Artificial Intelligence for Engineering Design, Analysis and Manufacturing

cambridge.org/aie

Hidden Markov model-based digital twin construction for futuristic manufacturing systems

Angkush Kumar Ghosh[1], AMM Sharif Ullah[2] and Akihiko Kubo[2]

[1]Graduate School of Engineering, Kitami Institute of Technology, 165 Koen-cho, Kitami, Hokkaido 090-8507, Japan and [2]Faculty of Engineering, Kitami Institute of Technology, 165 Koen-cho, Kitami, Hokkaido 090-8507, Japan

Research Article

Cite this article: Ghosh AK, Ullah AMMS, Kubo A (2019). Hidden Markov model-based digital twin construction for futuristic manufacturing systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **33**, 317–331. https://doi.org/10.1017/S089006041900012X

Received: 27 November 2018
Revised: 10 March 2019
Accepted: 10 March 2019
First published online: 3 May 2019

Key words:
Complex phenomena; digital twin; hidden Markov model; manufacturing systems; surface roughness

Author for correspondence:
AMM Sharif Ullah, E-mail: ullah@mail.kitami-it.ac.jp

Abstract

This paper addresses the construction of digital twins (exact mirror images of real-world in cyberspace) using hidden Markov models for the futuristic manufacturing systems known as Industry 4.0. The proposed digital twin consists of two components namely model component and simulation component. The model component forms a Markov chain that encapsulates the dynamics underlying the phenomenon by using some discrete states and their transition probabilities. The simulation component recreates the phenomenon using a Monte Carlo simulation process. The efficacy of the proposed digital twin construction methodology is shown by a case study, where the digital twin of the surface roughness of a surface created by successive grinding operations is described. The developers of the cyber-physical systems will be benefitted from the outcomes of this study because these systems need the computable virtual abstractions of the manufacturing phenomena to address the issues related to the maturity index of futuristic manufacturing systems (i.e., understand, predict, decide, and adopt).

Introduction

The aerospace community has introduced a system engineering concept called digital twin, which refers to an exact mirror image of a real-life aspect (e.g., flying of a spacecraft) in the cyberspace using the multi-scale, multi-physics, and probabilistic simulation that is aided by the sensor updates and historical data (Glaessgen and Stargel, 2012). This has inspired the research community of the futuristic manufacturing systems (e.g., Industry 4.0, smart manufacturing, and connected factory). As such, the digital twins of manufacturing aspects are expected to populate the systems (e.g., cyber-physical systems) that are needed for functionalizing the futuristic manufacturing systems (Grieves and Vickers, 2017; Ullah, 2019). The research on digital twin construction reports its myriad interplays with various aspects of futuristic manufacturing systems. Some of the recent articles are described below.

Rosen *et al.* (2015) described the role of digital twin for converting a manufacturing system from an automated system (sensor-actuator-execution) to an autonomous system by adding knowledge and skill bases to the sensor-actuator-execution module. Alam and Saddik (2017) proposed the architecture of the digital twins for functionalizing the cyber-physical systems where the key elements are physical things, digital twins of the physical things, complex things comprising of the hierarchically structured subsystems, relationship networks among the complex things, and integration nodes of the web services. Kritzinger *et al.* (2018) reviewed the research on the digital twin used in manufacturing, and showed that the usages of sensor signals, shape information, data formatting (RIDF, XML, AutomationML, and alike), semantic web technology, and simulation technology either for constructing the digital twins or for functionalizing it within the framework of futuristic manufacturing systems. Schroeder *et al.* (2016) have outlined the IT infrastructure for constructing a manufacturing digital twin using AutomationML. Uhlemann *et al.* (2017) described that there are three types of digital twins, namely, digital shadow (real-time data links), digital twin for data manipulation, and digital twin for process parameter optimization, which must work in a systematic manner for reducing lead time due to the data acquisition in manufacturing systems. Qi *et al.* (2018) provided a broader picture of digital twin by elucidating its existence at the three levels of systemization, namely, job-shop level, manufacturing system level, and manufacturing system-of-system level. Haag and Anderl (2018) described the concept of digital twin using the case of a bending beam test bench to incorporate it into the next generation manufacturing systems. Talkhestani *et al.* (2018) showed that there must be some universal resources called anchor points that contain the detailed information regarding the machine tools, software packages, and electric devices for checking the consistency of the digital twins meant for the job-shop environment. Scaglioni and Ferretti (2018) developed the digital twins of a serial

© Cambridge University Press 2019

CAMBRIDGE UNIVERSITY PRESS

machine tool structure using the FEM and of a material removal process using the kinematic analysis. The twins can be used to regulate the chatter vibration in milling. Schleich *et al.* (2017) described the concept of a digital twin from the perspective of the shape of an object. Botkina *et al.* (2018) developed the digital twin of a cutting tool by organizing the information of the components of a cutting tool using the international standard (ISO 13399), which is needed to functionalize the IoT devices for optimizing a machining operation. They also described that the digital twins provide the contents necessary for performing, coordinating, and optimizing manufacturing activities. Lu and Xu (2018) developed the semantic model of CNC machine tool using concept maps (Ullah *et al.*, 2013) (a personalized ontological network) that can be used to construct the digital twins of the manufacturing resources. Luo *et al.* (2018) proposed the concept of digital twin for making the CNC machine tool more intelligent where the knowledge acquisition and management modules are integrated with the control mechanism of the machine tool. Hu *et al.* (2018) showed how to connect different types of physical and virtual agents using MTConnect Protocol for creating the digital twin-based manufacturing systems. Kunath and Winkler (2018) reported that digital twin emerged from the development of simulation technology. They proposed a framework of digital twin where the data and information systems are integrated with both physical and virtual manufacturing systems (e.g., manufacturing equipment systems, material flow systems, value stream system, operating material system, and human resource system). Padovano *et al.* (2018) showed how to convert an existing manufacturing environment to a manufacturing environment that is in line with the concept of futuristic manufacturing systems using digital twins where the twins encapsulate and transfer knowledge required within the cyber-physical systems. They also showed an architecture to integrate the digital twins. Söderberg *et al.* (2017) described the digital twins for a sheet metal assembly line. They showed that for ensuring desired geometry of a sheet metal product, the geometric representation of the assembly, kinematic relations, FEA functionality, Monte Carlo simulation, material properties, and the links to the inspection databases must be integrated. Olivotti *et al.* (2018) proposed a concept called installed base (detailed knowledge of machines, components, and subcomponents associated with a manufacturing facility), which should be combined with the sensor data for building the digital twins of manufacturing services. Zhuang *et al.* (2018) described the role of the digital twin of an assembly shop floor as the core component of a cyber-physical system. The digital twins must be integrated with real-time data acquisition systems and the historical data in the form of big-data. Tao *et al.* (2018) introduced the concept called the digital twin data for resource management, production planning, and process control, and stressed the need of a systematic approach for constructing the digital twin data. Zheng *et al.* (2018) revisited the concept of digital twin from a broader perspective and introduced different types of digital twins (digital twins for product function/performance/testing, manufacturing process, production equipment, plant operation, and alike) for supporting the product lifecycle, which are linked to the physical world via information processing and network module.

Nevertheless, from the viewpoint of contents, the digital twins can be categorized into three categories, namely, object twin, process twin, and phenomenon twin (Ullah, 2019). An object twin is the computable virtual abstraction of the geometrical and topological structures of a product (e.g., a gear) or a facility (a machine tool, an assembly line, and so forth). A process twin is a computable virtual abstraction of a process or production plan (e.g., scheduling for machining a part at different workstations spread in different factories, a bill of materials, and so forth). On the other hand, the computable virtual abstraction of a manufacturing phenomenon (e.g., the phenomena related to material removal process, namely, cutting force, tool wear, cutting temperature, workpiece deformation, surface roughness, chatter vibration, and so forth) is called a phenomenon twin. The three categories of digital twins must populate the systems (e.g., cyber-physical systems) underlying futuristic manufacturing systems, as mentioned above.

However, manufacturing phenomena are very complex and exhibit stochastic features (Ullah, 2019). In a real-life setting, the phenomena are monitored by time series data generated from the outputs of various sensors (e.g., temperature sensor, pressure sensor, force sensor, deformation sensor, acoustic emission sensor, and so forth). When a phenomenon is studied in a laboratory setting either by performing an analysis or by conducting an experiment, the results are recorded using a set of time series. Therefore, the manifestation of a manufacturing phenomenon is most likely to be a set of time series with stochastic features. As a result, the digital twin of a phenomenon is expected to encapsulate the dynamics exhibited by the stochastic features of a set of time series. At the same time, the twin must be capable of simulating the phenomenon in the form of a time series whenever needed (e.g., while monitoring a relevant manufacturing process). This type of phenomena twins is hereinafter referred to as time-series-driven phenomena twins. An immediate question is what is the procedure to construct a time-series-driven phenomenon twin? One of the answers to this question is to use the hidden Markov models because they are powerful tools for modeling and simulation of time series exhibiting stochastic features, and, thereby, have been used for a long time (Baum and Petrie, 1966; Fraser, 2008; Visser, 2011; Li *et al.*, 2017; Nguyen, 2017; Petropoulos *et al.*, 2017). Based on this contemplation, this paper describes the utilization of the hidden Markov models in constructing the time-series-driven phenomena twins. Besides describing the methodology using some mathematical entities and their relationships, the paper also reports a case study showing the efficacy of the hidden Markov models in constructing the time-series-driven phenomena twins of the surface roughness of grinding operations. As such, the remainder of this paper is organized as follows.

The section "Methodology" is organized in three subsections to describe the methodology showing how to construct a hidden Markov model for an arbitrary time series. The section "Case Study" describes a case study where the methodology described in the section "Methodology" is applied to create the digital twins of the surface roughness of ground surface (i.e., the workpiece surface generated by successive grinding operations). This section also describes how to construct a meta-ontology of the constructed digital twin so that it can create useful contents for the semantic web, which is needed to functionalize the cyber-physical systems. The section "Summary and concluding remarks" provides the summary and concluding remarks of this study.

Methodology

This section describes a methodology showing how to construct a hidden Markov model using the information of an arbitrary time

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

series. For the sake of better understanding, this section first describes the fundamental idea, which is followed by the mathematical formulations and algorithms, respectively.

Fundamental idea

The fundamental idea means here a somewhat informal description of the hidden Markov model and its relationship with a time series.

Figure 1 schematically illustrates a hidden Markov model consisting of three modules. The first module is called a Markov chain. The second module is called a time series of latent variables. The last module is called a time series of observations. The descriptions of the three modules are as follows:

As seen in Figure 1, a Markov chain forms a network showing both a set of discrete states (which later appear as latent variables) and their transitions. Each state and its possible transitions are associated with the probabilities, which are also the parts of the Markov chain. For example, the Markov chain shown in Figure 1 consists of three discrete states having the probabilities $p(\text{Low}) = 0.3$, $p(\text{Moderate}) = 0.4$, and $p(\text{High}) = 0.3$. The transitions to states denoted as Low, Moderate, and High from the state Low exhibit the following transition probabilities: $p(\text{Low}|\text{Low}) = 0.4$, $p(\text{Low}|\text{Moderate}) = 0.4$, and $p(\text{Low}|\text{High}) = 0.2$. The summation of the state probabilities or the transition probabilities from a given state to other possible states is unit. There are other issues related to the hidden Markov chain (e.g., the order of the Markov chain). The case shown in Figure 1 corresponds to the first order Markov chain because the probability of the previous state determines the current state. This paper adopts this strategy. See Fraser (2008) and Visser (2011) for more details regarding the order of a Markov chain and other relevant issues.

Regarding the module called time series of the latent variables, the following remarks can be made. The latent variables are the results of a stochastic simulation process (i.e., Monte Carlo simulation of discrete states associated with the Markov chain). Thus, for the case shown in Figure 1, the latent variables belong to the set of discrete states, that is, $lv(0),\ldots,lv(i-1)$, $lv(i)$, $lv(i+1),\ldots \in$ {Low, Moderate, High}. These are called latent because one cannot observe (or not interested in observing) these variables, that is, they are simulated for the sake of computation. The probabilities (in reality, relative frequencies) of the latent variables must be consistent with the transition probabilities associated with the Markov chain. This means that for the case shown in Figure 1, when $lv(i)$ = Low, then the probability of $lv(i+1)$ = Low is equal to 0.4, $lv(i+1)$ = Moderate is equal to 0.4, and $lv(i+1)$ = High is equal to 0.2.

Lastly, regarding the module called the time series of the observations, the following remarks can be made. The observations $ob(0),\ldots,ob(i-1)$, $ob(i)$, $ob(i+1),\ldots$ are simulated using the information of the corresponding latent states, $lv(0),\ldots,lv(i-1)$, $lv(i)$, $lv(i+1),\ldots$. The observations are the outputs of the hidden Markov model, which is used to solve a problem.

When one constructs a hidden Markov model to encapsulate the dynamics underlying a given time series, the scenario shown in Figure 2 evolves. The scenario entails five steps, namely, data acquisition (defining the time series, return map, and latent variables), Markov chain construction, simulation of latent variables and observations, and a comparison between the simulated observations and given time series. This paper uses the above-mentioned steps for creating a time-series-driven phenomenon twin. For the sake of better understanding, a set of mathematical entities and their relationships are required, which are described in the following subsection.

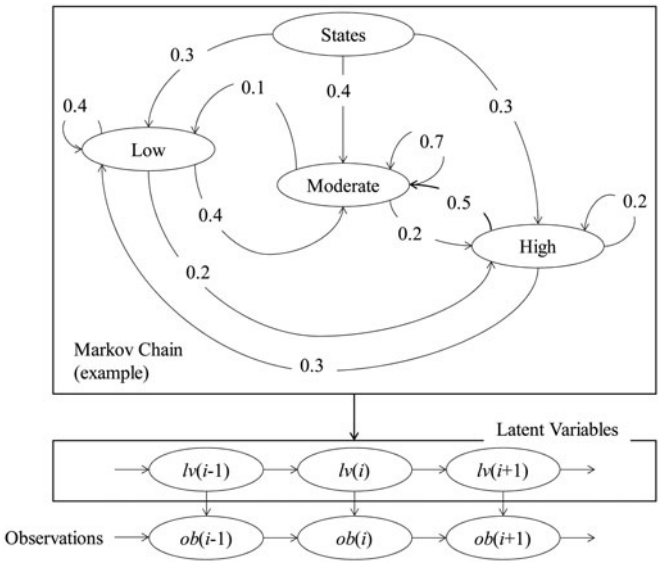


Fig. 1. The concept of hidden Markov model.

Mathematical formulations

This subsection describes the mathematical entities and the underlying relationships that are needed to create a time-series-driven phenomenon twin using a hidden Markov model. The goal is to define the processes underlying the data acquisition, Markov chain construction, simulation of latent variables and observations, and a comparison between simulated observations and given time series.

Let the manifestation of a phenomenon be a piece of *time series* denoted as $X = \{x(t) \in \mathfrak{R} \mid t = 0, \Delta t, 2\Delta t,\ldots, m \times \Delta t\}$ where Δt is known as *delay* or *interval*. The parameter t underlies a temporal or spatial entity, that is, a point of time or a distance. If preferred, the time series can be represented by indexing its elements using a *pointer*. In this case, $x(t)$ is replaced by $x(i)$ $(=x(t))$ where $t = i \times \Delta t$ and i is the pointer, which is an positive integer including 0, that is, $i = 0,1,\ldots$.

Let $U = [u_{\min}, u_{\max}] \in \mathfrak{R}$, defined as the *universe of discourse*, be an interval so that $X \supseteq U$. Let $x_{\min}$ be the minimum value of X, that is, $x_{\min} = \min(x(t) \mid \forall t \in \{0, \Delta t, 2\Delta t,\ldots\})$. Let $x_{\max}$ be the maximum value of X, that is, $x_{\max} = \max(x(t) \mid \forall t \in \{0, \Delta t, 2\Delta t, \ldots\})$. If $U = [u_{\min}, u_{\max}] = [x_{\min}, x_{\max}]$, then it is defined as the *exact interval case*.

Let $U_1, \ldots, U_n$ be n number of mutually exclusive intervals that partition U so that the following proposition denoted as P is true.

$$P = \left(\begin{array}{c} (U_1 \cup \ldots \cup U_n = U) \wedge (U_j < U_{j+1} | \, \forall j \in \{1, \ldots, n-1\}) \\ \wedge (U_k \cap U_l = \emptyset \mid \forall k \in \{1, \ldots, n\}, \forall l \in \{1, \ldots, n\} - \{j\}) \end{array} \right) \quad (1)$$

The partitions are the *states* (or latent states or variables) of X resulting in a *state vector* $SV = (U_1,\ldots,U_n)$. One can define the states in many ways making the proposition P true. One of the straightforward ways is to consider a *state interval* $\Delta u = (u_{\max} - u_{\min})/n$ and use it for defining the states in the following manner: $U_1 = [u_{\min}, u_{\min} + \Delta u)$, $U_2 = [u_{\min} + \Delta u, u_{\min} + 2\Delta u)$, $\ldots$, $U_n = [u_{\min} + (n-1) \times \Delta u, u_{\min} + n \times \Delta u]$.

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

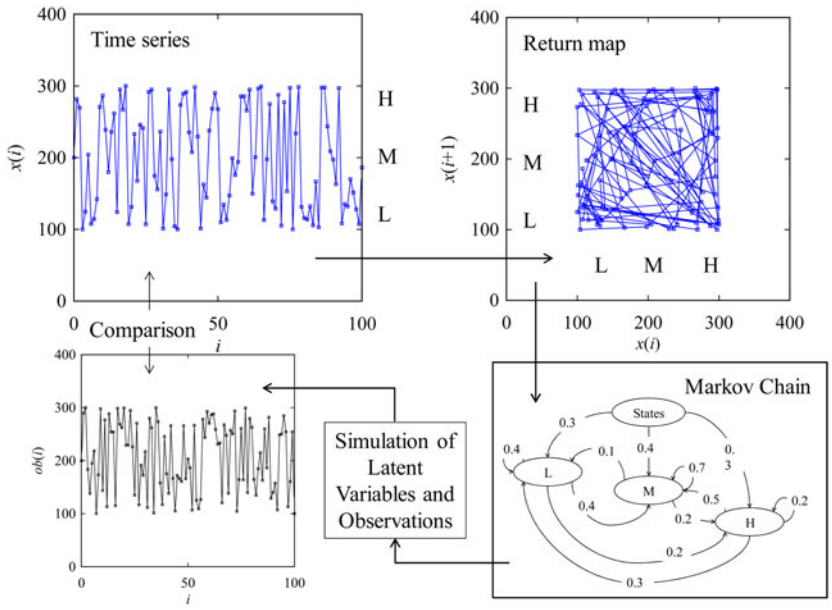


Fig. 2. Integration between hidden Markov model and time series.

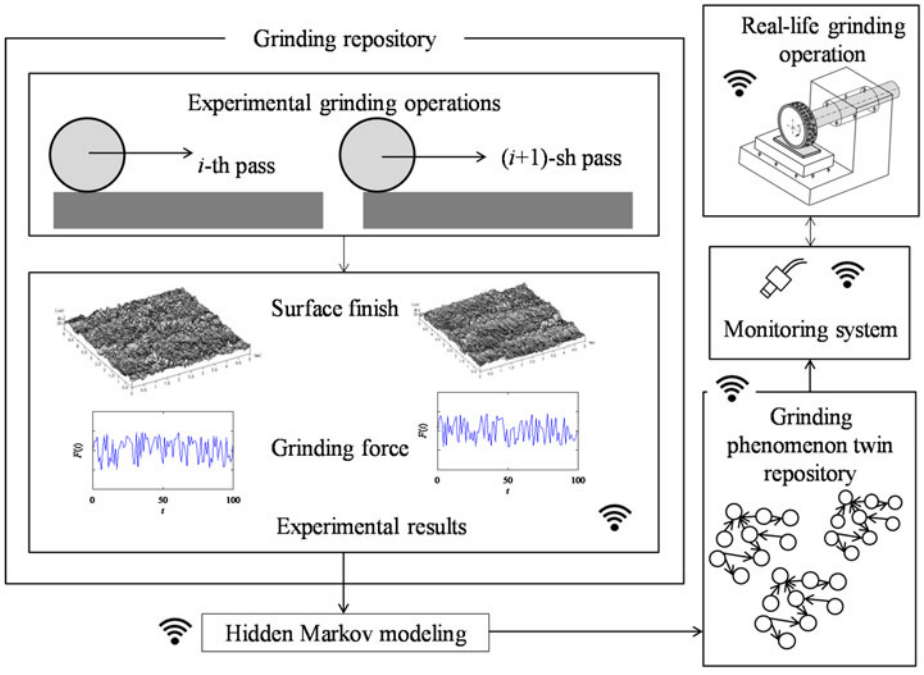


Fig. 3. An architecture of hidden Markov model-based grinding operation system.

Let $p(U_j) \in [0,1]$ be the probability of j-th state U_j in SV with respect to X, $j = 1,\ldots,n$. Thus, the following relationship holds.

$$\Phi(x(i), U_j) = \begin{cases} 1, & x(i) \in U_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$p(U_j) = \frac{\sum_{i=0}^{m} \Phi(x(i), U_j)}{m+1} \tag{3}$$

Thus, $p(U_j)$ is defined as the *state probability* of the j-th state U_j. As such, the summation of all state probabilities is equal to unit, that is,

$$p(U_1) + \ldots + p(U_n) = 1 \tag{4}$$

For the sake of computation (e.g., simulation), the state probabilities can be used to calculate the *cumulative state probability*. The cumulative state probability of the j-th state U_j in SV is defined as follows:

$$pc(U_j) = p(U_1) + \ldots + p(U_j) \tag{5}$$

As such, $pc(U_n) = 1$. The cumulative state probability can be used to calculate the *state probability interval* denoted as $pc_{\text{in}}(U_j)$.

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

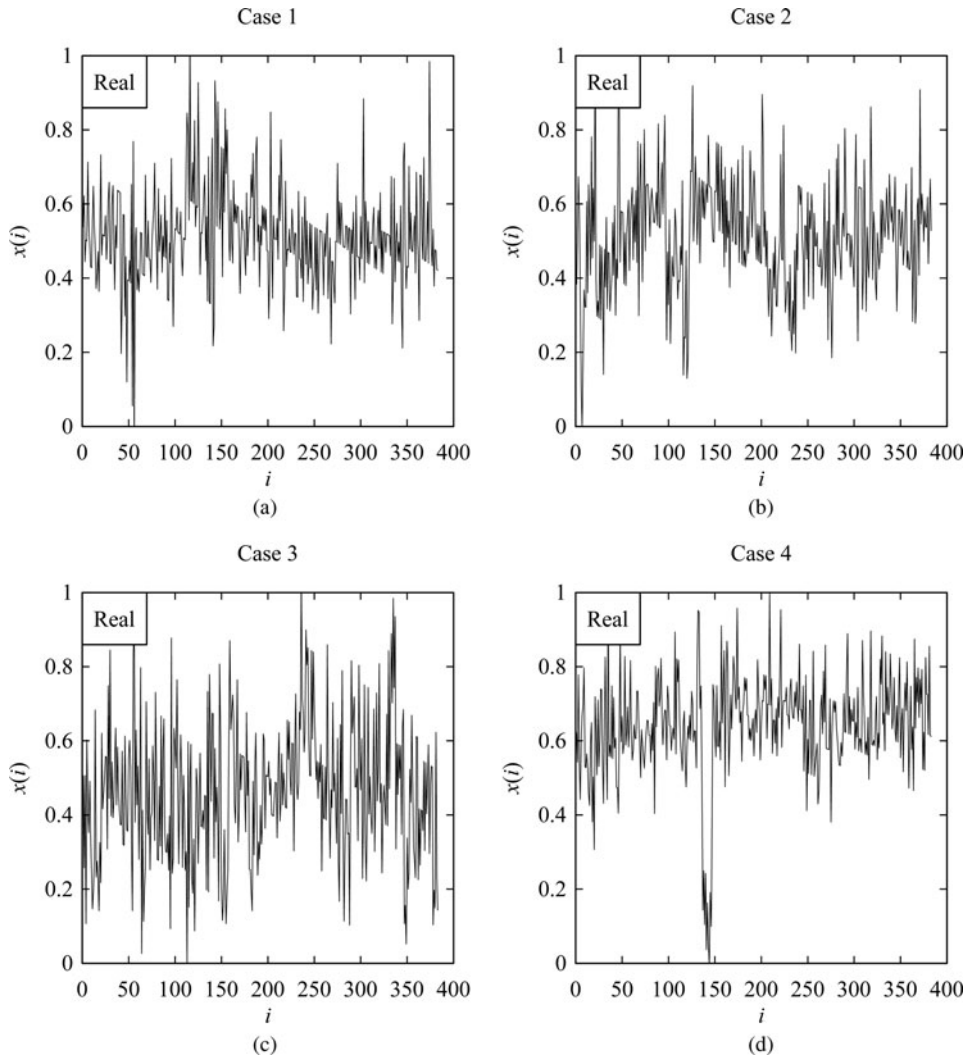


Fig. 4. The normalized ground surface heights (Ullah *et al.*, 2018). (a) Measured surface heights after the first pass. (b) Measured surface heights after the second pass. (c) Measured surface heights after the third pass. (d) Measured surface heights after the fourth pass.

The state probability interval of the first state U_1 in SV is as follows:

$$pc_{\text{in}}(U_1) = [0, pc(U_1)) \tag{6.1}$$

The state probability interval of the last state U_n in SV is as follows:

$$pc_{\text{in}}(U_n) = [pc(U_{n-1}), pc(U_n)] \tag{6.2}$$

The state probability intervals of the states other than U_1 and U_n in SV are as follows:

$$pc_{\text{in}}(U_j) = [pc(U_{j-1}), pc(U_j)) \quad \forall j \in \{2, \ldots, n-1\} \tag{6.3}$$

Let the set of tuples $\{(x(i),x(i+1)) \mid i = 1,2,\ldots\}$ (or $\{(x(t),x(t+i\Delta t)) \mid i = 1,2,\ldots\}$) be the *return* or *delay map* of the time series X. Therefore, each point $(x(i), x(i+1))$, $\exists i \in \{1,2,\ldots\}$ of the return map exhibits a transition. As a result, a *transition probability* denoted as $tp(U_o|\ U_j)$ ($\forall o, \forall j \in \{1,\ldots,n\}$) means the likelihood of the transition of X to the state U_o from the state U_j. Thus, the following relationships hold.

$$\Phi((x(i), x(i+1)), (U_o, U_j)) = \begin{cases} 1, & ((x(i) \in U_j) \wedge (x(i+1) \in U_o)) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$tp(U_o|U_j) = \frac{\sum_{i=0}^{m} \Phi((x(i), x(i+1)), (U_o, U_j))}{\sum_{i=0}^{m} \Phi(x(i), U_j)} \tag{8}$$

As such, the summation of all transition probabilities from a given state is equal to unit, that is, $tp(U_1|U_j) + \ldots + tp(U_n|U_j) = 1$, $\exists j \in \{1,\ldots,n\}$. This yields a *transition probability matrix* as follows:

$$M_{tp} = \begin{bmatrix} tp(U_1|U_1) & \cdots & tp(U_n|U_1) \\ \vdots & \ddots & \vdots \\ tp(U_1|U_n) & \cdots & tp(U_n|U_n) \end{bmatrix} \tag{9}$$

For the sake of computation (e.g., simulation), the transition probabilities can be used to calculate the *cumulative transition probability*. The cumulative transition probability is defined as

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

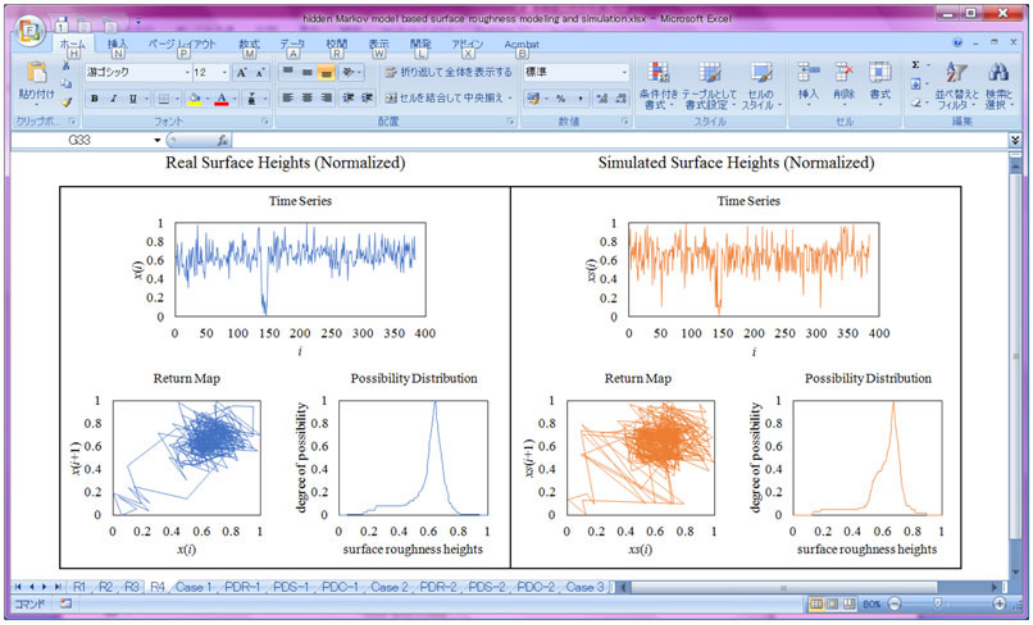


Fig. 5. Computing tool for implementing hidden Markov model-based surface roughness modeling and simulation.

follows:

$$tpc(U_o|U_j) = tp(U_1|U_j) + \ldots + tp(U_o|U_j) \tag{10}$$

As such, $tpc(U_n|U_j) = 1$. This yields the *cumulative transition probability matrix*, as follows:

$$M_{tpc} = \begin{bmatrix} tpc(U_1|U_1) & \cdots & tpc(U_n|U_1) \\ \vdots & \ddots & \vdots \\ tpc(U_1|U_n) & \cdots & tpc(U_n|U_n) \end{bmatrix} \tag{11}$$

The cumulative transition probability can be used to calculate the *transition probability interval* denoted as $tpc_{\text{in}}(U_o|U_j)$. The transition probability interval of the first states U_1 to any state U_j is as follows:

$$tpc_{\text{in}}(U_1|U_j) = [0, tpc(U_1|U_j)) \tag{11.1}$$

The transition probability interval of the last state U_n to any state U_j is as follows:

$$tpc_{\text{in}}(U_n|U_j) = [tpc(U_{n-1}|U_j), tpc(U_n|U_j)] \tag{11.2}$$

The transition probability intervals of any states to U_j (other than U_1 or U_n) are as follows:

$$tpc_{\text{in}}(U_o|U_j) = [tpc(U_{o-1}|U_j), tpc(U_o|U_j)) \ \forall o \in \{2, \ldots, n-1\} \tag{11.3}$$

Simulation

Using the mathematical entities and their relationships described in the previous subsection, one can formulate a Monte Carlo simulation process to simulate the latent variables $lv(.) \in (U_1, \ldots, U_n)$ and the observations $ob(.) \in \mathfrak{R}$. Note that the simulated observations will be denoted as $xs(.)$, not as $ob(.)$, to make the notation consistent with the time series, $x(.)$. The simulation process consists of the following nine steps. The first three steps, Steps 1,…,3, are related to the steps of the Markov chain formulation as shown in Figure 2. The other steps, Steps 4,…,9, are related to the steps of the simulation of latent variables and observations as shown in Figure 2.

Step 1	Define the time series $X = \{x(i) \in \mathfrak{R} \mid i = 0,\ldots,m\}$, universe of discourse U, state vector $SV = (U_1,\ldots,U_n)$, and the number of iteration $N \in \aleph^{+0}$.
Step 2	Calculate the state probabilities $p(U_j)$, cumulative state probabilities $pc(U_j)$, and state probability intervals $pc_{\text{in}}(U_j)$ so that $\forall j \in \{1,\ldots,n\}$.
Step 3	Calculate transition probabilities $tp(U_o\|U_j)$, cumulative transition probabilities $tpc(U_o\|U_j)$, and transition probability intervals $tpc_{\text{in}}(U_o\|U_j)$ so that $\forall o \in \{1,\ldots,n\}$ and $\forall j \in \{1,\ldots,n\}$.
Step 4	Initialize the simulation process by assigning $xs(i=0) \in U$ randomly.
Step 5	Determine the state of $xs(i) \rightarrow S(i)$ as follows: For $j = 1,\ldots,n$ If $xs(i) \in U_j$ Then $S(i) = U_j$ End For
Step 6	Generate a random number $r_i \in [0,1]$.
Step 7	Determine the transition state $S(i+1)$ as follows: For $o = 1,\ldots,n$ If $r_i \in tpc_{\text{in}}(U_o\|U_j)$ Then $S(i+1) = U_o$ End For
Step 8	Simulate $xs(i+1)$ as follow: $xs(i+1) = \max(\min(u_{\max}, f(U_o)), u_{\min})$
Step 9	Redefine the pointer $i = i+1$. If $i \leq N$-1 Then Go To Step 5. Otherwise Stop.

Note that in Step 8, a function $f(U_{(.)})$ is introduced. It produces a value based on the state $U_{(.)}$. When any other information is not available, $f(U_{(.)})$ randomly generates a real number from a normally distributed variable denoted as $rn_{(.)}(\mu(U_{(.)}),\sigma(U_{(.)}))$. Here,

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

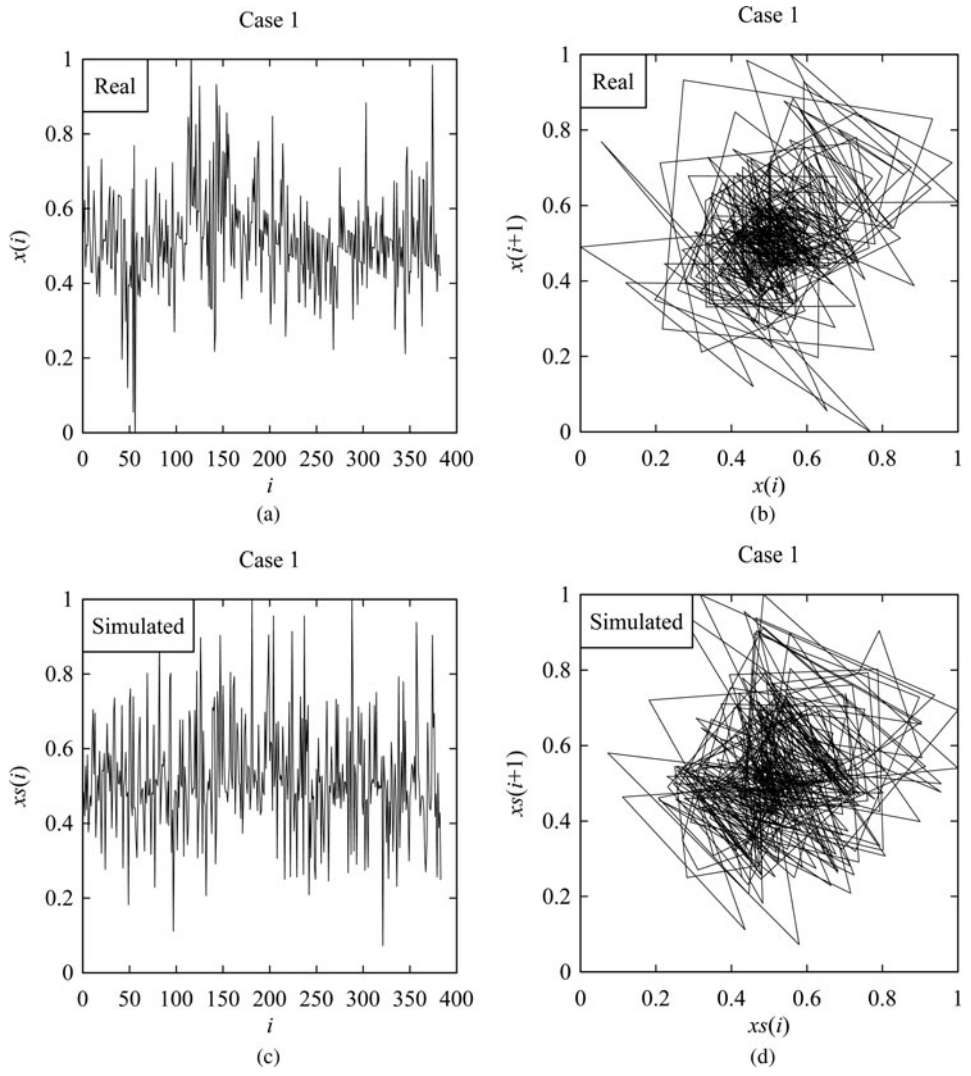


Fig. 6. Results corresponding to Case 1. (a) Real surface profile. (b) Return map of (a). (c) Simulated surface profile. (d) Return map of (c).

$\mu(U_{(\cdot)})$ and $\sigma(U_{(\cdot)})$ denote the mean and standard deviation, respectively. As such, the following formulation holds:

$$f(U_{(\cdot)}) = rn_{(\cdot)}((\mu_{(\cdot)}),(\sigma_{(\cdot)})) \tag{12}$$

The formulation of $f(U_{(\cdot)})$ defined in (12) is used in this article. [Other formulations of $f(U_{(\cdot)})$ can be used, as preferred].

It is worth mentioning that even though a simulated value $xs(i)$ belongs to one of the states say U_j, the next state $xs(i+1)$ may not belong to the same state. This means that the simulation process continues similar to a dynamical system (Ullah, 2017).

However, since the intention of constructing the hidden Markov model is to encapsulate the dynamics of the stochastic features underlying the given time series, the given time series $X = \{x(i) \in \Re \mid i = 0,\ldots,m\}$ and the simulated time series $S = \{xs(i) \in \Re \mid i = 0,1,\ldots\}$ must exhibit similar characteristics. The parameters by which one quantifies the characteristics depend on the phenomenon involved with the given time series X. In general, at least one parameter is needed to compare X and S in the time domain. In addition, one parameter is needed to quantify them in delay domain, that is, their return maps $\{(x(i), x(i+1)) \mid i = 0,1,\ldots,m-1\}$ and $\{(xs(i),xs(i+1)) \mid i = 0,1,\ldots\}$. However, one of the default choices by which one can compare the characteristics of X and S is an entity that is probability distribution-neutral representation of the uncertainty associated with X and S. Since a possibility distribution (e.g., a fuzzy number) is a probability distribution-neutral representation of the uncertainty, the possibility distributions induced from X and S can be used to compare them. The description of the induction of a fuzzy number (i.e., a possibility distribution) from a given set of data can be found in Ullah and Shamsuzzaman, 2013.

Case Study

The computing power of hidden Markov models has been playing an important role in studying the complex phenomena underlying design and manufacturing. For example, Liao *et al.* (2016) have developed a heuristic optimization algorithm using hidden Markov model coupled with simulated annealing for condition monitoring of machineries. Li *et al.* (2018) have developed a data-driven bearing fault identification methodology using an improved hidden Markov model and self-organizing map. Mba *et al.* (2018) developed a hidden Markov model-based methodology for condition monitoring of gearbox. Zhang *et al.* (2018) developed a methodology for predicting the residual life of the rolling machine elements using hidden Markov model. Xie

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

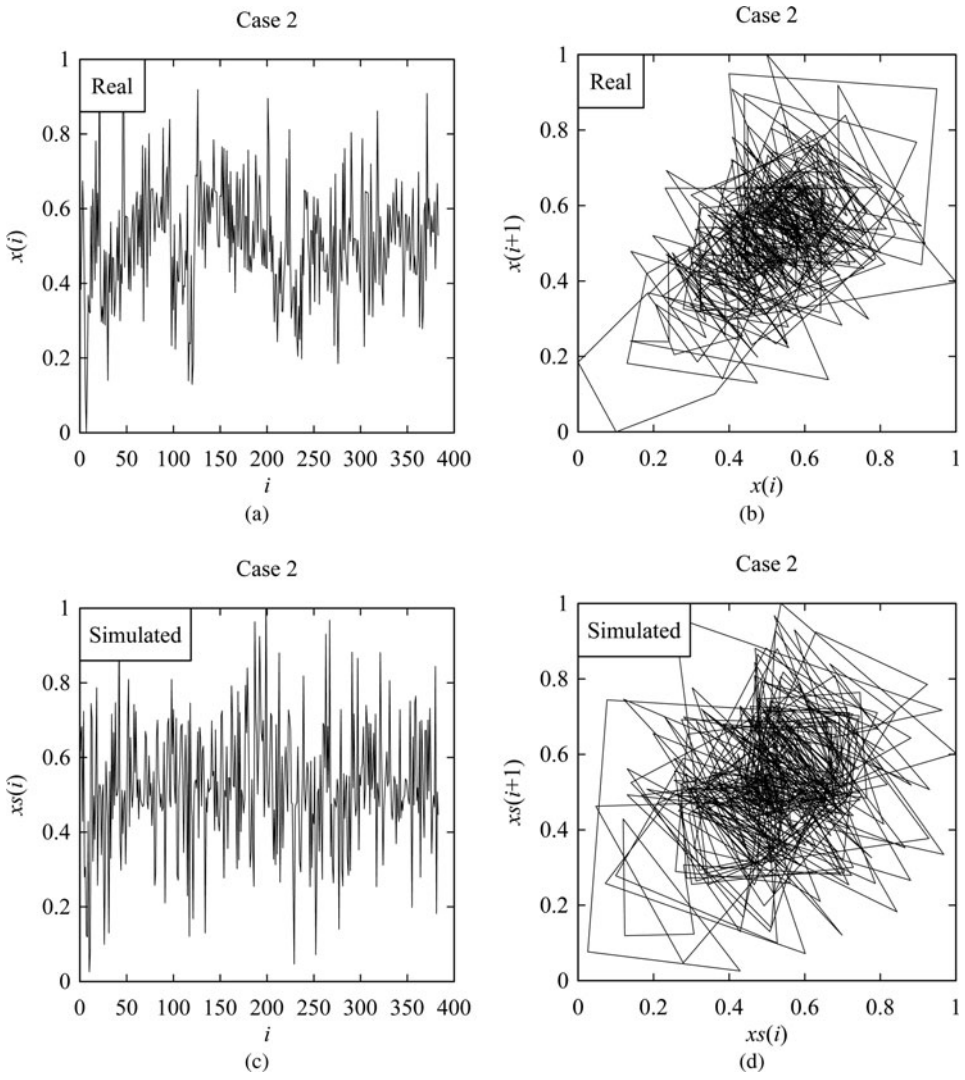


Fig. 7. Results corresponding to Case 2. (a) Real surface profile. (b) Return map of (a). (c) Simulated surface profile. (d) Return map of (c).

et al. (2016) described the hidden Markov model-based methodology for recognizing the machining states ensuring safe operations. Bhat *et al.* (2016) developed a hidden Markov model-based tool condition monitoring methodology ensuring the economical usages of cutting tools. Liao *et al.* (2006) developed a grinding wheel condition monitoring methodology where a hidden Markov model-based clustering approach was used to recognize the patterns found in the acoustic emission signals. Cai *et al.* (2018) developed a methodology using a hidden Markov model to identify the energy efficiency states while removing materials by milling ensuring eco-friendly machining operation. Kumar *et al.* (2018) integrated hidden Markov model with polynomial regression for predicting the useful life of cutting tools. Nevertheless, this case study shows how to apply the hidden Markov model (presented in the section "Methodology") for constructing a phenomenon twin of surface roughness. The description is as follows.

Surface roughness is a concept used to quantify the degree of the surface finish of a processed surface. Therefore, when one studies a manufacturing process in laboratory settings, the surface heights data are measured by using appropriate surface metrology equipment. Afterward, the measured surface heights data are processed to calculate different types of parameters for quantifying the surface roughness. A description of how the surface height data should be processed for calculating the conventional and nonconventional surface roughness quantification parameters can be found in Ullah *et al.* (2015). On the other hand, when one monitors a manufacturing process in real-life settings, the surface heights data are measured by using an appropriate sensor-driven system. The surface heights data are then processed to know whether the process produces the expected surface roughness. As far as the data processing is concerned, the surface heights data are a piece of time series. These data can be processed by using different approaches for the reconstruction of the surface heights [e.g., fractional Brownian motion-based simulation (Higuchi *et al.*, 2001), rule-based systems (Ullah and Harib, 2006), autocorrelation analysis (Chui *et al.*, 2013; Choi *et al.*, 2018), feature-based simulation (Ullah *et al.*, 2010), and integer-sequencedbased dynamical systems (Ullah, 2017)]. The methodologies mentioned above are highly customized, and require a large set of user-defined parameters. Thus, for simplifying the process of the reconstruction of the surface heights, one can use the hidden Markov model described in the previous section. This possibility is explored using the case of surface heights of ground surfaces (i.e., surfaces heights generated due to the application of a material removal process called grinding).

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

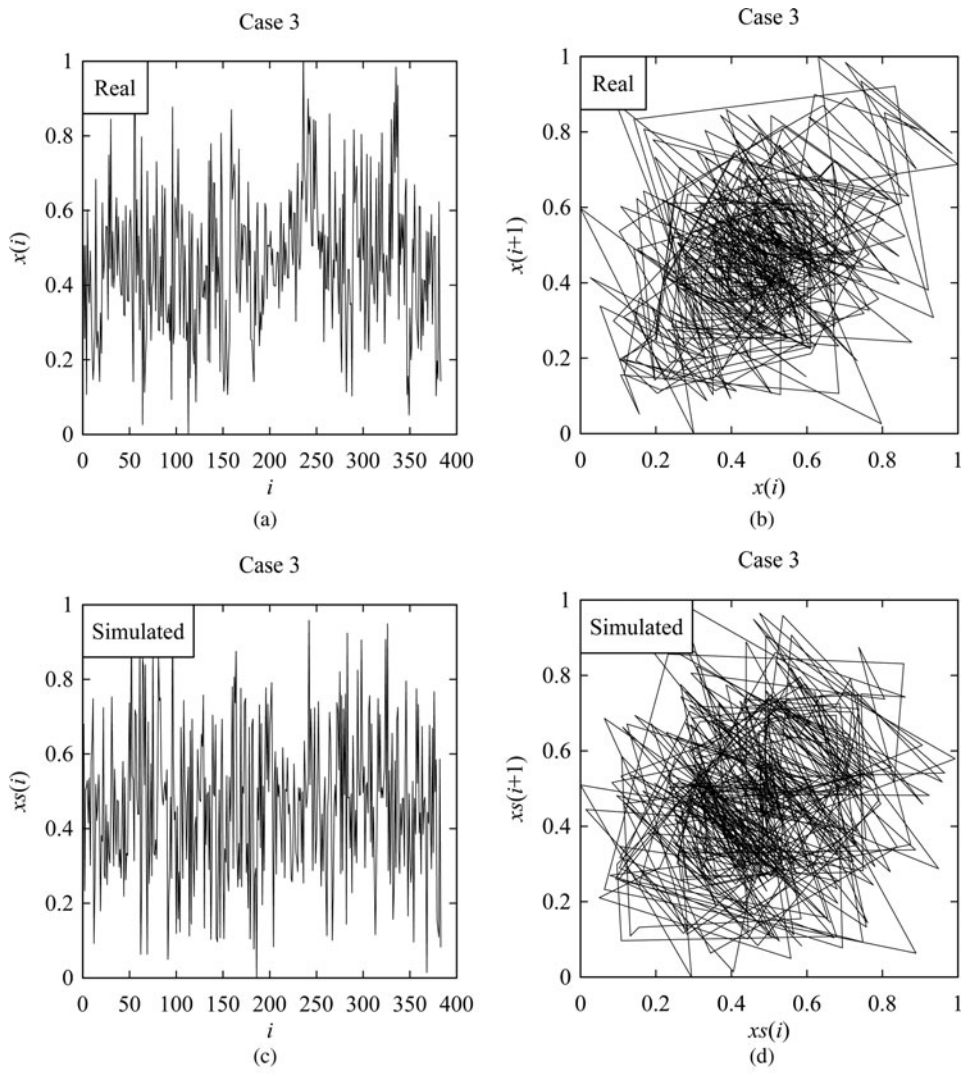


Fig. 8. Results corresponding to Case 3. (a) Real surface profile. (b) Return map of (a). (c) Simulated surface profile. (d) Return map of (c).

Grinding is a widely used material removal process that helps remove materials from the surfaces of the objects made of difficult-to-cut materials (e.g., stainless steels, ceramics, and so forth) ensuring a high surface finish. In grinding, a complex microscopic interaction between the abrasive grains attached on the circumferential surface of grinding wheel and work-surface takes place (Ullah *et al.*, 2018). As a result, in a grinding operation, a grinding wheel passes the same area of the work surface multiple times (as schematically illustrated in Fig. 3) so that the desired amount of materials is removed and better surface finish is achieved. This results in a gradual change in the surface roughness. A detailed description of this grinding mechanism can be found in Ullah *et al.* (2018).

Now, from the viewpoint of futuristic manufacturing systems, one can consider that there is a repository defined as the *grinding experiment repository* where the pass-by-pass grinding experiment results are stored for reuse, as schematically illustrated in Figure 3. It can be located in the manufacturing clouds for sharing as described by Wu *et al.* (2016). Nevertheless, using the information stored in the grinding experiment repository, one can construct the *digital twins of grinding phenomena* (e.g., the digital twins of the grinding force, grinding temperature, wear of grinding wheel, surface roughness, and so forth). This way, the user-dependency can be reduced in manufacturing clouds. As a result, the digital twins form another repository defined as the *grinding phenomenon twin repository*. Between these two repositories, the hidden Markov model plays its role as schematically illustrated in Figure 3. This means that some of the phenomena twins are constructed by using the hidden Markov model. It is worth mentioning that similar repositories can be built for other manufacturing processes, too, and placed in the manufacturing clouds. This will help make the manufacturing clouds more functional because manufacturing clouds still have limited usability since solving modeling and simulations problems in manufacturing clouds depends heavily on the user-experience (Wu *et al.*, 2016).

To see the efficacy of the hidden Markov model in creating a phenomenon twin of the surface roughness of grinding operations, consider the measured surface heights of a ground surface for the four consecutive passes, denoted as Cases 1,…,4 as reported in Ullah *et al.* (2018). For the sake of comparison, the heights are normalized and shown by the four plots in Figure 4. As seen in Figure 4, the surface heights are very irregular and stochastic. For constructing a hidden Markov model, the

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

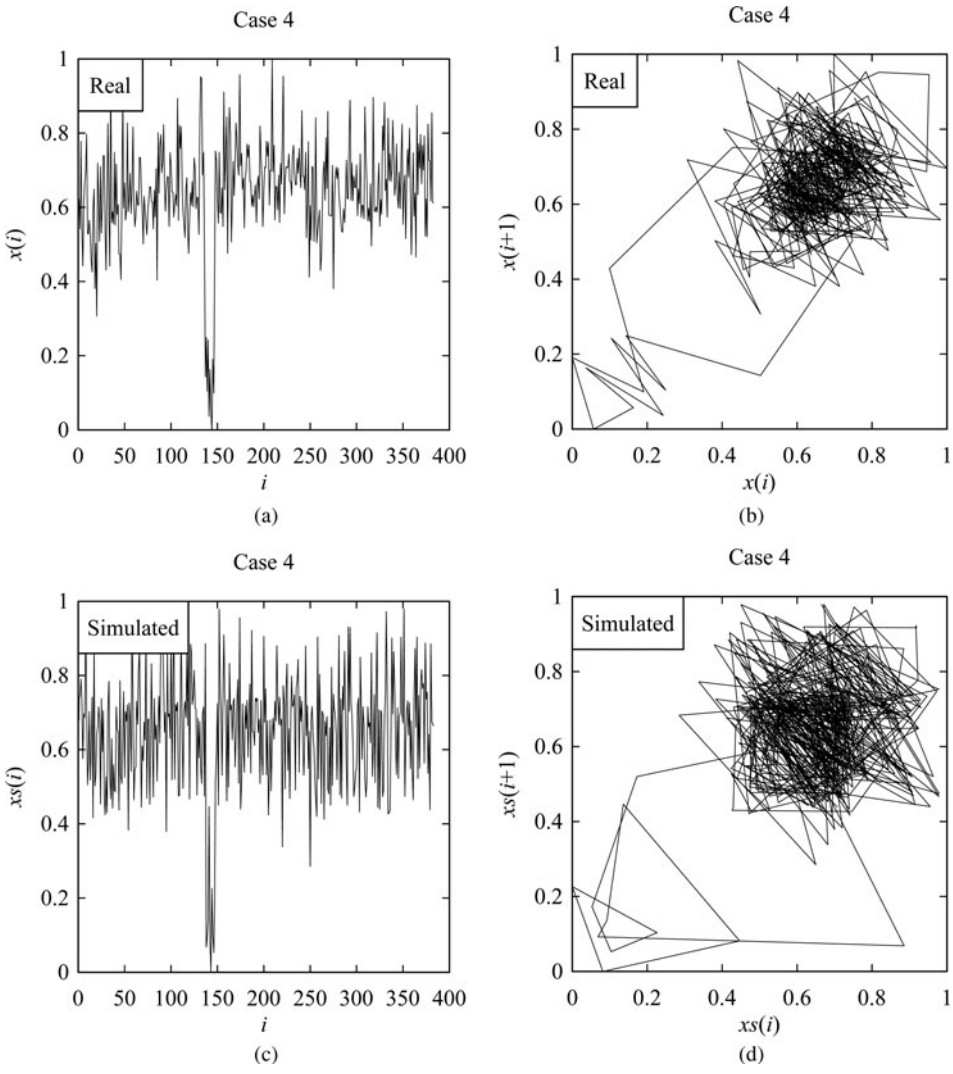


Fig. 9. Results corresponding to Case 4. (a) Real surface profile. (b) Return map of (a). (c) Simulated surface profile. (d) Return map of (c).

following five latent states are considered making the proposition P true [Eq. (1)]: $U_1 = [0,0.2)$, $U_2 = [0.2,0.4)$, $U_3 = [0.4,0.6)$, $U_4 = [0.6,0.8)$, and $U_5 = [0.8,1]$, that is, $U_1 \cup U_2 \cup U_3 \cup U_4 \cup U_5 = [0, 1]$. From the semantics sense, "U_1" represents "very low" heights, "U_2" represents "low" heights, "U_3" represents "moderate" heights, "U_4" represents "high" heights, and "U_5" represents "very high" heights. A user may use other formulations of the latent states.

In accordance with Eq. (9), the transition probability matrix (M_{tp}) for Cases 1,…,4 are constructed as given by Eqs (13)–(16).

$$M_{tp(\text{Case 1})} = \begin{bmatrix} 0 & 0.5 & 0.25 & 0.25 & 0 \\ 0 & 0.192 & 0.577 & 0.212 & 0.019 \\ 0.004 & 0.142 & 0.696 & 0.130 & 0.028 \\ 0.043 & 0.058 & 0.594 & 0.275 & 0.030 \\ 0 & 0.091 & 0.273 & 0.545 & 0.091 \end{bmatrix} \quad (13)$$

$$M_{tp(\text{Case 2})} = \begin{bmatrix} 0.333 & 0.334 & 0.333 & 0 & 0 \\ 0.044 & 0.324 & 0.5 & 0.117 & 0.015 \\ 0.010 & 0.171 & 0.540 & 0.245 & 0.034 \\ 0.011 & 0.078 & 0.555 & 0.323 & 0.033 \\ 0 & 0.083 & 0.584 & 0.25 & 0.083 \end{bmatrix} \quad (14)$$

$$M_{tp(\text{Case 3})} = \begin{bmatrix} 0.290 & 0.291 & 0.161 & 0.161 & 0.097 \\ 0.135 & 0.288 & 0.406 & 0.144 & 0.027 \\ 0.032 & 0.31 & 0.478 & 0.154 & 0.026 \\ 0.045 & 0.258 & 0.409 & 0.182 & 0.106 \\ 0 & 0.2 & 0.2 & 0.45 & 0.15 \end{bmatrix} \quad (15)$$

$$M_{tp(\text{Case 4})} = \begin{bmatrix} 0.625 & 0.25 & 0.125 & 0 & 0 \\ 0.4 & 0 & 0.2 & 0.4 & 0 \\ 0.009 & 0.009 & 0.375 & 0.5 & 0.107 \\ 0 & 0.009 & 0.257 & 0.635 & 0.099 \\ 0 & 0 & 0.306 & 0.638 & 0.056 \end{bmatrix} \quad (16)$$

In accordance with Eq. (11), the cumulative transition probability matrix (M_{tpc}) for Cases 1,…,4 are constructed as given by Eqs (17)–(20).

$$M_{tpc(\text{Case 1})} = \begin{bmatrix} 0 & 0.5 & 0.75 & 1 & 1 \\ 0 & 0.192 & 0.769 & 0.989 & 1 \\ 0.004 & 0.146 & 0.842 & 0.972 & 1 \\ 0.043 & 0.101 & 0.695 & 0.970 & 1 \\ 0 & 0.091 & 0.364 & 0.909 & 1 \end{bmatrix} \quad (17)$$

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

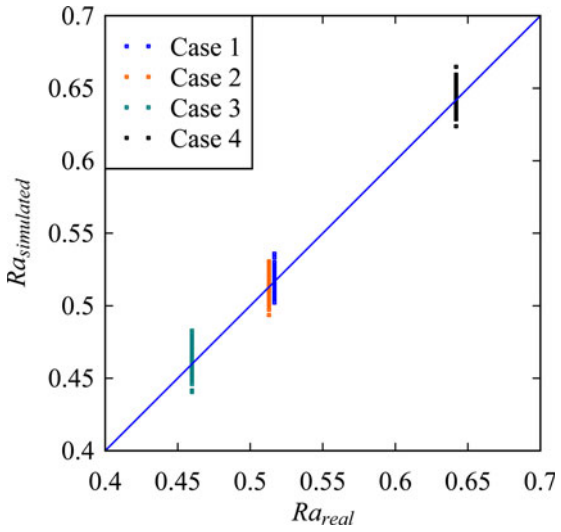


Fig. 10. Comparison between the real and the simulated surface roughness.

$$M_{tpc(\text{Case 2})} = \begin{bmatrix} 0.333 & 0.667 & 1 & 1 & 1 \\ 0.044 & 0.368 & 0.868 & 0.985 & 1 \\ 0.010 & 0.181 & 0.721 & 0.966 & 1 \\ 0.011 & 0.089 & 0.644 & 0.967 & 1 \\ 0 & 0.083 & 0.667 & 0.917 & 1 \end{bmatrix} \quad (18)$$

$$M_{tpc(\text{Case 3})} = \begin{bmatrix} 0.290 & 0.581 & 0.742 & 0.903 & 1 \\ 0.135 & 0.423 & 0.829 & 0.973 & 1 \\ 0.032 & 0.342 & 0.820 & 0.974 & 1 \\ 0.045 & 0.303 & 0.712 & 0.894 & 1 \\ 0 & 0.2 & 0.4 & 0.85 & 1 \end{bmatrix} \quad (19)$$

$$M_{tpc(\text{Case 4})} = \begin{bmatrix} 0.625 & 0.875 & 1 & 1 & 1 \\ 0.4 & 0.4 & 0.6 & 1 & 1 \\ 0.009 & 0.018 & 0.393 & 0.893 & 1 \\ 0 & 0.009 & 0.266 & 0.901 & 1 \\ 0 & 0 & 0.306 & 0.944 & 1 \end{bmatrix} \quad (20)$$

Using the results shown in Eqs (17)–(20), the transition probability intervals denoted as $tpc_{\text{in}}(U_o|U_j)$ ($\forall o \in \{1,\ldots,5\}$ and $\forall j \in \{1,\ldots,5\}$) are calculated for each case. For simulating the observations from the latent states, the values of the mean of the respective states are set as follows: $\mu(U_1) = 0.1, \mu(U_2) = 0.3, \mu(U_3) = 0.5, \mu(U_4) = 0.7$, and $\mu(U_5) = 0.9$. The values of the standard deviation are set as follows: $\sigma(U_1) = \sigma(U_2) = \sigma(U_3) = \sigma(U_4) = \sigma(U_5) = 0.05$. This formulation is based on Eq. (12). Using the above settings, the Monte Carlo simulations as defined by Steps 4,…,9 (see the section "Simulation") are carried out for each case. Figure 5 shows the screen-print of the spreadsheet-based computing tool that implements the simulation process. As seen in Figure 5, the real surface heights X (in this case see Case 4 in Fig. 4) and the simulated surface heights S are seen on the screen-print. The user needs to input the time series X and define the latent states, $U_1,\ldots,U_5$. The computing tool does the rest. The computing tool also plots the respective return maps and the possibility distributions for the sake of comparison.

However, Figures 6–9 shows the samples of the simulated surface heights in terms of both time series and return maps, along with the time series and return maps of the respective real heights. For comparing the characteristics of real surface heights X and simulated surface heights S, two parameters are considered in this section. The first parameter is the arithmetic mean deviation of the surface heights denoted as Ra (Ullah *et al.*, 2015), which is the most widely used surface roughness parameter. The other one is the possibility distributions (or fuzzy numbers) (Ullah and Shamsuzzaman, 2013).

The values of Ra are calculated for the real and simulated surface heights (i.e., for X and S) as shown in Figure 10. The simulations were carried out 100 times for each case to observe the variability. The unit-sloped straight line shown by the blue color in Figure 10 represents the ideal case, that is, Ra of the simulated surface heights is equal to that of the real surface heights. As seen in Figure 10, for all cases, more than 50% of the simulated results are greater than that of the real ones. This means that the hidden Markov model-based phenomenon twin for surface roughness produces a rougher surface than the real one.

On the other hand, the possibility distributions of the simulated surface heights resemble those of the real surface heights, as shown in Figure 11. This means that from the dynamical systems point of view, the characteristics of the real and simulated surface heights are quite similar although their degrees of roughness are not exactly the same.

Ontology

Apart from the comparison between the real and simulated surface heights, there are other important issues, for example, the issue of ontology. Ontology has been implemented within the framework of futuristic manufacturing systems, for example, see the articles Khilwani and Harding (2014), Ramos (2015), Lu and Xu (2018), Kim and Ahmed (2018). It provides a semantic representation of the relevant contents. It has become more relevant because of both the maturity indices of the web technology (semantic web or web 3.0/4.0, e.g., see Berners-Lee *et al.*, 2001) and the futuristic manufacturing systems (understand, predict, decide, and adopt, e.g., see Schuh *et al.*, 2017). In terms of the enterprise system architecture, an ontology is a semantic model of the contents (e.g., in this case, a semantic model of the digital twin) that forms the meta-model and meta-meta model (Tunjic *et al.*, 2018), that is, high-level description of the contents. Therefore, the semantic annotations (linguistically described semantic description) must accompany all concepts necessary for encoding the contents (Fill, 2017, 2018). On the other hand, as far as the semantic web—which will be used to functionalize the connections of cyber-physical systems (Alam and Saddik, 2017)—is concerned, the role of the ontology can be seen from a broader perspective. Compared to its predecessors (web 1.0/2.0), the semantic web has some additional features that make it self-contained and autonomous. These features form the upper layers of the semantic web layer-cake, namely, unified logic, proof, trust, and user-interface and application (Sizov, 2007). The purpose of these layers is to provide the *provenance* (Moreau *et al.*, 2015; Oliveira *et al.*, 2017) of the contents, incorporating the individuals, institutions, data sources, data manipulation methodologies and algorithms, and so forth regarding the contents. Using the provenance, the layers of unified logic, proof, trust, and user-interface are built. Therefore, an entity defined as meta-ontology, must be made available to the systems

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

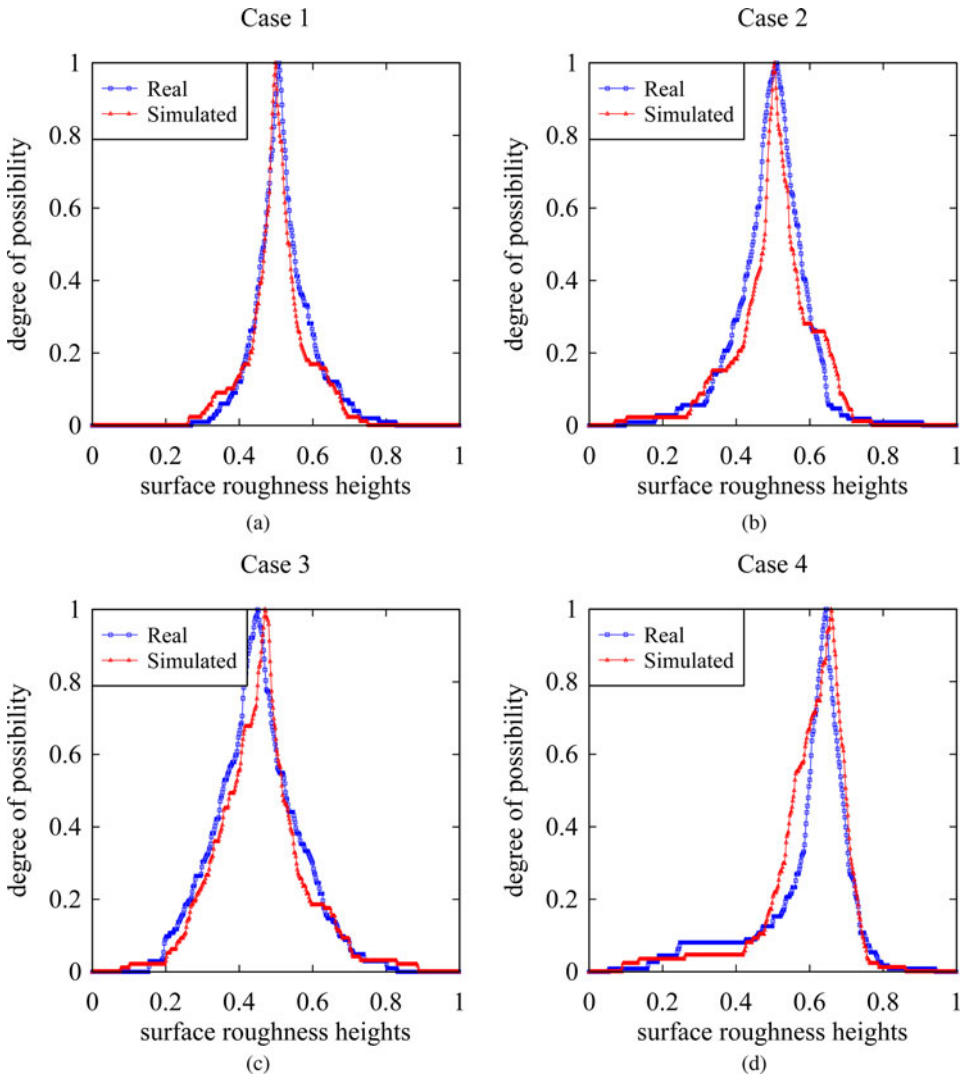


Fig. 11. Possibility distributions of the real and the simulated surface heights. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4.

developers of the cyber-physical systems using which the developers then construct the ontology of the contents to be used in the upper layers of semantic web layer-cake. Now the question is: what is the meta-ontology of the digital twin of surface roughness? One of the answers is presented as follows:

For the case shown in this section (i.e., digital twin of surface roughness) that entities that are subject to the provenance are as follows: a grinding operation, grinding wheel, workpiece, grinding conditions, surface metrology instrument, grinding machine, computing tool, surface heights, hidden Markov model, digital twin, surface roughness, five latent states, very low, low, moderate, high, very high, state probabilities, transition, one of the latent states, and transition probabilities. These entities can be considered the concepts of a concept map (Ullah *et al.*, 2013; Ullah, 2019) for providing the meta-ontology of the provenance. One of the possible outcomes is shown in Figure 12. Some of the concepts are repeated (e.g., surface heights) for the sake of better understanding. One can access this concept map from the following URL for integrating it to the systems of futuristic manufacturing systems: http://cmapspublic3.ihmc.us/rid=1SVM7NWKX-W2MYVP-1FWD/hidden-markov-model-of-surface-roughness.cmap. The meta-ontology shown in Figure 12 entails the following propositions: (1) a grinding operation entails a grinding wheel, workpiece, grinding conditions, surface metrology instrument, grinding machine; (2) surface metrology instrument measures the surface heights to represent surface roughness; (3) surface heights are normalized to construct a digital twin of surface roughness using hidden Markov model; (4) hidden Markov model consists of five latent states namely very low, low, moderate, high, and very high; (5) five latent states namely very low, low, moderate, high, and very high have the state probabilities; (6) transition of surface heights from one of the latent states to very low, low, moderate, high, and very high have transition probabilities; (7) hidden Markov model results in a computing tool to simulate surface heights. Each concept can be integrated with other concepts for making it comprehensible to the users who are not familiar to it. For example, with the concept of grinding wheel one can link a concept map that describes a grinding wheel and its features, construction, and so forth. The concept "hidden Markov model" can be linked to the articles or tutorials where a lucid and user-friendly description of it is available. The concept "computing tool" can be linked to a source from where the user can download the computing tool (e.g., the tool shown in Fig. 5). For more details on how to construct a good ontology for manufacturing, refer to Ullah (2019). It is worth mentioning that one can easily create the machine-

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

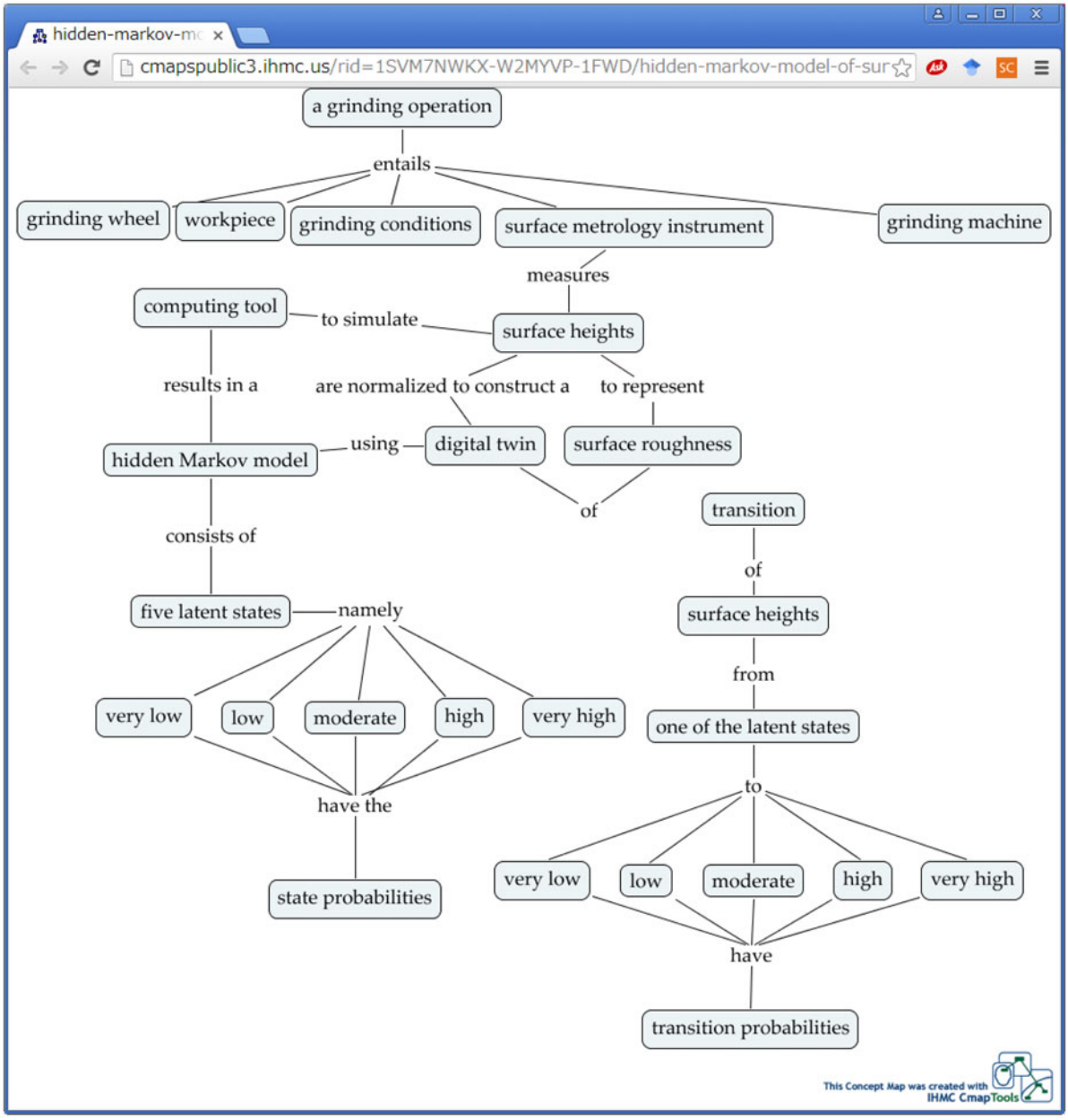


Fig. 12. The meta-ontology of the digital twin of the surface roughness due to grinding.

readable data (XML, OWL, and so forth) from the concept map shown in Figure 12.

Summary and concluding remarks

(1) The recent literature review on digital twins relevant to manufacturing suggests that digital twins must populate the cyberspace of the cyber-physical systems to functionalize the futuristic manufacturing systems. This means that the digital twins are the contents on which the systems associated with futuristic manufacturing systems (e.g., cyber-physical systems) act to perform the desired functions. It is however difficult to prescribe a unified methodology by which all kinds of digital twins can be created, used, and managed. This is somewhat a new topic for the manufacturing research community.

(2) As far as the complex phenomena are concerned, the hidden Markov model is one of the choices using which one can construct digital twins. A hidden Markov model consists of three components, namely, a Markov chain, a simulation process that simulates the latent states, and a simulation process that simulates the observation.

(3) It is possible to encapsulate the dynamics of the phenomenon (manifested in the form of a time series) using a Markov chain where the latent states are user-defined. Using a Monte Carlo simulation process, the latent states can be simulated in accordance with the Markov chain. Each simulated latent state can be translated into an observation (i.e., a numerical value) using a user-defined function. The simulated observations (a time series) are the outcomes of the hidden Markov model. If the time series of the observation resembles that of the phenomenon in

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

both qualitative and quantitative viewpoints, then it can be claimed that the constructed hidden Markov model is an exact mirror image of the underlying phenomenon, that is, it is phenomenon twin.

(4) The presented hidden Markov model is applied to construct a digital twin of the surface roughness of a ground surface (a surface created due to successive grinding operations). The surface heights given in the form of a time series is first used to construct a Markov chain using the five latent states (very low heights, low heights, moderate heights, high heights, and very high heights). Using a Monte Carlo simulation process that simulates the latent states in accordance with the constructed Markov chain, the five latent states are simulated randomly to produce a time series of latent states. Each latent state is then translated to an observation (i.e., a simulated surface height) considering that the height distribution follows a normal distribution. The mean and standard deviation are customized for each latent state. The time series of the simulated surface heights and the real surface heights are compared quantitatively using two parameters, i.e., arithmetic average surface roughness (*Ra*) and possibility distribution (probability-distribution neutral quantification of uncertainty). In terms of *Ra*, it is found that the hidden Markov model-based phenomenon twin of surface roughness produces a rougher surface than the real one. And, the possibility distributions of the simulated surface heights resemble those of the real surface heights. This means that from the dynamical systems point of view, the characteristics of the real and simulated surface heights are quite similar although they are not exactly the same from the viewpoint of *Ra*.

(5) A meta-ontology of the digital twin of the surface roughness is constructed using a semantic web embedded concept map. The concepts and the propositions underlying the concept map provide the provenance of the content (i.e., the digital twin). This is needed for building the upper layers, namely, unified logic, proof, trust, and user-interface and application of the semantic web. Thus, the representation of the digital twin in the form of the semantic web embedded concept map must populate the cyber-physical systems to functionalize the futuristic manufacturing systems called Industry 4.0.

Acknowledgments. An initial short version of the manuscript was published in the proceedings of the 17th International Conference on Precision Engineering, November 12-16, 2018, Kamakura, Japan (Paper A-1-6) and in the Proceedings of the 22nd Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES2018), December 20–22, 2018, Sapporo, Japan, CD–ROM, pp. 13–20. The first author is supported by the MEXT Scholarship.

References

Alam KM and Saddik AE (2017) C2PS: a digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE Access* **5**, 2050–2062.

Baum LE and Petrie T (1966) Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Statist.* **37**, 1554–1563.

Berners-Lee T, Hendler J and Lassila O (2001) The semantic web. *Sci. Am.* **284**, 34–43.

Bhat NN, Dutta S, Pal SK and Pal S (2016) Tool condition classification in turning process using hidden Markov model based on texture analysis of machined surface images. *Measurement* **90**, 500–509.

Botkina D, Hedlind M, Olsson B, Henser J and Lundholm T (2018) Digital twin of a cutting tool. *Proc. CIRP* **72**, 215–218.

Cai Y, Shi X, Shao H, Wang R and Liao S (2018) Energy efficiency state identification in milling processes based on information reasoning and hidden Markov model. *J. Clean. Prod.* **193**, 397–413.

Choi W, Shi F, Lowe MJS, Skelton EA, Craster RV and Daniels WL (2018) Rough surface reconstruction of real surfaces for numerical simulations of ultrasonic wave scattering. *NDT & E Int.* **98**, 27–36.

Chui MW, Feng YQ, Wang W, Li PL and Li ZC (2013) Numerical simulation of rough surface with crossed texture. *Appl. Mech. Mater.* **321–324**, 196–200.

Fill H-G (2017) SeMFIS: a flexible engineering platform for semantic annotations of conceptual models. *Semant. Web* **8**, 747–763.

Fill H-G (2018) Semantic annotations of enterprise models for supporting the evolution of model-driven organizations. *Enterprise Model. Info. Syst. Archit.* **13**, 5:1–5:25.

Fraser AM (2008) *Hidden Markov Models and Dynamical Systems*. Philadelphia: SIAM.

Glaessgen EH and Stargel DS (2012) The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Structures, Structural Dynamics, and Materials and Co-located Conferences*. Honolulu, Hawaii: American Institute of Aeronautics and Astronautics.

Grieves M and Vickers J (2017) Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In Kahlen F-J, Flumerfelt S and Alves A (eds), *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Cham: Springer International Publishing, pp. 85–113.

Haag S and Anderl R (2018) Digital twin – Proof of concept. *Manuf. Letter.* **15**(B), 64–66.

Higuchi M, Yamaguchi T, Yano A, Yamamoto N, Ueshima R, Matumori N and Yoshizawa I (2001) Development of design technology of porous superfinishing stone using fractal geometry (2nd report): geometric modeling of stone topography and design support system. *J. Japan Soc. Prec. Eng.* **67**(3), 428–432. (In Japanese)

Hu L, Nguyen N-T, Tao W, Leu MC, Liu XF, Shahriar MR and Sunny SMNA (2018) Modeling of cloud-based digital twins for smart manufacturing with MT connect. *Proc. Manuf.* **26**, 1193–1203.

Khilwani N and Harding JA (2014) Managing corporate memory on the semantic web. *J. Intell. Manuf.* **27**(1), 101–118.

Kim K-Y and Ahmed F (2018) Semantic weldability prediction with RSW quality dataset and knowledge construction. *Adv. Eng. Inform.* **38**, 41–53.

Kritzinger W, Karner M, Traar G, Henjes J and Sihn W (2018) Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* **51**(11), 1016–1022.

Kumar A, Chinnam RB and Tseng F (2018) An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools. *Comput. Ind. Eng.* **128**, 1008–1014.

Kunath M and Winkler H (2018) Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process. *Proc. CIRP* **72**, 225–231.

Li J, Pedrycz W and Jamal I (2017) Multivariate time series anomaly detection: a framework of hidden Markov models. *Appl. Soft Comput.* **60**, 229–240.

Li Z, Fang H, Huang M, Wei Y and Zhang L (2018) Data-driven bearing fault identification using improved hidden Markov model and self-organizing map. *Comput. Ind. Eng.* **116**, 37–46.

Liao TW, Hua G, Qu J and Blau PJ (2006) Grinding wheel condition monitoring with hidden Markov model-based clustering methods. *Mach. Sci. Technol. Int. J.* **10**, 511–538.

Liao W, Li D and Cui S (2016) A heuristic optimization algorithm for HMM based on SA and EM in machinery diagnosis. *J. Intell. Manuf.* **29**, 1845–1857.

Lu Y and Xu X (2018) Resource virtualization: a core technology for developing cyber-physical production systems. *J. Manuf. Syst.* **47**, 128–140.

Luo W, Hu T, Zhu W and Tao F (2018) Digital twin modeling method for CNC machine tool. *Proc. IEEE 15th International Conference on Networking, Sensing and Control, ICNSC,18*. Zhuhai, China: IEEE.

Mba CU, Makis V, Marchesiello S, Fasana A and Garibaldi L (2018) Condition monitoring and state classification of gearboxes using stochastic resonance and hidden Markov models. *Measurement* **126**, 76–95.

Moreau L, Groth P, Cheney J, Lebo T and Miles S (2015) The rationale of PROV. *J. Web Semant.* **35**, 235–257.

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press

Nguyen N (2017) An analysis and implementation of the hidden Markov model to technology stock prediction. *Risks* **5**, 62:1–62:18.

Oliveira W, Ambrósio LM, Braga R, Ströele V, David JM and Campos F (2017) A framework for provenance analysis and visualization. *Proc. Comput. Sci.* **108**, 1592–1601.

Olivotti D, Dreyer S, Lebek B and Breitner MH (2018) Creating the foundation for digital twins in the manufacturing industry: an integrated installed base management system. *Inf. Syst. E-Bus. Manage.* doi: 10.1007/s10257-018-0376-0.

Padovano A, Longo F, Nicoletti L and Mirabelli G (2018) A digital twin based service oriented application for a 4.0 knowledge navigation in the smart factory. *IFAC-PapersOnLine* **51**, 631–636.

Petropoulos A, Chatzis SP and Xanthopoulos S (2017) A hidden Markov model with dependence jumps for predictive modeling of multidimensional time-series. *Info. Sci.* **412–413**, 50–66.

Qi Q, Tao F, Zuo Y and Zhao D (2018) Digital twin service towards smart manufacturing. *Proc. CIRP* **72**, 237–242.

Ramos L (2015) Semantic Web for manufacturing, trends and open issues: toward a state of the art. *Comput. Ind. Eng.* **90**, 444–460.

Rosen R, Wichert Gv, Lo G and Bettenhausen KD (2015) About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine* **48**, 567–572.

Scaglioni B and Ferretti G (2018) Towards digital twins through object-oriented modelling: a machine tool case study. *IFAC-PapersOnLine* **51**, 613–618.

Schleich B, Anwer N, Mathieu L and Wartzack S (2017) Shaping the digital twin for design and production engineering. *CIRP Ann. Manuf. Technol.* **66**, 141–144.

Schroeder GN, Steinmetz C, Pereira CE and Espindola DB (2016) Digital twin data modeling with AutomationML and a communication methodology for data exchange. *IFAC-PapersOnLine* **49**, 12–17.

Schuh G, Anderl R, Gausemeier J, Hompel Mt and Wahlster W (eds) (2017) Industrie 4.0 maturity Index. In *Managing the Digital Transformation of Companies* (acatech STUDY). Munich: Herbert Utz Verlag, pp. 1–53.

Sizov S (2007) What makes You think that? The semantic web's proof layer. *IEEE Intell. Syst.* **22**, 94–99.

Söderberg R, Wärmefjord K, Carlson JS and Lindkvist L (2017) Toward a digital twin for real-time geometry assurance in individualized production. *CIRP Ann. Manuf. Technol.* **66**, 137–140.

Talkhestani BA, Jazdi N, Schloegl W and Weyrich M (2018) Consistency check to synchronize the digital twin of manufacturing automation based on anchor points. *Proc. CIRP* **72**, 159–164.

Tao F, Cheng J, Qi Q, Zhang M, Zhang H and Sui F (2018) Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **94**, 3563–3576.

Tunjic C, Atkinson C and Draheim D (2018) Supporting the model-driven organization vision through deep, orthographic modeling. *Enterprise Model. Info. Syst. Archit.* **13**, 7:1–7:39.

Uhlemann TH-J, Schock C, Lehmann C, Freiberger S and Steinhilper R (2017) The digital twin: demonstrating the potential of real time data acquisition in production systems. *Proc. Manuf.* **9**, 113–120.

Ullah AMMS (2017) Surface roughness modeling using Q-sequence. *Math. Comput. Appl.* **22**, 33:1–33:12.

Ullah AMMS (2019) Modeling and simulation of complex manufacturing phenomena using sensor signals from the perspective of Industry 4.0. *Adv. Eng. Inform.* **39**, 1–13.

Ullah AMMS and Harib KH (2006) Knowledge extraction from time series and its application to surface roughness simulation. *Info. Knowl. Syst. Manag.* **5**, 117–134.

Ullah AMMS and Shamsuzzaman M (2013) Fuzzy Monte Carlo simulation using point-cloud-based probability- possibility transformation. *Simulation* **89**, 860–875.

Ullah AMMS, Tamaki J and Kubo A (2010) Modeling and simulation of 3D surface finish of grinding. *Adv. Mater. Res.* **126–128**, 672–677.

Ullah AMMS, Arai N and Watanabe M (2013) Concept map and internet-aided manufacturing. *Proc. CIRP* **12**, 378–383.

Ullah AMMS, Fuji A, Kubo A, Tamaki J and Kimura M (2015) On the surface metrology of bimetallic components. *Machining Sci. Technol.* **19**, 339–359.

Ullah A, Caggiano A, Kubo A and Chowdhury M (2018) Elucidating grinding mechanism by theoretical and experimental investigations. *Materials* **11**, 274:1–274:19.

Visser I (2011) Seven things to remember about hidden Markov models: a tutorial on Markovian models for time series. *J. Math. Psychol.* **55**, 403–415.

Wu D, Terpenny J and Schaefer D (2016) Digital design and manufacturing on the cloud: a review of software and services. *Artif. Intell. Eng. Des. Anal. Manuf.* **31**, 104–118.

Xie F-Y, Hu Y-M, Wu B and Wang Y (2016) A generalized hidden Markov model and its applications in recognition of cutting states. *Int. J. Preci. Eng. Manuf.* **17**, 1471–1482.

Zhang S, Zhang Y and Zhu J (2018) Residual life prediction based on dynamic weighted Markov model and particle filtering. *J. Intell. Manuf.* **29**, 753–761.

Zheng Y, Yang S and Cheng H (2018) An application framework of digital twin and its case study. *J. Ambient Intell. Humanized Comput.* **10**, 1141–1153.

Zhuang C, Liu J and Xiong H (2018) Digital twin-based smart production management and control framework for the complex product assembly shop-floor. *Int. J. Adv. Manuf. Technol.* **96**, 1149–1163.

Angkush Kumar Ghosh received his BSME from the Khulna University of Engineering and Technology in 2013. Currently, he is pursuing his graduate studies leading to a doctoral degree in advanced manufacturing as a MEXT scholarship recipient at the Kitami Institute of Technology, Japan. Prior to his current role, he was a lecturer in the Department of Mechanical Engineering at the Sonargaon University, Dhaka. He researches artificially intelligent systems for Industry 4.0 and beyond.

AMM Sharif Ullah is currently a Professor at the Kitami Institute of Technology and directs its Advanced Manufacturing Laboratory. Prior to his current role, he was a full-time faculty member at the Asian Institute of Technology (2000–2002) and United Arab Emirates University (2002–2009). He received his BSME from the BUET in 1992 and MSME and PhD from the Kansai University in 1996 and 1999, respectively. He currently researches intelligent systems for Industry 4.0 and beyond, 3D printing, and design theory.

Akihiko Kubo is an Assistant Professor at the Kitami Institute of Technology and affiliated with its Advanced Manufacturing Laboratory. He received his Bachelor's degree in Mechanical Engineering from the Kushiro National College of Technology. He has been teaching courses related to precision machining and computer-aided design and manufacturing. His current research focuses on modeling and simulation of precision machining and grinding.

https://doi.org/10.1017/S089006041900012X Published online by Cambridge University Press