A Modeling and Analysis Framework To Support Monitoring, Assessment, and Control of Manufacturing Systems Using Hybrid Models

by

Miguel Saez

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2018

Doctoral Committee:

Professor Kira L. Barton, Co-Chair
Professor Dawn M. Tilbury, Co-Chair
Professor Jionghua Jin
Professor Kazuhiro Saitou
Francisco P. Maturana Ph.D., Rockwell Automation

Miguel Saez
migsae@umich.com
ORCID iD: 0000-0001-5826-3346

DEDICATION

This work is dedicated to my advisers who guided me through this process, and my family who supported me in every step of the journey.

ACKNOWLEDGEMENTS

I would like to thank Professor Dawn Tilbury and Professor Kira Barton for their support and unceasing patience. Their dedication and support was a motivation to excel in my research. I would also like to thank my committee members for their valuable comments and feedback that guided me through this scientific research.

I would like to thank Rackham Graduate School, The Department of Mechanical Engineering and the National Science Foundation for their financial support during my time at the University of Michigan.

I would also like to thank my parents for their continuous support and guidance in the pursuit of my goals. Finally, I would like to thank my wife, Karly, who helped me every step of the way and whose love was a true inspiration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The manufacturing industry has constantly been challenged to improve productivity, adapt to continuous changes in demand, and reduce cost. The need for a competitive advantage has motivated research for new modeling and control strategies able to support reconfiguration considering the coupling between different aspects of plant floor operations. However, models of manufacturing systems usually capture the process flow and machine capabilities while neglecting the machine dynamics. The disjoint analysis of system-level interactions and machine-level dynamics limits the effectiveness of performance assessment and control strategies.

This dissertation addresses the enhancement of productivity and adaptability of manufacturing systems by monitoring and controlling both the behavior of independent machines and their interactions. A novel control framework is introduced to support performance monitoring and decision making using real-time simulation, anomaly detection, and multi-objective optimization.

The intellectual merit of this dissertation lies in (1) the development a mathematical framework to create hybrid models of both machines and systems capable of running in real-time, (2) the algorithms to improve anomaly detection and diagnosis using context-sensitive adaptive threshold limits combined with context-specific classification models, and (3) the construction of a simulation-based optimization strategy to support decision making considering the inherent tradeoffs between productivity, quality, reliability, and energy usage. The result is a framework that transforms the state-of-the-art of manufacturing by enabling real-time performance monitoring, assessment, and control of plant floor operations. The control strategy aims to improve productivity and sustainability of manufacturing systems using multi-objective optimization. The outcomes of this dissertation were implemented in an experimental testbed. Results demonstrate the potential to support maintenance actions, productivity analysis, and decision making in manufacturing

systems. Furthermore, the proposed framework lays the foundation for a seamless integration of real systems and virtual models.

The broader impact of this dissertation is the advancement of manufacturing science that is crucial to support economic growth. The implementation of the framework proposed in this dissertation can result in higher productivity, lower downtime, and energy savings. Although the project focuses on discrete manufacturing with a flow shop configuration, the control framework, modeling strategy, and optimization approach can be translated to job shop configurations or batch processes. Moreover, the algorithms and infrastructure implemented in the testbed at the University of Michigan can be integrated into automation and control products for wide availability.

CHAPTER I

Introduction

The work in this dissertation aims to enhance the monitoring, assessment, and control of manufacturing systems to support productivity improvements. These enhancements are achieved through a control framework that integrates real-time performance assessment, condition monitoring, and simulation-based optimization. The details of this control framework will be described. The implementation and expected improvements will be discussed with case studies.

I.1 Motivation

The manufacturing industry is responsible for 22% of the Gross Domestic Product (GDP) and 36% of the total energy consumed in the U.S. [11]. However, results show that manufacturing performance as measured by Overall Equipment Effectiveness (OEE), the combination of availability, productivity, and quality metrics, is below 50% for various industry sectors [2]. Considering the key economic and environmental role of the manufacturing industry, low OEE values motivate the need to for improvement.

Manufacturing automation, reconfiguration, and energy efficient control policies have been identified as effective ways to improve productivity and reduce both the environmental footprint and cost with the potential to save $6.2 billion in the U.S. manufacturing industry [37]. However, the increasing complexity of machines and processes, and the need for rapid changes pose a challenge for the assessment, monitoring, and control of manufacturing systems.

I.2 State-Of-The-Art

Manufacturing system performance is affected by both the operation of independent machines as well as their interactions. As a result, a significant amount of research has been focused on the development of efficient methods to assess performance, monitor machine operation and health, and develop control strategies that include both machine- and system-level variables.

I.2.1 Performance Assessment

Manufacturing system performance can be evaluated using a variety of productivity assessment metrics. System-level performance is often evaluated based on the indicators: *throughput*, the number of parts made per unit time, *Work-in-Progress* (WIP), the amount of unfinished goods in the system, and *flow time*, the amount of time that a product spends in the system. Likewise, machine-level performance is generally evaluated based on *cycle time*, the amount of time a machine requires to process a part, and *availability*, the percentage of time a machine is available to process a part. Productivity expectations can be based on historical data or mathematical models.

OEE and Key Performance Indicators (KPI) have been extensively used in manufacturing to assess performance [71]. The expected values of these metrics are often defined based on historical data or enterprise-level requirements. In [83], machine downtime data was used to analyze the severity of a fault based on its impact over availability, showing the importance of monitoring machine-level performance to find the best system-level maintenance policy. In [34], KPI are developed to predict, diagnose, and evaluate the productivity of an industrial hot strip mill. These applications showed the advantages of continuous monitoring of performance metrics for improved productivity and quality. However, the use of historical data to assess performance might neglect changes in plant floor operating conditions.

An alternative method for assessing performance is to develop a mathematical model of the manufacturing system. A *model* is a representation of some underlying essence of a real world object based on governing equations, assumptions, and constraints. Models can be classified as static or dynamic. Static models are time-invariant and represent a system under steady-state conditions. Dynamic models are time-dependent and represent the system under both transient- and steady-state conditions. Production System Engineering (PSE) has developed a framework to build static models of manufacturing systems under the assumption of steady-state conditions and a single part type [79]. The PSE framework models machines and buffers as a Markov chain to estimate throughput and WIP. However, a Markov model does not capture the dynamics of a manufacturing system that contains a variety of machines and processes used to fabricate different part types [80]. A common platform for evaluating the behavior of a dynamic model over time is to run the model in a simulation environment. Discrete Event System (DES) models have been used to study the dynamics of manufacturing systems by simulating a sequence of events over time [61]. In [91], a simulation environment was used as a predictive tool for bottleneck detection and performance diagnosis. The simulation evaluated the impact of machine-level productivity and location on system-level performance. The combination of a simulated machine in a virtual environment and the physical machine in the real world is commonly known as Hardware in the Loop (HIL).

The combination of virtual and real environments in real time has led to the concept of "virtual fusion" [49]. Prior efforts to merge simulated and real-world systems focus on studying the discrete behavior of machines and systems. However, the connection between models of machine-level continuous dynamics and discrete machine- and system-level models has not been explored. Moreover, machine- and system-level simulations are often run separately, limiting the capacity to evaluate the effects of a sequence of events on the machine dynamics. A simulation capable of running in parallel and under the same operating conditions as the plant floor for real-time performance assessment has not previously been developed.

I.2.2 Condition Monitoring

In manufacturing, unplanned machine downtime can disturb the operation of the system and have a negative effect on productivity. Condition monitoring supports effective maintenance by detecting and diagnosing faults based on anomalous signal values. An *anomaly* is defined as an occurrence that is different from what is standard, normal, or expected. *Faults* are a specific class of anomalies that are associated with failures, malfunctions, or quality degradation. Anomaly and fault detection has been extensively studied using both physics-based and data-driven models [105]. The former is based on a continuous model representing physical parameters and machine dynamics. The latter is based on a statistical model developed using historical data.

A physics-based model for Fault Detection and Diagnosis (FDD) requires knowledge of equations that govern the machine dynamics. Physics-based models have been developed to detect faults in different types of machines (e.g., CNC, robots, gantries). However, due to the complexity of some manufacturing processes and changes in machine operations throughout a machine's lifecycle, this approach has not been fully implemented on the plant floor [60]. Data-driven models have been used to detect and diagnose anomalies using historical data. Extensive research has been done to monitor machining operations to detect anomalies based on different signal processing and analysis strategies [1]. However, in order to improve detection and diagnosis, some knowledge of the system, whether from physics-based models or experts, is required [39]. A comparison between physics-based and data-driven models [36] shows that both have advantages and disadvantages based in part on factors such as detail of data available, model development efforts, and implementation challenges.

Prior work on anomaly detection and diagnosis has not considered the different machine-part interactions or the combination of physics-based and data-driven models. Considering that manufacturing machines can perform different tasks and operate under different conditions, a single model might not be able detect anomalies under different operational contexts.

I.2.3 Control Strategies

Many well-known methods exist for controlling different aspects of the plant floor. Production planning and scheduling often control the timing of manufacturing tasks. The operation schedule can include productive tasks such as "part processing" and non-productive tasks such as "repair" [30], [44], [70]. These methods rely on models of the equipment that capture production capacity and reliability. However, aspects such as quality and energy consumption are often not considered. Extensions have been proposed to include energy usage considerations [136], although these methods are not yet used in industry [20]. Efforts to control manufacturing operations should consider the coupling between productivity, quality, reliability, and energy consumption.

Several modeling and control strategies that consider the coupling in manufacturing performance have been proposed. In [94], the study of machine-level quality-quantity coupling showed that increasing processing speed can have a negative effect on quality. In [25], the development of a maintenance policy that considered the quality-reliability coupling showed a positive effect on productivity. Results showed that defining an energy-efficient production schedule can help reduce energy consumption but have a negative effect on throughput [19], [23]. The trade-offs between different performance metrics can be balanced using a multi-objective optimization approach. The formulation of the optimization problem depends on the type of manufacturing system and the temporality of the control variables. For job shop systems, multi-objective optimization problems have been formulated for production control [96] [90]. For flow shop systems, multi-objective optimization problems have been formulated to define the plant configuration and layout [118] [114].

There is a need to bring together these divergent topics of production scheduling, maintenance scheduling, and energy usage on a plant-wide scale, using multidimensional modular models of the equipment in the system. Prior work neglected to develop a modeling strategy that captures the coupling between different performance metrics. Moreover, a control framework capable of evaluating the effect of machine- and system-level control variables on productivity, reliability, quality, and sustainability has yet to be defined.

I.3 Contributions

As manufacturing processes increase in complexity, so does the need to monitor and evaluate their performance. The *first contribution* of this dissertation is a method to monitor productivity using real-time simulation. A novel hybrid modeling strategy is introduced to further extend the concept of "virtual fusion" by studying discrete and continuous behaviors at the machine- and system-levels. The model uses Discrete Event Systems (DES) to estimate performance metrics at a system-level and Continuous Dynamics (CD) to monitor state and output variables at a machine-level. Data from the physical system is compared in real-time with data from the simulation. The concurrent and synchronous simulation provides a reference of expected plant floor performance.

The normal operation of a manufacturing system can be disturbed by anomalies. The *second contribution* of this dissertation introduces a framework for modeling manufacturing systems at the machine-level as Cyber-Physical Manufacturing Systems (CPMS). This modeling approach combines sensor data, context information, and expert knowledge to leverage both physics-based and data-driven models. In this work, anomaly detection is based on context-sensitive adaptive threshold limits, while root cause diagnosis is based on context-specific classification models. The proposed approach was implemented using sensor data and context information extracted using Internet-of-Things (IoT) at both an automotive assembly plant and university testbed.

In manufacturing, there are often trade-offs between different performance indicators. The *third contribution* of this dissertation is a control strategy formulated as a multi-objective optimization problem with decision variables at both the machine- and system-levels to improve productivity, reliability, and sustainability metrics. This dissertation combines both machine- and system-level variables to evaluate control strategies that improve energy efficiency, productivity, quality, and reliability. At a machine-level, the model combines both the discrete and continuous behavior to capture energy consumption and demand. At a system-level, the model considers the interactions between machines and buffers to estimate plant floor performance metrics. The control framework aims to evaluate repair, reconfiguration, and process alternatives.

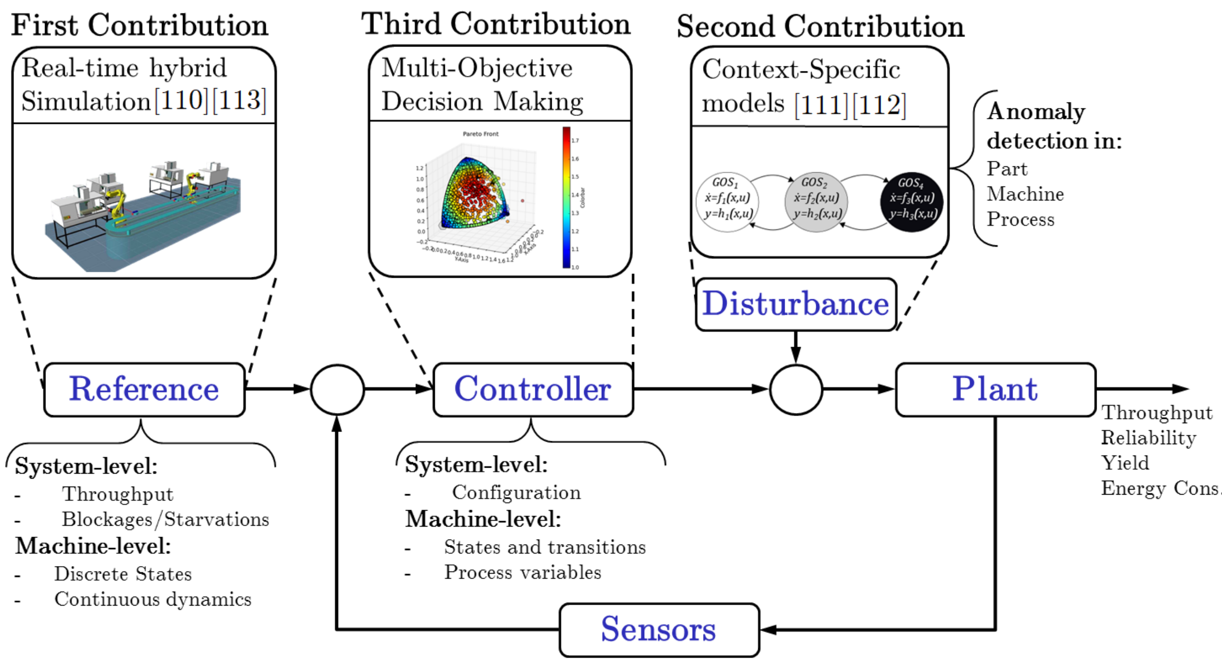


Figure I.1: Integrated Control Framework of Plant Floor Operations

I.4 Expected Impact

The aforementioned contributions create a control approach that provides a reference of expected performance in real time, the ability to detect and diagnose anomalies that can disturb the manufacturing system, and analysis of different "what-if?" scenarios that can be converted into control actions. Deployment can improve productivity, support effective maintenance actions, and reduce energy consumption. If adopted, plant floor operations will be more responsive to changes in operating conditions. The manufacturing control personnel will be able to make better decisions considering the performance trade-offs, the maintenance personnel will take more effective actions to prevent faults or react to anomalies, and the operations personnel will have better knowledge of the machine, part, and process conditions. Moreover, if all the components are integrated, plant floor operations will improve productivity, reliability, quality, and sustainability, driving manufacturing systems to a new level of efficiency. The approach outlined in this dissertation can be used with both centralized and decentralized control strategies by seamlessly integrating plant floor data, simulation models, expert knowledge, and automation.

I.5 Assumptions

The framework and analysis methodologies in this dissertation are subject to a number of assumptions regarding the type of manufacturing system, control infrastructure, and data extraction capabilities. First, this research only considers discrete manufacturing systems where products are processed as individual workpieces (e.g. vehicles) and not continuous manufacturing where products are processed in batches (e.g. oil). Moreover, this dissertation focuses on the flow-shop type of systems where resources are arranged based on the sequence of the operation. Here, the term "resource" refers to equipment in the manufacturing system capable of transporting a part, such as gantries and conveyors, or processing a workpiece, such as CNC machines. Second, this work assumes a centralized control strategy where the actions of different resources are coordinated by a single controller and a hierarchical structure is clearly defined. Resources are able to send messages to and receive commands from a system-level controller using a communication network. It is assumed that the physical location of the resources does not change, but the system can be reconfigured by changing the task assigned to individual machines. Third, this dissertation assumes that the operation of machines and systems can be monitored using sensor data and context information. Sensor data refers to a continuous signal that needs to be processed in order to gain insight such as encoder data or an energy signal. Context information refers to variables that explicitly describe the operation such as process step, part number, and machine state. It is assumed that both sensor data and context information can be extracted in real-time using industrial communication protocols such as OPC-UA [124] or MTConnect [128]. The term "real-time" means that measured signals can be observed and extracted in less than one second. Lastly, this research considers that knowledge of the machine dynamics, operation, and performance is available or can be obtained by a combination of process observations and data analysis. Moreover, the control logic that defines the functionality at both the machine- and system-levels can be created or is already available.

I.6 Limitations

The work presented in this dissertation has some limitations in the integration of models and real systems for assessment, monitoring, and control.

- The assessment framework presented here relies on models of discrete and continuous behavior. However, changes in these behaviors due to machine degradation are not considered. Given that some degradation can be expected throughout the machine lifecycle, the divergence of real machine operation and simulation outputs over time represents a limitation for implementation.

- The anomaly detection strategy in this dissertation requires a combination of sensor data, context information, and expert knowledge. The increasing complexity of the modeling framework and the need of subject matter expertise might represent a challenge for scalability.

- The control approach introduced in this research work is based on the evaluation of different combinations of machine- and system-level variables. As the number of machines and possible interactions increases, so does the model complexity and computational requirements. The computational time of the simulation-based optimization approach might limit the capacity to generate time-sensitive control actions in large manufacturing systems.

I.7 Dissertation Overview

The organization of this dissertation is as follows: Chapter II presents the monitoring and assessment strategy using real-time hybrid simulation. Chapter III presents the anomaly detection and diagnosis algorithms using context information. Chapter IV covers the integrated modeling framework and multi-objective optimization to support decision making. Finally, Chapter V addresses conclusions and future work.

CHAPTER II

Real-time Manufacturing Machine and System Performance Monitoring

This chapter presents the research work published in [113] [110] which extends the state-of-the-art in Hardware in the Loop (HIL) simulation for performance assessment of manufacturing systems.

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 [69] 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 [3]. 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 [45], [89]. 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 [60], [143]. 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 [98]. 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 chapter 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 [101]. However, extracting machine data remains a major implementation challenge. Recent developments in Industrial Internet of Things (IIoT) and communication protocols have simplified data collection, making possible advanced monitoring techniques [13], [10], [142].

In this chapter a new approach assessing the performance of manufacturing systems is proposed. One of the main challenges of using a simulation 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. 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.

A real-time performance assessment method using hybrid simulation was developed in [110] [113] with three main contributions:

The *first contribution* of this chapter 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 chapter is to introduce 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 both machine- and system-levels. 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* of this chapter 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 chapter is organized as follows. Section II.1 provides background on the research area. Section II.2 defines the hybrid simulation providing details of discrete and continuous machine models. Section II.3 describes the performance analysis rules and plant floor data extraction. Section II.4 demonstrates the validity of the approach through a case study using a University of Michigan testbed. Finally, Section II.5 summarizes the work.

II.1 Background

In this chapter, 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.

II.1.1 Models and Simulations 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 appropriate 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 [98]. 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 [62], 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 [68]. 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 [29] or improve productivity, however, for performance monitoring and assessment the need for a synchronous simulation running in real-time is evident.

As companies seek to expand system-level perspective of their operations, DES models have been extended to capture additional information such as machine Continuous Dynamics (CD). Industry's interest in hybrid systems, which is the combination of DES with CD, has grown in recent years [61]. Hybrid models capturing system and machine-level dynamics can help trace problematic behavior. In [107], 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 study plant floor reconfiguration are not considered.

II.1.1.1 Discrete Models of Manufacturing Systems

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 [108]. A discussion on the selection of the formalisms and analysis framework based on modeling viewpoint and concern aspect is presented by Broman et al. [17]. 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 [15], 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 [33]. 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 [54]. 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 [26]. 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 [103], [95]. 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 [4]. Similar work was presented in [86] 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 [61]. 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 [137]. 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 [104]. The productivity of a manufacturing system has been improved by combining DEVS models and Model Predictive Control (MPC) in semiconductor manufacturing [55]. The use of simulations helped maintain stable operation under nonlinear and stochastic plant dynamics. In [58], a DEVS model built on Matlab/SimEvent was used to analyze the fabrication process in a nuclear facility and evaluate efficiency. Plant models developed on DEVS thus far capture the time-driven transitions of discrete states but do not include the machine dynamics and are not real-time capable.

II.1.1.2 Continuous Models of Machine Dynamics

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 [133], [119]. In [64], 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 [63]. Simulation tools for virtual commissioning in real-time can be used to visualize control action and machine dynamics [93]. 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. [22] 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 [22]. A study of continuous dynamics has been used for anomaly detection by monitoring residuals between expected and real values. In [130] 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.

II.1.2 Manufacturing Systems Performance Analysis

The different performance metrics evaluated throughout a manufacturing system life cycle are compared by Leung *et al.* [78]. 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) [59]. MES links Enterprise Resource Planning (ERP) with plant floor equipment to monitor performance using Overall Equipment Effectiveness (OEE) and Key Performance Indicators (KPI) [71]. 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 [83] 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 [34], 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 [79]. 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 [80].

The importance of modeling and simulation of manufacturing systems to improve performance is demonstrated in [56]. 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 [91]. Nonetheless, the continuous machine dynamics are not included in the analysis, and the concurrent analysis between the real and simulated systems is not discussed.

II.1.3 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 [50]. 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 [131].

IIoT has grown in popularity for data extraction. In [128], 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 [10]. However, MTConnect has been limited to machine-level data extraction. In [8] 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 [141], 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 [140] 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 [122], 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 extracted via IIoT was not discussed. Moreover, merging the two environment has the potential to improve the performance analysis and control actions in manufacturing systems.

II.2 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.

II.2.1 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.

II.2.1.1 Discrete Models of Machines and Systems

In this dissertation we model manufacturing systems using the Discrete Event System Specification (DEVS) formalism [137]. 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 A:

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

where:

$U = \{e_{i1}, e_{i2}, ...\}$	Set of inputs
$Y = \{e_{o1}, e_{o2}, ...\}$	Set of outputs
$X = \{s_1, s_2, ...\}$	Set of states
δ_{int}:$S \times \{t_{adv}, \emptyset\} \rightarrow S$	Internal transition function
δ_{ext}:$X \times U \rightarrow S$	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

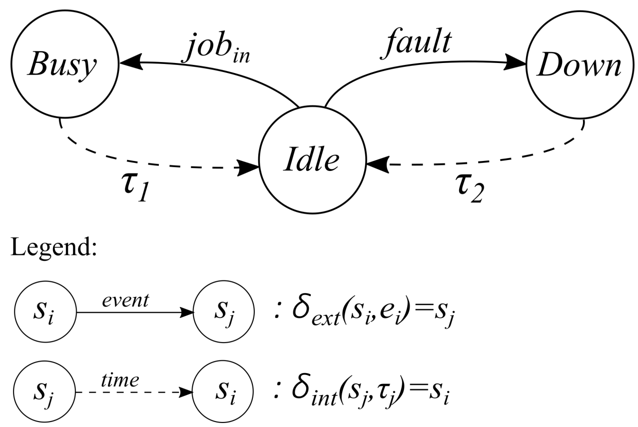


Figure II.1: Atomic Model of a Machine With Time- and Event-Driven Transitions

Atomic models for event generators, buffers, and processors are presented in [53]. 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. II.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. II.1 and represented by:

$$U = \{job_{in}, fault\}$$

$$Y = \{job_{out}, ready\}$$

$$S = \{Idle, Busy, Down\}$$

$$\delta_{int} = \begin{cases} \delta_{int}(Busy, \tau_1) = Idle \\ \delta_{int}(Down, \tau_2) = Idle \end{cases}$$

$$\delta_{ext} = \begin{cases} \delta_{ext}(Idle, job_{in}) = Busy \\ \delta_{ext}(Idle, fault) = Down \end{cases}$$

$$\lambda = \begin{cases} \lambda(\delta_{int}(Busy, \tau_1)) = job_{out} \\ \lambda(\delta_{int}(Down, \tau_2)) = ready \end{cases}$$

$$t_{adv} = \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2\}$$

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

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:

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

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 [137]. An example of a coupled model of two machines (M_1, M_2) and one buffer (B_1) is shown in Fig. II.2. The set of atomic models is defined by $\boldsymbol{A} = \{M_1, M_2, B_1\}$. 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.

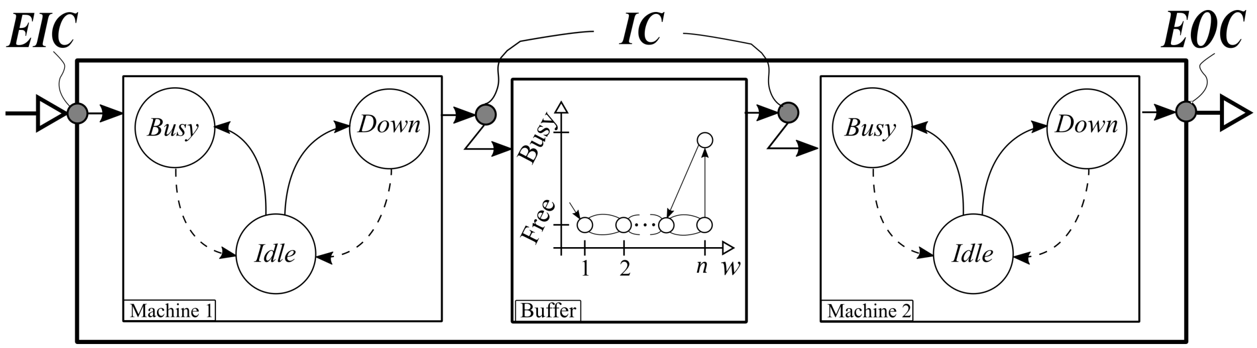


Figure II.2: Coupled Model of a Manufacturing System With Machines and Buffers

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 $t \to \infty$, $\Lambda(t)$ converges to a constant value Λ [21].

The focus of this chapter 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 ζ, Work-in-Process β, and Yield defined as the ratio between throughput and arrival rate (ζ/Λ).

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

Expected throughput ($\zeta(t)$) 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) [91]. Moreover, machine-level variables such as transition times can be combined into performance metrics (e.g: availability, efficiency) to dynamically monitor OEE [3]. 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).

II.2.1.2 Continuous Models of Machine Dynamics

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. II.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. 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 $\boldsymbol{T}$ for an industrial robot arm represented as a kinematic chain. The inputs are the position in the world coordinate frame $u = \{p_x, p_y, p_z, r_x, r_y, r_z\}$, the state variables are $x = \{q_1, \dot{q}_1, ..., q_6, \dot{q}_6\}$, and the output variable is torque $y = \{T_1, ..., T_6\}$. The transformation from inputs to state variables is done using inverse kinematics, and the output are calculated by [28]:

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

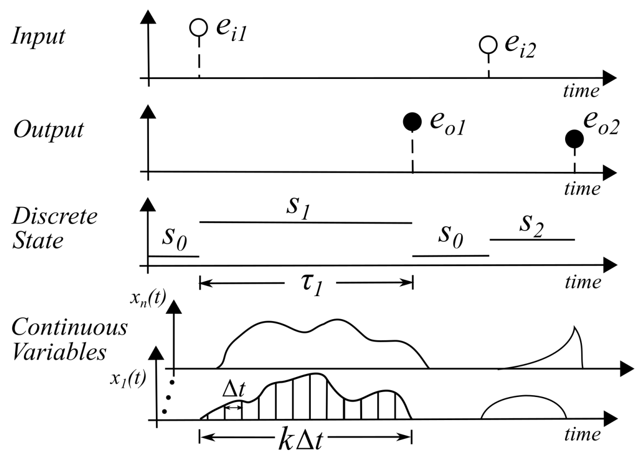


Figure II.3: Hybrid Discrete and Continuous Model

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 [133] or are available from the machine manufacturer.

II.2.1.3 Hybrid Model

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. II.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. The continuous outputs are matrices Θ and Γ of time-series vectors of 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, as described in the section II.2.

II.2.2 Real-time Synchronization

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

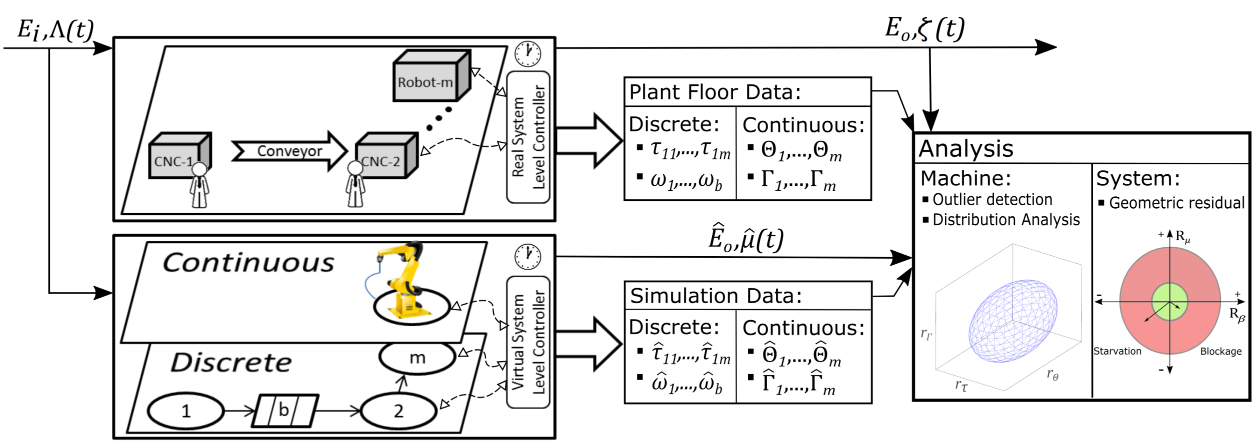


Figure II.4: Analysis Framework With Real and Virtual Environment

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 and comparison of solvers is discussed in [117]. Step size provides a metric to analyze simulation time. Smaller step sizes lead to longer simulation times with more accurate results. To assure the 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 can be detected based on zero-crossing detection [123].

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. II.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 research we use Rockwell Automation's *RSEmulate* to emulate a PLC operation and *SimKit* to describe the interaction between the emulated PLC and simulation [92]. 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. II.4, data streams from both the simulated and real system enable real-time analysis.

II.3 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.

II.3.1 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. The integration between plant floor and simulation is requires monitoring discrete and continuous signals

• 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 ζ. 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 [56] [91], 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. II.5. Machine- and system-level performance metrics are evaluated based on real tag values and simulation results.

II.3.2 Productivity and Health 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- levels at any point in time.

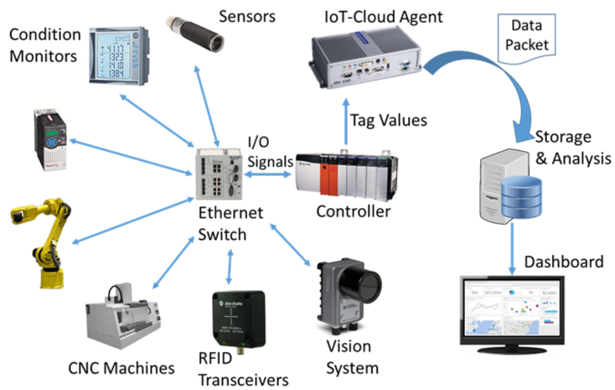


Figure II.5: Data Flow

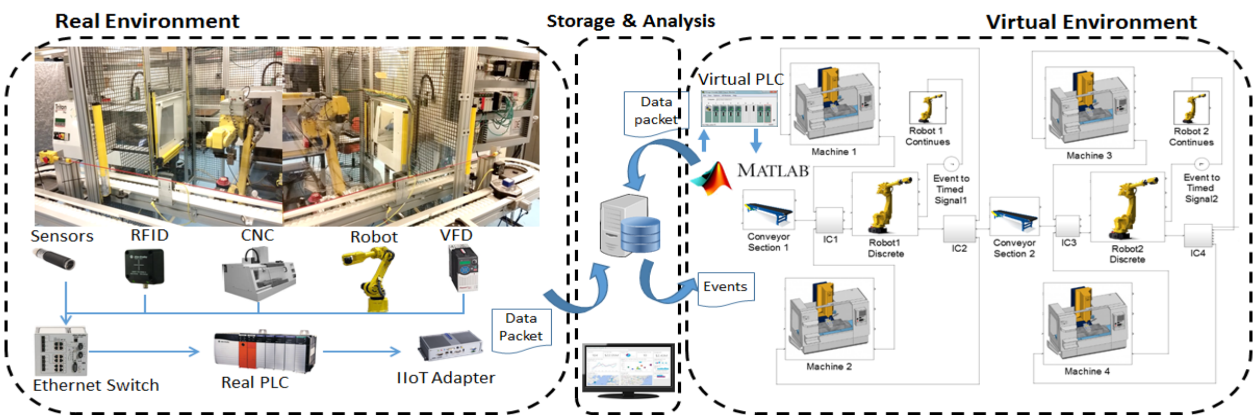


Figure II.6: Real and Virtual Environment

The novelty of this chapter 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 chapter, 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.

II.3.2.1 Machine-Level Performance Analysis

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:

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

• 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) [115] is used to align signal features. The residual between warped time-series is given by:

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

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

The analysis is based on monitoring vector $\mathbf{r}$ for each event in E_i and a substring of E_i defined by a sliding window $\mathbf{v}$. Performance evaluated by detecting outlier.

• Outliers 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 matrix. Outliers are detected using Olive-Hawkins method [100] and Mahalanabis distance $D(z)$ from a cluster in a multivariate space:

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

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. $\boldsymbol{\Sigma}$ 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. II.4

• Distribution Analysis: In this section 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:

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

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 [27]. The distance between distributions of consecutive sliding windows is used for hypothesis testing. The null hypothesis that both samples come from a common distribution is evaluated for 95% confidence intervals. A rejection of the null hypothesis identifies an abnormal distribution of residuals.

II.3.2.2 System-Level Performance Analysis

Productivity metrics such as throughput (ζ) and Work-in-Process (β) are key performance metrics [79]. 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 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) [60]. However, in this section 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. We define the system-level residual as:

$$\boldsymbol{r}(t) = \begin{bmatrix} r_\zeta(t) \\ r_\beta(t) \end{bmatrix} = \begin{bmatrix} \zeta(t) \\ \beta(t) \end{bmatrix} - \begin{bmatrix} \hat{\zeta}(t) \\ \hat{\beta}(t) \end{bmatrix}$$

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 ($r_\zeta(t)$) and Work-in-process ($r_\beta(t)$) can describe changes in the bottleneck location.

II.3.3 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. II.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 $r_\zeta(t)$ 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 [7] [56].

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 [9].

II.4 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 testbed is equipped with two Fanuc robots, four Denford CNC milling machines, and a conveyor loop [73]. 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 II.2.2. Data from the testbed and simulation were collected in real-time. Machine and system-level performance were evaluated following the analysis described in section II.2.3.

II.4.1 Modeling Manufacturing Systems

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

II.4.1.1 Machine-Level Model

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. II.7. As represented in the DEVS formalism example shown in section II.2.1, 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

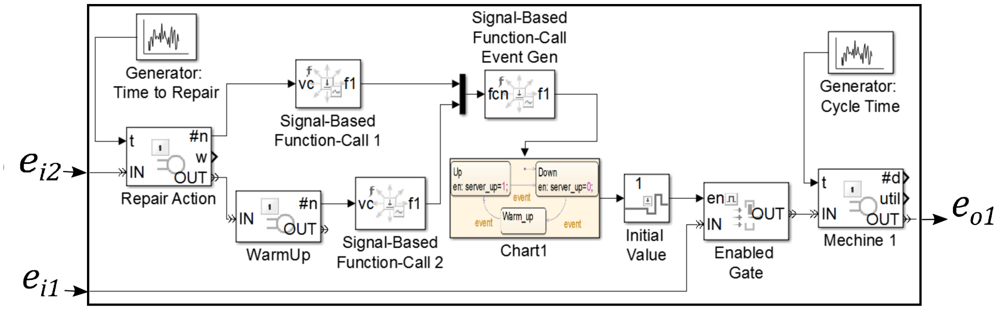


Figure II.7: Single Machine Discrete-Event Model

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 [28]. 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. II.8.

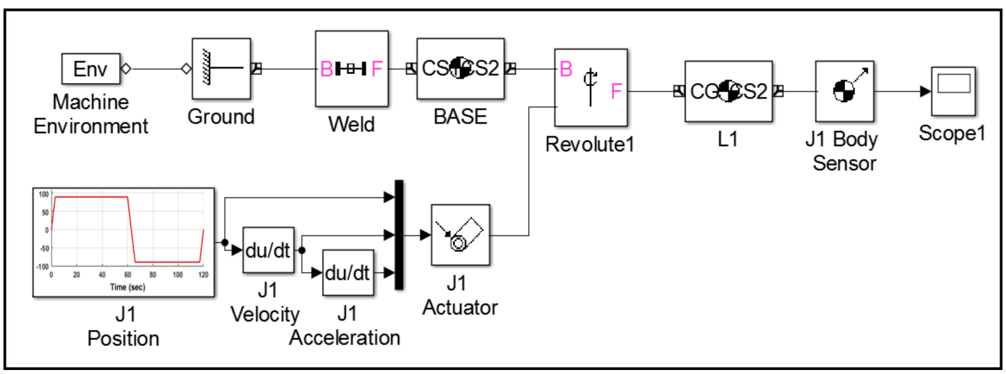


Figure II.8: Robot Joint Model

II.4.1.2 System-Level Model

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 II.6 demonstrates the interaction between machines.

II.4.1.3 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.

II.4.2 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.

II.4.2.1 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 II.6 shows the integration between real and virtual environment by sending external events from the testbed into the simulation and sending data packets containing discrete and continuous variables into the common repository.

II.4.2.2 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

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 II.1.

Table II.1: 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 II.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.

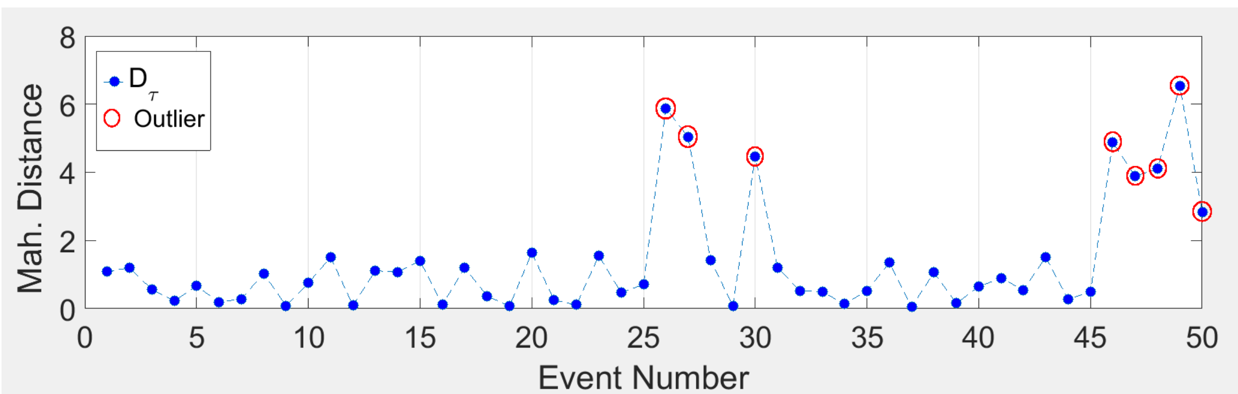


Figure II.9: Mahalanabis Distance of Cycle Time Residual

To study distributions, we used a sliding window of 5 samples. For each window we estimated a kernel using Eq. (II.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. II.10 and Table II.2 respectively.

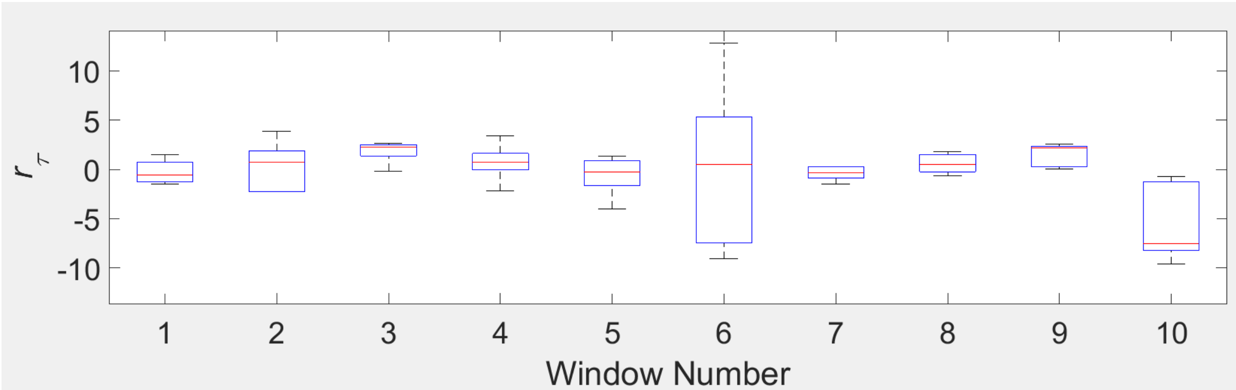


Figure II.10: Box Plot of Cycle Time Residual

Table II.2: 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. II.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. II.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.

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 II.11 shows the trajectory in each coordinate axis for both the real and virtual robot.

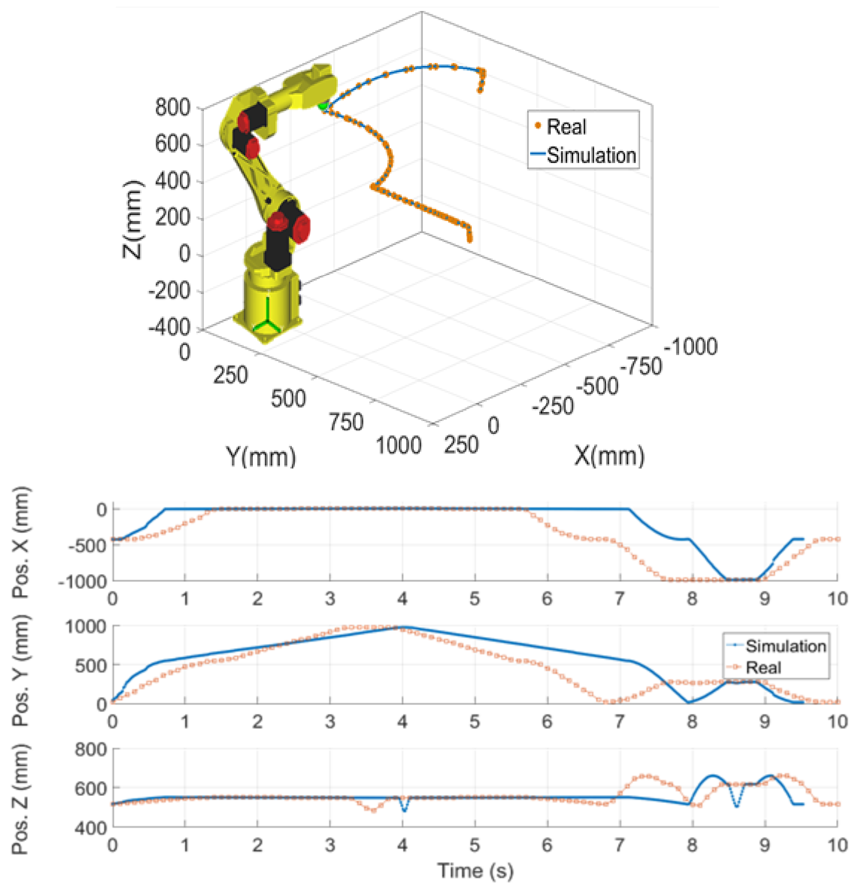


Figure II.11: Fanuc M-6iB: Physical and Virtual model

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. II.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-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.

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. II.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.

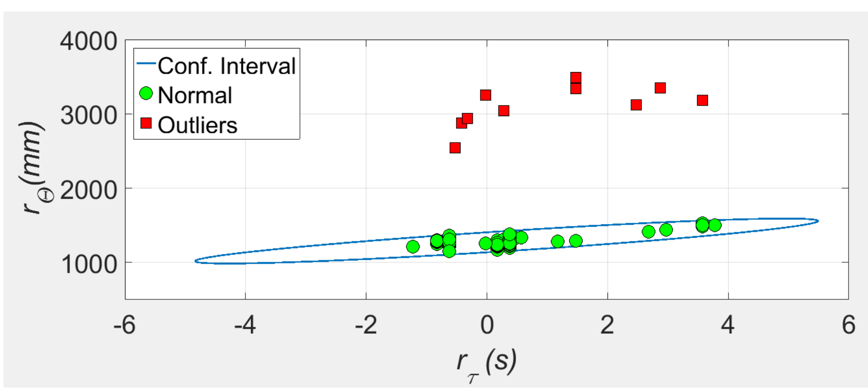


Figure II.12: Residuals of Trajectory and Cycle Time for Outliers Detection

II.5 Summary

This chapter presented a mathematical framework for real-time modeling and synchronous simulation of a manufacturing plant at both the machine- and system-levels. The hybrid model presented here merges discrete and continuous variables at the machine-level and considers the interactions at the system-level. 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 in the same context as the plant can help direct comparisons of expected performance metrics at any given point in time.

This chapter advances the state-of-the-art in hybrid modeling, simulation, and real-time synchronization along with the necessary techniques to analyze the performance of manufacturing systems. The proposed approach was demonstrated 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. The experimental validation of this framework demonstrated how to evaluate performance and detect anomalies in different machines. The research presented in this chapter has the potential to improve the analysis of OEE by using synchronous simulation to manage expectations.

CHAPTER III

Anomaly Detection and Diagnosis on Cyber-Physical Manufacturing Systems

This chapter presents the research work published in [111] [112] which extends the state-of-the-art in anomaly detection and diagnosis for condition monitoring of manufacturing equipment.

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. 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.

Anomaly or fault detection has been extensively studied using both physics-based [60] and data-driven [1] [39] models. A comparison between both modeling strategies showed that both have advantages and disadvantages and a single model might fail to capture all the machine operating conditions [36]. The goal of this chapter is to answer the following questions: 1) How to detect anomalies considering the different machine-part interactions? 2) How to improve the diagnosis of anomalies by considering the operational context in classification algorithms?

Recent advances in machine communication, data extraction and real-time analysis have enabled development of cyber-physical systems. A cyber-physical system is defined by the integration of cyber and physical components such as communication and control networks, sensors, and actuators in a multi-layer architecture [76]. In this chapter, a novel approach to model manufacturing operations as a hybrid system is presented. The model leverages local computing, communication, and control for CPS in manufacturing to estimate discrete states and continuous variables.

An anomaly detection and diagnosis framework merging sensor data, context information, and expert knowledge was developed in [111] [112] with three main contributions:

The *first contribution* of this chapter is a mathematical 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 of operational context and continuous dynamics.

The *second contribution* of this chapter is to develop a framework for anomaly detection and diagnosis based on context-sensitive adaptive threshold limits combined with context-specific classification models and knowledge-based rules.

The *third contribution* of this chapter is an experimental demonstration of the proposed framework to detect and diagnose anomalies contained within the part, machine, and process of a machining operation considering the context information

The remainder of this chapter is organized as follows. Section 1 provides background on the research area. Section 2 defines the modeling framework providing details of discrete states and continuous dynamics. Section 3 describes the anomaly detection and diagnosis methods. Section 4 presents a case study to validate the approach for anomaly detection and diagnosis in a machining application. Finally, Section 5 concludes the chapter and discusses other applications.

III.1 Background

In this chapter, an abstract model of manufacturing operations studied as Cyber-Physical Manufacturing Systems (CPMS) is presented for anomaly detection and diagnosis.

III.1.1 Cyber-Physical Manufacturing Systems

A cyber-physical manufacturing system (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 a manufacturing process. The analysis of CPMS requires data extraction and model development.

III.1.1.1 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 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 [124], while the MTConnect protocol has been developed specifically to extract controller data from CNC machines [127].

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 [24] 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 has also been used to improve manufacturing sustainability using MTConnect [129]. However, the capability of extracting sensor data and context information to provide insight into machine operations has not been fully developed for anomaly detection.

III.1.1.2 Modeling Cyber-Physical Manufacturing Systems

Cyber-Physical Systems 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 Finite State Machines (FSM) and hybrid Petri-nets. The formalism can be seen as the “semantics" linking the cyber and physical domains. In [17] different formalisms and tools to model CPS are discussed and compared. Physics-based models have been developed using the identification of states based on observed data without the need for prior knowledge [38]. Energy consumption has also been studied to generate a hybrid timed automaton based on historical data for estimation of expected behavior [99]. 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 [116] system level control of CPS for decision making shows how the implementation of communication networks and cloud computing can improve flexibility of material handling systems. Anomaly detection models have also been improved by studying CPS given that more data is made available for process monitoring. Different models have been suggested, 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 [99]. However, applications are still limited, and expert knowledge is needed for diagnosis in cases where results require operational context considerations.

III.1.2 Anomaly Detection

In manufacturing, anomaly and fault detection on machine tools has 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 statistical analysis of historical data. In [60], 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 increase in signal noise during the manufacturing operation caused by the machine-part interactions. In [57], 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 [14] 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 [109]. 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, selection of the proper clustering and classification algorithms for studying time-series data should be based on the type of data and application [84].

Limit-based methods for anomaly detection often require consideration of the impact of false positives and false negatives (type I and type II errors respectively). This consideration can be based on cost [97] [74] or risk [42] [31]. In manufacturing, the risks associated with part or process anomalies are evaluated using Failure Mode and Effect Analysis (FMEA) [43]. However, the ability to assign risk for specific threshold limits often requires expert knowledge.

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 technology such as IoT has granted access to context information that can complement both modeling strategies and anomaly detection and diagnosis algorithms. This chapter aims to improve modeling and analysis of CPMS for anomaly detection by using context information extracted from machine and system level controller.

III.2 Modeling Cyber-Physical Manufacturing Systems

The interconnection of information management systems and plant floor data has set the ground work for modeling and analysis of Cyber-Physical Manufacturing Systems. Information from the cyber domain, 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 a cyber-physical manufacturing system 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 Global Operational States (GOS), identification of Continuous Dynamics (CD) models, and definition of the hybrid system by specifying the CD for each GOS of the manufacturing operations.

III.2.1 Discrete States

Global Operational States (GOS) represent the discrete set of states characterized by the operational context of the machine. In this chapter we define GOS as the combination of functional, dynamic, and interactive states identified using implicit process descriptors and explicit process descriptors.

III.2.1.1 Implicit Descriptors

Implicit descriptors require interpretation of machine data and control logic by an expert to provide context. The implicit descriptors are defined as states in different domains: Functional (F), Dynamic (D), and Interactive (I) using Discrete Event System Specification (DEVS) [138]. Each domain is represented in an atomic model defined as a tuple H.

$H^i = (E^i, S^i, \delta^i)$ for $i \in F, D, I$ where:

$E^i = \{e_1, e_2, ...\}$ Set of events

$S^i = \{s_1, s_2, ...\}$ Set of states

$\delta : S \times E \rightarrow S$ Transition function

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

• 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, stand-by, positioning, processing, changing tool, setup).

• 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).

Identification of functional states requires some information about the control system. This information can be in the form of a Finite State Machine (FSM) or control logic in the PLC. Expert knowledge may help identify the states, events, and transitions relevant for anomaly detection.

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

• Dynamic state: 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).

• 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 (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 [67]

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

• 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).

• 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) $e^I = [Y_{pat}(1)...Y_{pat}(n)]^T$.

In a 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 anomaly detection and diagnostic processes. Identification of interactive states and events requires knowledge of the manufacturing process to identify data patterns. Given a matrix of continuous output variables $G = [Y(1)...Y(m)]^T$ collected during a manufacturing operation, the time instance when e^I has occurred can be obtained using the search algorithm in [111].

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 state explosion. However, some combinations are unfeasible (e.g., idle, constant speed, face milling). A data- or knowledge-driven approach can help reduce the number of possible combinations to consider. Knowledge of the control logic or the manufacturing operation can support limiting the number of combined states to a feasible set.

III.2.1.2 Explicit Descriptors

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

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 operational context.

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.

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 the explicit descriptors.

III.2.1.3 Global Operational State

The abstraction of manufacturing equipment as a CPS 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 chapter, we define the CPS abstraction at a machine level as a coupled model describing a Global Operational State (GOS).

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

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 $GOS_2 = \{Processing, Accelerating, NoInteraction, 1, 5, 1\}$

III.2.2 Continuous Dynamics

The continuous dynamic 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. The proper structure of $f(\cdot)$ and $h(\cdot)$ to capture the machine dynamics can be represented in a white-box, grey-box, or black-box model.

III.2.3 Hybrid Model

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

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

- $\boldsymbol{GOS}$ represents the discrete set of Global Operational States
- U is the continuous input space of the system in which the continuous input variables u take their values. For our purpose $U \subset \mathbb{R}^m$
- X is the continuous state space variable where $X \subset \mathbb{R}^n$
- Y is the continuous output space of y where $Y \subset \mathbb{R}^q$
- $F : \boldsymbol{GOS} \times X \times U \rightarrow TX$ is the mapping of U and X into TX that assigns a model of state variable evolution f to each GOS
- $H : \boldsymbol{GOS} \times X \times U \rightarrow Y$ 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 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.III.1. The discrete and continuous behavior are summarized in table III.1.

III.2.4 Scalability

Expert knowledge can help reduce model complexity by identifying three key aspects:

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 manufac-

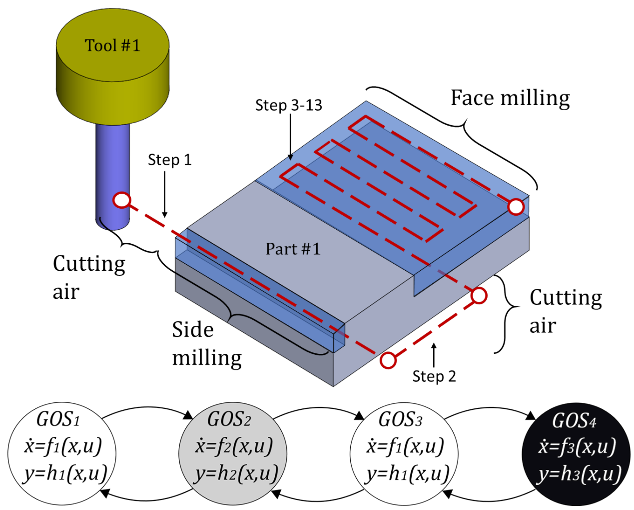


Figure III.1: Data Extraction, Analysis and Partitioning Process

Table III.1: 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

turing to reduce the number of states based on the process requirements and capabilities.

2. Dynamic models: The machine dynamics and the effect of 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 III.1, 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.

III.3 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. Anomalies are detected based on adaptive threshold limits by studying residuals between estimated and actual values. The root cause is diagnosed using supervised clustering or classification models where a specific classification model is assigned to each operational context.

III.3.1 Anomaly Detection

In this chapter 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 non-zero values under normal conditions. 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:

$\mu_y(t) = \sum_{i=1}^{n}(r_{y_i}(t)/n)$ 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.

III.3.1.1 Confidence Intervals

Based on 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%, 99%) to balance detection errors. The Receiver Operating Characteristic (ROC) curve can be used to evaluate accuracy of a binary classifier as determined by a discrimination threshold based on the ratio between true positives (detection) and a false positive (false alarm) [97].

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 operation
2. Evaluate the mean and standard deviation of the residual
3. Build ROC curve by assessing the true positive (TP) and false positive (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 detections and false alarms is defined by $m(TP, FP) = 1$.

If the cost associated with false negatives is larger than the cost of a false positive the optimal slope can be less than 1 (i.e.: $m(TP, FP) = 0.8$) [88].

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 $\boldsymbol{Z_y}$

III.3.1.2 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. 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 [42] [31]. In this chapter 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 trade-offs between detection errors. The risk coefficient can be assigned by an expert 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:

1. Evaluate design and process FMEA
2. Define the critical part features or process step based on high Risk Priority Number (RPN)
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.III.1 is presented in Fig.III.2. Considering the accuracy of physic-based or data-driven models in capturing the machine dynamics during different GOS, it is possible to have off-sets on mean residual values.

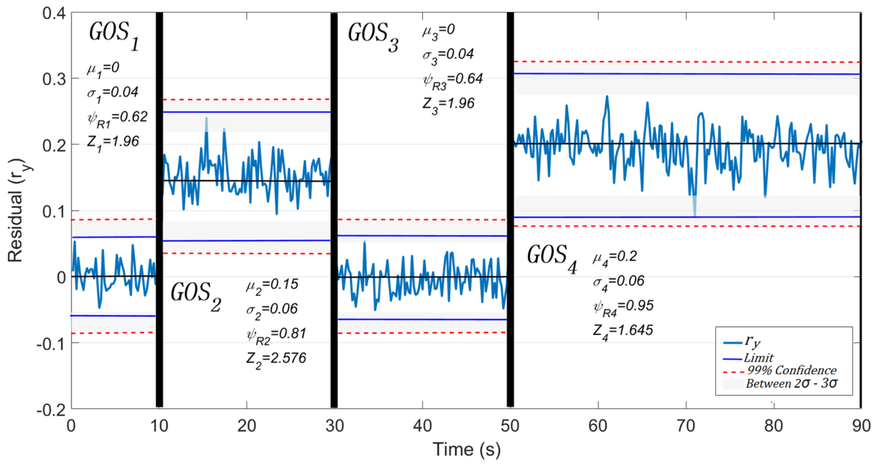


Figure III.2: Data Partitioning and Adaptive Threshold Limits

III.3.2 Root Cause Diagnosis

In a manufacturing operation, 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 non-stationary. Moreover, not all anomalies are equally likely to occur under different operating conditions.

Diagnosis of anomalies can be improved by considering the machine operational context as defined by explicit or implicit descriptors. Partitioning a non-stationary output signal by GOS can improve the diagnosis model by creating multiple stationary segments of similar operational context. Moreover, context-sensitive classification models can be developed be specifying a supervised learning or knowledge-base for each GOS. An example would be to use supervised classification methods for root cause diagnosis [132]. A Support Vector Machines (SVM) classification model can be developed for each partition, i.e. for each GOS_i a SVM_i is defined for $i \in \{1...p\}$. 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.

III.4 Implementation and Evaluation

The methodology presented in the previous section was implemented to detect anomalies in a machining operation. The experimental setup is based on a 3 axis CNC machine enabled with OPC-UA communication. The data was extracted using Rockwell Automation IoT adapter. The machine was studied as a cyber-physical manufacturing system by considering the control architecture, communication capabilities, and manufacturing operation. The model was developed using motion and electric power data from each drive, and part and process data, extracted from the machine controller. Figure III.3 describes the data extraction capabilities.

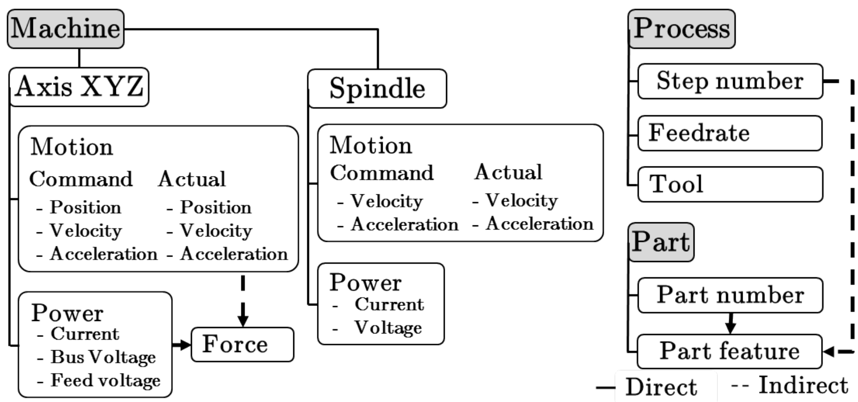


Figure III.3: IoT Data Extraction Schema

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. Detection was performed by monitoring the residual of output variables throughout the entire manufacturing operation, while diagnosis utilized classification models developed using context information. Figure III.4 shows the part, features, and tool trajectory. Table III.2 describes the manufacturing operation and tool used for each part feature.

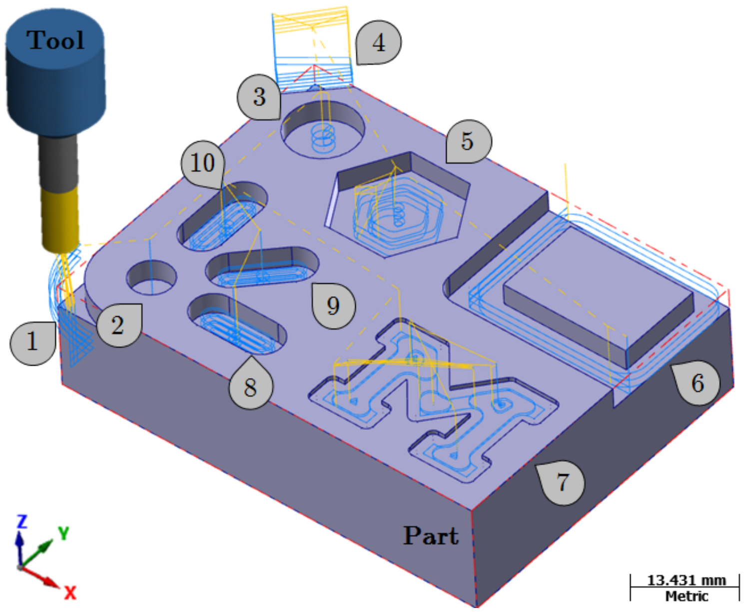


Figure III.4: Sample Part Machining Description

Table III.2: 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

III.4.1 Cyber-Physical Manufacturing System Model

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

III.4.1.1 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 process step. The data required to identify the descriptors were extracted from the machine and system controller. The atomic models are defined as follow

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. Figure III.5 shows the functional atomic model H^F including states, events, and transitions.

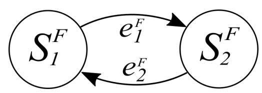


Figure III.5: Functional Atomic Model

$H^F = (U^F, S^F, \delta^F)$ where:

$U^F = \{e_1^F, e_2^F\}$ Set of events

$S^F = \{s_1^F, s_2^F\}$ Set of states

$s_1^F = Idle \quad s_2^F = Processing$

III.4.1.2 Dynamic

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

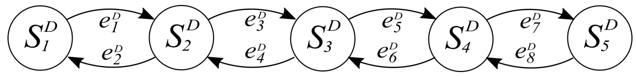


Figure III.6: Functional Atomic Model

$H^D = (U^D, S^D, \delta^D)$ where:

$U^D = \{e_1^D, e_2^D, \ldots, e_8^D\}$ Set of events

$S^D = \{s_1^D, s_2^D, \ldots, s_5^D\}$ Set of states

$s_1^D : CS = 0$ $\quad s_2^D : CS = 1.8$ $\quad s_3^D : CS = 2$

$s_4^D : CS = 2.5$ $\quad s_5^D : CS = 50$

III.4.1.3 Interactive

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. Figure III.7 shows the states and transitions.

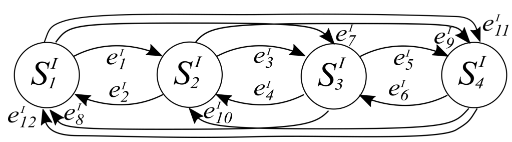
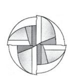


Figure III.7: Machine-Part Interaction States

$s_1^I = No.Int$  $s_2^I = End.Int$ 

$s_3^I = Side.Int_1$  $s_4^I = Side.Int_2$ 

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. Figure III.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. Using the partitioning algorithm presented in [111], interactive events within the manufacturing process were identified.

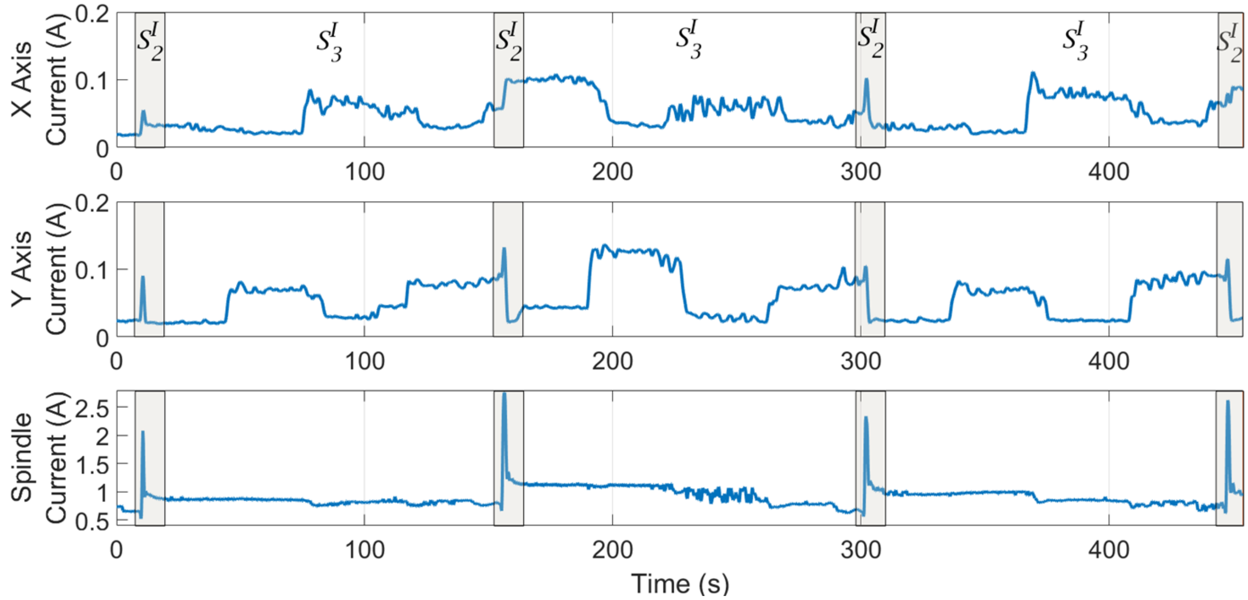


Figure III.8: Current of XY Drives and Spindle Partitioned by Interactive State

III.4.1.4 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 multi-model framework presented on section III was used.

Physics-based Models of the X and Y axis drives on the CNC machine. A one-mass model based on the physics of the electric drive is defined as [60]:

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

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

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

Data-driven Autoregressive models developed to study the current and voltage of the X and Y drives. The order of the models was estimated based on the Box-Jenkins analysis using time series data [48]. 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 (ARMAX) was defined as:

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

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

The parameters ϕ,β are polynomials with respect to the backward shift operator (B) identified by fitting norm-based models, n is the system delay, and ε is the disturbance [87].

III.4.1.5 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.

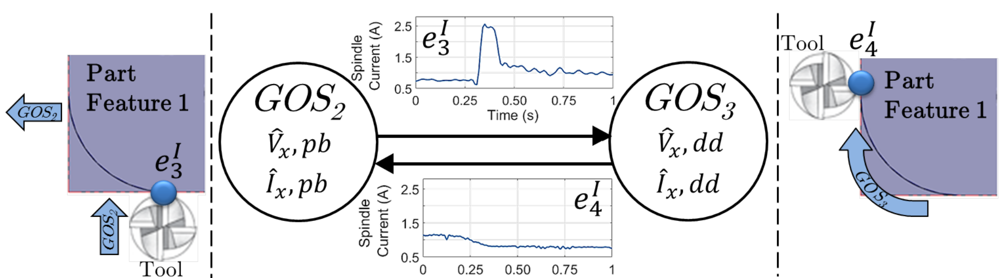


Figure III.9: Description of Hybrid Model With Interactive Events

Figure III.9 shows the discrete states and continuous dynamic model for machining part feature 1 (side milling - fillet) 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.

III.4.2 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:

- Tool: Worn tool, broken tool
- Part: Wrong material, wrong dimensions

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

III.4.2.1 Residual Analysis

For anomaly detection we implemented context-sensitive adaptive threshold limits presented in Section III.3.1. Context is defined by the GOS. The limits were defined by mean μ and standard deviation σ of the residuals determined by evaluating the output of the model to 20 independent data samples collected under normal operation. Figure III.10 shows the GOS, and residual of the output variables for three part features under normal and abnormal conditions. Table III.3 summarized the partitions, states, model and limits.

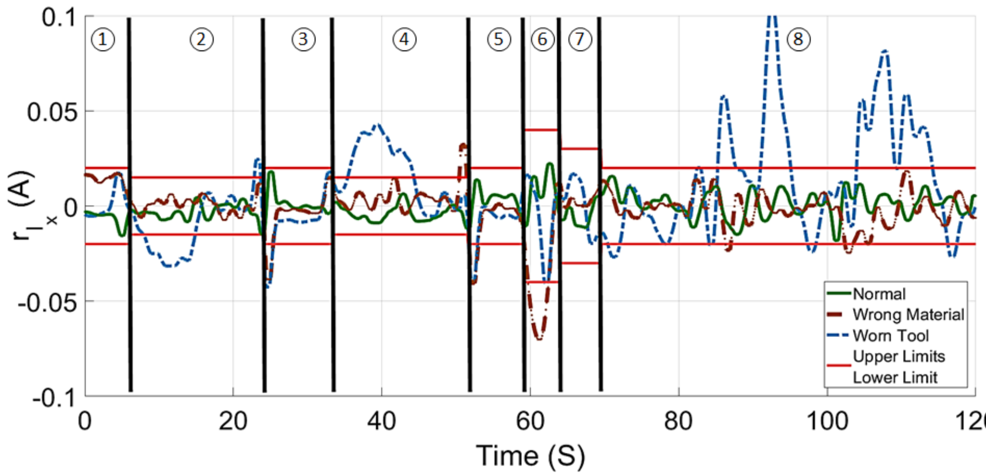


Figure III.10: Adaptive Threshold Limits of Electric Current Residual

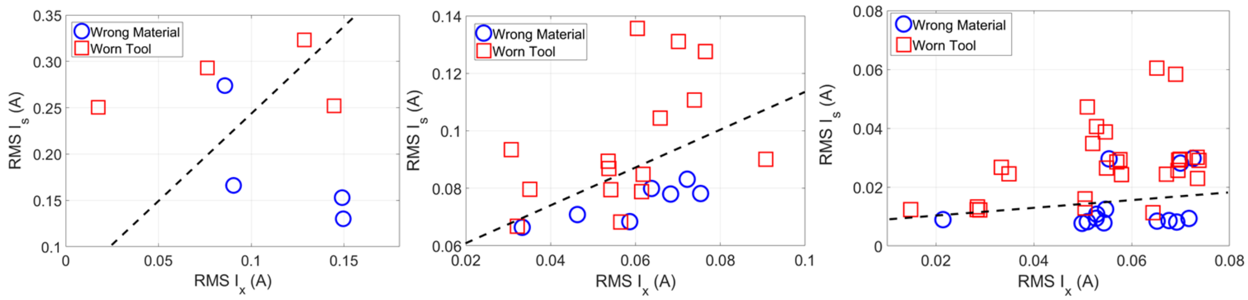
Table III.3: Residual Analysis Information

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

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.

III.4.2.2 Event Occurrence

Using historical data, we were able to identify the average and standard deviation time intervals associated with each GOS. Results showed that wrong part dimensions of -5mm on the X-axis and -0.8mm on the Z-axis caused an average delay of the machine-part interaction of 1.39 and 0.42 seconds respectively. A similar effect was observed when the part was poorly clamped causing the part shifted during the machining operation.



(a) (b) (c)

Figure III.11: Classification model for diagnosing wrong material or a worn tool: (a) features from 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%

III.4.3 Root Cause Diagnosis

In this study, classification and rule-based methods were used for root cause diagnosis.

III.4.3.1 Classification-based

Supervised learning was used to identify the root cause of residual values outside the normal thresholds. An SVM classification model 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 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. Figure III.11 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.

III.4.3.2 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. Figure III.12 shows the 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.

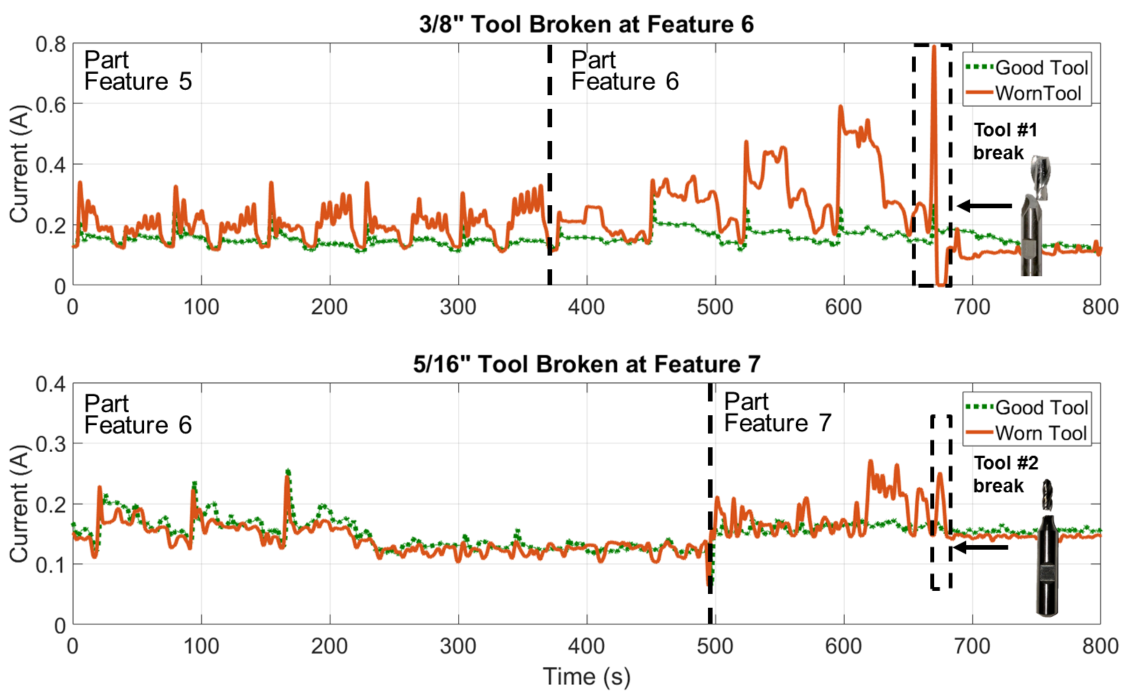


Figure III.12: Effect of Worn/Broken Tool on Spindle Current for Two Different Tool Sizes and Part Features

III.4.4 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 non-stationary 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 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.

III.5 Conclusion

In this chapter, we presented a modeling strategy to study cyber-physical manufacturing systems (CPMS) using a hybrid model. Discrete states are defined by the Global Operational States (GOS) based on implicit and explicit process descriptors. Continuous dynamics are estimated using both physics-based and data-driven models.

The main contribution of this chapter 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 in 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. A 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 enhance the detection and diagnosis of tool breakage.

The modeling approach was also implemented at an automotive assembly plant to detect backlash and monitor the time of subtasks in the manufacturing process [111]. Result show that using context information can support anomaly detection and diagnosis to prevent unexpected downtime, and productivity analysis that can lead to an increase in throughput.

CHAPTER IV

Modeling Framework to Support Decision Making on Smart Manufacturing Considering the Relationship Between Productivity, Quality, and Energy Consumption

This chapter presents research on modeling manufacturing systems that extends the state-of-the-art in decision making for control at both machine and system levels.

The economic, social, and environmental impact of the manufacturing industry motivates a constant pursuit of improvement. Considering that the manufacturing industry accounts for 32% of the total energy consumed in the United States, energy efficiency has been identified as an effective way to reduce both the environmental footprint and cost [11]. Moreover, the development and implementation of integrated energy control systems, automation, and robotics have the potential to save $6.2 billion a year on electricity charges in the U.S. manufacturing sector [37]. However, efforts to improve energy efficiency must be part of a holistic approach that considers other aspects such as productivity and quality.

The electricity charges in the industrial sector are based on consumption and demand [35]. Electricity consumption refers to the total amount of electric energy consumed over a certain period of time and it is measured in kilowatt-hours (kWh). Electricity demand represents the rate at which electric energy is consumed and it is measured in kilowatts (kW). Manufacturers are charged based on both the total energy consumed and the peak demand during a billing cycle, with the cost on demand evaluated at a much higher premium than the cost on consumption. In Chapter III we showed that energy is a continuous signal and the manufacturing operation could cause multiple peaks in the signal. Furthermore, the hybrid models developed in Chapter II can be extended to study energy consumption and demand rate based on the continuous machine dynamics.

Efforts to improve energy efficiency have led to the development of machine- and system-level models to evaluate different control strategies. At a machine-level, continuous models show that energy consumption can be divided into two parts: constant and variable consumption. The former is associated with the non-productive states and the latter with transients and productive states [144]. For the purpose of reducing variable energy consumption, previous research shows that adjusting the machine process variables can help reduce consumption during productive states [51], [12]. At a system-level, discrete models capture the consumption in different states (e.g.: processing, idle, warm-up). Energy saving strategies developed by controlling the states and events have the potential to reduce consumption by turning the machines on or off and pausing the operation [41]. However, the integration of both machine- and system-level models and control strategies has not been explored. Moreover, the performance of plant floor operations is often evaluated based on different indicators of productivity, quality, and reliability [52]. Research has shown that system-level performance is affected by the coupling between different aspects of machine-level manufacturing. In [94], the study of machine-level quality-quantity coupling indicates that increasing processing speed can have a negative effect on part quality and machine reliability. In [25], the implementation of a maintenance policy which considered the quality-reliability coupling had a positive effect on system-level productivity and part quality. In order to access the full energy-saving potential of the manufacturing industry, the trade-offs between plant floor productivity and energy consumption should be considered and balance using multi-objective optimization. Moreover, the formulation of the optimization problem depends on the type of manufacturing system and decision variables. For job shop systems, multi-objective optimization problems have been formulated for production control to define a schedule [96]. For flow shop systems, multi-objective optimization problems have been formulated to define the plant configuration [114]. However, the combination of schedule, configuration, and process variables has yet to be studied.

The development of a modeling framework that captures the variable electricity consumption and the intrinsic relationship between different performance metrics is not trivial. The goal of this chapter is to answer the following questions: 1) How can machines and systems be modeled to consider the coupling between energy consumption, productivity, quality, and reliability metrics? 2) How can control strategies of a manufacturing system be developed to consider both machine- and system-level variables?

In this chapter, we extend the hybrid modeling framework presented in Chapter II to include a machine performance model that defines the inputs of the dynamic model. The integration of both machine-level and system-level models enables the analysis of quantity-quality and quantity-energy consumption couplings. Furthermore, in the case where a fault or anomaly is detected using the approach presented in Chapter III, the model in this chapter is used to support decision making. This chapter presents two main contributions.

The *first contribution* of this chapter is a framework for modeling and simulating manufacturing systems that captures the relationship between productivity, reliability, quality, and energy consumption. The framework is based on a hybrid model of machine dynamics combining discrete states and continuous variables with manufacturing performance.

The *second contribution* is the formulation of a multi-objective optimization problem to support opportunistic decision making. The problem is formulated to evaluate both system-level configuration and machine-level operation.

The modeling and optimization framework is demonstrated in a case study based on a fully automated manufacturing testbed. The framework is used to define control actions that suggest machine-level operation and system-level configuration in a flow shop system.

The remainder of this chapter is organized as follows. Section 1 provides a background on the research area. Section 2 defines the modeling framework providing details of machine- and system-level models. Section 3 describes the multi-objective optimization method. Section 4 presents a case study to validate the approach for decision making. Finally, Section 5 summarizes the research.

IV.1 Background

In this chapter, a framework for modeling manufacturing systems is introduced. The model aims to support simulation-based optimization for decision making.

IV.1.1 Modeling Manufacturing Systems

A model is a representation of an underlying essence of a real world object based on governing equations, assumptions, and constraints. In manufacturing, models have been developed to study both machine- and system-level operations.

IV.1.1.1 Machine-Level Models

The use of models to evaluate machine operations has been approached in different ways. Dynamic models aim to study the evolution of state or output variables in response to specific inputs. Several methods have been developed to study machine dynamics. In [5], the development and implementation of dynamic models of machine tools is presented. Results demonstrate the ability of models of different complexity to estimate position and velocity in response to a sequence of commands from G-code programs. In [126], a model and simulation of industrial robots was used to evaluate the energy consumption of coordinated robot actions. Results demonstrate the advantage of combining a dynamic model with integer programming to develop energy saving operation schedules. For machines that operate in a distinct set of states, the machine dynamics have been studied as a hybrid system with discrete states and continuous dynamics [85]. However, dynamic models do not capture other performance aspects such as machine reliability or product quality.

Reliability models have been developed using historical data to estimate the probability of a machine failure [66]. In [40], probabilistic models based on Design of Experiments (DoE) were developed to study the effect of process variables on productivity and reliability. Feedrate and spindle speed proved to have an impact on failure rate and throughput. Quality models have been developed to study the effect of process variables on part quality. In [25], the ability of a machine to produce a part according to quality specifications over time was developed considering the component degradation and need for repairs. Prior work has shown that process variables have an effect on both machine dynamics and performance metrics. In order to study the effect of process variables on productivity, reliability, quality and energy consumption, there is a need to combine various machine models. However, a modeling framework merging both machine dynamics and performance has not yet been developed.

IV.1.1.2 System-Level Model

During the last few decades, several methods have been developed to analyze different aspects of manufacturing systems. A major area of interest is the analysis of configuration. Hard or physical configuration refers to the location of physical assets such as machines, robots, and material handling equipment. Soft or logical configuration refers to commands and control logic of the different machines that define the part route, schedule, and operation sequence. The configuration of the system defines the interactions between machines and has a direct effect on productivity.

The configuration of a manufacturing system has been analyzed using static and dynamic models. Static models are time-invariant and study the system under steady-state conditions. In [80], a system-level static model was developed to analyze the productivity of different configurations and machine cycle times. In [72], a methodology to evaluate productivity and reliability of reconfigurable manufacturing systems was presented. Results showed that the configuration of a manufacturing system has a direct effect on performance. Moreover, the selection of the proper configuration depends on the machine-level productivity and reliability. Dynamic models are time-dependent and study the system under both transient- and steady-state conditions. Discrete Event Systems (DES) have been used to model plant floor operations and evaluate their performance by simulating a sequence of events over time [61]. In [6], a dynamic model of a manufacturing system was developed to study the effect of control action and system configuration using a DES model. The simulation of a DES model of a manufacturing system is often used to study complex reconfiguration opportunities. However, the constraints in system-level interaction and machine-level operations were not considered. Both static and dynamic models have been used to analyze the reconfiguration of manufacturing systems, that is to modify the hard and soft configuration in response to changes in demand, product design, operating conditions [135]. However, system-level models often do not consider the effect of machine-level variables on productivity or quality and are not capable of estimating the continuous energy consumption. A modeling framework that considers both machine- and system-level operations is yet to be developed.

IV.1.2 Optimization of Manufacturing Systems

Optimization is the process of finding the value of input variables that best satisfy an objective function. Different techniques have been developed for optimization but broadly they can be classified into two: parametric (static) and control (dynamic) [46]. The selection of the proper optimization technique depends on the conditions of the problem.

In [139], a machining operation was studied as a Multiple-Input-Single-Output (MISO) system using a static optimization to find the set of process variables that improve product quality. Experimental data was used to develop a response surface to optimize the machining operation, and process variables such as feedrate and spindle speed were found to have a direct effect on product quality. However, studies of machining operations as Multiple-Input-Multiple-Output (MIMO) systems demonstrated that the set of solutions that improved product quality caused an increase in both cycle time and energy consumption [125].

At a system-level, optimization has been used to evaluate different plant floor configurations and schedules. In [75], plant floor configuration was studied using a combination of static optimization and simulation. The optimization used various performance criteria to evaluate productivity such as machine utilization, flow time, blocked time, and product lateness. Results indicate an advantage on combining simulation and optimization methods; however, other aspects that affect productivity such as production schedule were not considered. The combined analysis of plant floor configuration and production schedule drives a need to use a combination of static and dynamic optimization. In [114], genetic algorithms were combined with a DES simulation of the plant floor to evaluate different configurations, and dynamic programming was used to evaluate production schedules. However, due to the limitations in the DES model used to evaluate the configuration and schedule alternatives, the optimization focused only on productivity.

Multi-objective optimization supports the evaluation of various objective functions and helps balance the performance trade-offs. This approach is crucial when objective functions have conflicting goals. For example, in [82], the authors showed that improving productivity can increase energy consumption. Similarly, a maintenance policy which improves machine reliability has proven to reduce throughput [134]. The use of multi-objective optimization methods in manufacturing has been the focus of many researchers. In [90], manufacturing operations were studied using different performance assessment criteria. The authors describe the methodology for multi-objective optimization using normalization and weighted sums of objective functions, and the use of both static and dynamic optimization. However, the methodology focuses only on system-level variables and ignored the effect of machine-level operations on performance.

To summarize, current modeling and analysis frameworks focus mostly on isolated analysis of machine-level or system-level variables. Moreover, purely discrete models of machines neglect the continuous dynamics that impact the energy consumption. Thus, current modeling strategies limit the capability of optimization methods to improve plant floor operations.

IV.2 Modeling of Manufacturing Systems

Manufacturing systems operate by the synchronous interaction of machines and buffers to process a part. In this section, we present an approach to model manufacturing systems considering system-level interactions, machine-level operations, and part-level attributes as shown in Fig. IV.1.

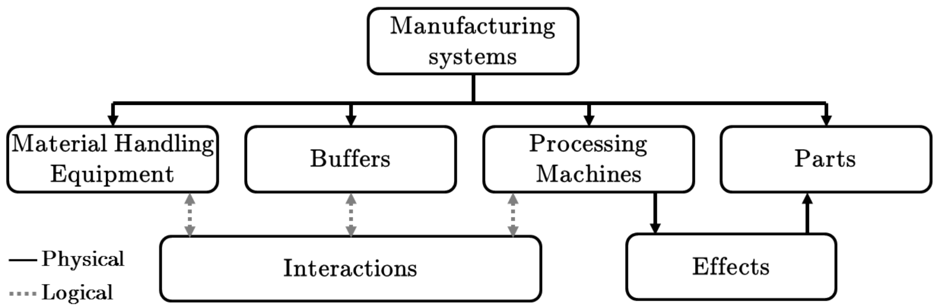


Figure IV.1: Diagram of The Elements Included In The Model of The Manufacturing System

This novel approach for developing and integrating system, machine, and part models requires three steps. First, identification of the properties, constraints, and requirements of the different elements in the plant floor. Second, formal representation of system, machines, buffers, and part. Lastly, analysis of the model to study productivity, reliability, quality, and energy consumption.

IV.2.1 Identification of Properties, Constraints, and Requirements

Prior to model development, knowledge about the different elements that operate in the plant floor and their interactions is required. This section describes the information that defines the properties, constraints, and requirements of the model and some possible sources. This information can be classified as system-, machine-, and part-level.

IV.2.1.1 System-level

The development of a system-level model requires knowledge about the configuration and machine interactions. The hard configuration determines the arrangement of machines in the factory and is often detailed on the plant floor layout. The soft configuration determines the virtual or network-based interaction between machines and can be obtained from the controller or process flow plan. The capabilities and constraints to reconfigure can be obtained from a system-level reconfiguration smoothness analysis [135].

IV.2.1.2 Machine-level

Modeling machine-level operations requires knowledge about the configuration, process variables, and dynamics. Both hard and soft configurations are often specified in the machine instruction sheet, a document that describes the manufacturing operation and process variables [77]. The machine dynamics can be identified using historical data [65].

IV.2.1.3 Part-Level

Modeling the physical part requires the knowledge of the different attributes, requirements, and constraints. Attributes are the properties that characterize the part such as dimensions and material, requirements are the specification of the different attributes such as tolerances. Both attributes and requirements are often described in the product design or drawing. Constraints are defined by the series of conditions that restrain the manufacturing of the part such as precedence are often specified in the Bill-Of-Process (BOP) or product ontology [121] [102].

IV.2.2 Model Development

Modeling a manufacturing system requires the representation of the system-level interaction, machine-level operations, and part-level attributes. In this chapter, we extend the framework presented in Chapter II.2 to introduce the machine- and system-level reconfiguration capabilities and part attributes. The model can be used to evaluate the effect of different decision or control variables on productivity, reliability, quality, and energy consumption.

IV.2.2.1 System

In this chapter, the interactions between machines and buffers from the plant layout or process flow plan in a DEVS coupled model [137] are presented. The system-level model extends the representation described in Chapter II.2 by specifying alternative configurations of the system. A DEVS coupled model of a reconfigurable manufacturing system is defined by a tuple G:

$$G = (\boldsymbol{U_d}, \boldsymbol{Y_d}, \boldsymbol{A}, EIC, EOC, IC, IC_o)$$

where

$\boldsymbol{U_d} = \{e_{i1}, e_{i2}, ...\}$: Set of discrete input events

$\boldsymbol{Y_d} = \{e_{o1}, e_{o2}, ...\}$: Set of discrete output events

$\boldsymbol{A} = \{M_1, ...B_1, ...\}$: Set of atomic models

$EIC = \{((G, e_{i1}), (M_1, e_{i1})), ...\}$: Set of external input couplings

$EOC = \{((M_m, e_{o1}), (G, e_{o1})), ...\}$: Set of external output couplings

$IC = \{((M_1, e_{o1}), (M_2, e_{i1})), ...\}$: Set of feasible internal couplings

$IC_o \in IC$: Set of currently enabled internal couplings

For modeling a reconfigurable manufacturing system we consider IC the set of feasible system-level configurations that define different process flows but do not require layout changes (e.g., soft-type reconfiguration). Figure IV.2 shows an example of a system-level model with two processing machines, one material handling robot, and two buffers.

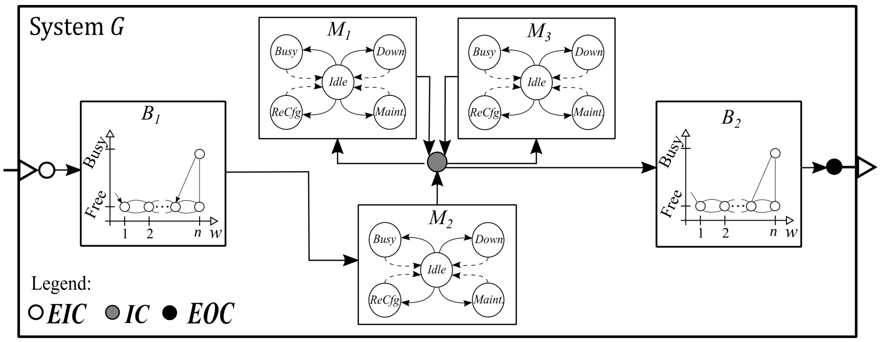


Figure IV.2: Example of a DEVS Model of a Reconfigurable Manufacturing System

Table IV.1: Representation of a Reconfigurable Manufacturing System

U	$\{job_{in}\}$	
Y	$\{job_{out}\}$	
M	$\{M_1, M_2, M_3, B_1, B_2\}$	
EIC	$\{(G, job_{in}), (M_1, job_in)\}$	
EOC	$\{((B_2, job_{out}), (G, job_{out}))\}$	
IC	$\{((M_1, job_{out}), (M_2, job_{in}),$ $((M_2, job_{out}), (M_3, job_{in}), ...\}$	M_1,M_3 series
	$\{((M_2, job_{out}), (M_1, job_{in}),$ $((M_2, job_{out}), (M_3, job_{in}, ...)$	M_1,M_3 parallel

The internal coupling (IC_o) of the system-level model defines the process flow. Based on the Internal Couplings in (IC) the system can operate in two different configurations with serial or parallel machines. The control logic in the material handling equipment (M_2) dictates if machines operate in series or parallel without the need for layout changes. Knowledge of the system-level properties, constraints, and reconfiguration smoothness is used to define the feasible internal couplings in IC.

Each configuration requires the assignment of part features or tasks to different machines [72]. Given p part features and m processing machines in the system, let $\mathcal{P}_{(i,j)}$ be a binary variable, where $\mathcal{P}_{(i,j)} = 1$ if part feature i is assigned to machine j and 0 otherwise. The assignment of part features to the machines in the system is represented by a matrix: $\mathcal{P}^{p \times m}$. The development of the part feature assignment matrix requires knowledge of the machine-level capabilities and constraints.

IV.2.2.2 Machines

Processing machines and material handling equipment are represented by two models: a dynamic model and a performance model. The dynamic model captures both the discrete and continuous behavior of the machine. The performance model captures the relationship between process and maintenance variables.

Dynamic Model In order to estimate both the energy consumption, which is a continuous signal, and productivity, which depends on discrete states and transitions, we model the machine dynamics as a hybrid system using the Hybrid Discrete Event System Specification (HDEVS) formalism [85]. The hybrid model aims to capture both the discrete and continuous states. An HDEVS atomic model with inputs, outputs, states, and transition functions is defined by a tuple M:

$M = (U_d, U_c, Y_d, Y_c, X_d, X_c, \delta_t, \delta_e, F, H, \lambda, t_{adv})$ where

$U_d = \{e_{i1}, e_{i2}, ...\}$: Set of discrete input events

$U_c = \{u_1, u_2, ...\}$: Set of continuous input variables

$Y_d = \{e_{o1}, e_{o2}\}$: Set of discrete output events

$Y_c = \{y_1, ..., y_m\}$: Set of continuous output variables

$X_d = \{s_1, s_2, ...\}$: Set of discrete states

$X_c = \{x_1, ..., x_n\}$: Set of continuous state variables

$\delta_t : X_d \times \{t_{adv}, \emptyset\} \rightarrow X_d$: Time-driven transition function

$\delta_e : X_d \times U_d \rightarrow X_d$: Event-driven transition function

$F : X_d \times X_c \times U_c \rightarrow \mathbb{R}^n$: Continuous state function

$H : X_d \times X_c \times U_c \rightarrow \mathbb{R}^m$: Continuous output function

$\lambda : \{\delta_t, \delta_e\} \rightarrow Y_d$: Discrete output function

$t_{adv} = \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, ...\}$: Set of transition times

The model considers discrete inputs $U_d = \Sigma_{cn} \bigcup \Sigma_{uc}$ that contain controllable (Σ_{cn}) and uncontrollable (Σ_{uc}) events. Controllable events are forced by the system-level controller. Uncontrollable events can occur at any time and are detected by the machine-level controller.

Machine productivity and reliability are affected by the discrete states (X_d) and transition times (t_{adv}). Energy consumption is described by the continuous outputs (Y_c). The discrete states can be defined based on the control logic and can be classified as productive and non-productive states. The transition times can be deterministic or stochastic variables and their value is determined by the performance model. The energy consumption is described by electric current (I) and voltage (V) studied as continuous output variables such that $Y_c = \{I, V\}$. The energy consumption analysis considers the relationship between the discrete states (X_d), the continuous input variables (U_c), and the continuous state variables (X_c).

Modeling machines requires the definition of possible states, the set of controllable and uncontrollable events, transition times, transition functions, and input and output variables. However, the development of a hybrid model of all the machines in the system may not be feasible. Some machines can be represented in lesser detail by modeling only their discrete behavior with a reduced number of states [81]. For example, a machine could be modeled with only two states $(Busy, Idle)$, and the transition times (τ_1, τ_2) can be defined by a probabilistic model using some parametric distribution such as Gaussian or exponential [94].

An example of an atomic model for a CNC machine with five possible states is shown in Fig.

IV.3. The productive state is $\{Processing\}$ and the non-productive states are $\{Idle, Reconfigure, Down, Maintenance\}$. The controllable events are $\{job_{in}, change, service\}$ and the uncontrollable event is $\{fault\}$. The continuous inputs are the table position in XYZ coordinates $U_c = \{p_x, p_y, p_z\}$, and the continuous outputs are the current and voltage $Y_c = \{I, V\}$. The representation of machine M is summarized in Table IV.2

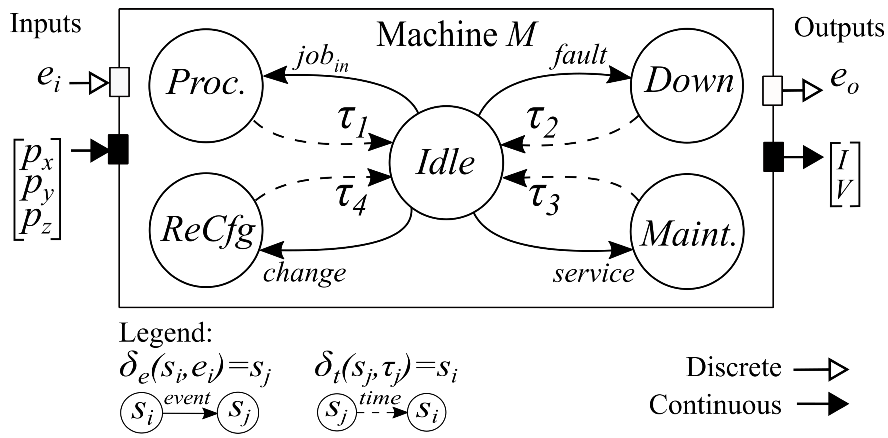


Figure IV.3: Example of Machine-level Model

Table IV.2: Representation of Machine-level Model

U_d	$\{job_{in}, fault, service, change\}$
U_c	$\{p_x, p_y, p_z\}$
Y_d	$\{job_{out}, ready\}$
Y_c	$\{I, V\}$
X_d	$\{Proc., Down, Maint., Reconfigure, Idle\}$
X_c	$\{q_x, \dot{q}_x, ...\}$
t_{adv}	$\{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \boldsymbol{\tau}_3, \boldsymbol{\tau}_4\}$

The transitions between discrete states can be time- or event-driven. For example, an event-driven transition caused by a part arrival is defined by $\delta_e(Idle, job_{in}) = Proc.$ The completion of the cycle time triggers a time-driven transition $\delta_t(Proc., \tau_1) = Idle$. The evolution of state variables while in a processing state is described by $\dot{q}_x = f(Proc., q_x, p_x, t)$ and the current consumption is described by $I = h(Proc., q_x, \dot{q}_x, t)$.

Performance Model In this section we define sets of process variables, maintenance tasks, and reconfiguration smoothness that determine the transition times and continuous inputs of the dynamic model. $\boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_i\}$ is the set of process variables such as feedrate on a CNC machine or moving speed on a robot arm. $\boldsymbol{\beta} = \{\beta_1, ..., \beta_j\}$ is the set of preventive maintenance tasks given the time in operation or number of parts processed, such as replacing cutting fluid after 5000 parts. $\boldsymbol{\gamma}$ is the reconfiguration smoothness [135]. The performance model estimates the transition times based on pre-defined functions developed using experimental analysis, process simulation, or expert knowledge. The effect of process variables on processing time has been studied using Design of Experiments (DoE) [51] [125] and Computer-Aided Manufacturing (CAM) software [65]. Service or repair time throughout the machine lifecycle has been determined experimentally by monitoring component-level degradation [66]. The reconfiguration time has been defined using expert knowledge based on a reconfiguration smoothness analysis [18].

For the example of a CNC machine model in Fig. IV.3, the processing time is a function of α_1: feedrate so that $\boldsymbol{\tau}_1 = f_p(\alpha_1)$. Service time is a function of β_1: replace cutting fluid such that $\boldsymbol{\tau}_2 = f_m(\beta_1)$. The reconfiguration time $\boldsymbol{\tau}_3$ is defined by an expert based on the number of changes required and the complexity of the changes. The process variables also affect the continuous inputs. For example, the feedrate in a CNC machine will affect the table position commands $U_c = \{p_x, p_y, p_z\}$ and influence the outputs $Y_c = \{I, V\}$. The effect of process variables on continuous inputs for different machines can be obtained using process simulation. The integration of dynamic and performance models aims to capture the effect of process variables, maintenance tasks, and reconfiguration smoothness on productivity, reliability, and energy consumption.

IV.2.2.3 Buffer

Buffers are represented as a DEVS atomic model based on a discrete set of inputs, outputs, states, transitions, and occupancy. A buffer is defined by a tuple B

$$B = (U_d, Y_d, X_d, w, \delta_e)$$

where

$U_d = \{e_{i1}, e_{i2}, ...\}$: Set of discrete input events

$Y_d = \{e_{o1}, e_{o2}, ...\}$: Set of discrete output events

$X_d = \{s_1, s_2, ...\}$: Set of discrete states

$w \in \mathbb{Z}^+$: Buffer occupancy

δ_e:$X_d \times U_d \rightarrow X_d$: Event-driven transition function

An example of a buffer B with capacity 10 is shown in Fig. IV.4. The discrete inputs and outputs are $U_d = \{job_{in}\}$ and $Y_d = \{job_{out}\}$ respectively. The buffer states are $X_d = \{Busy, Free\}$. Occupancy (w) defines the number of parts in the buffer, and δ_e is the transition function.

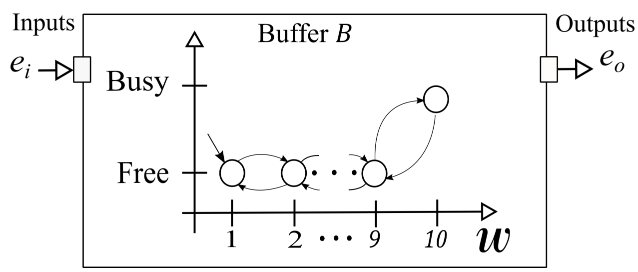


Figure IV.4: Example of a Buffer Represented as a DEVS Atomic model

IV.2.2.4 Part

In this chapter, parts are represented as entities with a series of attributes such as the part processing flow, processing time, features, and quality conditions. Entities are created using a DEVS entity generator [137] and are defined by a tuple O:

$$O = (L, L_o, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{\Psi},)$$

where

$L = \{M_1, M_2, B_1, ...\}$: Set of possible part locations

L_o: The current location of the part

$\boldsymbol{P} = \{P_1, P_2, ...\}$: Set of part features

$\boldsymbol{Q} = \{Q_1, Q_2, ...\}$: Set of quality conditions

$\boldsymbol{\Psi} = \{\tau_{1,1}, \tau_{1,2}...\}$: Set of processing times for each machine in the location set.

The value of each element in $\boldsymbol{Q}$ is a function of the machine process variables (α). The relationship between product quality and process variable has been studied as a quality-quantity coupling [7]. A function for each quality condition can be defined as $Q = f_Q(\boldsymbol{\alpha})$. The function f_Q can be obtained experimentally using DoE or historical data.

Part quality is evaluated based on the deviation from a mean specification of a quality attribute (e.g.: tolerance, surface roughness) so that $0 < Q \leq 1$. For example, Fig. IV.5 shows a part with four features. Each feature is processed at a different machine in the system and has a set of defined quality conditions. The attributes of O after being processed by machines M_1 and M_3 are shown in Table IV.3. As defined in Chapter II.2.1, the values in Ψ are the processing times at M_1 to M_3. The values in Q are defined by the deviation of each dimension from the required part specification.

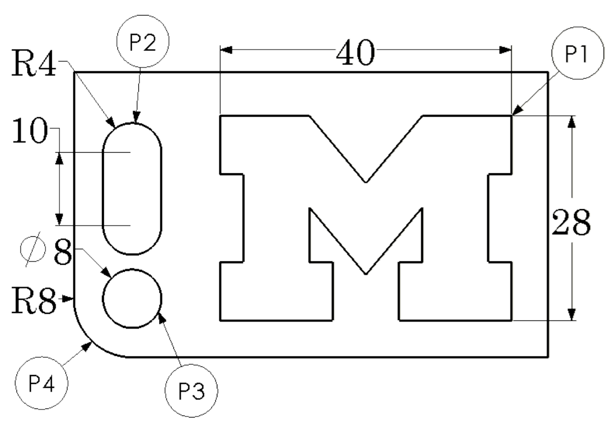


Figure IV.5: Example of a Part

Table IV.3: Attributes of Part Model O

L	$\{M_1, M_2, M_3\}$
L_o	B_2
$\boldsymbol{P}$	"*Logo*","*Slot*","*Hole*","*Fillet*"
$\boldsymbol{Q}$	[0.98,0.92,0.78,0.74,0.95,0.83]
$\boldsymbol{T}$	[148sec,12sec,109sec]

IV.2.3 Model Analysis

The modeling framework presented in this chapter aims to capture the effects of both machine- and system-level variables on productivity, reliability, quality, and energy consumption to support decision making. For example, in the case of a tool breakage as shown in Chapter III, the model can be used to evaluate the effect of a machine repair or system reconfiguration action along with changes in process variables.

Productivity and Reliability At the machine level, the discrete states (X_d), transition functions (Δ), and sequence of controllable events (E_{cn}) define the productivity and reliability. The integration of the dynamic and performance models is used to evaluate the effect of process variables, maintenance task, and reconfiguration smoothness ($\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$) over the transition times in t_{adv}. At the system level, the interactions represented by the internal coupling between machines and buffers in IC_o define the process flow and have a direct effect on productivity.

Quality A condition of the part estimated by a quality function f_Q defined by the quality-quantity and quality-reliability coupling of the machine. Quality conditions are estimated by the performance model based on a set of process variables $\boldsymbol{\alpha}$. The value for each element in $\boldsymbol{Q}$ is assigned to the part by processing machines as the part flows through the system.

Energy Consumption Described by the continuous output of the machine dynamic model. In order to estimate both electricity total consumption and demand rate, the current I and voltage

V are calculated based on continuous inputs in U_c. The sequence of controllable events in E_{cn}, continuous inputs X_c and process variables $\boldsymbol{\alpha}$ in the machine-level model affect the energy consumption.

Studying both machine- and system-level variables and their effect on plant floor performance in an integrated virtual environment can help improve the decision making process. The effect of different control actions and machine setups can be evaluated using simulation-based optimization or "what-if?" analyses.

IV.3 Multi-Objective Optimization

In this chapter we aim to optimize productivity, reliability, quality, and energy consumption. Considering that optimization in large-scale and stochastic scenarios is a complex problem, simulation-based optimization is utilized. The model developed in this chapter is used to evaluate the effect of machine-level events and processes variables, and system-level interactions on different objective functions. The simulation-based optimization can be used to support decision making or control action in response to under performance and anomalies detected using the approach presented in Chapter II and Chapter III respectively. The multi-objective decision making problem can be solved using a combination of optimization algorithms and simulation.

The simulation-based optimization approach to support decision making requires four steps. First, definition of the decision or control variables identified during the modeling the model development in Section IV.2.2. Second, formulation of the objective functions based on the performance requirements. Third, specification of machine- and system-level constraints identified in section IV.2.3. Fourth, selection of the optimization algorithm.

IV.3.1 Definition of the Control Variables

The decision variables define the system configuration, the machine operation, and the process variables. A solution is said to be feasible when the constraints are satisfied.

• System Configuration: The configurations of the manufacturing system are defined by the Internal Coupling IC. Considering k possible configurations, the set of possible configurations is defined by $S_{c1} \in \{IC_1, ..., IC_k\}$. The assignment of part features to machines in the system is represented in $\mathcal{P}^{p\times m}$. The set of possible part feature assignment is defined by $S_{c2} \in \{\mathcal{P}_1, ..., \mathcal{P}_q\}$

• Machine Operation Sequence: The sequence of events that forces the transitions to productive or non-productive states is defined by a string E. For m machines in the system, and s possible combination of events in a string, the set of strings of controllable events is defined by $S_e^j \in \{E_1^j, ..., E_s^j\}, \forall j = \{1, ..., m\}$

• Machine process variables: The operation of each machine depends on a set of process variables $\boldsymbol{\alpha}$. For m machines in the system and z possible combinations of process variables, the set of process variables is defined by $S_p^j \in \{\boldsymbol{\alpha}_1^j, ..., \boldsymbol{\alpha}_z^j\}, \forall j = \{1, ..., m\}$

The combination of system configuration, machine operation, and process variables is denoted by the set of solutions $\boldsymbol{S}$.

$$\boldsymbol{S} = \{S_{c1}, S_{c2}, S_e^j, S_p^j\}, \forall j = \{1, ..., m\}$$

Considering that evaluation of all possible combinations of solutions can be time consuming or unfeasible, the search space can be reduced by evaluating only a limited number of configuration and feature assignments [72]. Some guidelines for the reduction of the search space based on the machine- and system-level capabilities and constraints are:

Configuration The set of possible system configurations and part feature assignments can be reduced by evaluating the process flow and constraints, and cycle time balance. A process flow analysis defines the feasible interactions based on the plant layout and precedence constraints.

Moreover, the types of tools available in a machine and manufacturing operation capabilities might restrict the set of feasible part feature assignments. A cycle time balance analysis defines the part feature assignment in order to minimize the difference in cycle time throughout the machines [16].

Process variables The set of feasible process variables is often constrained by the type of manufacturing operation, material, and machine capabilities. For example, the feedrate and spindle speed of a machining operation is constrained by the load on the cutting edge of the tool. The ranges of feasible process variables for different materials, tool size, and operating condition (e.g., non-coolant or coolant fed) are often specified by the machine or tool manufacturer.

However, the reduction of the search space can add uncertainty regarding the optimality of the solution. Moreover, the evaluation of flow and process constraints may pose a scalability challenge when studying large manufacturing systems.

IV.3.2 Objective Functions

The formulation of an optimization problem aims to select the solution set that achieves the Pareto optimal solution to productivity, reliability, quality, and energy consumption functions. The objective functions are defined as follows:

IV.3.2.1 Productivity

Productivity can be studied from different perspectives such as resource utilization and product delivery. Furthermore, from a product delivery perspective, multiple indicators can be used for performance assessment. As described in Chapter II, some of the indicators that can be used to define an objective function are throughout, flow time, and lateness. The normalization of multi-criteria objective functions is described in [90] and expressed as $f'(.)$. For example, a productivity objective function to improve f_{Prod_1}: throughput and f_{Prod_2}: flow time can be defined by:

$$\text{maximize } f_{Prod}(\boldsymbol{S}) = f'_{Prod_1}(\boldsymbol{S}) - f'_{Prod_2}(\boldsymbol{S})$$

IV.3.2.2 Reliability

Reliability is a characteristic of the system that describes how effectively the machines operate over a time horizon. For each machine in the system, reliability R is calculated as the ratio between repair (τ_2) and service (τ_1) time and total time T in minutes or hours [90]. For example, in a system with m machines a reliability objective function can be defined as:

$$\text{maximize } f_{Rel}(\boldsymbol{S}) = \prod_{n=i}^{m} R_i, \quad \text{where } R_i = \tfrac{T-\tau_1-\tau_2}{T} \quad \text{for } i = \{1, ..., m\}$$

IV.3.2.3 Quality

Quality can be measured from two different perspectives: customer satisfaction and product specification compliance. In this chapter, we focus on product compliance as defined by the minimum lower bounds of quality conditions for the different part features. Given a lower bound for each quality condition ($\underline{Q}$) and N parts produced by the system, quality is evaluated as the number of parts with all the features above the lower bound.

$$Q_O = \begin{cases} 1, & \text{if } Q_j \geq \underline{Q}_j \forall i \in \{1, ..., p\} \\ 0, & \text{otherwise} \end{cases}$$

For example, for N parts produced by the system in a time interval (0,T], a quality objective function can be defined based on yield, the ratio between good and total number of parts produced, so that:

$$\text{maximize } f_{Qual}((\boldsymbol{S}) = \tfrac{1}{N} \sum_{n=1}^{N} Q_{O,n}$$

IV.3.2.4 Energy Consumption

The objective function to minimize energy consumption focuses on reducing both total consumption and demand rate.

• Total consumption: The total amount of electrical energy consumed over a time period (0,T] measured in KWh. For current (I) in amps and Voltage (V) in volts estimated as a discrete time-step (k), and T in hours, the total consumption function is defined as: $f_{e_{total}}(\boldsymbol{S}) = \sum_{k=0}^{T} I(k)V(k)/1000$

• Demand rate: The peak of energy consumption over a time window defined as:
$f_{e_{rate}}(\boldsymbol{S}) = Max\{I(k)V(k)/1000\}$

The electricity demand rate is charged at a higher premium than total consumption according to a load factor w_{e_r}. Considering both total consumption and demand, the energy consumption objective function is formulated as:

$$\text{minimize } f_{Energy}(\boldsymbol{S}) = f_{e_{total}}(\boldsymbol{S}) + w_{e_r} f_{e_{rate}}(\boldsymbol{S})$$

IV.3.3 Constraints

Define the conditions that the decision variables and simulation must satisfy.

Unidirectionality In a manufacturing system with unidirectional flow, parts can only be processed in a "one-way" sequence. The unidirectionality constraint defines that a machine or buffer output can only send parts downstream so: $IC_o = \{(A_i, e_o), (e_i, A_j)\}$ for $i < j$ and $A \in \{M, B\}$

Completeness The part must be processed completely before leaving the system. The completeness constraint for part feature assignment is formulated as: $\sum_{i=1}^{n_p} \mathcal{P}_{ij} = |\boldsymbol{P}|, \mathcal{P}_{ij} \in \{0, 1\}$

Precedence The part must be processed in a specific order. Immediate precedence constraints are defined by $C^{Pn}_{Pi,Pj}$ so feature P_n has to be processed after P_i and before P_j

Time bound The simulation outputs are evaluated over a finite time horizon. The discrete and continuous inputs are generated over timespan $t \in (0, T]$

Discrete Process Variables To reduce the search space, the process variables α for all the machines are restricted to a discrete set of values.

IV.3.4 Optimization Algorithms

The optimization problem formulated in this section requires the analysis of system-level configurations, the sequence of control actions, and machine parameters. To evaluate the effect of control or decision variables on performance, we combine the simulation and optimization algorithms. The optimization problem is solved using exhaustive search. This algorithm has shown to yield good results in multi-objective optimization but poses scalability challenges. However, other algorithms such as particle swarm, ant colony, and genetic algorithms could be used to solve the optimization problem.

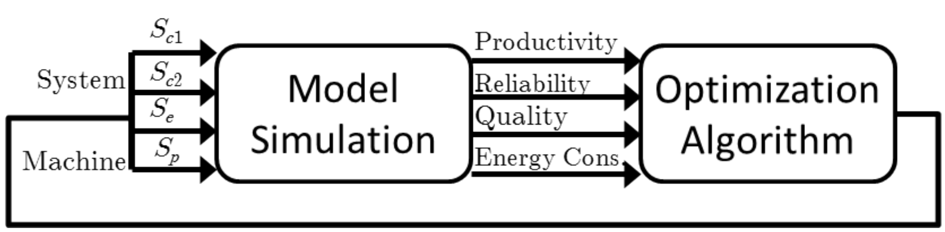


Figure IV.6: Simulation-Based Optimization Framework

IV.4 Implementation and Evaluation

To validate our proposed approach, we developed a model of a manufacturing testbed at the University of Michigan to study the system-level interactions and machine-level operations [73]. Experimental data was collected from the manufacturing process of a test part with multiple features. Performance was evaluated based on productivity, quality, reliability, and energy consumption. Multi-objective optimization was used to evaluate the trade-offs between these metrics.

IV.4.1 Modeling

The testbed was modeled in the Matlab environment using the framework described in Section IV.3. The system-level model was developed using SimEvents to represent the DEVS model. Machine-level models were developed using Simulink and StateFlow to represent the machine dynamics. Matlab was used to calculate the machine performance.

IV.4.1.1 System

The system-level model captures the machine interactions and reconfiguration capabilities. The testbed can produce parts on different configurations of serial or parallel machines without changing the layout. The configuration is determined by the pick-and-place operation of the robot that defines the interaction between CNC machines.

Input/Outputs The system-level input and output events are $\boldsymbol{U_d} = \{job_{in}\}$ and $\boldsymbol{Y_d} = \{job_{out}\}$ to represent part arrival and part departure, respectively.

Atomic Models The machines in the system are represented as atomic models using State Flow and SimEvents. Atomic models were developed for CNC machines (M), Robots (R), Conveyor (C), and Buffers (B). The set of atomic models is: $\boldsymbol{A} = \{M_1, ..., M_4, R_1, R_2, C, B_1, B_2\}$.

Internal Couplings The interactions between machines were represented as routing switches that enabled changes in configurations (e.g., parallel or serial). The current internal coupling IC_0 was defined by sending a command signal to the routing switch.

The model of the testbed with two cells capable of operating in series or parallel configurations is shown in Fig. IV.7

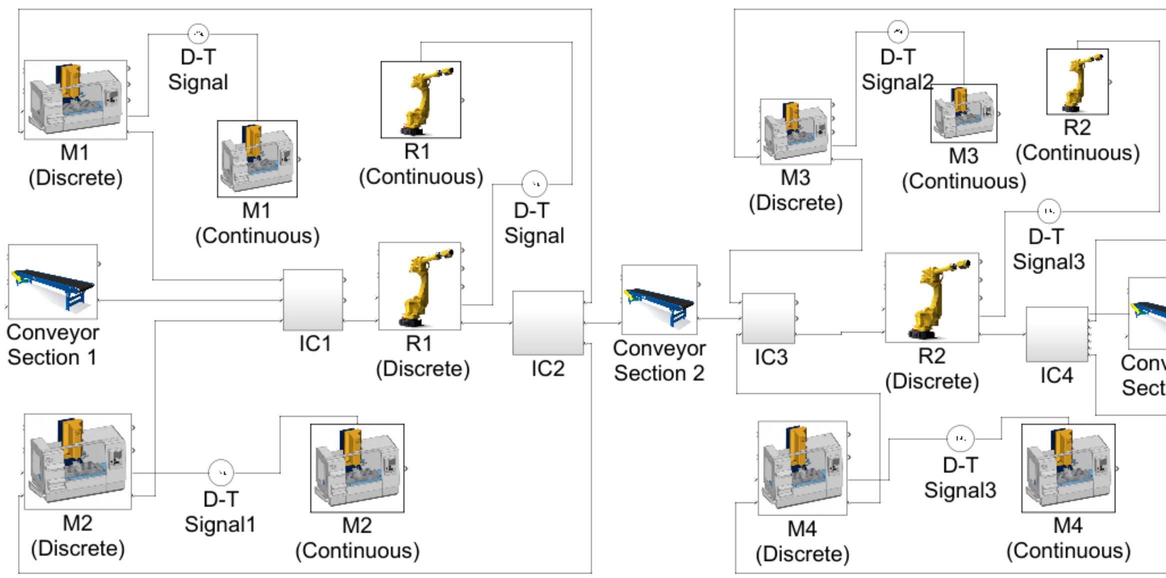


Figure IV.7: System-Level Model of Manufacturing Testbed

IV.4.1.2 Machines

The machine-level models aim to capture the machine dynamics and performance as described in Section IV.3. The hybrid model was developed using StateFlow and Simulink. The performance model was developed using Matlab functions.

CNC Machines Perform the machining operation. The models were developed based on knowledge of the controller, reconfiguration capabilities, historical data, and process simulation.

- Dynamic Model: Captures the discrete behavior and continuous dynamics of CNC machines.

The discrete states as described in Section IV.3 were built in StateFlow. The continuous dynamics were modeled using historical data to define a Simulink function.

$U_d = \{job_{in}, service, fault, change\}$

$U_c = \{p_x, p_y, p_z, v_s\}$

$Y_d = \{job_{out}\}$

$Y_c = \{I, V\}$

$X_d = \{Proc., Idle, Down, Maint., Idle, ReCfg\}$

$X_c = \{q_x, \dot{q}_x, q_y, \dot{q}_y, q_z, \dot{q}_z, \dot{q}_s\}$

• Performance Model: Estimate the effect of process variables ($\boldsymbol{\alpha}$), maintenance requirements $\boldsymbol{\beta}$, and reconfiguration smoothness $\boldsymbol{\gamma}$ on discrete transitions times and continuous inputs of the dynamic model.

$\boldsymbol{\alpha} = \{Feedrate, TravelSpeed, SpindleSpeed\}$

$\boldsymbol{\beta} = \{ReplaceTool\}$

$\boldsymbol{\gamma} = \{SoftReconfigurationSmoothness\}$

The effect of these variables over transition times was studied using simulation, historical data, and expert knowledge. Cycle time (τ_1) is a function of $\boldsymbol{\alpha}$ developed using process simulation. Repair time (τ_2) is a Gaussian random variable with parameters identified using historical data. Service time (τ_3) is a function of the maintenance variable $\boldsymbol{\beta}$. Reconfiguration time (τ_4) is a function of reconfiguration smoothness defined using expert knowledge.

The integration of dynamic and performance models enabled the analysis of quantity-energy and quantity-reliability coupling. The Quantity-Energy coupling is studied based on the effect of process variables on cycle time and energy consumption. Quantity-Reliability coupling defines the effect of process variables on the occurrence of a failure event. Moreover, the machine model aims

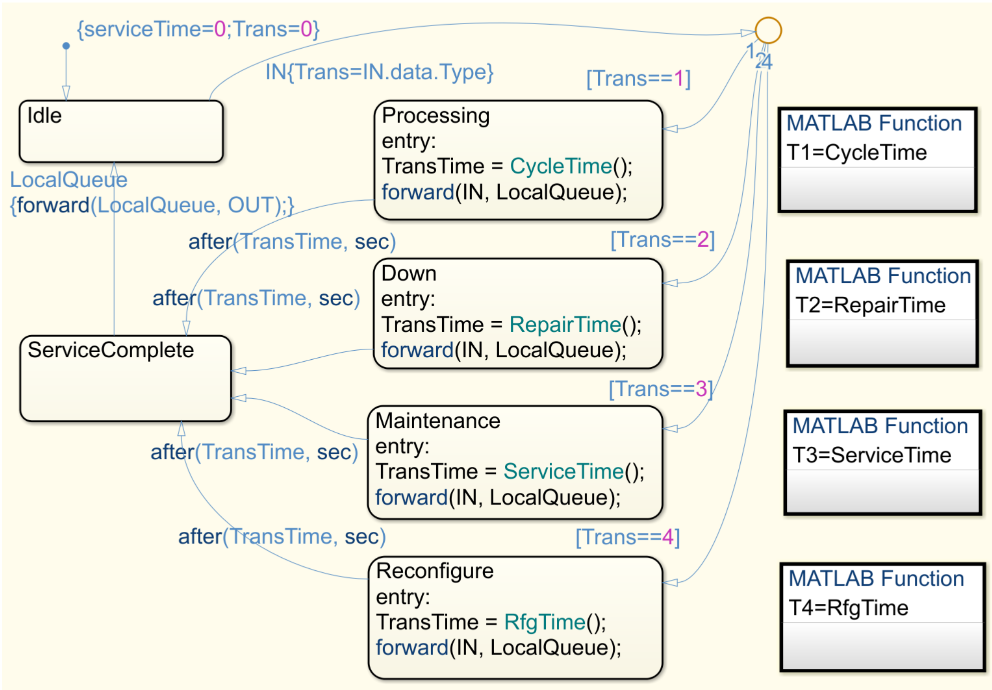


Figure IV.8: CNC Machine Model

to evaluate the time to reconfigure. The hard configuration of the machine is defined by the tool installed in the machine. The soft configuration is defined by the G-code program installed in the machine.

Robots Perform the pick-and-place operation. Robots take the parts from the conveyor and place them in the CNC machines and vice versa. The models were developed based on knowledge of the controller, and robot kinematics and dynamics.

• Dynamic Model: Developed using StateFlow to capture the discrete states, Matlab to study the robot kinematics, and Simulink to evaluate the machine dynamics and estimate energy consumption. The integration of these three software packages in a single environment supports the analysis of discrete behavior and continuous dynamics.

$U_d = \{job_{in}\}$

$U_c = \{p_x, p_y, p_z, o_x, o_y, o_z\}$

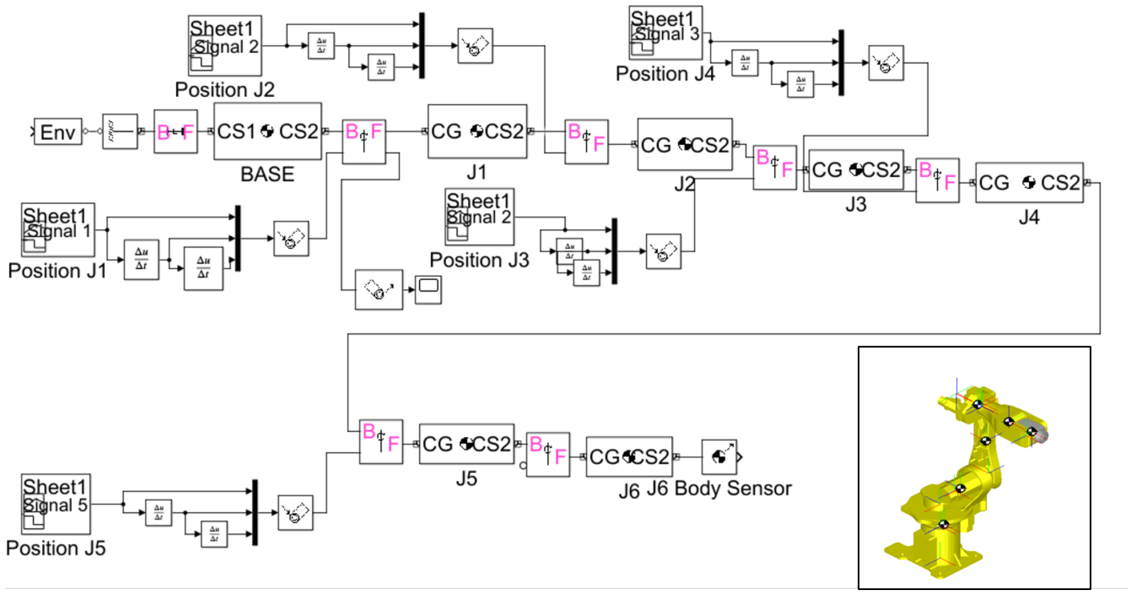


Figure IV.9: Dynamic Model of the 6 Degree-of-Freedom Robot

$Y_d = \{job_{out}\}$

$Y_c = \{I, V\}$

$X_d = \{Proc., Idle\}$

$X_c = \{q_1, ...q_6, \dot{q}_1, ..., \dot{q}_6\}$

• Performance Model: Estimate the effect of the point-to-point moving speed of the robot in the world coordinate frame over the continuous outputs and outputs. The performance model leverages process level simulation of the robot operation.

$\alpha = \{MovingSpeed\}$

The effect of moving speed over cycle time (τ_1) and input variables (U_c) was studied using a simulation of the different pick and place operations.

The integration of dynamic and performance models enabled the analysis of quantity-energy coupling. The robot model used to estimate the effect moving speed over cycle time is shown in Fig. IV.9

IV.4.1.3 Part

The case study focused on a part with 10 features manufactured using two different tools. Figure 4 shows the sample part. Part features P1 to P6 require a 3/8" diameter tool (Tool#1) and P7 to P10 require a 5/16" diameter tool (Tool#2).

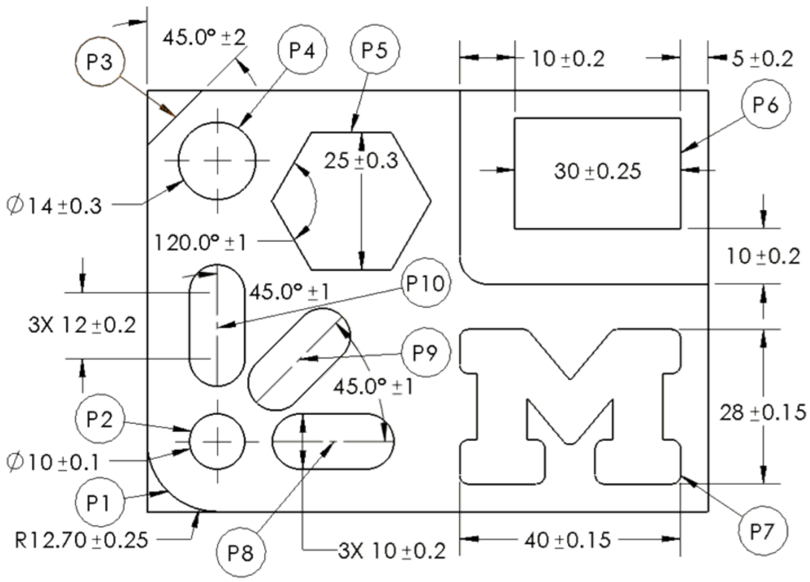


Figure IV.10: Case Study Sample Part

Features Manufacturing of the sample part requires specification of feature clusters for each machine. A part feature to machine assignment matrix $\mathcal{P}$ for a serial configuration of the system is:

$$
\begin{array}{c c}
\begin{array}{cccccccccc} P1 & P2 & P3 & P4 & P5 & P6 & P7 & P8 & P9 & P10 \end{array} & \\
\begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{pmatrix} & \begin{array}{c} M1 \\ M2 \\ M3 \\ M4 \end{array}
\end{array}
$$

Quality The quality condition of each feature was assigned by the CNC machines as the parts are processed in the system. The value of the quality condition is assigned by a function of feedrate and spindle speed $f_Q(\alpha_1, \alpha_2)$. The function can be obtained experimentally using DoE. Studying the effect of process variables over productivity and part quality enabled the analysis of quality-quantity couplings.

IV.4.2 Performance Analysis

In this case study, performance based on productivity, quality, reliability, and energy consumption metrics is evaluated. The effect of different control actions over performance is estimated using a simulation of the model.

- Productivity (f_{Prod}): Defined as the number of parts delivered by the system over a finite time horizon.

- Quality (f_{Qrod}): Evaluated based on the ratio between parts that meet minimum quality conditions for all the part features and the total number of parts delivered.

- Reliability (f_{Rel}): Calculated by the product of reliability for all the machines in the system.

- Energy consumption (f_{Energy}): Estimated considering the total consumption and demand rate of the machines and robots in the system.

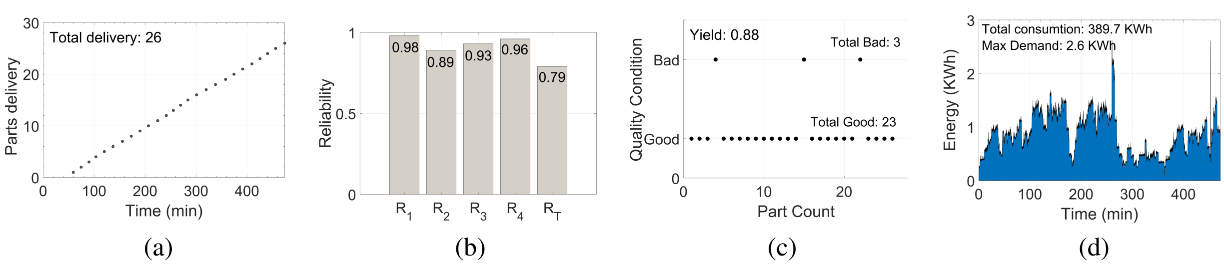


Figure IV.11: Simulated performance expectations over 8 hour period: (a) Part delivery over time (b) Reliability of each machine (c) Quality condition for each part (d) Energy Consumption and demand

A summary of the simulated performance of the system for processing the part over an 8 hour period is shown in Fig. IV.11. The machine-level models used to estimate productivity and energy consumption were developed using experimental data from the real machines and robots. The parametric distributions to define the failure and repair rate of the CNC machines were obtained from the data reported in [66].

IV.4.3 Optimization

The formulation of the optimization problem in this case study aims to support opportunistic decision making after the occurrence of a fault. In response to a machine fault, the model developed in the previous Section was used to evaluate reconfiguration, operation sequence, and process variables alternatives.

IV.4.3.1 Control Variables

The optimization algorithm defines the combination of feasible solutions to be evaluated by the simulation. The control or decision variables are:

Configuration The system is currently configured as four machines in series. In the case of a fault in one of the machines, the system can be reconfigured. The alternative configurations to operate as three machines in series, or two machines in parallel and one in series, and the tool installed at each machine are shown in Fig. IV.12. The reconfiguration analysis also considers the change in part feature assignment at each machine.

Sequence In response to a fault, the optimization algorithm evaluates a sequence of events as defined by the string E_{cn}. The events in E_{cn} control the transitions that dictates if the machines continue processing parts, are repaired, or reconfigured.

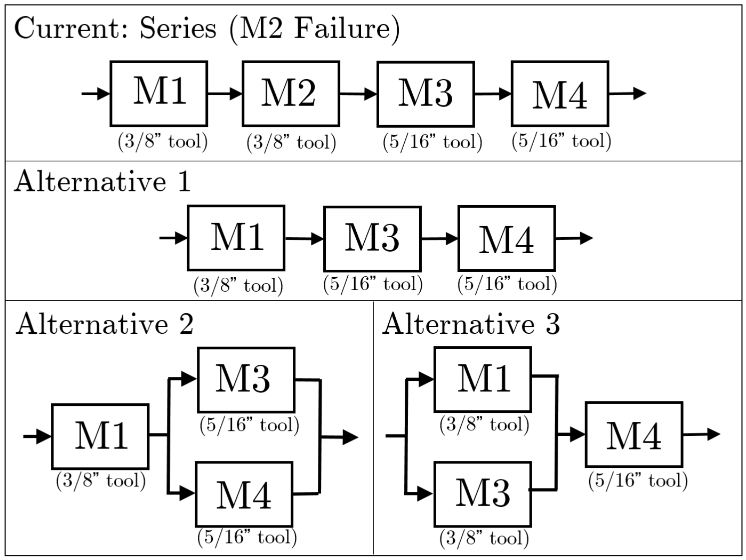


Figure IV.12: Configuration Alternatives

Process Variables The process variables that define the machine operation. For the CNC machines the discrete set of process variables are: α_1: Feedrate (mm/sec) $\alpha_1 \in \{1.5, 1.8, 2\}$, α_2: Spindle Speed (RPM) $\alpha_2 \in \{3000, 3200\}$, α_3: Travel speed (mm/sec) $\alpha_3 \in \{45, 50, 55\}$. For the robots, the process variable is α_4: moving speed (mm/sec) $\alpha_4 \in \{50, 100, 150\}$

IV.4.3.2 Objective Functions

The objective of the optimization is to evaluate the trade-offs between productivity (f_{Prof}), reliability (f_{Rel}), quality (f_{Qual}) and energy (f_{Energy}). The multi-objective optimization problem is formulated as:

$$Minimize(-f_{Prod}, -f_{Rel}, -f_{Qual}, f_{Energy})$$

IV.4.3.3 Results

In this case study, we simulated the manufacturing operation of the system over a 16 hour period. A fault was simulated in machine 2 after 7 hours of operation. The different combinations of reconfiguration, event sequence, and process variables were evaluated in order to define a control action. An exhaustive search was used to evaluate the different solution sets considering the performance trade-offs. Results show that the control action depends on the time horizon and the weight on the performance criteria. If the reconfiguration time is less than the repair time, the configuration alternative 3 in combination with an increase in feedrate resulted in higher part delivery in the short term. However for a longer time horizon, results indicate that repairing machine 2 in order to continue operating as four machines in series achieves higher part delivery. Figure IV.13 shows the part delivery over time for the different alternatives and the no-fault condition.

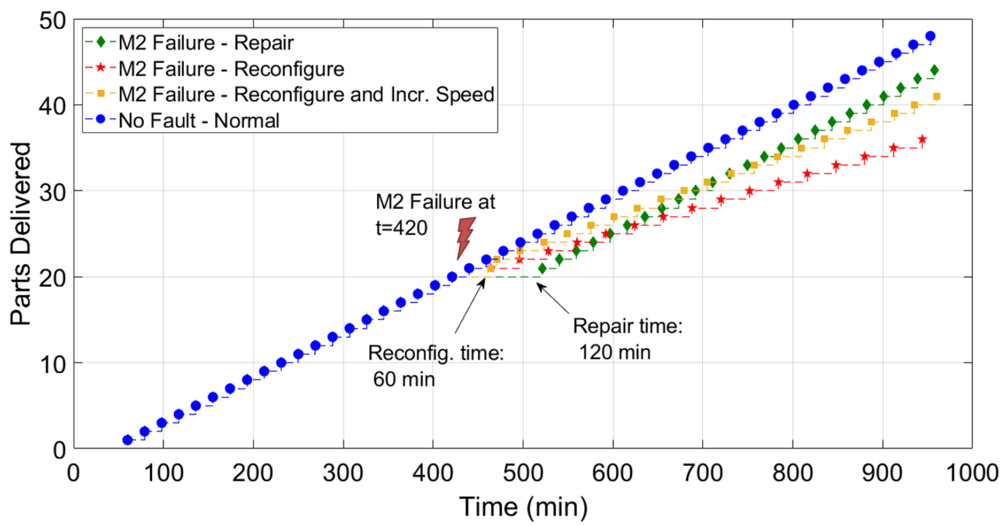


Figure IV.13: Effect of Control Action on Part Delivery Over Time

The control decision also had an effect on quality and energy consumption. The reconfiguration of the system and increase of CNC feedrate and the robot moving speed also increased the energy consumption (aggregation of total consumption and demand rate) and reduced the quality yield. This can be explained because increasing the speed of the other three machines to compensate for machine 2 being down increased the electricity demand rate and reduced part quality.

The summary of the effect of the different control actions on the performance assessment criteria is shown in Table IV.4. For this case study, a repair action resulted in higher productivity, quality, and less energy consumption at the end of the 16 hours when compared to reconfiguration to alternative 3 along with an increase in the speed of robots and CNCs. Moreover, for both reconfiguration scenarios, reliability drops due to machine 2 being down for 9 of the 16 hours.

Table IV.4: Performance Summary Table

Decision	Productivity	Reliability	Quality	Energy Cons.
Repair	44	0.875	0.908	794
Reconfigure	36	0.375	0.945	736
Reconfigure and Incr. Speed	41	0.375	0.878	887
Normal	48	1	0.958	836

IV.5 Conclusion

In this chapter, we presented a modeling framework to support multi-objective optimization of manufacturing systems. The model of the manufacturing system captures the system-level interaction, machine-level operation, and part-level attributes. The simulation of the model is used to evaluate the effect of system configuration, a sequence of events, and process variables over productivity, reliability, quality, and energy consumption.

The main contribution of this chapter is the introduction of a hybrid model of manufacturing systems. The novel modeling framework integrates the system, machine, and part models to study the quantity-quality and quantity-energy relationships. Moreover, the approach presented here extends the state-of-the-art of opportunistic decision making from system-level control to include both system- and machine-level control variables. The set of solutions obtained from the integration of simulation and optimization framework balances the trade-off between different performance criteria. The benefit of this framework is the ability to evaluate different control variables at the machine and system level to improve productivity and sustainability. The modeling approach was validated using simulation and historical data of a real manufacturing system. For a simulated case study, results indicate that the decision to repair, reconfigure, or change process variables depends on the time horizon to analyze and the weight of the performance analysis criteria. Furthermore, results showed that the moving speed of the robot and travel speed of the CNC had a minor effect on productivity but increased the demand rate of electric energy.

CHAPTER V

Conclusion and Future Work

Manufacturing is one of the main economic sectors, responsible for 9% of the total employment, 12% of the total GDP, and 70% of the private-sector research and development [11]. However, poor performance measured by OEE below 50% and stagnation of productivity growth indicates the need for improvement. Efforts to improve the different aspects of manufacturing have led to a significant amount of research to boost productivity, improve quality, and reduce cost.

The development of control strategies that support reconfigurable manufacturing has been identified as one of the focus areas to improve the performance of plant floor operations. Recent advances in Hardware-In-the-Loop (HIL) simulation, communication and data extraction protocols, and high-performance computing have enabled the development of novel control solutions. The combination of simulated and real-world systems has led to the virtual fusion which allows the seamless integration of simulation in the control loop to monitor and control manufacturing systems [49] [106]. The integration of Open Platform Communications (OPC) as a data source and Internet as a data gateway has granted access to data that has been used to detect anomalies and monitor energy consumption [120] [124]. High-performance computing (HPC) helps to reduce the computational time of complex problems which support time-sensitive control actions [47].

This dissertation introduced a novel control framework for manufacturing systems that support productivity improvement, system-level reconfiguration, and cost reduction. The framework integrates HIL simulation, anomaly detection, and multi-objective optimization using HPC. Experimental case studies were used to validate the research. The contributions and future work of this dissertation are described in the following sections.

V.1 Contributions

The contributions of this dissertation have focused on three main aspects. First, real-time hybrid simulation was used to assess performance at both the machine- and system-level based on the concurrent operation of the real and virtual environments. A hybrid model was developed to study the discrete and continuous behavior of machines. The simulation environment was synchronized to run parallel to the plant floor by defining external events from the real system, and internal events from the simulation. Second, a framework was developed for modeling machines as Cyber-Physical Systems to improve equipment monitoring for anomaly detection and productivity analysis. The modeling framework merges sensor data, context information, and expert knowledge to support a context-sensitive analysis. Third, a control strategy was introduced that considers the coupling between production, quality, reliability, and energy consumption metrics. The control strategy includes both machine- and system-level variables and their impact on plant floor performance. These three contributions are summarized in the following subsections.

V.1.1 Real-time hybrid simulation

The first contribution presented in Chapter II and [113] [110] is a method for the development of a real-time hybrid simulation capable of running synchronous and concurrent with the real system. The combination of data from the real system and its simulation can be used to analyze the performance of manufacturing systems operating under non-stationary conditions. The real-time hybrid simulation extends the capabilities of Hardware-in-the-Loop (HIL) technology and improves the performance assessment of plant floor operations at both the machine- and system-levels. The integration between real and virtual environments was implemented in a manufacturing testbed to evaluate processing time and health of CNC machines and robots.

The development of the real-time hybrid simulation requires a model of the plant and the integration of the virtual and real environments. The model captures the discrete and continuous behavior of machines and their interactions. The integration of real and virtual environments is achieved by the communication between a real and an emulated controller to incorporate internal and external events. The real-time hybrid simulation method has the potential to transform the monitoring and performance assessment of manufacturing systems by supporting more accurate production expectation management, detection of blockages or starvations, and machine health evaluation. The implementation of the aforementioned contribution impacts the manufacturing industry by boosting productivity and reducing downtime.

V.1.2 Context-sensitive anomaly detection and diagnosis

This contribution presented in Chapter III and [111] [112] introduces a framework which combines both physics-based and data-driven models for anomaly detection based on the identification of machine-part interactions. The framework merges sensor data, context information, and expert knowledge to develop context-sensitive models. Both sensor data and context information are extracted using machine-to-machine communication protocols. Expert knowledge is used to specify a discrete set of states that define the operational context of the machines. This work contributes to the field of model-based anomaly detection and diagnosis by introducing context-sensitive adaptive threshold limits, and context-specific classification algorithms. The modeling and analysis framework was implemented in both an automotive assembly plant and a university testbed.

The development of the context-sensitive analysis framework requires the identification of the operational context defined by the machine functionality, dynamics, and interactions. The operational context of the machine is used to define a set of discrete states. The continuous dynamics for each state are studied using physics-based or data-driven models. The implementation of the proposed framework on the factory floor has the potential to support more effective maintenance actions by monitoring the condition of machines, parts, and processes in a manufacturing operation. The contribution will impact the manufacturing industry by reducing downtime and improving machine reliability.

V.1.3 Multi-Objective optimization for decision making

The innovation of this contribution presented in Chapter IV is a modeling and control strategy which considers the coupling between production, quality, reliability and sustainability metrics to support plant floor decision making. The novel control strategy studies the effect of various machine- and system-level variables over different performance metrics and balances them using a multi-objective optimization problem. This research contributes to the field of control of manufacturing systems by merging the control of machine states, process variables, and system configuration. The framework was validated using a combination of real machine data and simulation.

The development of the decision making framework requires the creation of models of both the machines and systems and the formulation of a multi-objective optimization problem. The models capture the effects of machine process variables and system configurations over different performance metrics. The optimal set of control variables is obtained using simulation-based optimization. The modeling and control strategies presented here can support plant floor decision-making by studying process variables, maintenance actions, or system reconfiguration to reduce energy consumption and improve productivity. The implementation of the multi-objective optimization framework will impact the operations and control of manufacturing system by reducing the cost associated with electricity charges and improve flexibility and productivity.

V.2 Future Work

Several areas of future work build directly on this research. Some suggestions to further extend the advancement in the assessment, monitoring, and control of manufacturing systems are discussed in the remainder of this section.

V.2.1 Time-varying models

One of the challenges of using simulation to evaluate the performance of a real manufacturing system is that the models used in the simulation do not consider the normal or expected degradation of mechanical and electrical components. Thus, time-invariant models will cause the simulation outputs to diverge from the performance of the real machines over time. An extension of the machine-level models in the virtual environment could include a characterization of degradation. The development of time-varying models can improve the accuracy of the simulation to support better machine-level health assessment and system-level performance analysis. This approach will have an impact on the plant floor operations by improving machine availability and productivity leading to better OEE.

V.2.2 Integration of Cyber and Physical domains

The development of cyber-physical systems opens the door to new modeling strategies for anomaly detection. However, the interconnection between the cyber and physical domains can result in new types of anomalies not detectable by current methods. For anomaly detection, the modeling and analysis of cyber-physical systems can be extended to include the effect of components of the cyber domain such as control logic or communication networks on physical assets. This integration may require the development of new data reduction and feature extraction methods capable of identifying the key variables that best detect the domain of different anomalies. The integration of cyber and physical domains presents an opportunity to develop models that leverage the Global Operational State (GOS) for context-sensitive models presented in this dissertation. An integrated anomaly detection method that considers both the cyber and physical domains will impact manufacturing systems by reducing unexpected downtime and improve product quality.

V.2.3 Optimality verification

The approach taken in this thesis to develop a control strategy requires the evaluation of sets of machine- and system-level variables. The optimal solution is obtained using heuristic optimization algorithms and simulation. However, this approach does not guarantee optimality and can be computationally expensive. The continuation of this research might focus on the development of hybrid optimization methods which combine dynamic programming and heuristic algorithms to improve the performance of the optimization. Moreover, the simulation-based approach can be improved by combining static and dynamic models to evaluate the configuration of the system and the machine process variables. Additional research could also be completed by considering the effect of human and organizational factors in a manufacturing system. The inclusion of information such as availability of manual labor would extend the capabilities of the model. Improving the modeling and optimization method would impact the manufacturing system by balancing the trade-offs between different performance metrics and helping to reduce cost.

Bibliography

[1] Jose Vicente Abellan-Nebot and Fernando Romero Subirón. A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology*, 47(1-4):237–257, 2010.

[2] M. Munir Ahmad and Nasreddin Dhafr. Establishing and improving manufacturing performance measures. *Robotics and Computer-Integrated Manufacturing*, 18(3-4):171–176, 2002.

[3] M Munir Ahmad and Nasreddin Dhafr. Establishing and improving manufacturing performance measures. *Robotics and Computer-Integrated Manufacturing*, 18(3):171–176, 2002.

[4] 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*, 6(6):621–639, 1990.

[5] Y Altintas, C Brecher, M Weck, and S Witt. Virtual machine tool. *CIRP Annals-manufacturing technology*, 54(2):115–138, 2005.

[6] Anthony I Anosike and David Z Zhang. Dynamic reconfiguration and simulation of manufacturing systems using agents. *Journal of Manufacturing Technology Management*, 17(4):435–447, 2006.

[7] Jorge Arinez, Stephan Biller, Semyon M Meerkov, and Liang Zhang. Quality/quantity improvement in an automotive paint shop: A case study. *IEEE Transactions on automation science and engineering*, 7(4):755–761, 2010.

[8] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[9] Alfred Bauer, Jimmie Browne, R Bowden, and James Duggan. *Shop floor control systems: from design to implementation*. Springer Science & Business Media, 2012.

[10] Nils Bengtsson, J Michaloski, F Proctor, Guodong Shao, and Sid Venkatesh. Towards data-driven sustainable machining: Combining mtconnect production data and discrete event simulation. In *ASME 2010 International Manufacturing Science and Engineering Conference*, pages 379–387. American Society of Mechanical Engineers, 2010.

[11] Suzanne Berger. *Making in America: from innovation to market*. MIT Press, 2013.

[12] Rajesh Kumar Bhushan. Optimization of cutting parameters for minimizing power consumption and maximizing tool life during machining of al alloy sic particle composites. *Journal of Cleaner Production*, 39:242–254, 2013.

[13] Zhuming Bi, Li Da Xu, and Chengen Wang. Internet of things for enterprise systems of modern manufacturing. *IEEE Transactions on Industrial Informatics*, 10(2):1537–1546, 2014.

[14] André Carvalho Bittencourt, Kari Saarinen, Shiva Sander-Tavallaey, Svante Gunnarsson, and Mikael 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*, 24(8):1032–1041, 2014.

[15] Anders Blomdell, Gunnar Bolmsjo, Torgny Brogardh, Per Cederberg, Mats Isaksson, Rolf Johansson, Mathias Haage, Klas Nilsson, Magnus Olsson, Tomas Olsson, and Anders Robertsson. Extending an industrial robot controller: implementation and applications of a fast open sensor interface. *IEEE Robotics & Automation Magazine*, 12(3):85–94, 2005.

[16] Nils Boysen, Malte Fliedner, and Armin Scholl. A classification of assembly line balancing problems. *European journal of operational research*, 183(2):674–693, 2007.

[17] David Broman, Edward A Lee, Stavros Tripakis, and Martin Törngren. Viewpoints, formalisms, languages, and tools for cyber-physical systems. In *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling*, pages 49–54. ACM, 2012.

[18] Manfredi Bruccoleri, Zbigniew J Pasek, and Yoram Koren. Operation management in reconfigurable manufacturing systems: Reconfiguration for error handling. *International Journal of Production Economics*, 100(1):87–100, 2006.

[19] AAG Bruzzone, D Anghinolfi, M Paolucci, and F Tonelli. Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. *CIRP Annals-Manufacturing Technology*, 61(1):459–462, 2012.

[20] Katharina Bunse, Matthias Vodicka, Paul Schönsleben, Marc Brülhart, and Frank O Ernst. Integrating energy efficiency performance in production management–gap analysis between industrial needs and scientific literature. *Journal of Cleaner Production*, 19(6):667–679, 2011.

[21] Christos G Cassandras and Stephane Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[22] Tag Kim 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.

[23] Guorong Chen, Liang Zhang, Jorge Arinez, and Stephan Biller. Energy-efficient production systems through schedule-based operations. *IEEE Transactions on Automation Science and Engineering*, 10(1):27–37, 2013.

[24] Jihong Chen, Jianzhong Yang, Huicheng Zhou, Hua Xiang, Zhihong Zhu, Yesong Li, Chen-Han Lee, and Guangda Xu. CPS modeling of CNC machine tool work processes using an instruction-domain based approach. *Engineering*, 1(2):247–260, 2015.

[25] Yong Chen and Jionghua Jin. Quality-reliability chain modeling for system-reliability analysis of complex manufacturing processes. *IEEE Transactions on Reliability*, 54(3):475–488, 2005.

[26] YuFeng Chen, ZhiWu Li, and MengChu Zhou. Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach. *IEEE Transactions on Automation Science and Engineering*, 11(2):549–563, 2014.

[27] William Jay Conover and WJ Conover. *Practical nonparametric statistics*. Wiley New York, 1999.

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

[29] Lorenzo Damiani, Pietro Giribone, Roberto Revetria, and Alessandro 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*, volume 2, 2014.

[30] Houshang Darabi, Mohsen A. Jafari, and Shomit S Manapure. Finite automata decomposition for flexible manufacturing systems control and scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 33(2):168–175, 2003.

[31] Adiel Teixeira De Almeida, Cristiano Alexandre Virgínio Cavalcante, Marcelo Hazin Alencar, Rodrigo José Pires Ferreira, Adiel Teixeira de Almeida-Filho, Thalles Garcez, and Thalles Vitelli. *Multicriteria and multiobjective models for risk, reliability and maintenance decision analysis*. Springer, 2016.

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

[33] Frank DiCesare and Alan Desrochers. Modeling, control, and performance analysis of automated manufacturing systems using Petri nets. *Control and Dynamic Systems*, 47:121–172, 2012.

[34] Steven X. Ding, Shen Yin, Kaixiang Peng, Haiyang Hao, and Bo Shen. A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. *IEEE Transactions on Industrial Informatics*, 9(4):2239–2247, 2013.

[35] Joost R Duflou, John W Sutherland, David Dornfeld, Christoph Herrmann, Jack Jeswiet, Sami Kara, Michael Hauschild, and Karel Kellens. Towards energy and resource efficient manufacturing: A processes and systems approach. *CIRP Annals-Manufacturing Technology*, 61(2):587–609, 2012.

[36] Marc Engeler, Daniel Treyer, David Zogg, Konrad Wegener, and Andreas Kunz. Condition-based maintenance: Model vs. statistics a performance comparison. *Procedia CIRP*, 57:253–258, 2016.

[37] John Engler and Jerry Jasinowski. *Efficiency and Innovation In U.S. Manufacturing Energy Use*. National Association of Manufacturers, 2014.

[38] Ana Paula Estrada-Vargas, Ernesto López-Mellado, and Jean-Jacques Lesage. A black-box identification method for automated discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, 2015.

[39] Sebastian Faltinski, Holger Flatt, Florian Pethig, Björn Kroll, Asmir Vodenčarević, Alexander Maier, and Oliver Niggemann. Detecting anomalous energy consumptions in distributed manufacturing systems. In *Industrial Informatics, 2012 10th IEEE International Conference on*, pages 358–363. IEEE, 2012.

[40] Theodor Freiheit and S Jack Hu. Impact of machining parameters on machine reliability and system productivity. *Journal of manufacturing science and engineering*, 124(2):296–304, 2002.

[41] Nicla Frigerio and Andrea Matta. Energy saving policies for a machine tool with warm-up, stochastic arrivals and buffer information. In *Automation Science and Engineering, 2014 IEEE International Conference on*, pages 646–651. IEEE, 2014.

[42] Hans J Geissler, Philipp Hölzl, Sascha Marohl, Ferdinand Kuhn-Régnier, Uwe Mehlhorn, Michael Südkamp, and E Rainer de Vivie. Risk stratification in heart surgery: comparison of six score systems. *European Journal of Cardio-thoracic surgery*, 17(4):400–406, 2000.

[43] Warren Gilchrist. Modelling failure modes and effects analysis. *International Journal of Quality & Reliability Management*, 10(5), 1993.

[44] Hamid Reza Golmakani, James K Mills, and Beno Benhabib. Deadlock-free scheduling and control of flexible manufacturing cells using automata theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(2):327–337, 2006.

[45] Jun Gong, Vittaldas V Prabhu, and Wenxin Liu. Simulation-based performance comparison between assembly lines and assembly cells with real-time distributed arrival time control system. *International Journal of Production Research*, 49(5):1241–1253, 2011.

[46] Abhijit Gosavi. Simulation-based optimization. *Parametric Optimization Techniques and Reinforcement Learning*, 2003.

[47] Georg Hager and Gerhard Wellein. *Introduction to high performance computing for scientists and engineers*. CRC Press, 2010.

[48] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[49] William S Harrison and Dawn Tilbury. Virtual fusion: Hybrid process simulation and emulation-in-the-loop. In *ASME 2008 9th Biennial Conference on Engineering Systems Design and Analysis*, pages 263–270. American Society of Mechanical Engineers, 2008.

[50] William S Harrison, Dawn M Tilbury, and Chengyin 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*, 9(1):96–109, 2012.

[51] Keyan He, Renzhong Tang, and Mingzhou Jin. Pareto fronts of machining parameters for trade-off among energy consumption, cutting force and processing time. *International Journal of Production Economics*, 185:113–127, 2017.

[52] K.K.B. Hon. Performance and evaluation of manufacturing systems. *CIRP Annals-Manufacturing Technology*, 54(2):139–154, 2005.

[53] Gyung Pyo Hong and Tag Gon Kim. A framework for verifying discrete event models within a DEVS-based system development methodology. *Transactions*, 13(0):1, 1996.

[54] Hesuan Hu and MengChu Zhou. A Petri net-based discrete-event control of automated manufacturing systems with assembly operations. *IEEE Transactions on Control Systems Technology*, 23(2):513–524, 2015.

[55] Dongping Huang, Hessam S Sarjoughian, Wenlin Wang, Gary Godding, Daniel E Rivera, Karl G Kempf, and Hans Mittelmann. Simulation of semiconductor manufacturing supply-chain systems with DEVS, MPC, and KIB. *IEEE Transactions on Semiconductor Manufacturing*, 22(1):164–174, 2009.

[56] Samuel H Huang, John P Dismukes, J Shi, QI Su, Mousalam A Razzak, Rohit Bodhale, and D Eugene Robinson. Manufacturing productivity improvement using effectiveness metrics and simulation analysis. *International Journal of Production Research*, 41(3):513–527, 2003.

[57] Sunan Huang, Kok Kiong Tan, and Tong Heng Lee. Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter. *IEEE Transactions on Industrial Electronics*, 59(11):4285–4292, 2012.

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

[59] Sergii Iarovyi, Wael M Mohammed, Andrei Lobov, Borja Ramis Ferrer, and Jose L Martinez Lastra. Cyber–physical systems for open-knowledge-driven manufacturing execution systems. *Proceedings of the IEEE*, 104(5):1142–1154, 2016.

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

[61] Mohsen Jahangirian, Tillal Eldabi, Aisha Naseer, Lampros K Stergioulas, and Terry Young. Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1):1–13, 2010.

[62] 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*, 36(9):2609–2626, 1998.

[63] Young Hun Jeong, Byung-Kwon Min, Dong-Woo Cho, and Sang Jo Lee. Motor current prediction of a machine tool feed drive using a component-based simulation model. *International Journal of Precision Engineering and Manufacturing*, 11(4):597–606, 2010.

[64] Anders Jönsson, Johan Wall, and Göran Broman. A virtual machine concept for real-time simulation of machine tool dynamics. *International Journal of Machine Tools and Manufacture*, 45(7):795–801, 2005.

[65] Aini Abdul Kadir, Xun Xu, and Enrico Hämmerle. Virtual machine tools and virtual machining—a technological review. *Robotics and computer-integrated manufacturing*, 27(3):494–508, 2011.

[66] AZ Keller, ARR Kamath, and UD Perera. Reliability analysis of CNC machine tools. *Reliability Engineering*, 3(6):449–473, 1982.

[67] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22, 2004.

[68] Yeong-Dae Kim, Sang-Oh Shim, Bum Choi, and Hark Hwang. Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing*, 16(2):290–298, 2003.

[69] Jurgen Kletti. *Manufacturing execution systems-MES*. Springer, 2007.

[70] Avenir Kobetski and Martin Fabian. Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. *Discrete Event Dynamic Systems*, 19(3):287–315, 2009.

[71] Margarete T Koch, Henning Baars, Heiner Lasi, and Hans-Georg Kemper. Manufacturing execution systems and business intelligence for production environments. In *Americas Conference on Information Systems*, page 436, 2010.

[72] Yoram Koren and Moshe Shpitalni. Design of reconfigurable manufacturing systems. *Journal of manufacturing systems*, 29(4):130–141, 2010.

[73] Ilya Kovalenko, Miguel Saez Saez, Kira Barton, and Dawn Tilbury. SMART: A system-level manufacturing and automation research testbed. *Smart and Sustainable Manufacturing Systems*, 1(1):232–261, 2017.

[74] Mark Last, Alla Sinaiski, and Halasya Siva Subramania. Predictive maintenance with multi-target classification models. In *Asian Conference on Intelligent Information and Database Systems*, pages 368–377. Springer, 2010.

[75] Ateekh-Ur-Rehman Lateef-Ur-Rehman. Manufacturing configuration selection using multicriteria decision tool. *The International Journal of Advanced Manufacturing Technology*, 65:625–639, 2013.

[76] Jay Lee, Behrad Bagheri, and Hung-An Kao. Recent advances and trends of cyber-physical systems and big data analytics in industrial informatics. In *International Proceeding of Int Conference on Industrial Informatics*, pages 1–6, 2014.

[77] Andrew Lee-Mortimer. Six sigma: a vital improvement approach when applied to the right problems, in the right environment. *Assembly Automation*, 26(1):10–17, 2006.

[78] YT Leung and Rajan Suri. Performance evaluation of discrete manufacturing systems. *Control Systems Magazine, IEEE*, 10(4):77–86, 1990.

[79] Jingshan Li and Semyon M Meerkov. *Production systems engineering*. Springer Science & Business Media, 2008.

[80] Jingshan Li, Semyon M Meerkov, and Liang Zhang. Production systems engineering: main results and recommendations for management. *International Journal of Production Research*, 51(23-24):7209–7234, 2013.

[81] Wei Li, Ramamurthy Mani, and Pieter J Mosterman. Extensible discrete-event simulation framework in simevents. In *Proceedings of the 2016 Winter Simulation Conference*, pages 943–954. IEEE Press, 2016.

[82] Yang Li, Qing Chang, Jun Ni, and Michael P Brundage. Event-based supervisory control for energy efficient manufacturing systems. *IEEE Transactions on Automation Science and Engineering*, 2016.

[83] Yang Li, Qing Chang, Guoxian Xiao, and Jorge Arinez. Data-driven analysis of downtime impacts in parallel production systems. *IEEE Transactions on Automation Science and Engineering*, 12(4):1541–1547, 2015.

[84] T Warren Liao. Clustering of time series data: a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[85] Seong Yong Lim and Tag Gon Kim. Hybrid modeling and simulation methodology based on DEVS formalism.

[86] Christoph Lindemann, Gianfranco Ciardo, Reinhard Geman, and Giinter Hommel. Performability modeling of an automated manufacturing system with deterministic and stochastic Petri nets. *IEEE International Conference on*, 1993.

[87] Lennart Ljung. *System identification toolbox: user's guide*. Citeseer, 1995.

[88] Neil A Macmillan and C Douglas Creelman. *Detection theory: A user's guide*. Psychology press, 2004.

[89] Iraj Mahdavi, Babak Shirazi, and Maghsud Solimanpur. Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems. *Simulation Modelling Practice and Theory*, 18(6):768–786, 2010.

[90] Behnam Malakooti. *Operations and production systems with multiple objectives*. John Wiley & Sons, 2013.

[91] Alok Mathur, GS Dangayach, ML Mittal, and Milind K Sharma. Performance measurement in automated manufacturing. *Measuring business excellence*, 15(1):77–91, 2011.

[92] Francisco Maturana, Leopoldo Paredes, and Axel Rodriguez. Integration of simulation of a machine for industrial automation, June 26 2014. US Patent Application No. 13/834,850.

[93] Francisco P Maturana, Maulik Kalolia, and Polo Paredes. A Virtual Design Engineering Environment for Control System Design and Validation. In *Industrial Applications of Holonic and Multi-Agent Systems*, pages 118–127. Springer, 2013.

[94] Semyon M Meerkov and Liang Zhang. Bernoulli production lines with quality-quantity coupling machines: monotonicity properties and bottlenecks. *Annals of Operations Research*, 182(1):119–131, 2011.

[95] John Moody and Panos J Antsaklis. *Supervisory control of discrete event systems using Petri nets*, volume 8. Springer Science & Business Media, 2012.

[96] Ghasem Moslehi and Mehdi Mahnam. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129(1):14–22, 2011.

[97] James R Moyne. Method and apparatus for optimizing profit in predictive systems, February 28 2017. US Patent 9,582,828.

[98] Ashkan Negahban and Jeffrey S Smith. Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, 33(2):241–261, 2014.

[99] Oliver Niggemann, Gautam Biswas, John S Kinnebrew, Hamed Khorasgani, Sören Volgmann, and Andreas Bunte. Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control. In *DX@ Safeprocess*, pages 185–192, 2015.

[100] David J Olive. A resistant estimator of multivariate location and dispersion. *Computational statistics & data analysis*, 46(1):93–102, 2004.

[101] Djamila Ouelhadj and Sanja Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4):417–431, 2009.

[102] Hervé Panetto, Michele Dassisti, and Angela Tursi. ONTO-PDM: Product-driven ontology for product data management interoperability within manufacturing process environment. *Advanced Engineering Informatics*, 26(2):334–348, 2012.

[103] Euisu Park, Dawn M Tilbury, and Pramod 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)*, 31(2):168–188, 2001.

[104] Hyeong-Tae Park, Jong-Geun Kwak, Gi-Nam Wang, and Sang C Park. Plant model generation for PLC simulation. *International Journal of Production Research*, 48(5):1517–1529, 2010.

[105] Ying Peng, Ming Dong, and Ming Jian Zuo. Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4):297–313, 2010.

[106] Nicholas M Putman, Francisco Maturana, Kira Barton, and Dawn M Tilbury. Virtual fusion: a hybrid environment for improved commissioning in manufacturing systems. *International Journal of Production Research*, 55(21):6254–6265, 2017.

[107] Luis Rabelo, Magdy Helal, Albert Jones, and Hyeung-Sik Min. Enterprise simulation: a hybrid system approach. *International Journal of Computer Integrated Manufacturing*, 18(6):498–508, 2005.

[108] Peter JG Ramadge and W Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

[109] Matthias Roth, Stefan Schneider, Jean-Jacques Lesage, and Lothar Litz. Fault detection and isolation in manufacturing systems with an identified discrete event model. *International Journal of Systems Science*, 43(10):1826–1841, 2012.

[110] Miguel Saez, Francisco Maturana, Kira Barton, and Dawn Tilbury. Real-time hybrid simulation of manufacturing systems for performance analysis and control. In *Automation Science and Engineering, 2015 IEEE Conference on*, pages 526–531. IEEE, 2015.

[111] Miguel Saez, Francisco Maturana, Kira Barton, and Dawn Tilbury. Anomaly detection and productivity analysis for cyber-physical systems in manufacturing. *Automation Science and Engineering, 2017 13th IEEE Conference on*, pages 23–29, 2017.

[112] Miguel Saez, Francisco P Maturana, Kira Barton, and Dawn M Tilbury. Context-sensitive modeling and analysis of cyber-physical manufacturing systems for anomaly detection and diagnosis. *To appear in IEEE Transactions on Automation Science and Engineering*, 2018.

[113] Miguel Saez, Francisco P Maturana, Kira Barton, and Dawn M Tilbury. Real-time manufacturing machine and system performance monitoring using internet of things. *IEEE Transactions on Automation Science and Engineering*, (99):1–14, 2018.

[114] Kazuhiro Saitou, Samir Malpathak, and Helge Qvam. Robust design of flexible manufacturing systems using, colored Petri net and genetic algorithm. *Journal of intelligent manufacturing*, 13(5):339–351, 2002.

[115] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[116] Shivakumar Sastry, Michael S Branicky, and PS Sastry. Cloud conveyors system: a versatile application for exploring cyber-physical systems. In *Control of Cyber-Physical Systems*, pages 43–62. Springer, 2013.

[117] Lawrence F Shampine and Mark W Reichelt. The Matlab ODE suite. *SIAM journal on scientific computing*, 18(1):1–22, 1997.

[118] JS Shang and PR Tadikamalla. Multicriteria design and control of a cellular manufacturing system through simulation and optimization. *International Journal of Production Research*, 36(6):1515–1528, 1998.

[119] Yuan Shaoqiang, Liu Zhong, and Li Xingshan. Modeling and simulation of robot based on Matlab/SimMechanics. In *Control Conference, 2008. CCC 2008. 27th Chinese*, pages 161–165. IEEE, 2008.

[120] Seung-jun Shin, Jungyub Woo, and Sudarsan Rachuri. Predictive analytics model for power consumption in manufacturing. *Procedia CIRP*, 15:153–158, 2014.

[121] Joshi Mandar Shridhar and Shailendra Ravi. Virtual manufacturing: an important aspect of collaborative product commerce. *Journal of Advanced Manufacturing Systems*, 1(01):113–119, 2002.

[122] Fei Tao, Ying Zuo, Senior Member, and Lin Zhang. IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud. 10(2):1547–1557, 2014.

[123] James H Taylor. Rigorous handling of state events in Matlab. In *Control Applications, 1995., Proceedings of the 4th IEEE Conference on*, pages 156–161. IEEE, 1995.

[124] Nunzio Marco Torrisi and João Fernando Gomes de Oliveira. Remote monitoring for high-speed CNC processes over public IP networks using CyberOPC. *The International Journal of Advanced Manufacturing Technology*, 60(1-4):191–200, 2012.

[125] Chorng-Jyh Tzeng, Yu-Hsin Lin, Yung-Kuang Yang, and Ming-Chang Jeng. Optimization of turning operations with multiple performance characteristics using the Taguchi method and grey relational analysis. *Journal of materials processing technology*, 209(6):2753–2759, 2009.

[126] Alberto Vergnano, Carl Thorstensson, Bengt Lennartson, Petter Falkman, Marcello Pellicciari, Francesco Leali, and Stephan Biller. Modeling and optimization of energy consumption in cooperative multi-robot systems. *IEEE Transactions on Automation Science and Engineering*, 9(2):423–428, 2012.

[127] Athulan Vijayaraghavan, David Dornfeld, et al. Addressing process planning and verification issues with MTconnect. *Laboratory for Manufacturing and Sustainability*, 2009.

[128] Athulan Vijayaraghavan, Will Sobel, Armando Fox, David Dornfeld, and Paul Warndorf. Improving machine tool interoperability using standardized interface protocols: Mt connect. In *International Symposium on Flexible Automation*, 2008.

[129] Konstantin Vikhorev, Richard Greenough, and Neil Brown. An advanced energy management framework to promote energy awareness. *Journal of Cleaner Production*, 43:103–112, 2013.

[130] Asmir Vodenčarević, Hans Kleine Büning, Oliver Niggemann, and Alexander Maier. Using behavior models for anomaly detection in hybrid systems. In *Information, Communication and Automation Technologies, 2011 XXIII International Symposium on*, pages 1–8. IEEE, 2011.

[131] Pavel Vrba and Vladimír Marik. Simulation in agent-based manufacturing control systems. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 2, pages 1718–1723. IEEE, 2005.

[132] Achmad Widodo and Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6):2560–2574, 2007.

[133] Chifu Yang, Zhengmao Ye, O Ogbobe Peter, and Junwei Han. Modeling and simulation of spatial 6-DOF parallel robots using Simulink and Simmechanics. In *Computer Science and Information Technology, 2010 3rd IEEE International Conference on*, volume 4, pages 444–448. IEEE, 2010.

[134] Zimin Yang, Dragan Djurdjanovic, and Jun Ni. Maintenance scheduling for a manufacturing system of machines with adjustable throughput. *IIE transactions*, 39(12):1111–1125, 2007.

[135] Ayman MA Youssef and Hoda A ElMaraghy. Assessment of manufacturing systems reconfiguration smoothness. *The International Journal of Advanced Manufacturing Technology*, 30(1-2):174–193, 2006.

[136] Eleni Zampou, Stathis Plitsos, Angeliki Karagiannaki, and Ioannis Mourtos. Towards a framework for energy-aware information systems in manufacturing. *Computers in Industry*, 65(3):419–433, 2014.

[137] Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[138] Bernard P Zeigler, Hae Sang Song, Tag Gon Kim, and Herbert Praehofer. DEVS framework for modeling, simulation, analysis, and design of hybrid systems. In *International Hybrid Systems Workshop*, pages 529–551. Springer, 1994.

[139] Julie Z Zhang, Joseph C Chen, and E Daniel Kirby. Surface roughness optimization in an end-milling operation using the Taguchi design method. *Journal of Materials Processing Technology*, 184(1-3):233–239, 2007.

[140] Yingfeng Zhang, Cheng Qian, Jingxiang Lv, and Ying Liu. Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. *IEEE Transactions on Industrial Informatics*, 13(2):737–747, 2017.

[141] Yingfeng Zhang, Wenbo Wang, Naiqi Wu, and Cheng Qian. Iot-enabled real-time production performance analysis and exception diagnosis model. *IEEE Transactions on Automation Science and Engineering*, 13(3):1318–1332, 2016.

[142] Yingfeng Zhang, Geng Zhang, Junqiang Wang, Shudong Sun, Shubin Si, and Teng Yang. Real-time information capturing and integration framework of the internet of manufacturing things. *International Journal of Computer Integrated Manufacturing*, 28(8):811–822, 2015.

[143] Bo Zhao and Yuanchun Li. Local joint information based active fault tolerant control for reconfigurable manipulator. *Nonlinear dynamics*, 77(3):859–876, 2014.

[144] Lirong Zhou, Jianfeng Li, Fangyi Li, Qiang Meng, Jing Li, and Xingshuo Xu. Energy consumption model and energy efficiency of machine tools: a comprehensive literature review. *Journal of Cleaner Production*, 112:3721–3734, 2016.