous state and output variables, respectively. The proper structure of $f(\cdot)$ and $h(\cdot)$ to capture the machine dynamics can be represented in a physics-based or data-driven model. Physics-based models require prior knowledge of the machine dynamics. In [2], the structure and parameter estimation to develop physics-based models for different machines is presented. Data-driven models the historical data instead of prior knowledge of the machine dynamics. In [34], different types of data-driven models are discussed. In this work, we leverage prior art in the development of continuous models to develop a multimodel framework. Different continuous models are defined within various discrete states to develop a hybrid model.

C. Hybrid

In this paper, we define a model of the CD of a machine while operating in a specific context. Combining the discrete state and CD into a model leads to the hybrid system representation defined by the tuple M

$M = (\boldsymbol{GOS}, U, Y, X, F, H)$, where:

1) $\boldsymbol{GOS}$ represents the discrete set of GOS;
2) U is the continuous input space of the system in which the continuous input variables u take their values. For our purpose $U \subset \mathbb{R}^m$;
3) X is the continuous state space variable where $X \subset \mathbb{R}^n$;
4) Y is the continuous output space of y where $Y \subset \mathbb{R}^q$;

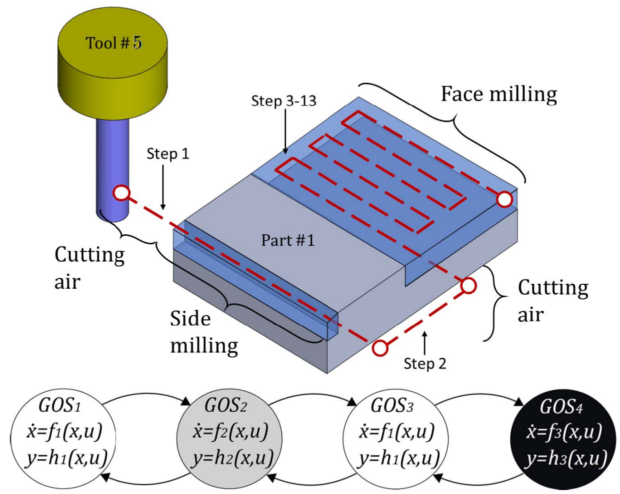

Fig. 1. Example of a hybrid model for the analysis of a machining operation.

TABLE I
HYBRID SYSTEM DESCRIPTION

	GOS_1	GOS_2	GOS_3	GOS_4
S_F	Proc.	Proc.	Proc.	Proc.
S_D	Const.	Const.	Const.	Const.
S_I	No int	Side mill.	No int	Face mill.
p	1	1	1	1
t	5	5	5	5
s	1	1	1,2,3	3-13
F	f_1	f_2	f_1	f_3
H	h_1	h_2	h_1	h_3

5) $F : \boldsymbol{GOS} \times X \times U \to TX$ is the mapping of U and X into TX that assigns a model of state variable evolution f to each GOS;
6) $H : \boldsymbol{GOS} \times X \times U \to Y$ is the mapping of U and X into Y that assigns a model of output variables h to each GOS.

A simple example is a machining operation of the part number 1 using tool number 5 following a sequence of steps 1 to 13. The machine, part, and process are modeled as a hybrid system presented in Fig. 1. Discrete and continuous behaviors are summarized in Table I.

D. Scalability

Expert knowledge can be obtained through process observation, analysis of the part manufacturing process, and review of the machine control sequence and logic. Information about the machine, part, and process can help reduce the complexity of the model by identifying an unfeasible set of states due to the machine control logic reachability constraints. For example, the logic on the programmable controller might limit the operational speed during specific cycles or process steps. Also, the command or processing steps to manufacture a specific part might require a specific type of machine–part interaction such as face tool interaction to drill a hole or high speed for face milling. Expert knowledge can help reduce model complexity by identifying two key aspects of the hybrid model, which are as follows.

1) *Discrete States:* Modeling all possible implicit and explicit descriptors of the GOS could result in a state

explosion. A knowledge-based approach can leverage the repetitive action of manufacturing to reduce the number of states based on the process requirements and capabilities.

2) *Dynamic Models:* The machine dynamics and the effect of the machine–part interaction during the manufacturing process can be captured by a limited number of models. A library of physics-based and data-driven models can then be used to monitor the manufacturing process while operating in different discrete states.
3) *Hybrid System:* As shown in Table I, the models in F can be shared between GOS as the dynamic model f_1 is used for studying the machine in GOS_1 and GOS_3. Moreover, the mapping between discrete states and dynamics models developed using a knowledge-based approach can help identify what model from the library best captures the operation on a discrete state.

IV. Anomaly Detection and Diagnosis

Identification of the proper operational state and context can help the evaluation of machine data for anomaly detection. In this work, a context-sensitive analysis framework is proposed for the detection of static anomalies. Anomalies are detected based on adaptive threshold limits by studying residuals between estimated and actual values at a single point in time. The root cause is diagnosed using supervised clustering or classification models where a specific classification model is assigned to each operational context.

A. Detection

In this work, anomalies are detected by evaluating residual values within specified intervals called thresholds. Residuals at time t are the difference between measured signals $Y(t)$ and estimated outputs $\hat{Y}(t)$. The proper dynamic model to generate the estimated output for each operational context is defined by the hybrid model. The residual generation for the output variables can then be defined as

$$r_y(t) = Y(t) - \hat{Y}(t).$$

Noise in the measured signal and model errors could lead to nonzero values under normal conditions. Filtering the signal to reduce noise and using a set of n measured values as a reference for normal or expected performance, it is possible to define the mean μ_y and standard deviation σ_y of the residual as

$$\mu_y(t) = \sum_{i=1}^{n}(r_{y_i}(t)/n) \quad \text{and } \sigma_y^2 = \sum_{i=1}^{n}(r_{y_i}(t) - \mu_y(t))^2/n.$$

Context-sensitive adaptive threshold limits are defined to separate normal and abnormal values. These limits are based on confidence in the model and risks associated with the operational context as defined by the GOS.

1) Confidence Intervals: Based on the experimental data, the confidence intervals describe the likelihood that residual values fall within a specific range. The confidence intervals for GOS_i are defined based on mean (μ_i), standard deviation (σ_i), and standard score (Z_i) as

$$\Delta r_{yi} = \mu_i \pm Z_i \sigma_i.$$

The score Z_i defines the confidence level (e.g., 90%, 95%, and 99%) to balance detection errors. The Receiver Operating Characteristic (ROC) curve can be used to evaluate the accuracy of a binary classifier as determined by a discrimination threshold based on the ratio between true positives (TPs) (detection) and an FP (false alarm) [28].

Guidelines: The Z-score defines the classification limits between normal and abnormal performance as the number of standard deviations from the mean of the expected residual. Optimal Z-score can be obtained by:

1) collecting data from normal and abnormal operations;
2) evaluate the mean and standard deviation of the residual;
3) build ROC curve by assessing the TP and FP for Z $\in \{0.1, \ldots, 3.0\}$;
4) calculate the slope $m(TP, FP)$ of the ROC curve for every Z-score;
5) the optimal Z-score balancing the tradeoffs between detections and false alarms is defined by $m(TP, FP) = 1$.

If the cost associated with false negatives is larger than the cost of an FP the optimal slope can be less than 1 [i.e., $m(TP, FP) = 0.8$] [35].

As part of a manufacturing operation, it is possible to have multiple tasks with different combinations of processes, machine setups, and parts. The confidence in a dynamic model capturing the behavior of input or output variables might be different based on the operational context. The confidence intervals for each state in $\boldsymbol{GOS}$ are defined by mean $\boldsymbol{\mu_y}$, variance $\boldsymbol{\sigma_y}^2$, and score $\boldsymbol{Z_y}$.

2) Process Risk Analysis: Using relational identifiers of specific steps or tasks in the manufacturing process can help map the risks associated with anomalous performance based on information from the FMEA. The data extracted out of the machine regarding both part and process can be used to change the allowable threshold for the output variables residuals r_y.

Different techniques to assess risk are presented in [30], [31]. In this work, we introduce a risk coefficient ψ_R to modify the detection limits for each GOS so that

$$\Delta r_y = \mu_i \pm \psi_{R_i} Z \sigma_i.$$

The risk coefficient modifies the classification limits defined by the confidence intervals based on prior risk analysis. The confidence intervals as defined by the Z-score can be calculated based on the tradeoffs between detection errors. The risk coefficient can be assigned based on the negative impact of an anomaly over the part's performance or process safety.

Guidelines: The risk coefficient ψ_R is defined by evaluating the severity of part or process failure based on FMEA. The value of ψ_R can be selected based on the following.

1) Evaluate design and process FMEA.
2) Define the critical part features or process step based on high-risk priority number.
3) Assign $\psi_R < 1$ to the GOS associated with critical part features or process steps.

The vector $\boldsymbol{\psi}_R$ defines the risk coefficient for each operational context in $\boldsymbol{GOS}$. An example of context-sensitive adaptive threshold limits for the part and process in Fig. 1 is

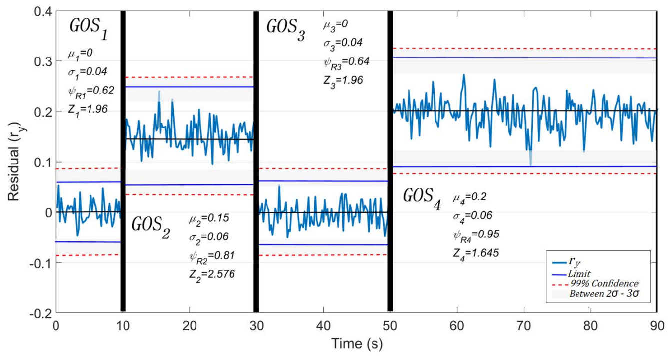

Fig. 2. Data partitioning and adaptive threshold limits.

presented in Fig. 2. Considering that the accuracy of physics-based or data-driven models during various GOS could be different, it is possible to have offsets on mean residual values.

B. Diagnosis

In a manufacturing operation, the abnormal behavior could be related to problems in the part, machine, tool, or process. Identifying the root cause using data-driven methods could be a challenge partially because changes in speed, task, and machine–part interaction cause the signal to be nonstationary. Moreover, not all anomalies are equally likely to occur under different operating conditions.

Partitioning a nonstationary output signal by GOS can improve the diagnosis model by creating multiple segments of similar operational context. After an abnormal condition has been detected in a specific GOS, a classification model is used for root cause diagnosis. In this work, we introduce context-sensitive classification models for diagnosis by 1) partitioning the signals; 2) extracting features from the different partitions of the signals; and 3) defining a specific classification model for partitions of GOS. The selection of the features to be extracted from the continuous signal such as peak value, rms, or decay time can be sensitive to the operational context of the machine as defined in the GOS partition. Moreover, different classification models can be defined for various partitions. An example would be to use supervised classification methods for root cause diagnosis [36]. A support vector machine (SVM) classification model can be developed for each partition, i.e., for each GOS_i, an SVM_i is defined for $i \in \{1 \ldots p\}$. Moreover, understanding the process and different machine–part interactions can help improve anomaly diagnosis by defining the most likely failure mode of each GOS and the effect that different anomalies have over features of a signal in the time or frequency domain.

V. Implementation and Evaluation

The methodology presented in Sections III and IV was implemented to detect anomalies in a machining operation. The experimental setup is based on a three-axis CNC machine enabled with OPC-UA communication. Using Rockwell Automation IoT adapter, we were able to extract position, velocity, acceleration, current, and voltage from each drive on the CNC machine, along with part and process information

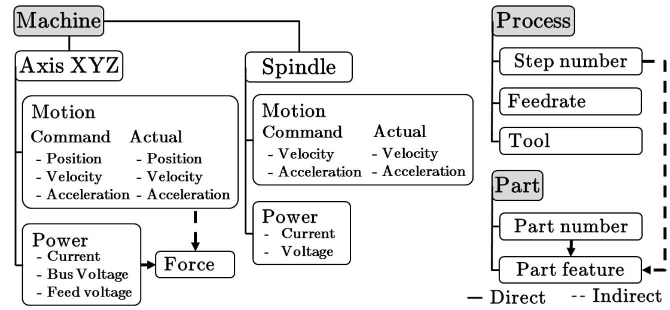

Fig. 3. IoT data extraction schema.

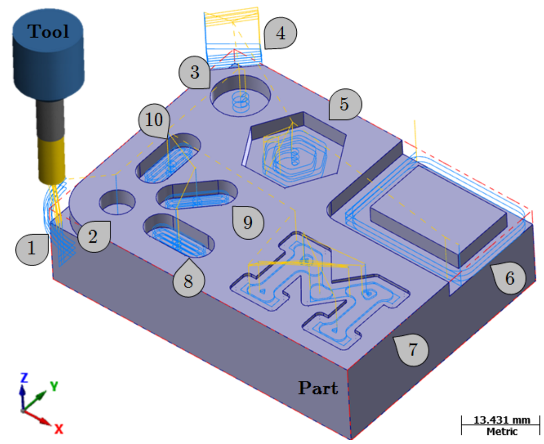

Fig. 4. Sample part machining description.

such as the part number and G-code command. The continuous signals were preprocessed using a Finite Impulse Response (FIR) filter. The machine was studied as a CPMS by considering the control architecture, communication capabilities, and manufacturing operation. The model was developed using a combination of continuous signals and context information described in Fig. 3. The validity of the model was evaluated by comparing the error between the model output and data from the real system under normal operating conditions.

The case study focused on a part with multiple features manufactured using different tools and machining operations. The study aims to detect and diagnose anomalies on the machine, part, or process. The detection was performed by monitoring the residual of output variables throughout the entire manufacturing operation, while diagnosis utilized classification models developed using context information. Fig. 4 shows the part, features, and tool trajectory. Table II describes the manufacturing operation and tool used for each part feature.

A. Cyber-Physical Manufacturing System Model

The manufacturing operation was modeled as a hybrid system based on discrete states and CD. The discrete states were defined by the operational context of the machine according to the Global Operation States GOS, and the CD in each GOS were studied by either physics-based (pb) or data-driven (dd) models.

TABLE II
SAMPLE PART AND PROCESS INFORMATION

Feature Number	Operation	Tool Number	Tool Diameter	Feedrate	Process Step
1	Side milling (fillet)	1	3/8”	2	5 to 89
2	Drilling	1	3/8”	1.8	90 to 94
3	Circular milling	1	3/8”	1.8	95 to 137
4	Side milling (chamfer)	1	3/8”	2	138 to 213
5	Pocket milling	1	3/8”	1.8	214 to 399
6	End milling	1	3/8”	2.5	400 to 474
7	Pocket milling	2	5/16”	1.5	475 to 764
8	Slot cutting (X axis)	2	5/16”	2	765 to 937
9	Slot cutting (45 deg)	2	5/16”	2	938 to 1151
10	Slot cutting (Y deg)	2	5/16”	2	1152 to 1317

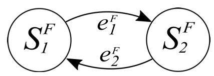

Fig. 5. Functional atomic model.

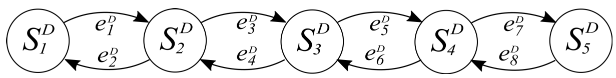

Fig. 6. Dynamic atomic model.

1) Discrete States: Defined by the combination of implicit (functional, dynamic, and interactive states) and explicit descriptors (part, tool, and process step) to specify the GOS. The implicit descriptors were defined using PLC logic, cutting speed, and tool–part interaction. The explicit descriptors were defined by part number, tool number, and line of the G-code program. The data required to identify the descriptors were extracted from the machine and system controller. The atomic model for each domain is defined as follows.

a) Functional: An atomic model of functional states built using information from the control logic. The functional states were machine $Idle$ or $Processing$. The transition between states was triggered by events $PartArrival$ and $PartDeparture$. The occurrence of an event was detected by a Presence Sensor (PS) mounted in the CNC machine. Fig. 5 shows the functional atomic model H^F including states, events, and transitions:

$$\begin{aligned} H^F &= (U^F, S^F, \delta^F) \quad \text{where} \\ U^F &= \{e_1^F, e_2^F\} \quad \text{Set of events} \\ S^F &= \{s_1^F, s_2^F\} \quad \text{Set of states} \\ s_1^F &= \text{Idle} \quad s_2^F = \text{Processing}. \end{aligned}$$

2) Dynamic: The atomic model for dynamic states included cutting and traveling speeds of the manufacturing operation. The cutting speed is defined as the rate at which the cutting tool passes along a workpiece. The speed is calculated as the magnitude of the velocity vector, $CS = (\dot{q}_x^2 + \dot{q}_y^2 + \dot{q}_z^2)^{1/2}$. The states were segmented by speed and acceleration for each drive. Fig. 6 presents the dynamic model

$$\begin{aligned} H^D &= (U^D, S^D, \delta^D) \quad \text{where} \\ U^D &= \{e_1^D, e_2^D, \ldots, e_8^D\} \quad \text{Set of events} \\ S^D &= \{s_1^D, s_2^D, \ldots, s_5^D\} \quad \text{Set of states} \\ s_1^D &: CS = 0 \quad s_2^D : CS = 1.8 \quad s_3^D : CS = 2 \\ s_4^D &: CS = 2.5 \quad s_5^D : CS = 50. \end{aligned}$$

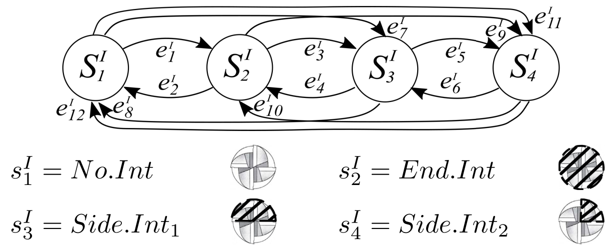

Fig. 7. Machine–part interaction states.

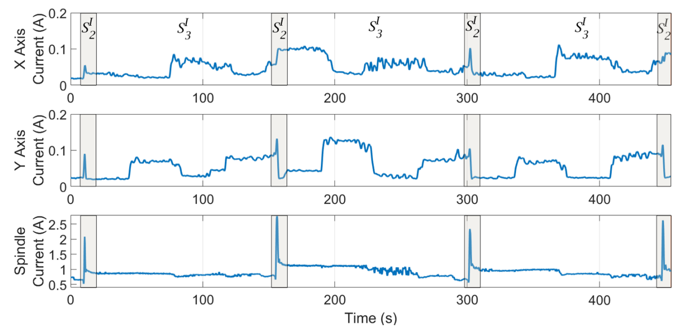

Fig. 8. Current of XY drives and spindle partitioned by interactive state.

3) Interactive: It is defined by the contact between tool and workpiece which is distinct for different machining operations. The states and operations, in this case, study include $NoInteraction$ for “cutting air” operations, $EndInteraction$ for drilling operations, and $SideInteractions$ for pocket or shoulder milling operations. Fig. 7 shows the states and transitions.

Interactive events are defined by the characteristic effects that machine–part interactions have over output signals. Process observation and signal analysis methods were combined to identify patterns that describe the effect of changes in interaction over output signals (e.g., current, voltage, etc.). Fig. 8 shows the current signature of the X-axis, Y-axis, and Spindle while machining part feature 6. Events are characterized by time-series patterns such as a spike in spindle current and a drop in the Y-axis current. Using the partitioning algorithm presented in [7], interactive events within the manufacturing process were identified.

4) Continuous Dynamics: State variables include position q and velocity $\dot{q}$, and the output variables were current I and voltage V. Considering that the dynamics of the machine and signal noise are different depending on the machine–part interaction, the multimodel framework presented in Section III was used.

a) Physics-based: Models of the X- and Y-axes drives on the CNC machine. A one-mass model based on the physics of the electric drive is defined as [2]

$$\hat{V}(t) = \psi \dot{q}(t) + L\dot{I}(t) + RI(t) \tag{1}$$

$$\hat{I}(t) = (J\ddot{q}(t) + M_{F1}\dot{q}(t) + M_{F0}\sin(\dot{q}(t)))/\psi \tag{2}$$

where the measured signals are speed $\dot{q}$, acceleration $\ddot{q}$, armature voltage V, and armature current I. The identified machine parameters are magnetic flux ψ, armature

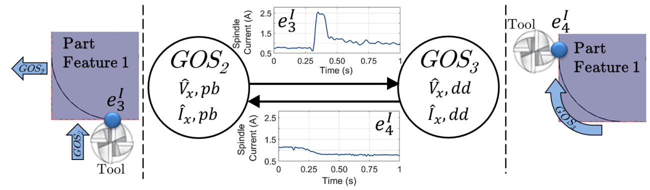

Fig. 9. Description of hybrid model with interactive events.

inductance L, armature resistance R, overall moment of inertia J, and friction coefficients M_{F0} and M_{F1}.

b) Data-Driven: Autoregressive models were developed to study the current and voltage of the X- and Y-axes drives. The order of the models was estimated based on the Box–Jenkins analysis using time series data [34]. The model was developed to estimate current (I) and voltage (V) based on previous observations, and exogenous inputs velocity ($\dot{q}$) and acceleration ($\ddot{q}$). An autoregressive model with independent predictors (autoregressive–moving-average model with exogenous inputs) was defined as

$$\boldsymbol{\phi}_V(B)\hat{V}(t) = \boldsymbol{\beta}_V(B)\dot{q}(t-n) + \varepsilon(t) \tag{3}$$

$$\boldsymbol{\phi}_I(B)\hat{I}(t) = \boldsymbol{\phi}_{I1}(B)\dot{q}(t-n) + \boldsymbol{\phi}_{I2}(B)\ddot{q}(t-n) + \varepsilon. \tag{4}$$

The parameters ϕ and β are polynomials with respect to the backward shift operator (B) identified by fitting norm-based models with regularization, n is the system delay, and ε is the system disturbance [37].

5) Hybrid Model: Used to specify which continuous model to use in each discrete state. Each part feature involved multiple GOS, but only two types of models (physics-based and data-driven) are defined based on interactive state S^I. The value of some model parameters such as friction or autoregressive terms changed based on the dynamic state S^D.

Fig. 9 shows the discrete states and continuous dynamic model for machining part feature 1 (side milling—filet) represented as a hybrid system. Two different GOS are defined. GOS_2 captures the operational context with no machine–part interaction when the machine is "cutting air" and the tool is traveling to the part entry point. During GOS_2, the machine dynamics are estimated using a physics-based model. The interactive event e_3^I is characterized by a spike in the spindle current consumption caused by the contact between the tool and the part and indicates the transition to GOS_3. During GOS_3, the tool is machining the part, and the machine dynamics are estimated using a data-driven model. The interactive event e_4^I is characterized by a drop in the spindle current consumption and indicates the transition back to GOS_2.

B. Anomaly Detection

This case study aims to detect anomalies by monitoring residuals and event occurrence. The models used to estimate the output variables are defined by the operational context of the machine and characterized by the GOS. In this case study, we evaluate the abilities to detect the following anomalies:

1) *Tool:* worn tool and broken tool.
2) *Part:* wrong material and wrong dimensions.

These anomalies can be detected by monitoring the magnitude of the residual and time intervals between occurrences of interactive events.

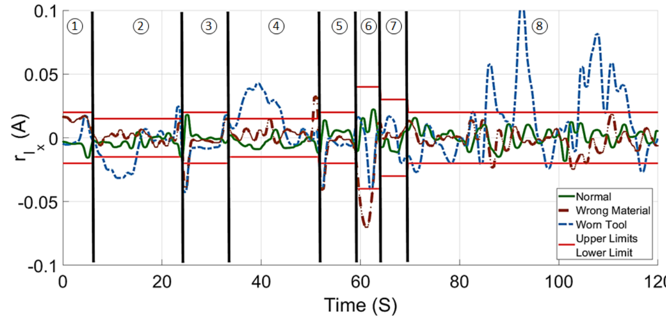

Fig. 10. Adaptive threshold limits of electric current residual.

TABLE III
RESIDUAL ANALYSIS INFORMATION

Partition	Feature	State	Interaction	Model	Limits (A)
1	1	GOS_2	No Interaction	$\hat{I}_{x,pb}$	± 0.21
2	1	GOS_3	Side Interaction	$\hat{I}_{x,dd}$	± 0.15
3	1	GOS_2	No Interaction	$\hat{I}_{x,pb}$	± 0.21
4	1	GOS_3	Side Interaction	$\hat{I}_{x,dd}$	± 0.15
5	2	GOS_2	No Interaction	$\hat{I}_{x,pb}$	± 0.21
6	2	GOS_4	End Interaction	$\hat{I}_{x,dd}$	± 0.43
7	3	GOS_5	No Interaction	$\hat{I}_{x,pb}$	± 0.29
8	3	GOS_6	Side Interaction	$\hat{I}_{x,dd}$	± 0.2

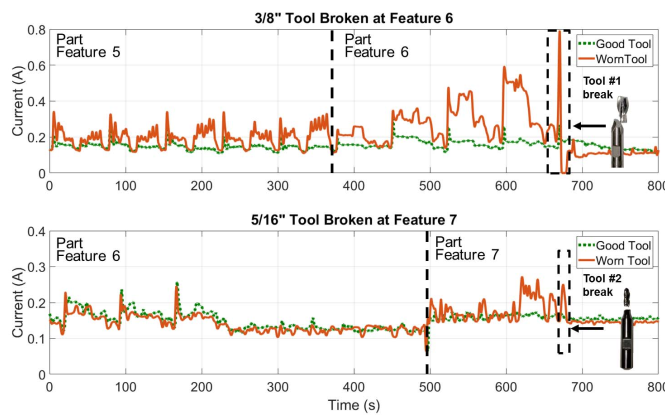

Fig. 11. Effect of worn or broken tool on spindle current for two different tool sizes and part features.

1) Residual Analysis: For anomaly detection, we implemented context-sensitive adaptive threshold limits presented in Section IV-A. The context is defined by the discrete states described in the GOS. The limits on residual were defined by mean μ and standard deviation σ estimated by evaluating the output of the continuous dynamic model for 20 independent data samples collected under normal operation. Fig. 10 shows the GOS and residual of the output variables for three part features under normal and abnormal conditions. Table III summarizes the partitions, states, model, and limits.

Results illustrate that both the wrong material and worn tool conditions cause the residual to exceed the threshold during a GOS that involves a machine–part interaction. The root cause was identified using supervised learning classification models to differentiate between these two conditions.

2) Event Occurrence: The time at which an interactive event occurs can be used to identify anomalies. Changes in the part geometry, machine fixture location, orientation, or tool

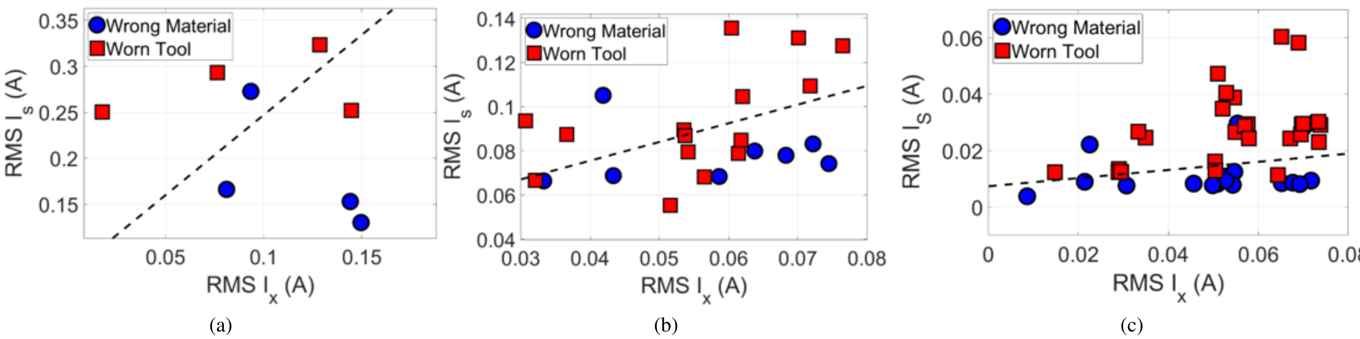

Fig. 12. Classification model for diagnosing wrong material or a worn tool. (a) Features from the entire signal, accuracy 75%. (b) Features extracted using signals partitioned by part feature, accuracy 81.2%. (c) Signal partitioned by part feature and GOS during side interaction and multiple passes, accuracy 93.6%.

condition might affect the time instance in which the machine and part interact. The time when the interactive event should occur and the time lapse of each GOS under normal conditions can be identified using historical data. As part of the case study, we identified the average and standard deviation time intervals associated with each GOS. Results showed that wrong part dimensions of -5 mm on the X-axis and -0.8 mm on the Z-axis caused an average delay on the occurrence of the interactive event of 1.39 and 0.42 s, respectively. A similar effect was observed when the part was poorly clamped causing the part to shift during the machining operation and changing the duration of an interactive state. Abnormal duration of the time interval of an interactive state can complement the anomaly detection and diagnosis process.

C. Root Cause Diagnosis

In this study, classification and rule-based methods were used to perform root cause diagnosis. After an anomaly was detected, context information was used to decouple the failure modes as not all the anomalies are equally likely to occur in different GOS and could affect the output signal in different ways.

1) Classification-Based: Supervised learning was used to identify the root cause of residual values outside the normal thresholds. A linear SVM for binary classification was trained using key characteristics in the time domain such as mean, max, peak-to-peak, and rms, and features on the frequency domain such as peak magnitude and frequency. The orthogonal transformation was used to define the set of variables that best describe the difference between different failure modes was defined to improve the classification accuracy. A soft margin, to define hyperplane that separates many, but not all data points were specified using L1-norm minimization.

The signals we studied were current and voltage from the XY drives and spindle. A total of 36 features were used to develop the classification model. Fig. 12 shows the classification hyperplane and rms values of spindle and X drive current. The results showed that considering the context information helped improve the diagnosis. The accuracy of the classification model improved from 75% when using the entire signal to 93.6% when the signal was partitioned by GOS. Partitioning the signal by part feature and GOS, and using only the states associated with side interactions S_2^I and S_3^I helped isolate the signal to stationary conditions of similar operational context.

2) Rule-Based: In this work, we used process observation and signal analysis to define the characteristics of the peak in spindle current such as max magnitude, rise time, rise level, fall time, and fall level for different part features prior to breakage. Magnitudes and patterns were used to define context-sensitive diagnosis rules. Fig. 11 shows different effects of tool breakage while machining feature 6 with a 3/8" diameter mill bit and feature 7 with a 5/16" diameter mill bit. The effect of tool breakage over spindle current is distinct for each part feature due to the different tool size and machine–part interactions involved in the manufacturing operations. The difference in magnitude between the two graphs can be explained by the distinct spindle current consumption required to increase the torsional shear stress above the failure point for the different tools. The pattern of the current consumption prior to failure could be explained by the particular interaction between the tool and the part for machining each part feature.

D. Discussion

In a manufacturing operation, anomalies can be caused by problems in the machine, part, tool, or process. In this work, anomalies in the part and tool were detected and diagnosed using a context-sensitive modeling framework. For detection, we implemented residual analysis using both physics-based and data-driven models. Results showed that anomalies related to part material or tool condition can be detected by monitoring the magnitude of the residual. Anomalies caused by changes in part dimensions or orientation had no effect on the residual but affected the time intervals between interactive events.

The nonstationary condition of the signal when studying the entire process represents a challenge for root cause diagnosis. Features extracted from the entire signal do not show a clear difference between the wrong material and worn tool. However, considering the GOS of the machine helped partition the signal and develop context-specific classification models. Moreover, knowledge of the magnitude and pattern of spindle current consumption prior to tool breakage for each part feature and GOS helped develop diagnosis rules. Results showed the advantages of using context information to improve the diagnosis of some anomalies. The steps for anomaly detection and diagnosis using the modeling framework here presented can be summarized to; first modeling, define the machine GOS and continuous dynamic models, second anomaly detection, Monitor the residual between estimated variables and machine data within the limits specified for each GOS, third diagnosis, partition the data by GOS and extract signal features for each partition for classification.

VI. Conclusion

In this paper, we presented a modeling strategy to study CPMS using a hybrid model. Discrete states are defined based on implicit and explicit process descriptors as GOS. CD are described using both physics-based and data-driven models.

The main contribution of this work is a framework to improve anomaly detection and diagnosis. Anomaly detection is based on residual analysis considering the GOS to define context-sensitive adaptive threshold limits. Root cause diagnosis is based on context-specific classification models. The benefit of this framework is the ability to diagnose anomalies in the machine, part, or tool to support effective maintenance actions. Timely and effective maintenance action can help reduce downtime and improve manufacturing productivity. The modeling approach was implemented in a machining operation. Results demonstrated that context information improved the classification accuracy from 75% to 94%, and enhanced the detection and diagnosis of tool breakage.

Future work will focus on expanding the modeling framework, testing scalability, model verification, and implementing additional data extraction techniques. The effect of hidden or nonobservable states in the machine controller will be explored in the continuation of this work. This research work will be extended to study other machines, including a wider range of anomalies, and developing predictive models to detect dynamic anomalies.

References

[1] F. Lopez *et al.*, "Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1885–1892, Oct. 2017.

[2] R. Isermann, *Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. New York, NY, USA: Springer, 2011.

[3] J. V. Abellan-Nebot and F. R. Subirón, "A review of machining monitoring systems based on artificial intelligence process models," *Int. J. Adv. Manuf. Technol.*, vol. 47, nos. 1–4, pp. 237–257, 2010.

[4] S. Faltinski, H. Flatt, F. Pethig, B. Kroll, A. Vodenčarević, A. Maier, and O. Niggemann, "Detecting anomalous energy consumptions in distributed manufacturing systems," in *Proc. IEEE 10th Int. Conf. Ind. Informat.*, Jul. 2012, pp. 358–363.

[5] M. Engeler, D. Treyer, D. Zogg, K. Wegener, and A. Kunz, "Condition-based maintenance: Model vs. statistics a performance comparison," *Procedia CIRP*, vol. 57, pp. 253–258, Jan. 2016.

[6] J. Lee, B. Bagheri, and H.-A. Kao, "Recent advances and trends of cyber-physical systems and big data analytics in industrial informatics," in *Proc. Int. Conf. Ind. Informat. (INDIN)*, 2014, pp. 1–6.

[7] M. Saez, F. Maturana, K. Barton, and D. Tilbury, "Anomaly detection and productivity analysis for cyber-physical systems in manufacturing," in *Proc. IEEE Conf. Autom. Sci. Eng.*, Aug. 2017, pp. 23–29.

[8] N. M. Torrisi and J. F. G. de Oliveira, "Remote monitoring for high-speed CNC processes over public IP networks using CyberOPC," *Int. J. Adv. Manuf. Technol.*, vol. 60, nos. 1–4, pp. 191–200, 2012.

[9] A. Vijayaraghavan and D. Dornfeld, "Addressing process planning and verification issues with MTconnect," *Lab. Manuf. Sustainability*, Jun. 2009.

[10] J. Chen *et al.*, "CPS modeling of CNC machine tool work processes using an instruction-domain based approach," *Engineering*, vol. 1, no. 2, pp. 247–260, 2015.

[11] K. Vikhorev, R. Greenough, and N. Brown, "An advanced energy management framework to promote energy awareness," *J. Cleaner Prod.*, vol. 43, pp. 103–112, Mar. 2013.

[12] D. Broman, E. A. Lee, S. Tripakis, and M. Törngren, "Viewpoints, formalisms, languages, and tools for cyber-physical systems," in *Proc. 6th Int. Workshop Multi-Paradigm Modeling*, 2012, pp. 49–54.

[13] R. G. Qiu and S. B. Joshi, "A structured adaptive supervisory control methodology for modeling the control of a discrete event manufacturing system," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 29, no. 6, pp. 573–586, Nov. 1999.

[14] G. Frey and L. Litz, "Formal methods in PLC programming," in *Proc. IEEE Int. Conf. Syst., Man*, vol. 4, Oct. 2000, pp. 2431–2436.

[15] V. Vyatkin and H.-M. Hanisch, "Verification of distributed control systems in intelligent manufacturing," *J. Intell. Manuf.*, vol. 14, no. 1, pp. 123–136, 2003.

[16] B. P. Zeigler, T. G. Kim, and H. Praehofer, *Theory of Modeling and Simulation*. New York, NY, USA: Academic, 2000.

[17] E. Kofman, M. Lapadula, and E. Pagliero, "PowerDEVS: A DEVS-based environment for hybrid system modeling and simulation," School Electron. Eng., Universidad Nacional de Rosario, Santa Fe Province, Argentina, Tech. Rep. LSD0306, 2003.

[18] D. Huang *et al.*, "Simulation of semiconductor manufacturing supply-chain systems with DEVS, MPC, and KIB," *IEEE Trans. Semicond. Manuf.*, vol. 22, no. 1, pp. 164–174, Feb. 2009.

[19] C. H. Sung, J. H. Hong, and T. G. Kim, "Interoperation of DEVS models and differential equation models using HLA/RTI: Hybrid simulation of engineering and engagement level models," in *Proc. Spring Simulation Multiconf.* Philadelphia, PA, USA: SIAM, 2009, p. 150.

[20] H. Saadawi and G. Wainer, "On the verification of hybrid DEVS models," in *Proc. Symp. Theory Modeling Simulation-DEVS Integrative M&S Symp.* Philadelphia, PA, USA: SIAM, 2012, p. 26.

[21] A. P. Estrada-Vargas, E. López-Mellado, and J.-J. Lesage, "A black-box identification method for automated discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1321–1336, Jul. 2015.

[22] O. Niggemann, J. S. Kinnebrew, H. Khorasgani, S. Volgmann, A. Bunte, and G. Biswas, "Data-driven monitoring of cyber-physical systems leveraging on big data and the Internet-of-Things for diagnosis and control," in *Proc. DX Safeprocess*, 2015, pp. 185–192.

[23] S. Sastry, M. S. Branicky, and P. Sastry, "Cloud conveyors system: A versatile application for exploring cyber-physical systems," in *Control of Cyber-Physical Systems*. New York, NY, USA: Springer, 2013, pp. 43–62.

[24] S. Huang, K. K. Tan, and T. H. Lee, "Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4285–4292, Nov. 2012.

[25] A. C. Bittencourt, K. Saarinen, S. Sander-Tavallaey, S. Gunnarsson, and M. Norrlöf, "A data-driven approach to diagnostics of repetitive processes in the distribution domain—Applications to gearbox diagnostics in industrial robots and rotating machines," *Mechatronics*, vol. 24, no. 8, pp. 1032–1041, 2014.

[26] M. Roth, S. Schneider, J.-J. Lesage, and L. Litz, "Fault detection and isolation in manufacturing systems with an identified discrete event model," *Int. J. Syst. Sci.*, vol. 43, no. 10, pp. 1826–1841, 2012.

[27] T. W. Liao, "Clustering of time series data—A survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, 2005.

[28] J. R. Moyne, "Method and apparatus for optimizing profit in predictive systems," U.S. Patent 9 582 828, Feb. 28, 2017.

[29] M. Last, A. Sinaiski, and H. S. Subramania, "Predictive maintenance with multi-target classification models," in *Proc. Asian Conf. Intell. Inf. Database Syst.* New York, NY, USA: Springer, 2010, pp. 368–377.

[30] H. J. Geissler *et al.*, "Risk stratification in heart surgery: Comparison of six score systems," *Eur. J. Cardio-Thoracic Surg.*, vol. 17, no. 4, pp. 400–406, 2000.

[31] A. T. de Almeida, C. A. V. Cavalcante, M. H. Alencar, R. J. P. Ferreira, A. T. de Almeida-Filho, and T. V. Garcez, *Multicriteria and Multiobjective Models for Risk, Reliability and Maintenance Decision Analysis*. New York, NY, USA: Springer, 2016.

[32] W. Gilchrist, "Modelling failure modes and effects analysis," *Int. J. Qual. Rel. Manage.*, vol. 10, no. 5, pp. 153–169, May 1993.

[33] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," in *Data Mining in Time Series Databases*, vol. 57. Singapore: World Scientific, 2004, pp. 1–22.

[34] J. D. Hamilton, *Time Series Analysis*, vol. 2. Princeton, NJ, USA: Princeton Univ. Press, 1994.

[35] N. A. Macmillan and C. D. Creelman, *Detection Theory: A User's Guide*. Essex, U.K.: Psychology Press, 2004.

[36] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mech. Syst. Signal Process.*, vol. 21, no. 6, pp. 2560–2574, 2007.

[37] L. Ljung, *System Identification Toolbox: User's Guide*. Natick, MA, USA: Citeseer, 1995.

Miguel A. Saez received the bachelor's degree in mechanical engineering from La Universidad del Zulia, Maracaibo, Venezuela, in 2008, and the M.Eng. degree in global automotive and manufacturing from the University of Michigan, Ann Arbor, MI, USA, in 2015, where he is currently pursuing the Ph.D. degree in mechanical engineering.

From 2007 to 2010, he worked as a Product Engineer at Dana Holding Corporation, Valencia, Venezuela, leading multidisciplinary design and manufacturing projects, coordinating cost-saving projects using CAD/CAE tools, and testing NVH for vehicles powertrain. From 2010 to 2013, he worked as a Senior Manufacturing Engineer at General Motors, Valencia, managing a capital investment project for new vehicle programs and design, testing, and installing semiautomated manufacturing systems.

Francisco P. Maturana (M'16) received the Ph.D. degree in intelligent manufacturing systems from the University of Calgary, Calgary, AB, Canada.

He is currently an Industrial Internet of Things (IoT) Architect and a Senior Principal Engineer with the Center of Excellence and Strategic Development, Rockwell Automation, Cleveland, OH, USA. He is involved in developing industrial- and enterprise-level solution architectures in both Platform as a Service and Infrastructure as a Service platforms. He holds more than 40 patents in advanced computing and numerous international publications. His research interests include distributed computing, artificial intelligence, cloud/edge analytics and big data, multiagent systems, and smart manufacturing.

Dr. Maturana is a member of the Industrial Internet Consortium (IIC) and the Computer Simulation International Society.

Kira Barton received the B.Sc. degree in mechanical engineering from the University of Colorado, Boulder, CO, USA, in 2001, and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of Illinois, Urbana–Champaign, Champaign, IL, USA, in 2006 and 2010, respectively.

From 2010 to 2011, she was a Post-Doctoral Researcher with the University of Illinois, Urbana–Champaign. She joined the Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA, where she is currently an Assistant Professor. She is involved in conducting research in modeling, sensing, and control for applications in advanced manufacturing and robotics, with a specialization in iterative learning control and microadditive manufacturing.

Dr. Barton was a recipient of an NSF CAREER Award in 2014, the 2015 SME Outstanding Young Manufacturing Engineer Award, the 2015 University of Illinois, Department of Mechanical Science and Engineering Outstanding Young Alumni Award, and the 2016 University of Michigan, Department of Mechanical Engineering Department Achievement Award.

Dawn M. Tilbury (F'08) received the B.S. degree in electrical engineering *(summa cum laude)* from the University of Minnesota, Minneapolis, MN, USA, in 1989, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1992 and 1994, respectively.

In 1995, she joined the faculty of the University of Michigan, Ann Arbor, MI, USA, where she is currently a Professor of Mechanical Engineering with a joint appointment in Electrical Engineering and Computer Science. She has published more than 150 articles in refereed journals and conference proceedings. Her research interests lie broadly in the area of control systems, including applications to robotics and manufacturing systems.

Dr. Tilbury has been a fellow of the ASME since 2012. She is a Life Member of SWE.