Real-time Manufacturing Machine and System Performance Monitoring Using Internet of Things

Miguel Saez[1], Francisco Maturana[2], Kira Barton[1] and Dawn Tilbury[1]

***Abstract*—This paper introduces a framework to assess the performance of manufacturing systems using hybrid simulation in real-time. Continuous and discrete variables of different machines are monitored to analyze performance using a virtual environment running synchronous to plant floor equipment as a reference. Data is extracted from machines using Industrial Internet of Things (IIoT) solutions. Productivity and reliability of a physical system are compared in real-time with data from a hybrid simulation. The simulation uses Discrete Event Systems (DES) to estimate performance metrics at a system level, and Continuous Dynamics (CD) at a machine level to monitor input and output variables. Simulation outputs are used as a reference to detect abnormal conditions based on deviations of real outputs in different stages of the process. This monitoring method is implemented in a fully automated manufacturing system testbed with robots and CNC machines. Machines are integrated on an Ethernet/IP control network using a Programmable Logic Controller (PLC) to coordinate actions and transfer data. Results demonstrated the capacity to perform real-time monitoring and capture performance errors within confidence intervals.**

***Note to Practitioners*— Estimating expected performance of a manufacturing system processing different parts across multiple machines is a complex problem due to the lack of closed-form equations. Existing solutions focus on monitoring stochastic variables such as production or failure rate, or machine dynamics in separate environments often running asynchronous to the real system. This paper addresses the problem of monitoring and assessing the performance of complex manufacturing systems in real-time. The proposed framework uses a real-time hybrid simulation of manufacturing at a machine and system level. The hybrid approach is based on a discrete and continuous model of manufacturing equipment integrated to run synchronously with the real plant floor operation. Data from both the virtual and real environments is merged to assess performance. Deviations from expected values represent an error that can trigger a warning signal to production, maintenance, and/or manufacturing personnel at the plant regarding health and productivity of plant operations.**

Index terms: Discrete Event Systems, Real-time Hybrid Simulation, Manufacturing Automation and Control, Cyber-Physical Systems

I. Introduction

The performance of manufacturing systems is affected by both the behavior of independent machines and by their interactions. Plant floor operational efficiency is often controlled by monitoring some performance indicators and taking corrective actions when the system deviates from expectations. Metrics such as throughput, processing time, reliability, and quality are usually monitored in the plant floor to assess performance. Manufacturing Execution Systems (MES) have been identified as a solution to monitor and supervise factory operations [1] using Overall Equipment Effectiveness (OEE) as the performance indicator. This indicator is applied in a production environment at the system level to assess availability, productivity, and quality [2]. However, collecting, processing, and analyzing data from the plant floor is a complex problem, particularly when the system operates under non-steady state conditions such as changes in demand, machine failure, rescheduling, or system reconfiguration. To close the loop for controlling manufacturing systems, it is necessary to have a reference for the expected performance in real-time and compare it to actual plant floor data. Considering that production requirements and machine operations can change rapidly in a flexible manufacturing system, monitoring and assessment tools should be able to adapt and run synchronously to the plant floor.

Manufacturing systems producing individual or separate parts are often modeled using Discrete Event System (DES) formalisms. However, the manufacturing operation can be studied at two different levels. At a system level where the operation is mainly discrete, performance is analyzed by specifying a sequence of events into a DES model [3], [4]. At a machine level, models are often hybrid to capture a discrete set of states and transitions along with some continuous dynamics. Performance at a machine level can then be estimated by simulating the effect of discrete and continuous input variables over some machine output variables under specific working conditions [5], [6]. The use of simulation to evaluate the performance of a manufacturing system under different scenarios using a plant or controller model has grown in recent years [7]. However, often the machine and system-level simulations run separately, limiting the capacity to evaluate the effects of a sequence of events on the machine dynamics. Moreover, if the simulations are not running in parallel and in the same context as the plant floor operation, the simulation outputs might not be a valid reference for real-time performance assessment. The goal of this work is to answer the following questions: 1) How to evaluate performance at both the machine and system levels in real-time when operating under non-steady state conditions? 2) How to correlate system level performance metrics to machine variables?

Machine and system level data from discrete and continuous variables can be used for real-time monitoring, performance assessment, and anomaly detection. Different variables related to machine productivity and dynamics have been used to evaluate machine health, detect faults, and control production [8]. However, extracting machine data remains a major

[1]Mechanical Engineering Department, University of Michigan, Ann Arbor, MI. {migsae, bartonkl, tilbury}@umich.edu
[2]Rockwell Automation, Cleveland, OH. fpmaturana@ra.rockwell.com

implementation challenge. Recent developments in Industrial Internet of Things (IIoT) and communication protocols have simplified data collection, making possible advanced monitoring techniques [9], [10], [11].

In this work, a new approach for modeling and assessing the performance of manufacturing systems is proposed. One of the main challenges of using a virtual model to evaluate the performance of a real system is the asynchronous execution of the virtual environment. Moreover, performance assessment using simulation could be inaccurate due to the differences in operational context between the real and virtual environments. To address these challenges, the proposed models run in real-time which is defined as "true" or "wall clock" time, and concurrent which is defined as actions developing at the same time in both plant floor and simulation. The plant floor model captures the stochastic timed behavior of manufacturing processes and machine dynamics combined into a single environment. The proposed framework combines modeling of hybrid systems with plant floor data extraction using IIoT to solve some of the synchronization and performance assessment challenges. By monitoring both the machine and system level variables with synchronous operation between the real and virtual environments, managers on the plant floor are supported in their decision making. Analysis of simulation outputs obtained under the same context as the plant floor provides a reference for expected OEE performance in real-time for both steady-state and non-steady-state operating conditions.

Initial work by the authors for real-time performance assessment using hybrid simulation was developed in [12]. This manuscript extends [12] with three main contributions.

The *first contribution* of this paper is a mathematical framework for hybrid models and a simulation environment capable of running in real-time. The model captures stochastic operational time and deterministic machine dynamics within a single virtual environment capable of running in synchrony with real manufacturing systems.

The *second contribution* of this paper is to present a set of rules to assess performance based on data from both virtual and real environments. Statistical testing is described for model validation and performance analysis at a machine and system level. Rules are defined to identify abnormal conditions at a machine level and the impact of these abnormal conditions on system-level performance measures.

The *third contribution* is an experimental demonstration of the proposed hybrid framework to detect anomalies and monitor performance at a machine level considering system-level interactions. The model has been evaluated on a fully automated manufacturing testbed using IIoT to extract data from the machines and the system controller.

The remainder of this paper is organized as follows. Section 2 provides background on the research area. Section 3 defines the hybrid simulation providing details of discrete and continuous machine models. Section 4 describes the performance analysis rules and plant floor data extraction. Section 5 demonstrates the validity of the approach through a case study using a University of Michigan testbed. Finally, Section 6 concludes the paper and discusses other applications and future work.

II. Background

In this paper, performance analysis at both machine and system levels is based on comparing data from real and virtual environments. This background section reviews the state-of-the-art for modeling and simulation of discrete and continuous plant floor operations, performance evaluation, and automation for data extraction.

A. Models and Simulation of Manufacturing Systems

The goal of modeling manufacturing systems is to gain insight into a specific aspect of the operation such as productivity, safety, or controllability. Multiple methods have been developed to model plant floor dynamics or controller actions, often using a Discrete Event System (DES) formalism. Selection of the adequate formalism depends on the required analysis. With the proper DES model, the system response to a specific set of inputs can be studied using simulation. A comparison of modeling formalisms and simulation tools for several types of manufacturing systems to study aspects such as planning and scheduling, real-time control, and optimization was presented in [7]. The results show an increasing trend in the use of simulation to support plant floor decision making and highlight the difficulties of real-time analysis due to complexity, stochastic nature, and data collection challenges. In [13], simulation and real-time machine information was used to develop scheduling and dispatching rules. Simulation has been implemented in semiconductor fabrication to develop dispatching rules in real-time in reaction to unexpected events [14]. However, it is common to call a simulation "real-time" when inputs are received from the plant floor in real-time and trigger a simulation that runs asynchronous to plant floor systems. The results can be used to forecast performance over a fixed period of time. Asynchronous simulations are helpful when studying mass-production systems to reduce cost [15] or improve productivity, however, for performance monitoring and assessment it is evident the need of a synchronous simulation running in real-time.

As companies seek to expand system level perspective of their operations, DES models have been extended to capture additional information about the plant floor such as machine Continuous Dynamics (CD), financial information, and maintenance strategy. Industry's interest in hybrid systems, which is the combination of DES with CD, has grown in recent years [16]. Hybrid models capturing system and machine level interaction can help trace problematic behavior and identify the root cause. In [17], an enterprise-level hybrid simulation is presented to study the discrete operation and the continuous dynamics of inventory, production, and sales. CD captured the long-term effects while DES showed the short-term effects of a decision. However, machine dynamics and interactions needed to monitor machine health and identify the need of maintenance action, or plant floor reconfiguration are not considered in the model.

1) Discrete Models: For modeling processes where manufacturing machines and systems can be described by a set of discrete states, different DES formalisms with event or time-driven transitions have been developed. A detailed comparison of some of these formalisms can be found in [18]. A discussion on the selection of the formalisms and analysis framework based on modeling viewpoint and concern aspect is presented by Broman *et al* [19]. The study shows that syntax of some formalisms might be better suited for performance analysis, model checking, or controller design. Selection for the proper abstract representation depends on the analysis requirements.

Some formalisms such as Finite State Machines (FSM) and Petri Nets (PN) have been extensively implemented in the design phase of a manufacturing system life cycle for control verification. In [20], performance of a manufacturing operation was improved by studying the robot dynamics in a discrete set of conditions and programming an FSM as part of the control strategy. However, due to challenges in scalability, and constraints in capturing concurrent activities, FSM has limitations when modeling large manufacturing systems [21]. PN is a graphical tool used for modeling large DES that operates with concurrent tasks. Controllability and possibility of deadlock or livelock in automated manufacturing systems has been studied using hierarchical PN models [22]. Moreover, the optimal configuration of the controller can be obtained by formulating the PN structure as an integer linear programming model to find a deadlock-free setup [23]. Finding proper controller configurations can help improve utilization and productivity of a manufacturing system. However, the latter is affected by other aspects such as machine processing time and reliability which are often not included in basic PN models. PN have been used successfully for modeling the controllers [24], [25]. Some features of the plant such as processing time have been used to evaluate production rate, downtime and work-in-process of a manufacturing system for different layouts and production mix by extending PN models [26]. Similar work was presented in [27] including machine degradation models to estimate the effect of failure rates over utilization and work-in-progress. However, due to increased complexity when adding different features at a machine and system level, PN could have limitations in modeling a plant.

Much of the research on modeling plant floor operations of manufacturing systems has focused on throughput analysis, production scheduling, process planning, and performance measurement [16]. Some DES formalisms developed specifically for simulation purposes have an increasing trend in industry applications. Discrete Event System Specification (DEVS) has been used for modeling and simulation of a wide class of dynamics systems. Some of the key features of DEVS for modeling manufacturing systems are modular and hierarchical configurations, capacity to capture the deterministic and stochastic event- or timed-based transitions, ability to handle concurrent tasks, analysis of continuous dynamics, and a wide range of available software packages able to interact with other applications [28]. In manufacturing applications, DEVS has been used to model automated plant operations, where the simulation interacts with inputs and outputs of the logic controller for process verification [29]. The productivity of a manufacturing system has been improved by combining DEVS models and Model Predictive Control (MPC) in the semiconductor manufacturing domain [30]. The use of simulations helped maintain stable operation under nonlinear and stochastic plant dynamics. In [31], a DEVS model built on Matlab/SimEvent was used to analyze the fabrication process in a nuclear facility and gain insight into an efficient operation schedule. Plant models developed on DEVS thus far capture the time-driven transitions of discrete states, however, the models do not include the machine dynamics and are not real-time capable.

2) Continuous Models: Some key performance indicators of a manufacturing process can be modeled based on continuous dynamics. Robot, conveyor, and CNC machine dynamics can be modeled using differential equations to monitor state-variables and outputs. Different multi-body systems have been modeled using equations of motion for kinematic and dynamic analysis [32]. A model of a 6-DOF parallel robot built in SimMechanics allowed joint position as a function of time to be monitored [33], [34]. In [35], a virtual CNC machine (including electrical and mechanical components) was modeled to simulate position error over time. Component level dynamic simulations have been used to simulate output variables given an input. In a CNC machine, motor torque or current has been simulated given a position command [36]. Simulation tools for virtual commissioning in real-time can be used to visualize control action and machine dynamics [37]. This approach has been used to reduce time in design and validation stages, but is yet to be extended to real-time performance monitoring to leverage the capacity to synchronize controller and simulation.

When a system is better described by the evolution of continuous variables while operating in a specific discrete state it can be modeled as a hybrid system. Sung *et al.* [38] developed a framework for simulation of hybrid systems for high-level architectures using analog-to-event and event-to-analog converters. The approach developed based on DES and CD identified the need to study hybrid systems for machine level applications [38]. A study of continuous dynamics has been used for anomaly detection by monitoring residuals between expected and real values. In [39] an anomaly detection algorithm based on modeling machines as hybrid systems and studying residuals between the model and current values of continuous variables from the plant is presented. However, these models are neither fully synchronized nor running in real-time.

B. Manufacturing Systems Performance Analysis

The different performance metrics evaluated throughout a manufacturing system life cycle are compared by Leung *et al.* [40]. In design and validation stages, performance metrics include the possibility of deadlock, reliability, and quality. Once functionality of the process has been verified and the manufacturing system is operational, performance is defined by productivity metrics such as utilization, production rate, work-in-process (WIP) and part flow time.

Production metrics and machine health are often monitored by a Manufacturing Execution System (MES) [41]. MES links Enterprise Resource Planning (ERP) with plant floor equipment to monitor performance using Overall Equipment Effectiveness (OEE) and Key Performance Indicators (KPI) [42].

The former combines availability, productivity, and quality in a single metric. The latter is used to monitor product or process variables that characterize performance. In [43] machine downtime data was used to analyze severity of a fault based on availability impact, showing the importance of monitoring machine level performance to find the best maintenance policy in parallel production systems. In [44], a data-driven KPI is developed to predict, diagnose, and evaluate the performance of an industrial hot strip mill. The mathematical model was implemented to predict exit strip thickness as a function of process variables. These applications showed the advantages of using a dynamic approach and the importance of data-driven decision making to improve manufacturing performance and part quality. However, they do not provide an insight into the expected performance in real-time.

Manufacturing system performance analysis is a complex problem. The interaction of multiple machines and buffers can be difficult to predict. Production System Engineering (PSE) has developed an analytical solution to study throughput, WIP, and blockage and starvation for a system operating under steady-state conditions with a single part type [45]. PSE models machines using parametric distributions of productivity and reliability, along with buffer capacity, to identify bottlenecks based on blockages and starvations. At a machine level, this modeling method can be implemented in other frameworks, and presents a systematic approach for process improvement. However, a PSE approach to modeling system level interactions as Markov chains may fail to capture the dynamics of a manufacturing system with different machines and multiple stages processing various part types [46].

The importance of modeling and simulation of manufacturing systems to improve performance is demonstrated in [47]. Results show how productivity of complex manufacturing systems measured by OEE is affected by machine level performance as described by cycle time, downtime, and quality. Moreover, productivity is sensitive to machine location in the system. A detailed analysis of the relationship between equipment timing and location over system level performance highlights the capability of simulation as a predictive tool for bottleneck detection and performance diagnosis [48]. Nonetheless, the continuous machine dynamics are not included in the analysis, and the concurrent analysis between the real and simulated systems is not discussed.

C. Plant Floor Automation and Data Extraction

Collecting information from the plant floor and calculating performance metrics in real-time can be challenging without proper automation and control. Manufacturing systems generate a large amount of data that can be used for performance analysis. Sensors, condition monitors, and machines connected to the system level controller generate data that is can be used in the estimation of states and machine health. Communication between simulated and real environment can be used to test extended versions of a manufacturing system [49]. In the real system, a programmable Logic Controller (PLC) often supports manufacturing control by coordinating tasks between machines or devices based on a low-level logic program [50].

IIoT has grown in popularity for data extraction. In [51], MTConnect protocol was implemented to extract state variables and outputs from machine controllers. Data from these state variables and simulation results were used to evaluate the most sustainable manufacturing setup [52]. However, MTConnect has been limited to machine level data extraction. In [53] the implementation of IIoT in a manufacturing system with a focus on Radio Frequency Identifier (RFID) for data collection was discussed. An RFID tag carries process information that is used by the controller to trigger the proper CNC and robot programs. In [54] , plant floor data describing processing tasks and parts was extracted using RFID. Information of discrete states was used to update a PN model and trigger transitions in real-time. In [55] process time, quality, and cost calculations helped manage the expectation of plant floor operations for reconfiguration. However, the data extracted from the system was only the machine's discrete variables (e.g: machine states, events); the continuous state variables while operating in a specific discrete state were not included.

Some of the challenges for implementation of IIoT are discussed in [56], where standardization, security, and data synchronization are highlighted. The implementation of IIoT enables moving data from the resource to application layer to identify, monitor, and manage manufacturing resources. However, the interaction between a real plant floor and a virtual plant model using data extract via IIoT was not discussed. Moreover, merging the two environment has the potential to improve the performance analysis and control actions in manufacturing systems.

III. Hybrid Simulation Model

To effectively evaluate manufacturing system productivity and machine operations, a hybrid model combining discrete and continuous parameters in real-time is developed. Machines are modeled using discrete event systems with continuous dynamics. System level behavior is studied by extending the discrete event model to capture the interactions of multiple components such as machines and buffers. This novel approach using real-time hybrid simulation to monitor and assess manufacturing performance requires two steps: first, modeling single machines and system interactions, and second, real-time synchronization of the virtual and real environments. Note: If x is a variable in the physical domain, $\hat{x}$ is the corresponding simulation variable.

A. Modeling machines and interactions

Machines are modeled as hybrid systems to capture both discrete and continuous behavior. Given that manufacturing equipment often operates on a discrete set of states with event- or time-driven transitions, the asynchronous behavior is modeled as a Discrete Event System (DES). The continuous dynamics (CD) inside some states are studied using differential equations to capture the rate of change of certain variables in a synchronous model. Both discrete and continuous models are then merged in a single simulation environment.

1) Discrete: In this paper we model manufacturing systems using the Discrete Event System Specification (DEVS) formalism [28]. The formalism models discrete-event systems based on inputs, outputs, states, and transition functions. DEVS is based on two types of models: atomic and coupled. The atomic models describe individual component behavior, while coupled models describe the connection or interaction of several atomic components. An atomic model of each component in the system is represented as a tuple H:

$H = (U, Y, S, \delta_{int}, \delta_{ext}, \Delta, \lambda, t_{adv})$ where:

$U = \{e_{i1}, e_{i2}, ...\}$	Set of inputs
$Y = \{e_{o1}, e_{o2}, ...\}$	Set of outputs
$S = \{s_1, s_2, ...\}$	Set of states
$\delta_{int}: S \times \{t_{adv}, \emptyset\} \rightarrow S$	Internal transition function
$\delta_{ext}: S \times U \rightarrow S$	External transition function
$\Delta = \{\delta_{int}, \delta_{ext}\}$	Set of Transition Functions
$\lambda : \Delta \rightarrow Y$	Output function
$t_{adv} = \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, ...\}$	Set of transition times

Atomic models for event generators, buffers, and processors are presented in [57]. A machine with event- and time-driven transition functions is defined as follows:

- $\delta_{int}(s_j, \tau_j) = s_i$ defines a transition from s_j to s_i after some time advance τ_j.
- $\delta_{ext}(s_i, e_i) = s_j$ defines an transition from s_i to s_j given that input event e_i has occurred.
- $\lambda(\delta_{int}) = e_o$ defines an output function of transition δ_{int} which results an output event e_o
- $\boldsymbol{\tau}_j$ is a random variable in space of probability distribution functions ξ over $\mathbb{R}^+$, so that $\boldsymbol{\tau}_j \in \xi$. For example, in Fig. 1 the time to process a job (*cycle time*) and time to recover from a fault (*time to repair*), are specified by the realization of the random variables τ_1 and τ_2 respectively.

Modeling machines requires a description of all possible states. A simple example of an atomic model for a machine m with only three possible states is shown in Fig. 1 and represented by:

$$
\begin{aligned}
U &= \{job_{in}, fault\} \\
Y &= \{job_{out}, ready\} \\
S &= \{Idle, Busy, Down\} \\
\delta_{int} &= \begin{cases} \delta_{int}(Busy, \tau_1) = Idle \\ \delta_{int}(Down, \tau_2) = Idle \end{cases} \\
\delta_{ext} &= \begin{cases} \delta_{ext}(Idle, job_{in}) = Busy \\ \delta_{ext}(Idle, fault) = Down \end{cases} \\
\lambda &= \begin{cases} \lambda(\delta_{int}(Busy, \tau_1)) = job_{out} \\ \lambda(\delta_{int}(Down, \tau_2)) = ready \end{cases} \\
t_{adv} &= \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2\}
\end{aligned}
$$

Buffers are defined by a set of states and the maximum capacity of events in queue. An example of a buffer b with two states (Busy, Free) and occupancy $w_b \in \mathbb{Z}^+$ is shown in Fig. 2. In this paper, machine and buffer parameters such as states, transition times, and buffer capacity are based on historical or experimental data and operation analysis.

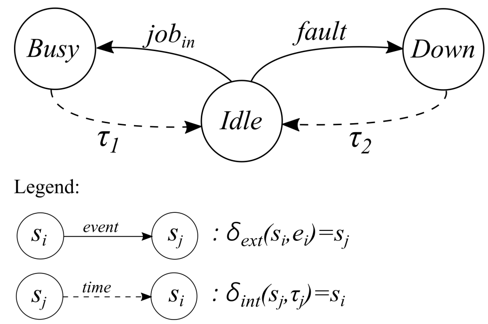

Fig. 1. Machine Discrete Event System Model

System level interactions are represented in a coupled model by specifying the interconnection of several atomic models. A DEVS coupled model is defined by a tuple G:

$$G = (\boldsymbol{U}, \boldsymbol{Y}, \boldsymbol{M}, EIC, EOC, IC, Select)$$

where $\boldsymbol{U}$ is a set of system input events, $\boldsymbol{Y}$ is a set of system output events, and $\boldsymbol{M}$ is a set of DEVS atomic models (i.e: buffers and machines). Coupling relations EIC, EOC, IC represent machine interconnections that are specified by the manufacturing process flow to map inputs and outputs. EIC are external input couplings, connecting external or system level inputs to component inputs. EOC are external output couplings, connecting component level outputs to external outputs. IC are internal couplings, interconnecting components output to other components inputs. $Select$ is a tie-breaking function specifying hierarchy. Several examples are presented in [28]. An example of a coupled model of two machines (m_1, m_2) and one buffer (b_1) is shown in Fig. 2. The set of atomic models is defined by $\boldsymbol{M} = \{m_1, m_2, b_1\}$. The set of system level input and output events are defined by $\boldsymbol{U} = \{u_1\}$ and $\boldsymbol{Y} = \{y_2\}$ respectively. These events are defined as inputs and outputs of m_1 and m_2, and their interactions are defined by EIC, EOC and IC.

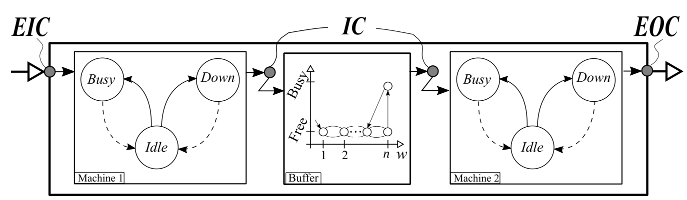

Fig. 2. Coupled Discrete Event System Model

Given a string of inputs events $E_i \in \boldsymbol{U}^*$ arriving at a rate Λ, the DEVS coupled model generates a string of output events $E_o \in \boldsymbol{Y}^*$ after some time t. The number of arrivals and departures N_i and N_o respectively, are defined by the length of E_i and E_o in the interval $(0, t]$. A common assumption is that a system operates under steady-state conditions, so that as $t \rightarrow \infty$, $\Lambda(t)$ converges to a constant value Λ [58].

The focus of this work is to study manufacturing systems operating under non-steady-state conditions with variable arrival rates $(\Lambda(t))$ of different events in the input string (E_i).

Moreover, performance metrics are monitored as a function of time and synchronization between the real and virtual environments supports concurrent performance analysis. Given a timespan T, buffer occupancy w of B number of buffers, system level performance is characterized by Throughput μ, Work-in-Process β, and Yield defined as the ratio between throughput and arrival rate (μ/Λ).

$$\text{Arrival Rate:} \quad \Lambda(t) = \frac{\Delta N_i}{\Delta t} = \frac{N_i(t) - N_i(t-T)}{t-(t-T)}$$
$$\text{Throughput:} \quad \mu(t) = \frac{\Delta N_o}{\Delta t} = \frac{N_o(t) - N_o(t-T)}{t-(t-T)}$$
$$\text{Work-in-Process:} \quad \beta(t) = \sum_{i=1}^{B} w_i(t)$$

Expected throughput ($\mu(t)$) and work-in-process ($\beta(t)$) can be used to identify system level features such as blockage and starvation, and calculation of Overall Throughput Effectiveness (OTE) [48]. Moreover, machine level variables such as transition times can be combined into performance metrics (e.g: availability, efficiency) to dynamically monitor OEE [2]. The effect of machine transition times over system level performance metrics will depend on the internal couplings as defined by the plant floor layout or process flow (e.g: parallel or series subsystems).

2) Continuous: Machine continuous dynamics (CD) are studied along with their discrete-state representation. The dynamic model captures parameters that can help evaluate machine performance given the context of a specific input event. State-space variables and outputs are studied based on a deterministic model. In the most basic form, machines can be studied as a system with input, outputs, and state variables. The dynamics of a machine can be described by a differential equation of the form $\dot{x} = f(x, u, t)$, where x is the continuous state variable vector, u is the continuous input vector, and t represents continuous time. For a machine with discrete states shown in Fig. 1, given that $0 < t < \tau_1$, continuous input signals $u(t)$ are related to input events in U, and $x(t)$ describes variables that operate in a state of S. In this work continuous state variables are studied in discrete-time at some time t and a short time later $t + \Delta t$. Time is discretized at a fundamental step size Δt so that state variables are calculated at $x(k\Delta t)$ where $k \in \mathbb{Z}^+$ represents the discrete-time unit. For an input signal $u(k)$ the machine dynamic model results in a vectors of n state variables $x(k)$ and n' outputs $y(k)$.

$$x(k) = [x_1(k) \cdots x_n(k)]^T \quad y(k) = [y_1(k) \cdots y_{n'}(k)]^T$$

An example of a dynamic model of the actuator force $\boldsymbol{Q}$ for an industrial robot arm represented as a kinematic chain is given by [59]:

$$\boldsymbol{Q} = \boldsymbol{M}(q)\ddot{q} + \boldsymbol{C}(q,\dot{q})\dot{q} + \boldsymbol{F}(\dot{q}) + \boldsymbol{G}(q) + \boldsymbol{J}(q)^T g \qquad (1)$$

The model requires identification of key parameters such as joint inertia matrix $\boldsymbol{M}(q)$, Coriolis and centripetal coupling matrix $\boldsymbol{C}(q,\dot{q})$, friction force $\boldsymbol{F}(\dot{q})$, gravity loads $\boldsymbol{G}(q)$, and state variables $\{q,\dot{q}\}$ denoting position and velocity in the joint space respectively. Different machine models have been studied in [33] or are available from the machine manufacturer.

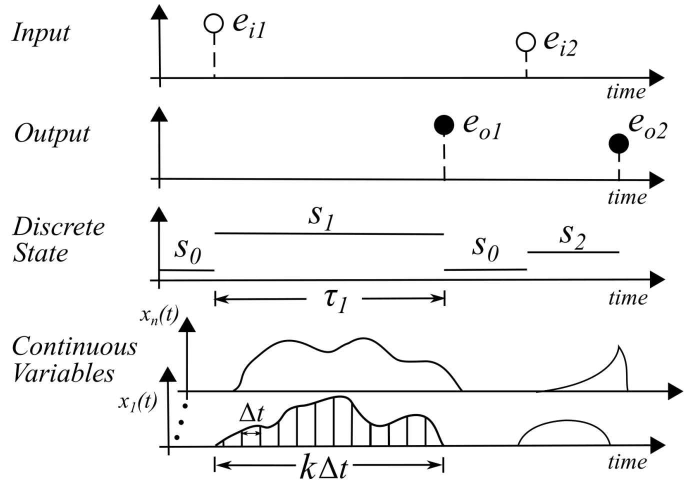

Fig. 3. Hybrid Discrete and Continuous Model

3) Hybrid: Machine and system level performance can be studied in parallel by merging discrete and continuous models. In the hybrid workspace, a CD model is defined for set of discrete states. An event triggers a state transition and dynamic action. Initiation of the continuous dynamic simulation requires signal conversion from event-based to time-based along with input specifications. For manufacturing equipment, the series of tasks or programs to perform in each state defines the dynamic model inputs. As shown in Fig. 3, an event e_{i1} in the discrete model triggers both a transition from s_0 to s_1 and dynamic model initiation (set $k = 0$) simultaneously. At stochastic time τ_1 the DES transitions back to state s_o while the CD model runs until some deterministic time $k\Delta t$.

A hybrid model of a single machine given a single input event $e_{i1} \in \boldsymbol{U}$ results in discrete and continuous outputs. The discrete output is a vector Ψ of cycle time to process the event. The continuous outputs are matrices Θ and Γ of time-series vectors of machine state variables and outputs respectively.

$$\Psi = [\tau_1] \quad \Theta = [x(1) \cdots x(k)] \quad \Gamma = [y(1) \cdots y(k)]$$

Discrete and continuous models differ in the way time is managed. DES is asynchronous and skips time intervals where the machine status does not change. CD are synchronous and are studied in discrete-time. Discrepancy between DES and CD running time are solved by synchronizing the hybrid model to run in real-time.

B. Real-time Synchronization

In this work the term real-time refers to "true" or "wall clock" time while concurrent refers to actions developing at

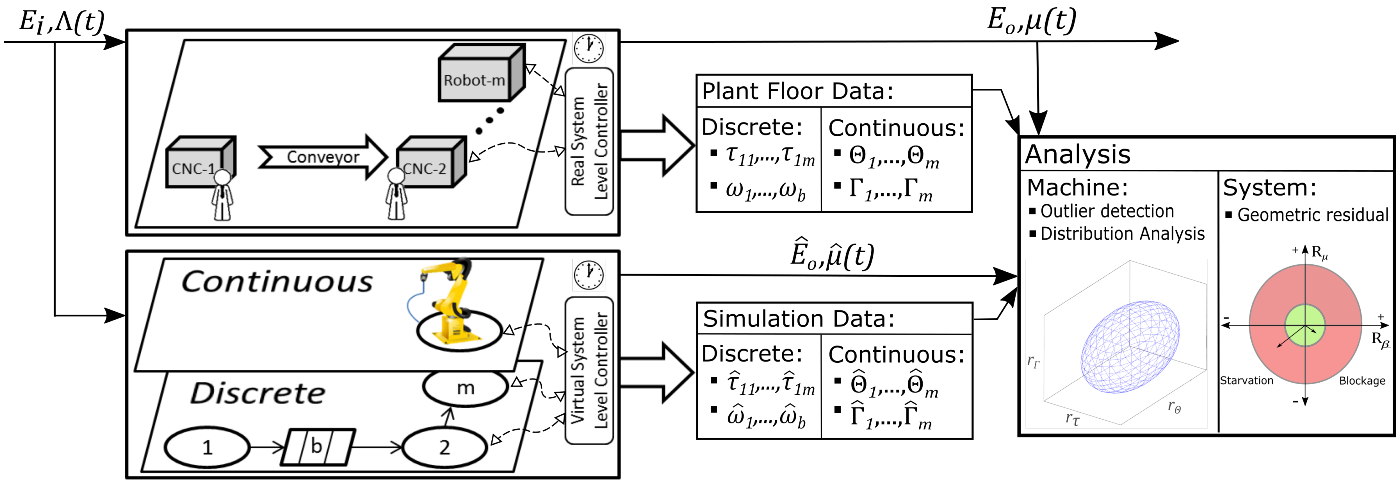

Fig. 4. Analysis Framework With Real and Virtual Environment

the same time in both real and virtual environments. This novel approach to monitor manufacturing system performance requires synchronization of the virtual environment to run in real-time and concurrent to plant floor operations. The latter is accomplished by monitoring the string of events from the physical system representing the production sequence or schedule and using them as inputs to the model. Having the simulation running in real-time under the same operational context as the plant floor enables direct comparison between the virtual and real environments for performance analysis.

Real-time execution of the virtual environment is achieved by creating a virtual controller that emulates the time progressing, and supervisory actions over events and signals of the real controller. During simulation, time advances at constant steps. The validity of simulation outputs will depend on both model accuracy and computation time length. It is important to define a time-step and solver that prevent an "overrun" defined as computation of variables and outputs exceeding real-world time of the system at a certain state.

A detailed description of solvers and performance comparison is discussed in [60]. Step size provides a metric to analyze simulation time. Smaller step sizes leads to longer simulation times with more accurate results. To assure the final outcome of the simulation is not compromised by the time-step size, first run the simulation using variable step size, then find the minimum step size requirement throughout the simulation running time. Fix the step size to the obtained minimum and re-run the simulation. Finally evaluate output accuracy and running time. Inappropriate selection of the solver, step size, or a non real-time capable model can cause the solver to skip solutions at a specific time-step and create gaps or discontinuity in the continuous dynamics of the machine. Discontinuities in state variables at each time step can be detected based on zero-crossing detection [61].

Synchronization between the real and virtual environment is accomplished by capturing a string of events from the plant floor PLC, and using these events as inputs to the simulation. Discrete and continuous signals used by the PLC for control actions can be captured and extracted by an IIoT adapter connected in the control network. Using these signals in the simulation requires plant floor data to be mapped to events. A simple example is a presence sensor that enables signals to be mapped to an input event job_{in} shown in Fig. 1. The IIoT adapter converts plant floor signals into data packets. In the virtual environment, the DEVS event generator model is configured to read and interpret the packets, and create events that initiate the simulation. Control actions for events such as change of state, task, or specific trajectory in the simulation can be programmed in the virtual PLC. Using plant floor events as inputs in the model assures concurrent execution of the simulation so that both real and virtual environments operate in the same context.

In this work we use Rockwell Automation *RSEmulate* to emulate a PLC operation and *SimKit* to describe the interaction between the emulated PLC and simulation [62]. Discrete and continuous signals are extracted from the real PLC using Rockwell Automation IIoT adapter, and are interpreted by the simulation using Matlab. As shown in Fig. 4, data streams from both the simulated and real system enable real-time analysis.

IV. Shop Floor Performance Analysis

Machine and system level performance assessment is based on residual analysis. Implementation requires plant floor data extraction of performance metrics and machine variables. In this section we discuss our data extraction strategy from the real system, and the performance analysis rules based on comparing plant floor and real-time simulation data.

A. Shop Floor Integration

In an automated manufacturing system, a PLC serves as supervisor or system level controller coordinating tasks based on machine states and Input/Output signals. Data from sensors and machines can be stored temporarily in "tags", an in-memory location. Tags can store binary or numeric values required for performance analysis.

- Discrete signals: Events and states are monitored based on binary signal values from a sensor. Common automation components such as presence sensors can be used to identify part arrival or departure events based on enable or disable signals.
- Continuous signals: State variables and outputs can be monitored based on digital signals. Data from condition sensors such as encoders, current transformers, temperature or pressure sensors is used to monitor machine continuous variables in discrete-time.

Machine level performance analysis is based on studying the actual transition times τ_j together with time-series matrices of state variables Θ and outputs Γ. Assuming that machine states can be monitored by the system level controller, τ_j is obtained by monitoring the time that a binary signal from the machine is enabled while in state s_j. State variables Θ and output variables Γ are monitored during the time interval $0 < k\Delta t_R \leq \tau_j$, where Δt_R is the fundamental time-step of the real system determined by the PLC scan rate.

System level performance analysis requires monitoring of real buffer occupancy w and throughput μ. At a specific time, the number of units stored in the buffer $w(t)$ is obtained either by direct measurement or calculation $w(t) = N_i(t) - N_o(t)$. Throughput is obtained by calculating the number of parts produced per unit of time.

Considering the effect of machine level timing over system performance [47] [48], variables at both machine and system level are temporarily stored in tags for later extraction. For example, transition times τ_1 and τ_2 are monitored to estimate machine availability which can affect the system throughput.

As noted earlier, we use an IIoT agent to extract data from the PLC. The advantage of extracting the PLC data is that multiple tags containing information from different machines can be combined into a single data packet. Packets are sent to a local repository for analysis as shown in Fig. 5. Machine and system level performance metrics are evaluated based on real tag values and simulation results.

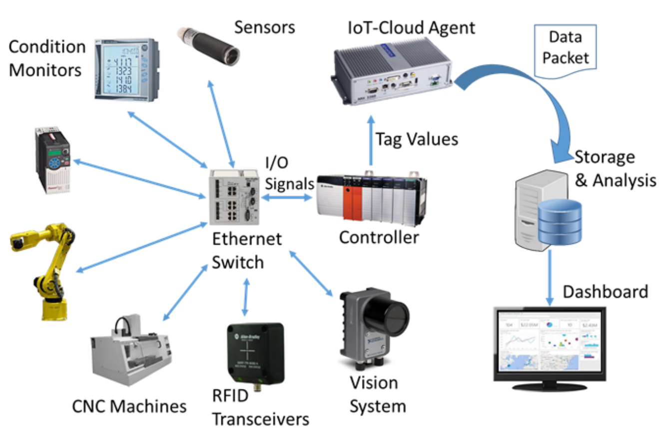

Fig. 5. Data Flow

B. Shop Floor Performance Analysis

Analyzing productivity and health in a manufacturing system with variable demand of different parts processed across multiple machines can be challenging. Here is where the synchronous interactions between the real and virtual environments gains importance as the simulation provides insight into the desired performance at both the system and machine level at any point in time.

The novelty of this work is the development of a framework that enables the comparison of plant floor data with real-time simulation data to analyze performance at both machine and system levels. For the z^{th} event on a string, we monitor discrete variables such as cycle time τ_1, continuous state-variables Θ and output-variables Γ, given a discrete time-step k. Multiple strategies can be used for performance analysis. In this work, we discuss a few of the possible multivariate analysis techniques that leverage real-time data from the real and virtual systems. At a machine level, residuals of discrete and continuous variables are studied for performance analysis. At a system level, a geometric framework to analyze residuals of production metrics is proposed.

1) Machine Level: Given a string of input events E_i, we monitor residuals of discrete and continuous variables for each event in the string.

- Discrete variables: The difference between transition times of a real machine τ_i and virtual machine $\hat{\tau}_i$ to process the z^{th} event in the string, is calculated as:

$$r_{\tau_i}(z) = \tau_i(z) - \hat{\tau}_i(z) \tag{2}$$

- Continuous variables: The difference between the time-series of the state variables $(\Theta, \hat{\Theta})$ and outputs $(\Gamma, \hat{\Gamma})$ generated by the real and virtual machines when processing the z^{th} event in the string are evaluated. Consider that the length of a time-series matrix obtained from the real and virtual systems k and $\hat{k}$ respectively are not necessarily the same $(k \neq \hat{k})$. The Dynamic Time Warping (DTW) [63] is used to align signal features. The residual between warped time-series is given by:

$$r_{\Theta}(z) = \sqrt{\sum_{i=1}^{n} (\Theta_{i,k}(z) - \hat{\Theta}_{i,\hat{k}}(z))^2} \tag{3}$$

$$r_{\Gamma}(z) = \sqrt{\sum_{i=1}^{n'} (\Gamma_{i,k}(z) - \hat{\Gamma}_{i,\hat{k}}(z))^2} \tag{4}$$

In this work a multivariate approach to monitor discrete and continuous machine parameters is proposed. The analysis is based on monitoring vector $\mathbf{r}$ for each event in E_i and a sub-string of E_i defined by a sliding window $\mathbf{v}$. Outlier detection and distribution analysis are used to evaluate performance.

- Outlier detection: A set of historical or experimental values of machine variables under normal operation define the range of allowable variation. Limits are calculated based on 95% confidence intervals of the covariance

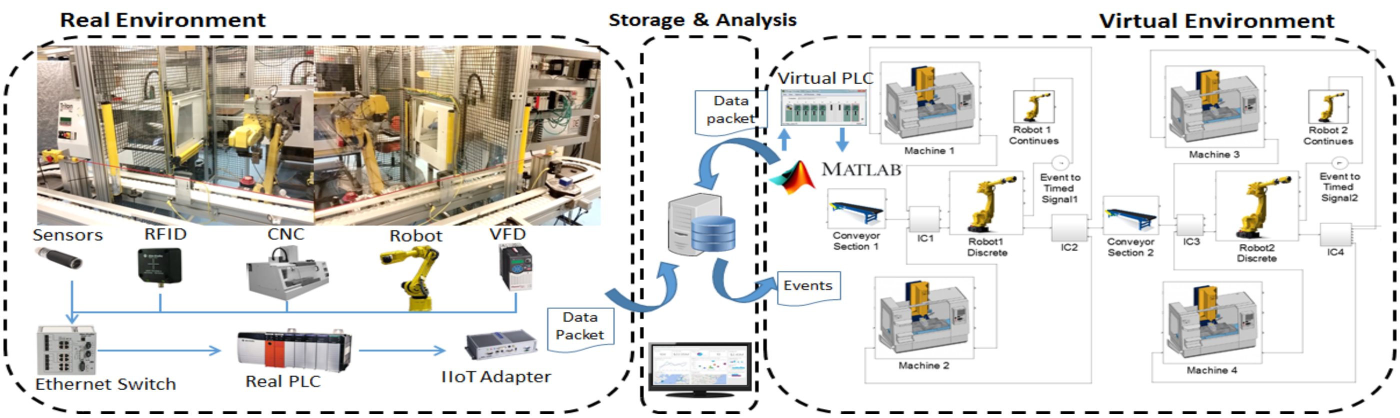

Fig. 6. Real and Virtual Environment

matrix. Outliers are detected using Olive-Hawkins method [64] and Mahalanabis distance $D(z)$ from a cluster in a multivariate space:

$$D(z) = \sqrt{(\mathbf{r}(z) - \mathbf{r}_0)^T \mathbf{\Sigma}^{-1} (\mathbf{r}(z) - \mathbf{r}_0)} \qquad (5)$$

where $\mathbf{r}(z) = [r_{\tau_i}(z), r_\Theta(z), r_\Gamma(z)]$ is a vector combining residuals of: transition time, state variables, and outputs for the z^{th} event. $\mathbf{\Sigma}$ is the robust covariance matrix, r_0 is a vector that identifies the cluster centroid. An example of the 95% confidence interval ellipsoid for a three dimensional residual cluster is shown in Fig. 4.

- Distribution Analysis: In this work we use Kernels, a non-parametric probability density function (pdf). Kernel Density Estimation (KDE) is an iterative process that does not require prior assumption of data distribution and is defined by:

$$\hat{f}(r) = \frac{1}{jh} \sum_{i=1}^{j} K\left(\frac{r - r(i)}{h}\right), \qquad (6)$$

where $(r(1), r(2), ..., r(j)) \in \mathbf{v}$ are experimental or historical sample data of sliding window $\mathbf{v}$, h is a smoothing parameter, and K is the kernel. Kolmogorov-Smirnov (KS) test is used to compare two sample KDE of residuals [65]. The distance between distributions of consecutive sliding windows is used for hypothesis testing. The null hypothesis that both samples come from a common distributions is evaluated for 95% confidence intervals. A rejection of the null hypothesis identifies an abnormal distributions of residuals.

2) System Level: Productivity metrics such as throughput (μ) and Work-in-Process (β) are key performance metrics [45]. We use synchronous simulation as a reference to monitor these metrics for a manufacturing system operating under non-steady state conditions. To the best of our knowledge, no closed form equation exists to model manufacturing systems, or to correlate between performance metrics.

A geometric framework for detecting "faults" and estimating possible root causes is defined. The direction of the residual vector provides insight on possible fault types, while length is proportional to the fault magnitude. Directional residual analysis has been studied in Fault Detection and Isolation (FDI) [5]. However, in this work a residual vector is not decomposed into known fault vectors for isolation. Nonetheless, expert knowledge can use residual analysis of a multivariate space to assess manufacturing system performance. In this work we define the system-level residual as:

$$R(t) = \begin{bmatrix} R_\mu(t) \\ R_\beta(t) \end{bmatrix} = \begin{bmatrix} \mu(t) \\ \beta(t) \end{bmatrix} - \begin{bmatrix} \hat{\mu}(t) \\ \hat{\beta}(t) \end{bmatrix}$$

Considering that the simulation outputs are generated in real-time and concurrent with the plant floor operation, residuals describe deviation from the expected performance. For example, an increase on throughput residual ($R_\mu(t)$) and Work-in-process ($R_\beta(t)$) can describe changes in the bottleneck location.

C. Shop Floor Management

Having a reference of expected performance or OEE metrics in real-time and under the same operational context of the plant can support shop floor management and decision making. As shown in Fig.4, direction and magnitude of the system level residual vector can help identify issues such a bottleneck shift or process delays causing starvation and negatively impacting throughput. To support shop floor management, changes in system throughput detected by $R_\mu(t)$ can be traced back to shifts in machine transition times r_{τ_i} to identify delays on cycle time or time to repair. Management can then assess the need for additional resources on a specific task, changes in the process flow or creation of new workstations [66] [47].

The root cause of machine under-performance can be identified by analyzing the residuals of continuous input ($r_\Theta(z)$) and output ($r_\Gamma(z)$) variables (e.g: velocity, torque). Moreover, the residuals of continuous input and output variables can be used to assess machine health. The need for a maintenance action could trigger a change on the work schedule of specific machines or part re-routing [67].

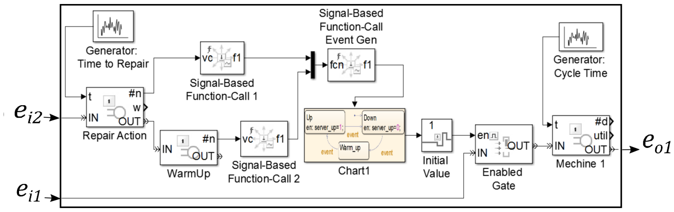

Fig. 7. Single Machine Discrete-Event Model

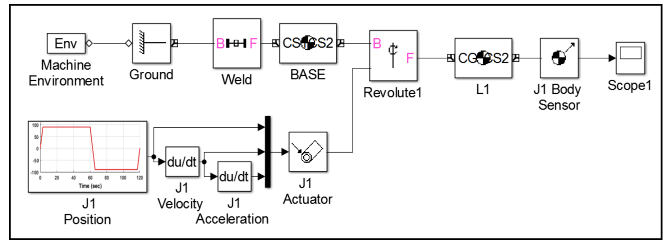

Fig. 8. Robot Joint Model

V. Implementation and Evaluation

A hybrid simulation of an automated manufacturing testbed installed at the University of Michigan was built to run in real-time. The physical tesbed is equipped with two Fanuc robots, four Denford CNC milling machines, and a conveyor loop [68]. The system is controlled with a Rockwell ControlLogix PLC connected over Ethernet/IP. The PLC receives input signals from the robots, a Variable Frequency Drive (VFD) controlling conveyor speed, and sensors installed in different locations across the conveyor. Parts are transported by the conveyor on pallets. As parts go through the system, sensors are triggered to initiate logic-driven operations embedded on the PLC such as a robot pick-and-place action and CNC machining.

We modeled the testbed in Matlab/Simulink environment using the framework described in section III-A. Data from the testbed and simulation were collected in real-time. Machine and system level performance were evaluated following the analysis techniques described in section IV.

A. Modeling

Machine and system level interactions were modeled in a Matlab/Simulink environment.

1) Machine Level: DES models of CNC machines, robots, and conveyors were created using SimEvents and StateFlow after identifying possible states, inputs, outputs, and transition times. A DES model for a single CNC machine is shown in Fig. 7. As represented in the DEVS formalism example shown in section III-A, transition times τ_1 and τ_2 are scalars representing cycle time and repair time respectively are generated given random variables $\boldsymbol{\tau_1}$ and $\boldsymbol{\tau_2}$.

Machine dynamics were simulated using SimMechanics. A parametric model of a 6 Degree-of-Freedom (DoF) robotic arm was created based on geometry and material information from a 3D model. Trajectory requirements for a specific task such as pick-and-place in the robot workspace was transformed to position commands in the joint state space based on inverse kinematics using Matlab Robotics Toolbox [69]. The model inputs were commanded joint position, velocity, and acceleration, and the model output was joint torque. An example of a single joint model is shown in Fig. 8.

2) System Level: Machine interactions were defined based on coupling relations between Input and Output events. To capture productivity metrics, parts were abstracted as events moving between machines given a specific process flow. For this case study, a single part type modeled as event e_{i1} was processed by CNC machines, robots, and conveyors. Figure 6 demonstrates the interaction between machines.

3) Real-Time Synchronization: Real-time synchronization is done by the selection of a proper solver and time-step size for the simulation, using Rockwell RSEmulate to define a virtual PLC and control logic, and SimKit to define the interactions between the virtual controller and simulation. We first ran the simulation using variable step size and solver ode23t (Dormand-Prince) to assess the required step size and computation time. Then we fixed the time-step size and re-ran the simulation to evaluate output errors and no zero-crossing events. Lastly, we assessed simulation results using a fixed time-step size and a less computationally expensive solver, ode4 (Runge-Kutta). Then the simulation was integrated with a virtual PLC created using RSEmulate by correlating PLC logic to simulation parameters using SimKit. The integration of Simulink, SimKit, and RSEmulate support real-time execution of the simulation. Synchronization can be accomplished by extracting events from the real PLC via Rockwell IIoT adapter and using them as inputs in the simulation to assure concurrent operation between the virtual and real environments.

B. Plant Floor Performance Analysis

For analysis, we compared data from the real and virtual testbed. Data from the real testbed was collected from the PLC using and IoT adapter. Data from the virtual testbed was generated in real-time. Both real and virtual datasets were analyzed to assess performance.

1) Plant Floor Integration: Plant floor data was collected using an IIoT adapter. Variables for performance analysis were temporarily stored on tags inside the PLC. Data was collected in discrete time based on the PLC scan rates that defined the fundamental step size $\Delta t_R = 100ms$. Discrete variables such as transition time (τ_1) were monitored using a "Timer-on-Delay" (TOD) function. To control transitions in the CNC machine, additional logic was added on the PLC to trigger and monitor CNC tasks. The machining task and TOD was initiated by a binary signal from the PLC. Once the machining task was completed, the CNC sent a binary signal back to the PLC that stopped the TOD. Cycle time τ_1 was computed by the PLC logic as the accumulated time in the TOD. For our implementation, an IIoT agent in the control network connected to the PLC collected and sent variables in data packets. Continuous variables in Θ were extracted from machine controller. Robot position data in the machine controller was defined as a monitoring variable. Data was extracted by writing a computer program based on Fanuc's PC Developer Kit (PCDK) that enabled Ethernet communication from a desktop computer to the robot controller to monitor pre-defined variables. During a part moving task, data from the robot controller extracted at a fundamental step size $\Delta t_R = 100ms$ were sent to a repository. Figure 6 shows the dataflow of real and virtual environment.

2) Performance Analysis: Our case study was based on univariate and multivariate analyses of different performance metrics. For CNC machines, we monitored cycle time, the time that the machine was in state "Busy" while processing a part. For Robots, we monitored cycle time and state variables, the time in state "Busy" while performing a pick-and-place operation, and position in the world coordinate frame

a) Milling Machine: Univariate analysis for productivity assessment was done based on cycle times τ_1 and $\hat{\tau}_1$. Transition time residual r_{τ_1} was calculated using (2). $\hat{\tau}_1$ was obtained from a DES model after identification of states and transition times pdf from 50 cycles under normal operation. A string of 50 input events (e_{i1}) was sent to both real and virtual machines. For testing purposes, the feedrate of some cycles was randomly changed to simulate an anomaly. Testing results are summarized in Table I.

TABLE I. TESTING SUMMARY

Cycles	%Feedrate	Avg. τ_1 (s)	Std. τ_1 (s)
1-25	50	198.4	1.3
26-30	40-60	197.7	7.7
31-45	50	198.8	1.1
46-50	60-70	190.8	3.0

Performance analysis was based on distance using Eq. (5). Figure 9 shows cycle time residual distance from the cluster centroid for each event. Events outside the 95% confidence interval were labeled as outliers. Changes in feedrate affected the cycle time residual and can be detected using the proposed framework.

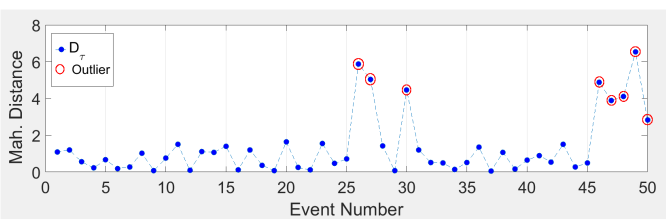

Fig. 9. Mahalanabis Distance of Cycle Time Residual

To study distributions, we used a sliding window of 5 samples. For each window we estimated a kernel using Eq. (6). A two-sample KS-test was performed between subsequent windows to evaluate statistically significant differences between the two distributions. The distributions and p-values for each time window are shown in Fig. 10 and Table II respectively.

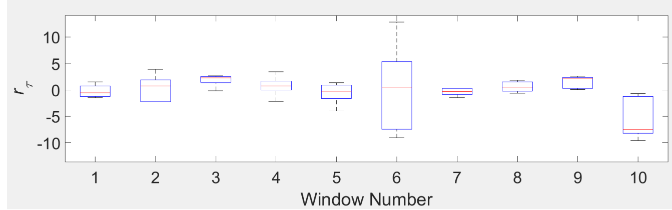

Fig. 10. Mahalanabis Distance of Cycle Time Residual

TABLE II. TWO-SAMPLE KS-TEST RESULT

Window #	1	2	3	4	5	6	7	8	9	10
p-value	0.679	0.963	0.809	0.660	0.679	0.044	0.122	0.152	0.784	0.002
Condition	OK	OK	OK	OK	OK	NO OK	OK	OK	OK	NO OK

Changes in either the mean or distribution of the cycle time residual could have a negative effect on system throughput. Reduction of mean cycle time as detected in window number 10 from Fig. 9 could increase work-in-process causing a blockage in the system. A change in cycle time distribution as detected in window number 6 from Fig. 9 can affect idle time of subsequent operations. Under-performance of the system that is captured by machine and system-level residuals can lead to improvements in plant floor control actions determined at the managerial level. For example, the identification of an important residual could lead to a change in the conveyor speed, production routing or schedule of a maintenance action.

b) Robot: A multivariate analysis for productivity and health assessment was done based on cycle time τ_1 and state-variables during a pick-and-place task from the conveyor to the CNC machine. The task was programmed in the real robot (Fanuc M6i-B) using the teach pendant and initiated by a signal from the PLC (e.g., part available for pick-up at CNC). End-effector position of the real robot was extracted from the robot controller at a scan rate of $\Delta t_R = 100ms$. End-effector position of the virtual robot was computed at a fundamental time-step $\Delta t_V = 10ms$. A simulation of the virtual robot operation was initiated by an input event and trajectory. Figure 11 shows the trajectory in each coordinate axis for both the real and virtual robot.

Position data was collected over 75 cycles. To simulate an anomaly, the trajectory of some cycles in the real robot were modified to add a jerky motion and trajectory changes. We compared the temporal position vector of each axis (XYZ) by overlapping output signals from the physical robot with the simulated trajectory. Due to differences in time-step size between the real (Δt_R) and virtual (Δt_V) robots, and variable cycle times of the real robot, the position vectors had unequal length. As shown in Fig. 11, the simulated and real trajectories have some differences in the number of samples due to differences in time-step size. However, we expect to capture anomalies on state-variables as long as the simulated values are obtained at a smaller time-step than the values sampled from the real robot $\Delta t_V < \Delta t_R$. Differences in the location of features in the trajectory between the real and virtual robots are caused by minor discrepancies in the robot geometry. DTW was used to normalize and align features in the temporal position vectors. The residual between time-

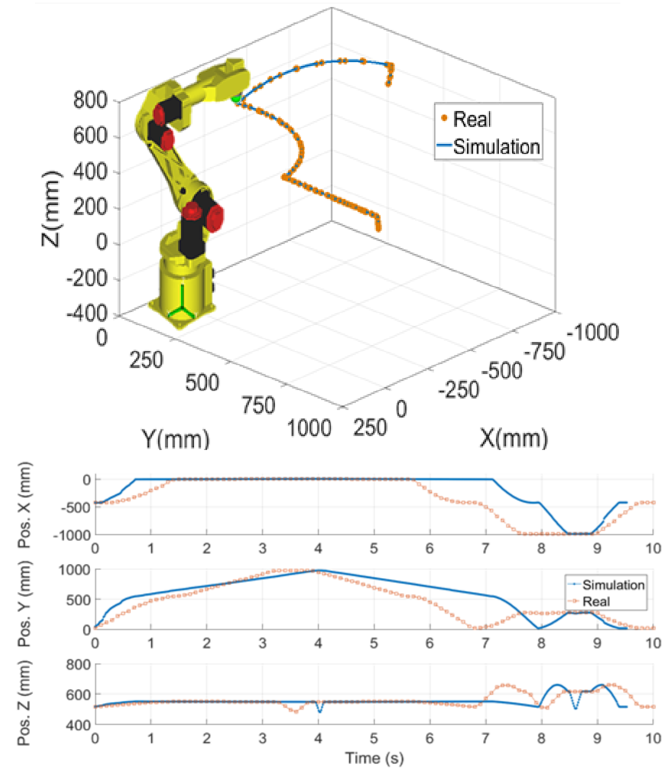

Fig. 11. Fanuc M-6iB: Physical and Virtual model

series $(\Theta, \hat{\Theta})$ was computed using Eq.(3). Statistical analysis aimed to monitoring changes in the expected residual rather than absolute changes. This approach reduces noise from the expected variation between the simulation and real systems. Selection of simulation step-size and robot controller scan rate can limit the type of anomalies that can be detected.

Abnormal cycles with changing trajectories were identified using a multivariate analysis approach with cycle time residual r_τ and state-variables residual r_Θ. The proposed framework was able to detect changes in the trajectory, even when those changes had little effect on cycle time. Fig. 12 shows the residual analysis and 95% confidence interval; outliers were detected based on distance from the cluster centroid for all abnormal cycles. Moreover, outliers in position residual can be used as indicator of machine health, to identify the need of a maintenance action. Changes in position accuracy for robot operation might be of particular interest in welding or machining operations.

VI. Conclusions

This paper presents a mathematical framework for real-time modeling and synchronous simulation of a manufacturing plant at both the machine and system-levels. This hybrid model merges discrete and continuous variables at the machine level and considers system level interactions. This novel approach leverages Industrial Internet of Things (IIoT) to monitor events on the plant floor and synchronize the real and virtual environments. Enabling the virtual environment to run in real-time and

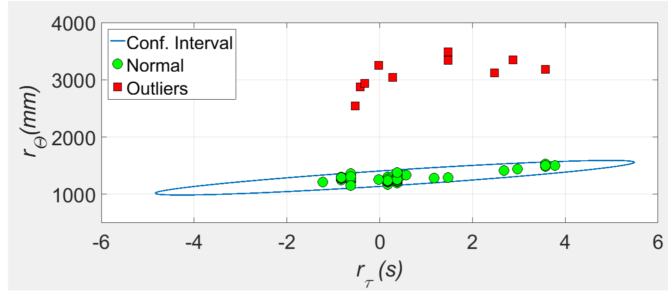

Fig. 12. Angular displacement over time: Real vs Simulation

in the same context as the plant can help direct comparisons of expected performance metrics at any given point in time. New advancements in hybrid modeling, simulation, and real-time synchronization along with the necessary techniques to analyze the performance of the system were provided. An evaluation of the proposed approach demonstrated a real-world implementation of the framework on a physical testbed equipped with CNC machines, conveyor, and robots using IIoT for data extraction. A DES model of equipment was built using SimEvents, while the CD of a robot were modeled using SimMechanics. Plant floor information was extracted using a Rockwell Automation IIoT adapter. Performance regarding machine cycle time and continuous variables were compared in the real and virtual environments to analyze residuals.

This work presents the first demonstration of a hybrid model simulated in real-time and concurrent to the plant floor. Experimental validation of this framework demonstrated how to evaluate performance and detect anomalies in different machines. This work has the potential to improve analysis of OEE by using synchronous simulation to manage expectations. Moreover, this framework provides insight into performance at a machine and system level that could be used to identify the need for actions such as maintenance, reconfiguration, or rescheduling. Future work will consider additional discrete variables such as quality and total energy consumption and continuous variables such as current signature analysis. Furthermore, the effect of simulation sampling rate on the detection accuracy of different classes of anomalies is an open area of research for future investigation.

Acknowledgment

This research was supported in part by the NSF under grant 1462910. The authors would like to thank Rockwell Automation for supporting this research. We would like to particularly acknowledge Robert Ulmer for his assistance.

References

[1] J. Kletti, *Manufacturing execution systems-MES*. Springer, 2007.

[2] M. M. Ahmad and N. Dhafr, "Establishing and improving manufacturing performance measures," *Robotics and Computer-Integrated Manufacturing*, vol. 18, no. 3-4, pp. 171–176, 2002.

[3] J. Gong, V. V. Prabhu, and W. Liu, "Simulation-based performance comparison between assembly lines and assembly cells with real-time distributed arrival time control system," *International Journal of Production Research*, vol. 49, no. 5, pp. 1241–1253, 2011.

[4] I. Mahdavi, B. Shirazi, and M. Solimanpur, "Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems," *Simulation Modelling Practice and Theory*, vol. 18, no. 6, pp. 768–786, 2010.

[5] R. Isermann, *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media, 2011.

[6] B. Zhao and Y. Li, "Local joint information based active fault tolerant control for reconfigurable manipulator," *Nonlinear dynamics*, vol. 77, no. 3, pp. 859–876, 2014.

[7] A. Negahban and J. S. Smith, "Simulation for manufacturing system design and operation: Literature review and analysis," *Journal of Manufacturing Systems*, vol. 33, no. 2, pp. 241–261, 2014.

[8] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.

[9] Z. Bi, L. Da Xu, and C. Wang, "Internet of things for enterprise systems of modern manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1537–1546, 2014.

[10] N. Bengtsson, J. Michaloski, F. Proctor, G. Shao, and S. Venkatesh, "Towards data-driven sustainable machining: Combining mtconnect production data and discrete event simulation," in *ASME 2010 International Manufacturing Science and Engineering Conference*. American Society of Mechanical Engineers, 2010, pp. 379–387.

[11] Y. Zhang, G. Zhang, J. Wang, S. Sun, S. Si, and T. Yang, "Real-time information capturing and integration framework of the internet of manufacturing things," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 8, pp. 811–822, 2015.

[12] M. Saez, F. Maturana, K. Barton, and D. Tilbury, "Real-Time Hybrid Simulation of Manufacturing Systems for Performance Analysis and Control," 2015.

[13] K.-C. Jeong and Y.-D. Kim, "A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules," *International Journal of Production Research*, vol. 36, no. 9, pp. 2609–2626, 1998.

[14] Y.-D. Kim, S.-O. Shim, B. Choi, and H. Hwang, "Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 2, pp. 290–298, 2003.

[15] L. Damiani, P. Giribone, R. Revetria, and A. Testa, "An Innovative Model for Supporting Energy-based Cost Reduction in Steel Manufacturing Industry Using Online Real-time Simulation," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, 2014.

[16] M. Jahangirian, T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young, "Simulation in manufacturing and business: A review," *European Journal of Operational Research*, vol. 203, no. 1, pp. 1–13, 2010.

[17] L. Rabelo, M. Helal, A. Jones, and H.-S. Min, "Enterprise simulation: a hybrid system approach," *International Journal of Computer Integrated Manufacturing*, vol. 18, no. 6, pp. 498–508, 2005.

[18] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[19] D. Broman, E. A. Lee, S. Tripakis, and M. Törngren, "Viewpoints, formalisms, languages, and tools for cyber-physical systems," in *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling*. ACM, 2012, pp. 49–54.

[20] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson *et al.*, "Extending an industrial robot controller: implementation and applications of a fast open sensor interface," *IEEE Robotics & Automation Magazine*, vol. 12, no. 3, pp. 85–94, 2005.

[21] F. DiCesare and A. A. Desrochers, "Modeling, control, and performance analysis of automated manufacturing systems using petri nets," *Control and Dynamic Systems*, vol. 47, pp. 121–172, 2012.

[22] H. Hu and M. Zhou, "A petri net-based discrete-event control of automated manufacturing systems with assembly operations," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 513–524, 2015.

[23] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by petri nets: A set classification approach," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 549–563, 2014.

[24] E. Park, D. M. Tilbury, and P. P. Khargonekar, "A modeling and analysis methodology for modular logic controllers of machining systems using petri net formalism," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, no. 2, pp. 168–188, 2001.

[25] J. Moody and P. J. Antsaklis, *Supervisory control of discrete event systems using Petri nets*. Springer Science & Business Media, 2012, vol. 8.

[26] R. Y. Al-Jaar and A. A. Desrochers, "Performance evaluation of automated manufacturing systems using generalized stochastic petri nets," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 6, pp. 621–639, 1990.

[27] C. Lindemann, G. Ciardo, R. Geman, and G. Hommel, "Performability modeling of an automated manufacturing system with deterministic and stochastic Petri nets," *IEEE International Conference on*, 1993.

[28] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[29] H.-T. Park, J.-G. Kwak, G.-N. Wang, and S. C. Park, "Plant model generation for plc simulation," *International Journal of Production Research*, vol. 48, no. 5, pp. 1517–1529, 2010.

[30] D. Huang, H. S. Sarjoughian, W. Wang, G. Godding, D. E. Rivera, K. G. Kempf, and H. Mittelmann, "Simulation of semiconductor manufacturing supply-chain systems with devs, mpc, and kib," *IEEE Transactions on Semiconductor Manufacturing*, vol. 22, no. 1, pp. 164–174, 2009.

[31] Hyo Jik Lee, Sung Hyun Kim, Hee Sung Park, Chang Hwan Choi, Ji Sup Yoon, and B. S. Park, "Operation Analysis of Headed Process Facility Using Discrete Event Simulation," in *International Conference on Industrial Informatic*, 2008.

[32] J. G. De Jalon and E. Bayo, *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Springer Science & Business Media, 2012.

[33] C. Yang, Z. Ye, O. O. Peter, and J. Han, "Modeling and simulation of spatial 6-{DOF} parallel robots using Simulink and Simmechanics," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 4. IEEE, 2010, pp. 444–448.

[34] Y. Shaoqiang, L. Zhong, and L. Xingshan, "Modeling and simulation of robot based on Matlab/SimMechanics," in *Control Conference, 2008. CCC 2008. 27th Chinese*. IEEE, 2008, pp. 161–165.

[35] A. Jönsson, J. Wall, and G. Broman, "A virtual machine concept for real-time simulation of machine tool dynamics," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7, pp. 795–801, 2005.

[36] Y. H. Jeong, B.-K. Min, D.-W. Cho, and S. J. Lee, "Motor current prediction of a machine tool feed drive using a component-based simulation model," *International Journal of Precision Engineering and Manufacturing*, vol. 11, no. 4, pp. 597–606, 2010.

[37] F. P. Maturana, M. Kalolia, and P. Paredes, "A Virtual Design Engineering Environment for Control System Design and Validation," in *Industrial Applications of Holonic and Multi-Agent Systems*. Springer, 2013, pp. 118–127.

[38] T. K. Changho Sung, "Framework for simulation and hybrid systems interpolation of discrete and continuous simulation using {HLA/RTI}," *25th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*, 2011.

[39] A. Vodenčarević, H. K. Büning, O. Niggemann, and A. Maier, "Using behavior models for anomaly detection in hybrid systems," in *Information, Communication and Automation Technologies (ICAT), 2011 XXIII International Symposium on*. IEEE, 2011, pp. 1–8.

[40] Y. Leung and R. Suri, "Performance evaluation of discrete manufacturing systems," *Control Systems Magazine, IEEE*, vol. 10, no. 4, pp. 77–86, 1990.

[41] S. Iarovyi, W. M. Mohammed, A. Lobov, B. R. Ferrer, and J. L. M. Lastra, "Cyber–physical systems for open-knowledge-driven manufacturing execution systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1142–1154, 2016.

[42] M. T. Koch, H. Baars, H. Lasi, and H.-G. Kemper, "Manufacturing execution systems and business intelligence for production environments." in *AMCIS*, 2010, p. 436.

[43] Y. Li, Q. Chang, G. Xiao, and J. Arinez, "Data-Driven Analysis of Downtime Impacts in Parallel Production Systems," vol. 12, no. 4, pp. 1541–1547, 2015.

[44] S. X. Ding, S. Yin, K. Peng, H. Hao, and B. Shen, "A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2239–2247, 2013.

[45] J. Li and S. M. Meerkov, *Production systems engineering*. Springer Science & Business Media, 2008.

[46] J. Li, S. M. Meerkov, and L. Zhang, "Production systems engineering: main results and recommendations for management," *International Journal of Production Research*, vol. 51, no. 23-24, pp. 7209–7234, 2013.

[47] S. H. Huang, J. P. Dismukes, J. Shi, Q. Su, M. A. Razzak, R. Bodhale, and D. E. Robinson, "Manufacturing productivity improvement using effectiveness metrics and simulation analysis," *International Journal of Production Research*, vol. 41, no. 3, pp. 513–527, 2003.

[48] A. Mathur, G. Dangayach, M. Mittal, and M. K. Sharma, "Performance measurement in automated manufacturing," *Measuring business excellence*, vol. 15, no. 1, pp. 77–91, 2011.

[49] W. S. Harrison, D. M. Tilbury, and C. Yuan, "From hardware-in-the-loop to hybrid process simulation: an ontology for the implementation phase of a manufacturing system," *Automation Science and Engineering, IEEE Transactions on*, vol. 9, no. 1, pp. 96–109, 2012.

[50] P. Vrba and V. Marik, "Simulation in agent-based manufacturing control systems," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1718–1723.

[51] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld, and P. Warndorf, "Improving machine tool interoperability using standardized interface protocols: Mt connect," *Laboratory for Manufacturing and Sustainability*, 2008.

[52] N. Bengtsson, "Msec2010-34178 Mtconnect Production Data and Discrete Event Simulation," pp. 1–9, 2010.

[53] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.05.010

[54] Y. Zhang, W. Wang, N. Wu, and C. Qian, "Iot-enabled real-time production performance analysis and exception diagnosis model," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1318–1332, 2016.

[55] Y. Zhang, C. Qian, J. Lv, and Y. Liu, "Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 737–747, 2017.

[56] F. Tao, Y. Zuo, S. Member, and L. Zhang, "IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud," vol. 10, no. 2, pp. 1547–1557, 2014.

[57] G. P. Hong and T. G. Kim, "A framework for verifying discrete event models within a devs-based system development methodology," *Transactions*, vol. 13, no. 0, p. 1, 1996.

[58] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[59] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.

[60] L. F. Shampine and M. W. Reichelt, "The matlab ode suite," *SIAM journal on scientific computing*, vol. 18, no. 1, pp. 1–22, 1997.

[61] J. H. Taylor, "Rigorous handling of state events in matlab," in *Control Applications, 1995., Proceedings of the 4th IEEE Conference on*. IEEE, 1995, pp. 156–161.

[62] F. Maturana, L. Paredes, and A. Rodriguez, "Integration of simulation of a machine for industrial automation," Jun. 26 2014, uS Patent App. 13/834,850.

[63] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[64] D. J. Olive, "A resistant estimator of multivariate location and dispersion," *Computational statistics & data analysis*, vol. 46, no. 1, pp. 93–102, 2004.

[65] W. J. Conover and W. Conover, "Practical nonparametric statistics," 1980.

[66] J. Arinez, S. Biller, S. M. Meerkov, and L. Zhang, "Quality/quantity improvement in an automotive paint shop: A case study," *IEEE Transactions on automation science and engineering*, vol. 7, no. 4, pp. 755–761, 2010.

[67] A. Bauer, J. Browne, R. Bowden, and J. Duggan, *Shop floor control systems: from design to implementation*. Springer Science & Business Media, 2012.

[68] I. Kovalenko, M. S. Saez, K. Barton, and D. Tilbury, "SMART: A system-level manufacturing and automation research testbed," *Smart and Sustainable Manufacturing Systems*, 2017.

[69] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.

TABLE III. NOMENCLATURE

Discrete	
Machine Level	
H	Atomic model
U	Set of inputs
Y	Set of outputs
S	Set of states
δ_{int}	Internal transition function
δ_{ext}	External transition function
Δ	Set of transitions functions
λ	Output function
t_{adv}	Set of transition times
e_i	Input event
e_o	Output event
s_j	State
τ_j	Transition time
w	Buffer occupancy
System Level	
G	Coupled model
$\boldsymbol{U}$,$\boldsymbol{U}^*$	Set of inputs, Kleene closure of set of inputs
$\boldsymbol{Y}$,$\boldsymbol{Y}^*$	Set of outputs, Kleene closure of set of outputs
$\boldsymbol{M}$	Set of atomic models
EIC	External input coupling
EOC	External output coupling
IC	Internal coupling
$Select$	Tie-breaker function
E_i	String of input events
E_o	String of output events
N_i	Number of arrivals
N_o	Number of departures
Λ	Arrival rate
μ	Throughput
β	Work-in-process
Ψ	Vector of cycle times
Continuous	
$x(k)$	State-variable in discrete time
$y(k)$	Output-variable in discrete time
$\Theta(z)$	Time-series matrix of state-variables to process z^{th} event in E_i
$\Gamma(z)$	Time-series matrix of outputs to process z^{th} event in E_i

TABLE IV. NOMENCLATURE CONT.

Analysis	
$r_\tau(z)$	State-variable in discrete time
$r_\Theta(z)$	Output-variable in discrete time
$r_\Gamma(z)$	Time-series matrix of state-variables to process z^{th} event in E_i
$D(z)$	Time-series matrix of outputs to process z^{th} event in E_i
$\boldsymbol{r}(z)$	Residual vector to process z^{th} event in E_i
$\boldsymbol{r}_o$	Residual cluster centroid
$\boldsymbol{\Sigma}$	Robust covariance matrix
$f(r)$	Probability density function
$R(t)$	System level residual vector
$R_\mu(t)$	Throughput residual
$R_\beta(t)$	Work-in-process residual

MiguelSaez Miguel Saez received his Mechanical Engineering degree from La Universidad del Zulia, Venezuela in 2008, the M.Eng in Global Automotive and Manufacturing from University of Michigan, USA in 2015 and is now working towards the Ph.D in Mechanical Engineering. He worked as a Product Engineer at Dana Holding Corporation, Venezuela from 2007 to 2010, leading multidisciplinary design and manufacturing projects, coordinating cost saving projects using CAD/CAE tools, and testing NVH for vehicles powertrain. He worked as a Senior Manufacturing Engineer at General Motors, Venezuela from 2010 to 2013, managing a capital investment project for new vehicle programs and design, testing and installing semi-automated manufacturing systems.

Francisco P. Maturana is a senior research engineer at the Rockwell Automation Advanced Technology Laboratory. His research interests include distributed computing and communication protocols for multi-agent systems, FIPA, and self-organizing manufacturing systems. He received his PhD in intelligent manufacturing systems from the University of Calgary. Hes a member of the IEEE and the Computer Simulation International Society.

Kira Barton is an Assistant Professor in the Department of Mechanical Engineering at the University of Michigan. She received her B.Sc. in Mechanical Engineering from the University of Colorado at Boulder in 2001. She continued her education in mechanical engineering at the University of Illinois at Urbana-Champaign and completed her M.Sc. and Ph.D. degrees in 2006 and 2010, respectively. She held a postdoctoral research position at the University of Illinois from Fall 2010 until Fall 2011, at which point she joined the Mechanical Engineering Department at the University of Michigan at Ann Arbor. Kira conducts research in modeling, sensing, and control for applications in advanced manufacturing and robotics, with a specialization in Iterative Learning Control and micro-additive manufacturing. Kira is the recipient of an NSF CAREER Award in 2014, 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 received the B.S. degree in Electrical Engineering, summa cum laude, from the University of Minnesota in 1989, and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 1992 and 1994, respectively. In 1995, she joined the faculty of the University of Michigan, Ann Arbor, where she is currently Professor of Mechanical Engineering with a joint appointment in Electrical Engineering and Computer Science. Her research interests lie broadly in the area of control systems, including applications to robotics and manufacturing systems. She has published more than 150 articles in refereed journals and conference proceedings. She was elected Fellow of the IEEE in 2008 and Fellow of the ASME in 2012, and is a Life Member of SWE.